

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный технический университет»

Факультет Электроники и вычислительной техники

Кафедра Программное обеспечение автоматизированных систем

Согласовано

Утверждаю

и. о. зав. кафедрой

(должность гл. специалиста предприятия)

О. А. Сычев

(подпись)

(инициалы, фамилия)

(подпись)

(инициалы, фамилия)

« » 20 г.

« » 20 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе бакалавра на тему
(наименование вида работы)

Разработка мобильного приложения для контроля тренировочного процесса
и приема биологически активных добавок спортсменами (Backend).

Автор Чупинин Антон Павлович
(подпись и дата подписания) (фамилия, имя, отчество)

Обозначение ВКРБ-09.03.04-10.19-16-24
(код документа)

Группа ПрИн-466
(шифр группы)

Направление 09.03.04 – Программная инженерия,
Разработка программно-информационных систем
(код и наименование направления, наименование программы (профиля))

Руководитель работы Гилка В.В.
(подпись и дата подписания) (инициалы и фамилия)

Консультанты по разделам:

(краткое наименование раздела) (подпись и дата подписания) (инициалы и фамилия)

(краткое наименование раздела) (подпись и дата подписания) (инициалы и фамилия)

Нормоконтролер: Кузнецова А.С.
(подпись и дата подписания) (инициалы и фамилия)

Волгоград 2024 г.

Кафедра Программное обеспечение автоматизированных систем

и. о. зав. кафедрой

(подпись)

(инициалы, фамилия)

« » 20 Г.

на выпускную квалификационную работу бакалавра

(наименование вида работы)

Студент Чупинин Антон Павлович

(фамилия, имя, отчество)

Код кафедры	10.19	Группа	ПрИн-466
-------------	-------	--------	----------

Тема Разработка мобильного приложения для контроля тренировочного
процесса и приема биологически активных добавок спортсменами (Backend).

Утверждена приказом по университету « 01 » сентября 20 23 г. № 1074-ст

Срок представления готовой работы (проекта)

(дата, подпись студента)

Исходные данные для выполнения работы (проекта)

Задание, выданное научным руководителем кафедры «ПОАС»

Содержание основной части пояснительной записки

1 Анализ проблемы и существующих решений

2 Предлагаемый способ решения поставленной задачи

Выводы

3 Проектирование и разработка приложения

Выводы

4 Тестирование разработанного приложения

Выводы

Заключение

Приложение А - Справка о результатах проверки выпускной

квалификационной работы на наличие заимствований

Приложение Б - Техническое задание

Приложение В - Руководство системного программиста

Перечень графического материала

- 1) Актуальность работы
- 2) Описание существующей проблемы в заданной предметной области.
- 3) Цели и задачи
- 4) Объект и предмет исследования
- 5) Практическая значимость.
- 6) Обзор аналогов
- 7) Как решалось и предлагаемое решение
- 8) Функциональные требования.
- 9) Входные и выходные данные
- 10) Макет основных экранов модуля
- 11) Структура модуля
- 12) Требования к Бэкенд части приложения
- 13) Аутентификация
- 14) Реализация уведомлений
- 15) Абстрактная структура и формат данных
- 16) Диаграмма классов Backend-части приложения
- 17) Результаты опросов независимых пользователей
- 18) Инструментальные средства, язык, библиотеки

Руководитель работы (проекта)

(подпись и дата подписания)

Гилка В.В.

(инициалы и фамилия)

Консультанты по разделам:

(краткое наименование раздела)

(подпись и дата подписания)

(инициалы и фамилия)

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:

и. о. зав. кафедрой ПОАС

_____ О.А. Сычев

«___» _____ 20 ____ г.

Разработка мобильного приложения для контроля тренировочного процесса
и приема биологически активных добавок спортсменами (Backend).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ВКРБ–09.03.04–10.19–16–24–81

Листов 75

Руководитель работы

_____ Гилка В.В.

«___» _____ 20 ____ г.

Нормоконтролер

Исполнитель

студент группы ПриИ-466

_____ Кузнецова А.С.

_____ Чупинин А. П.

«___» _____ 20 ____ г.

«___» _____ 20 ____ г.

Волгоград, 2024 г.

Аннотация

Настоящий документ является пояснительной запиской к выпускной квалификационной работе бакалавра на тему: ««Разработка мобильного приложения для контроля тренировочного процесса и приема биологически активных добавок спортсменами (Backend)»».

В работе обосновывается актуальность выбранной темы, проводится анализ предметной области, а также представлен анализ существующих аналогов и программных решений, описан процесс проектирования, разработки и тестирования разработанной программы.

Документ включает в себя страниц — 75, рисунков — 16, таблиц — 3, приложений — 3.

Ключевые слова: биологически активные добавки, модуль управления приёмом БАДов, мобильное приложение, Backend-часть мобильного приложения, база данных, Android, напоминания, push-уведомления, разработка мобильного приложения, система управления базами данных, алгоритмы обработки данных, масштабируемость, устойчивость.

Содержание

Введение	9
1 Анализ существующих подходов в контроле тренировочного процесса приема БАД спортсменами	11
1.1 Введение в исследование	11
1.2 Теоретический анализ	13
1.2.1 Определение основных терминов и понятий, связанных с предметной областью	13
1.2.2 Рассмотрение теоретических основ и принципов, лежащих в основе предмета исследования	14
1.2.2.1 Теоретические основы контроля тренировочного процесса	14
1.2.2.2 Теоретическая обоснованность управления приемом БАД	16
1.2.2.3 Технологические аспекты Backend-части мобильного приложения	17
1.2.2.4 Системы управления базами данных в контексте мобильного приложения	19
1.3 Существующие подходы к анализу	21
1.3.1 Анализ существующих методологий и подходов, применяемых в анализе исследуемых явлений	21
1.3.1.1 MyTherapy	21
1.3.1.2 MediSafe	23
1.3.1.3 Мои таблетки	25
1.4 Текущие тенденции и вызовы	27
1.4.1 Анализ текущих вызовов и проблем, стоящих перед исследователями и практиками в данной области	27
2 Предлагаемый процесс контроля тренировочного процесса и приема биологически активных добавок	29
2.1 Решение проблемы контроля спортивной активности и улучшение физической формы пользователя	29

2.2 Подбор и анализ компонентов для модуля, связанного с приёмом биологически активных добавок	30
2.3 Задачи модуля управления биологически активными добавками	32
2.4 Задачи для Backend-части приложения:	33
Выводы	35
3 Реализация модуля управления приема БАДов и Backend-части мобильного приложения	36
3.1 Требования к функциональным характеристикам	36
3.2 Требования к нефункциональным характеристикам	37
3.3 Варианты использования разрабатываемого сервиса	37
3.4 Общая структура устройства мобильного приложения	39
3.5 Проектирование Backend-части мобильного приложения	40
3.5.1 Проектирование схемы базы данных	40
3.5.2 Проектирование веб-сервиса	42
3.5.2.1 Взаимодействие веб-сервиса с базой данных	43
3.5.2.2 Проектирование аутентификации.	44
3.5.2.3 Реализация обработки запросов.	45
3.5.3 Реализация напоминаний	47
3.5.4 Диаграмма классов	49
3.6 Проектирование модуля управления приемом БАДов	50
3.7 Используемые языки разработки и программные средства	52
Выводы	52
4 Тестирование модуля, отвечающего за управление приема БАДов и Backend-части мобильного приложения.	55
4.1 Ручное тестирование модуля для управления работы с БАДами	55
4.2 Юзабилити-тестирование	61
4.3 Нагрузочное тестирование	64
Выводы	66
Заключение	67
Список использованных источников	69

Приложение А - Справка о результатах проверки выпускной квалификационной работы на наличие заимствований	73
Приложение Б - Техническое задание	74
Приложение В - Руководство системного программиста	75

Введение

Современный спорт характеризуется не только стремлением к новым рекордам и достижениям, но и возрастающими требованиями к комплексной подготовке атлетов. Это включает в себя не только физические тренировки, но и строгий контроль питания, а также использование биологически активных добавок (БАДов) [1]. Правильно подобранные БАДы могут значительно улучшить результаты тренировок, ускорить процессы восстановления после физических нагрузок и повысить общий тонус организма.

Однако, несмотря на потенциальные преимущества, использование БАДов требует строгого соблюдения режима дозировок и времени приема. Неправильное употребление добавок может не только снизить эффективность тренировок, но и привести к негативным последствиям для здоровья [2]. В этом контексте ключевую роль играет точный и систематический контроль за приемом БАДов, который должен осуществляться на регулярной основе.

Существующие приложения на рынке мобильных технологий часто фокусируются на отслеживании физической активности или контроле диеты, но редко предоставляют возможности для интегрированного управления и мониторинга всего комплекса задач, связанных с тренировочным процессом и приемом БАДов. Это создает определенные трудности для спортсменов, которым приходится использовать несколько приложений одновременно.

Именно здесь и проявляется актуальность разработки нового мобильного приложения, которое бы объединило в себе функциональность по контролю тренировок и приему биологически активных добавок в единой экосистеме. Такой подход позволит не только сделать процесс более удобным и эффективным, но и значительно повысить качество спортивной подготовки.

Цель работы заключается в совершенствовании процесса приема биологически активных добавок (БАДов) спортсменами и разработке

эффективного Backend-программного обеспечения для мобильного приложения, контролирующего тренировочный процесс.

Для достижения этой цели предполагается решение следующих задач:

- провести анализ текущих режимов приема БАДов среди спортсменов, чтобы понять их потребности и проблемы, с которыми они сталкиваются;
- определить ключевые функции режима приема БАДов;
- проектирование и разработка Backend-системы, которая будет обрабатывать данные о тренировках и приеме БАДов;
- разработать модуль для приложения контроля процесса тренировок, который обеспечит доступ к функциям управления приемом БАДов;
- провести комплексное тестирование всех компонентов системы для выявления и устранения возможных ошибок и несоответствий спецификациям.

Объектом исследования является процесс управления тренировочной активностью и приёмом биологически активных добавок спортсменами.

Предметом исследования является Backend-часть мобильного приложения, отвечающая за обработку данных, взаимодействие с сервером, и обеспечение функциональности по контролю тренировок и приему БАД.

Методы исследований. Для решения поставленных задач были использованы методы математического моделирования, системного анализа, программной инженерии, объектно-ориентированного программирования, технологии проектирования человеко-машинного взаимодействия, тестирования, и интеграции для разработки и оценки эффективности Backend-системы.

Практическая ценность работы заключается в том, что разработанный Backend-продукт не только предоставит спортсменам инструмент для учета и анализа приема БАД, но и способствует повышению эффективности тренировок, а также обеспечивает ценную аналитику для специалистов в области физической подготовки.

1 Анализ существующих подходов в контроле тренировочного процесса приема БАД спортсменами

1.1 Введение в исследование

Актуальность выбранной темы исследования обусловлена рядом факторов, которые подчеркивают ее важность и неотложность в современном контексте. В последние десятилетия наблюдается увеличение числа людей, стремящихся к активному образу жизни и заботе о своем здоровье. Этот тренд включает в себя спортивные занятия, правильное питание и использование биологически активных добавок (БАД).

Спорт становится неотъемлемой частью повседневной жизни многих людей. С каждым годом процент людей в России, занимающиеся спортом, растет. Так, в 2023 в России систематически занимаются спортом или физической культурой около 76 миллионов человек [3]. Спортсмены и любители физической активности активно внедряют инновационные подходы для улучшения своих тренировочных процессов. Множество людей, занимающиеся спортивной активностью, предпочитают заниматься самостоятельно, не пользуясь услугами специалистов, поэтому нуждаются в удобных инструментах для контроля тренировочного процесса, а также контроле приема биологически активных добавок, которые могут повлиять на прогресс в развитии спортсменов физических навыков.

В качестве инструмента, который поможет пользователю контролировать, тренировочный процесс могут выступать дневники в бумажном виде, умные часы и фитнес трекеры и различные мобильные приложения. Последнее является хорошим выбором, потому что оно позволяет удобно отслеживать и анализировать прогресс тренировок и прием БАДов в реальном времени, предоставляя напоминания и рекомендации. В мобильное приложение также возможно добавить интеграцию с другими

устройствами и платформами, обеспечивая комплексный подход к управлению здоровьем и фитнесом. Мобильные приложения становятся неотъемлемым инструментом для спортсменов, предоставляя им доступ к важной информации в режиме реального времени.

Так, разработка эффективного Backend-решения для мобильного приложения, ориентированного на контроль тренировочного процесса и прием БАД, становится ключевой задачей. Backend-часть обеспечивает стабильную работу приложения, обмен данными и взаимодействие с пользовательским интерфейсом. От Backend-части зависит безопасность хранения данных, своевременная синхронизация информации.

Выбранная тема исследования актуальна не только в контексте современных тенденций в физической активности и здоровье, но и представляет собой стратегически важный аспект для развития мобильных технологий в сфере спорта и заботы о здоровье. В данном контексте Backend-решение для мобильного приложения будет не только соответствовать текущим потребностям пользователей, но и способствовать прогрессивному развитию индустрии здоровья и фитнеса.

Разработка мобильного приложения – это многогранный процесс, который требует всестороннего понимания текущих тенденций и особенностей сочетания современных технологий с потребностями пользователей в сфере спорта и физической активности. Обзор предметной области является неотъемлемым этапом в процессе создания Backend-части приложения, так как изучение современных трендов в области спорта позволяет выявить актуальные потребности и предпочтения пользователей. Взаимодействие с новыми тенденциями в фитнес-индустрии и здоровом образе жизни становится ключевым элементом успешного мобильного приложения.

Также, понимание того, как пользователи взаимодействуют с мобильными устройствами и какие функциональности они ценят, определяет пользовательский опыт. Это важно для создания удобного и интуитивно

понятного интерфейса, что, в свою очередь, влияет на популярность и эффективность приложения.

Не менее важным является изучение успешных кейсов в данной области помогает выделить стратегии, которые привели к успеху других приложений, что может включать в себя функциональные особенности, маркетинговые подходы или инновационные решения, которые могут быть адаптированы к создаваемому мобильному приложению.

Наконец, ознакомление с текущими подходами и технологиями, используемыми в аналогичных приложениях, дает представление о том, какие инструменты и решения наилучшим образом соответствуют задачам Backend-части, что позволяет избежать повторения ошибок и выбрать оптимальные решения.

1.2 Теоретический анализ

1.2.1 Определение основных терминов и понятий, связанных с предметной областью

В данном разделе осуществляется теоретический анализ ключевых терминов и понятий, которые имеют прямое отношение к предметной области разработки мобильного приложения для контроля тренировочного процесса и приема биологически активных добавок (БАД).

Контроль тренировочного процесса представляет собой систематическое и целенаправленное воздействие на тренировочные активности спортсмена с целью достижения определенных спортивных результатов. Он включает в себя мониторинг физических нагрузок, адаптацию программ тренировок и оценку эффективности занятий.

Биологически активные добавки – это продукты, предназначенные для дополнения рациона питания и обогащения организма необходимыми веществами, такими как витамины, минералы, аминокислоты [4]. Они используются спортсменами для повышения эффективности тренировок, улучшения восстановления и поддержания общего здоровья.

Backend в мобильных приложениях представляет собой серверную часть мобильного приложения, отвечающую за обработку данных, бизнес-логику, и взаимодействие с базой данных [5]. В контексте приложения для контроля тренировочного процесса Backend обеспечивает передачу, хранение и обработку данных о тренировках, биологически активных добавках, времени их употребления, а также взаимодействие с функциональностью приложения.

Таким образом, этот теоретический анализ служит основой для дальнейшего понимания и использования данных понятий в разработке и исследовании. Разъяснение ключевых терминов уточняет понимание предметной области и обеспечивает единое теоретическое основание для дальнейших шагов исследования.

1.2.2 Рассмотрение теоретических основ и принципов, лежащих в основе предмета исследования

1.2.2.1 Теоретические основы контроля тренировочного процесса

Контроль тренировочного процесса в сфере спорта – это комплексная задача. Для ее решения необходимо понимание физиологических особенностей организма спортсменов, применение современных методов мониторинга физической активности и эффективные стратегии оценки затрат энергии. Рассмотрим теоретические основы, лежащие в основе этого важного аспекта тренировочного процесса.

Контроль тренировочного процесса начинается с понимания, каким образом физиологические системы организма реагируют на физическую активность. Рассмотрение основных физиологических процессов, таких как работа сердечно-сосудистой системы, дыхательная функция, и обмен веществ, является ключевым для эффективного контроля тренировочного процесса.

Понимание, сколько энергии тратится во время тренировок, играет важную роль в оптимизации тренировочного процесса. Различные методы оценки затрат энергии, такие как метаболический эквивалент, позволяют более точно анализировать интенсивность тренировок и разрабатывать персонализированные программы.

Использование современных технологий, таких как носимые устройства (фитнес-трекеры, умные часы), предоставляет возможность непрерывного мониторинга физической активности. Анализ собранных данных о шагах, расстоянии, частоте сердечных сокращений и других параметрах позволяет тренерам и спортсменам следить за прогрессом и корректировать тренировочные нагрузки.

Разработка и применение алгоритмов для анализа данных, собранных в процессе мониторинга, является важным теоретическим аспектом контроля тренировочного процесса. Алгоритмы должны учитывать индивидуальные особенности спортсменов, а также динамику изменения показателей физической активности.

Измерение пульса является классическим методом контроля физической нагрузки. Теоретические основы включают понимание связи между пульсом и интенсивностью тренировки, а также выбор оптимальных зон пульса для достижения конкретных целей тренировок.

Оценка затрат энергии должна учитывать не только количество сжигаемых калорий, но и эффективность тренировки в контексте достижения поставленных целей. Теоретические аспекты разработки стратегий оценки эффективности тренировок помогают определить оптимальные подходы для каждого спортсмена.

Разбор теоретических основ контроля тренировочного процесса предоставляет необходимый фундамент для разработки эффективной Backend-части мобильного приложения, способной предоставлять точные и персонализированные данные по физической активности спортсменов.

1.2.2.2 Теоретическая обоснованность управления приемом БАД

Управление приемом биологически активных добавок (БАД) в контексте тренировочного процесса требует глубокого понимания основных принципов питания, которые играют важную роль в поддержании физической активности и восстановлении организма после тренировок. Давайте рассмотрим теоретические аспекты, ориентированные на оптимизацию приема БАД для спортсменов.

Теоретическая обоснованность управления приемом БАД начинается с изучения физиологии питания для спортсменов. Это включает в себя понимание основных макро- и микроэлементов, необходимых для эффективного функционирования организма в условиях увеличенной физической активности.

Теоретический анализ направлен на определение потребностей спортсменов в витаминах и минералах, которые могут быть усилены тренировочной нагрузкой. Это включает в себя изучение влияния различных веществ на энергетический обмен, образование костной ткани, а также процессы восстановления.

Теоретическая обоснованность управления приемом БАД также включает в себя изучение роли белков и аминокислот в спортивном питании. Различные типы тренировок и виды спорта требуют разнообразных количеств белка для обеспечения роста мышц, восстановления и достижения оптимальной производительности.

Теоретический аспект анализа включает в себя рассмотрение того, какие биологически активные добавки оказывают на физическую активность. Это включает в себя изучение эффектов на выносливость, силу, скорость восстановления и снижение риска травм.

Управление приемом БАД также требует теоретической обоснованности индивидуализации рекомендаций. Изучение особенностей каждого спортсмена, его метаболизма, стилей тренировок и целей позволяет разрабатывать персонализированные стратегии приема добавок.

Теоретический аспект включает в себя понимание биохимических процессов в организме, в том числе усвоения и обработки добавок. Это позволяет разрабатывать стратегии приема, максимально эффективные для каждого спортсмена.

Теоретическая обоснованность управления приемом БАД является критическим элементом для разработки эффективного Backend-решения мобильного приложения, способного предоставлять рекомендации по приему биологически активных добавок, соответствующие индивидуальным потребностям каждого спортсмена.

1.2.2.3 Технологические аспекты Backend-части мобильного приложения

Разработка Backend-части мобильного приложения требует тщательного рассмотрения различных технологических аспектов, чтобы обеспечить эффективное, масштабируемое и безопасное функционирование. Рассмотрим технологические аспекты Backend-части мобильного приложения подробнее.

Одним из главных аспектов проектирования Backend-части как мобильного приложения, так и веб-приложения является выбор СУБД (Системы Управления Базами Данных) [6]. Различают реляционные и

нереляционные СУБД. Реляционные базы данных представляют собой совокупность строго структурированных связанных таблиц, в котором столбец – это поле определенного типа, а строка представляет собой отдельную запись. В свою очередь, нереляционные базы данных содержат динамическую схему данных, которая может меняться в любой момент времени. Из этого можно сделать вывод, что реляционные базы данных подходят для структурированных данных, в то время как нереляционные обеспечивают гибкость при хранении данных, но их использование может быть недостаточно эффективным.

Важным аспектом при проектировании Backend-части приложения является проектирование алгоритмов обработки данных и оптимизация запросов, достигающаяся за счёт тщательного анализа запросов к базе данных для оптимизации их выполнения, особенно при работе с объемными данными, например, данными о тренировках и приеме БАД.

При проектировании Backend-части мобильного приложения следует уделить внимание предоставлению возможностей по горизонтальному и вертикальному масштабированию. Для этого необходимо заложить при проектировании методы, обеспечивающие возможность увеличения производительности системы при росте объема данных и количества пользователей.

Необходимо также ответственно отнестись к выбору методов резервного копирования для наиболее эффективных стратегий резервного копирования данных в целях предотвращения потери информации. Вероятно, необходимо озаботиться и внедрением принципов шифрования данных для обеспечения конфиденциальности пользовательской информации, а также применении методов обеспечения целостности данных, чтобы гарантировать правильность и достоверность информации.

С точки зрения Backend-части мобильного приложения необходимо предусмотреть реализацию механизмов контроля доступа для защиты данных от несанкционированного доступа.

Также требуется разработка методов обновления данных в режиме реального времени для обеспечения актуальности информации на стороне клиента и использование технологий оптимизации сетевого взаимодействия для снижения задержек при передаче данных.

В целях повышения производительности возможно применение механизмов кэширования для ускорения доступа к часто используемым данным и постоянное улучшение запросов и использование индексов для оптимизации производительности системы.

Одной из главных задач при проектировании является разработка открытых API для удобной интеграции Backend-части с внешними сервисами, такими как информационные платформы о здоровье и базы данных питательных добавок.

Теоретическая обоснованность технологических аспектов Backend-части мобильного приложения играет ключевую роль в разработке надежной, производительной и безопасной системы, способной эффективно обрабатывать данные, обеспечивать масштабируемость и соответствовать современным требованиям информационной безопасности.

1.2.2.4 Системы управления базами данных в контексте мобильного приложения

В контексте мобильного приложения, где требуется эффективная обработка данных тренировок и приема БАД, рассмотрим различные типы баз данных. Основными вариантами являются реляционные базы данных, представленные PostgreSQL [7], и NoSQL решения, такие как MongoDB. Выбор PostgreSQL перед другими реляционными базами данных обоснован тем, что PostgreSQL имеет открытый исходный код, активное сообщество, а

также тем, что PostgreSQL зарекомендовала себя, как надежная и безопасная база данных.

MongoDB - это современная, документно-ориентированная база данных, которая разработана для удобного хранения и управления большими объемами данных. Она является одной из самых популярных NoSQL баз данных благодаря своей гибкости, масштабируемости и производительности, а также имеет активное сообщество пользователей и разработчиков, что обеспечивает обширную поддержку, регулярные обновления и богатый выбор инструментов для разработки. Реляционные базы данных хорошо подходят для структурированных данных, в то время как NoSQL базы данных предоставляют большую гибкость при работе с изменяющимися данными, что важно для приложения, учитывающего разнообразные параметры тренировочного процесса и приема БАД.

Важным критерием при выборе системы управления базами данных для мобильного приложения является производительность и скорость обработки данных. PostgreSQL обеспечивает высокую производительность при выполнении сложных запросов, что сказывается на скорости выполнении операций, связанных с данными о тренировках, приеме БАДов, а также на быстродействии работы мобильного приложения. Помимо этого, PostgreSQL, как и большинство популярных реляционных баз данных, поддерживает принципы ACID (атомарность, согласованность, изолированность, долговечность) [8], что обеспечивает стабильность операций и предотвращает потерю или повреждение информации. Это является критически важным для многих приложений, особенно в финансовой или бизнес-сфере. MongoDB, с другой стороны, обладает высокой масштабируемостью и способностью эффективно обрабатывать неструктурированные данные, что может быть полезно при создании приложения, управляющим прием БАДов спортсменом.

Таким образом, на основе теоретического анализа, предпочтительным выбором для Backend-части мобильного приложения является использование PostgreSQL для данных о тренировках и для данных о приеме БАД.

1.3 Существующие подходы к анализу

1.3.1 Анализ существующих методологий и подходов, применяемых в анализе исследуемых явлений

Стоит отметить, что приложений, специализирующихся на трекинге БАДов практически нет. Существуют приложения, специализирующие на трекинге употребления лечебных средств и все они имеют примерно один и тот же функционал. Проведём ревью существующих аналогов мобильных приложений по этой тематике:

1.3.1.1 MyTherapy

MyTherapy – это мобильное приложение, которое напоминает о приеме лекарств в назначенное пользователем время. Приложение имеет минималистичный дизайн, интерфейс приложения не перегружен лишними деталями и удобен для пользователя [9]. На рисунке 1 продемонстрирован внешний вид экрана со списком напоминаний.

Помимо напоминаний о приеме лекарств, приложение позволяет сделать напоминания о мероприятии, измерении, контроле симптомов. Настройка напоминаний довольно гибкая: можно выбрать единицу измерения употребления лекарственных средств, время приема, период (1 раз в день, несколько раз в день, конкретные дни недели и так далее). Из минусов: приходится постоянно создавать новое лекарственное средство вместо того, чтобы сохранять их где-то и иметь возможность повторно создавать напоминания с этим же лекарственным средством.

Приложение также содержит раздел статистики, в котором пользователь может просмотреть дни приема лекарств, а также изменение биомедицинских показателей. Рисунок 2 демонстрирует внешний вид данного раздела.

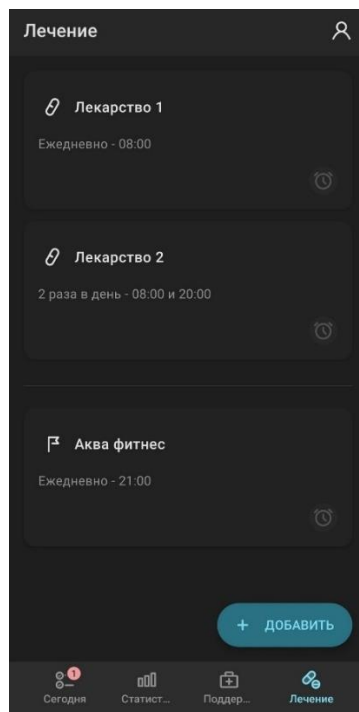


Рисунок 1 — Список напоминаний

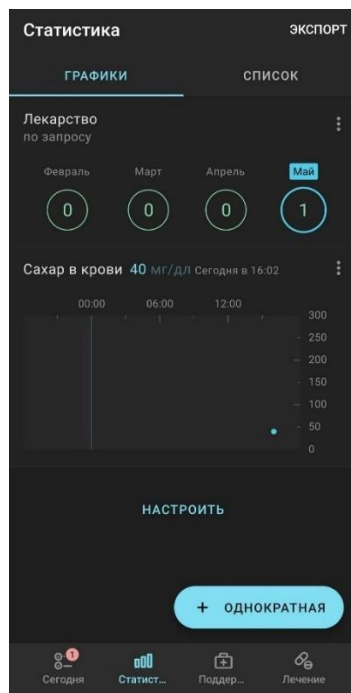


Рисунок 2 — Раздел статистики

К сожалению, в приложении отсутствует дневник, в котором пользователь может оставлять заметки о своем здоровье, что является небольшим упущением создателей проекта.

В ходе изучения составляющих приложения MyTherapy были выявлены следующие достоинства и недостатки.

Достоинства:

- приятный дизайн;
- удобный пользовательский интерфейс;
- гибкая настройка напоминаний;
- возможность просматривать статистику о приеме препаратов и изменении биомедицинских показателей.

Недостатки:

- нет возможности повторно использовать ранее созданные записи о препаратах в напоминаниях;
- отсутствует дневник здоровья;
- нет возможности модифицировать приложение.

1.3.1.2 MediSafe

Приложение MediSafe создано для трекинга приема лекарственных препаратов. Приложение имеет современный дизайн, но при этом интерфейс не совсем дружелюбен для пользователя [10]. Список медикаментов и связанных с ними напоминаний представлен на рисунке 3.

В приложение довольно богатый функционал, но ощущается приложение перегруженным. Чтобы просто добавить одно напоминание необходимо пролистать около 4–5 экранов.

В MediSafe присутствует гибкая настройка напоминаний о приеме препаратов (каждые X дней, по определенным дням неделям и так далее).

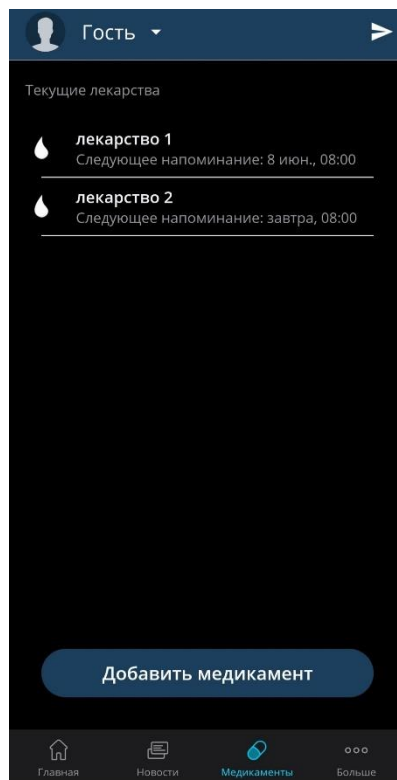


Рисунок 3 — Список напоминаний

Созданные ранее лекарства сохраняются и есть возможность использовать одно и то же лекарство в нескольких напоминаниях. Приложение также содержит раздел заметок и при этом еще и содержит раздел трекинга здоровья, который по функционалу ничем не отличается от заметок.

Проанализировав, составляющие приложения MediSafe были выделены достоинства и недостатки.

Достоинства:

- современный дизайн;
- возможность добавления собственных препаратов;
- наличие дневника здоровья;
- гибкие настройки напоминаний.

Недостатки:

- перегруженный интерфейс пользователя;

- множество лишних функций, некоторые из которых дублируют друг друга (заметки, трекинг здоровья);
- нет возможности модифицировать приложение.

1.3.1.3 Мои таблетки

Мои таблетки — это еще одно приложение, нацеленное на создание напоминаний о приеме лекарственных препаратов. Приложение имеет приятный минималистичный дизайн. На рисунке 4 продемонстрирован главный экран приложения [11].

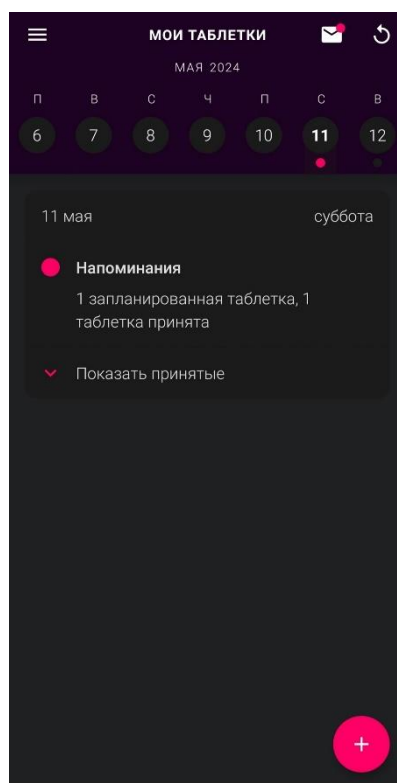


Рисунок 4 — главный экран

Функциональных возможностей в приложении не так много. Можно создавать напоминания о приеме лекарств, а также создавать заметки. Кроме того, создание заметок отсутствует в бесплатной версии. Для возможности создания заметок придется покупать ежемесячную подписку. Таким образом,

и так не богатый функционал приложения становится еще меньше, а впечатление потенциальных пользователей о продукте может испортиться.

Стоит заметить, что создание напоминаний о приеме лекарств имеет несколько недостатков. Приложение имеет не такие гибкие возможности создания напоминаний, как у ближайших конкурентов. Помимо этого, у приложения есть проблема с лекарствами. У пользователя нет возможности выбрать список уже созданных лекарств: для каждого напоминания приходится создавать лекарство снова и снова.

Таким образом, анализ составляющих приложения “Мои таблетки” помог выявить достоинства и недостатки данного приложения.

Достоинства:

- современный дизайн;
- присутствует функция создания заметок/дневник здоровья;

Недостатки:

- в бесплатной версии отсутствует часть функций;
- нет возможности добавления собственных лекарственных средств;
- нет возможности модифицировать приложение.

1.3.2 Основные преимущества и недостатки приложений для контроля приема биологически активных добавок.

В данной работе были рассмотрены различные приложения, предназначенные для контроля приема биологически активных добавок. Эти приложения обладают различными функциями и характеристиками, которые могут существенно влиять на удобство и эффективность их использования. Основные преимущества и недостатки этих приложений сведены в таблицу 1, которая представлена ниже.

Таблица 1 - Таблица сравнения аналогов

Критерий сравнения	MyTherapy	MediSafe	Мои таблетки
Неперегруженный интерфейс	+	-	+
Наличие дневника здоровья	-	+	+/-
Повторное использование собственных лекарств	-	+	-
возможность интеграции и доработки	-	-	-
Доступность к полному функционалу по бесплатной версии	+	+	+

1.4 Текущие тенденции и вызовы

1.4.1 Анализ текущих вызовов и проблем, стоящих перед исследователями и практиками в данной области

В настоящее время исследователи и практики в области разработки мобильных приложений для контроля тренировочного процесса и управления приемом биологически активных добавок (БАД) сталкиваются с рядом значительных вызовов и проблем, которые оказывают влияние на эффективность и функциональность создаваемых систем.

Одним из главных вызовов является сложность интеграции данных о тренировках и приеме БАД из различных источников. Это включает в себя необходимость синхронизации с фитнес-трекерами, умными весами, медицинскими устройствами и другими сенсорами. Различные устройства предоставляют данные в разном формате, используют разные протоколы связи и требуют разработки эффективных механизмов обмена данными.

Необходимость обеспечения стабильной и надежной синхронизации данных между различными источниками становится одним из ключевых технических аспектов.

С ростом объема персональных данных о здоровье и физической активности пользователей возникает серьезная проблема обеспечения высокого уровня конфиденциальности и безопасности этих данных. Системы, собирающие и обрабатывающие медицинскую информацию, подвергаются повышенным требованиям в области защиты данных. Это включает в себя соблюдение законодательства о конфиденциальности, использование современных методов шифрования, анонимизацию данных и реализацию прочих мер безопасности для предотвращения утечек и несанкционированного доступа.

Разработка Backend-части мобильных приложений в области контроля приема биологически активных добавок сопряжена с рядом вызовов и проблем, которые требуют особого внимания и комплексного подхода. Эти вызовы включают обеспечение безопасности данных, масштабируемость системы, высокую производительность и надежность, а также соблюдение нормативных требований и стандартов. Решение данных проблем является неотъемлемой частью процесса разработки и требует внедрения современных технологий и практик.

Внедрение технических инноваций играет ключевую роль в решении возникающих проблем при разработке Backend-части мобильных приложений. Это включает использование передовых технологий и инструментов. Облачные технологии, предоставляют масштабируемую инфраструктуру, которая позволяет легко адаптироваться к увеличению числа пользователей и объема данных. Безопасность данных является критически важным аспектом при разработке Backend-части мобильных приложений. Соблюдение строгих мер контроля за обработкой персональных данных является обязательным для соответствия нормативным требованиям.

2 Предлагаемый процесс контроля тренировочного процесса и приема биологически активных добавок

2.1 Решение проблемы контроля спортивной активности и улучшение физической формы пользователя

Как уже отмечалось выше, на мобильном рынке, в частности, на рынке смартфонов с операционной системой Android, уже существуют различные приложения, предназначенные отдельно для контроля тренировок и отдельно для отслеживания приема биологически активных добавок (БАДов). Эти приложения позволяют пользователям эффективно управлять своими тренировками и отслеживать прием БАДов, а также их влияние на организм. Однако, использование двух отдельных приложений может быть неудобным для пользователей из-за необходимости постоянного переключения между ними и разброса информации.

Решение данной проблемы заключается в интеграции двух ключевых аспектов спортивного образа жизни – тренировок и приема БАДов – в единое мобильное приложение, что позволит пользователям удобно отслеживать свой спортивный прогресс и контролировать прием БАДов на протяжении всего времени. Путем объединения этих двух функциональных модулей в одном приложении мы создаем среду, где спортсмены могут получать полную картину своей спортивной деятельности и ее влияния на их здоровье и спортивные результаты.

Используя приложение, совмещающие в себе функции по контролю тренировочного процесса, а также по контролю употребления биологически активных добавок, спортсмены получают возможность отслеживать свой прогресс и результаты в тренировочном процессе, а также увидеть, как их занятия и прием БАДов влияют на их общее состояние и физическую форму. Это поможет им лучше понять, какие изменения в их режиме тренировок и

диете могут привести к желаемым результатам. Таким образом, пользователи смогут самостоятельно адаптировать свои методы тренировок и рацион питания, основываясь на собственном опыте и наблюдениях.

Моя задача состоит в создании модуля по контролю приёма БАДов и Backend-части приложения, обеспечивающую взаимодействие пользователей с данными о приеме биологически активных добавок и контроле тренировочного процесса. Модуль БАД будет предоставлять возможность пользователям вводить информацию о принятых БАДах, включая их название, дозировку, рекомендации по применению и время приема. Пользователь будет иметь возможность создавать напоминания о приёме того или иного БАДа в определенное время, а приложение будет его оповещать об этом с помощью систем уведомлений. Также будет возможность делать заметки, вести дневник здоровья, по которому пользователь может отслеживать изменения своего организма.

Backend-сервер будет обрабатывать запросы от мобильных устройств, осуществляя валидацию и сохранение введенных данных о тренировках и приеме БАДов. Кроме того, серверная часть приложения будет ответственна за обеспечение безопасности данных, используя соответствующие механизмы шифрования и аутентификации.

2.2 Подбор и анализ компонентов для модуля, связанного с приёмом биологически активных добавок

Модуль управления биологически активными добавками (БАДами) представляет собой важную часть функциональности нашего приложения. Этот модуль состоит из трех основных компонентов: создание записей о БАДах, добавление заметок к ним и настройка напоминаний о приеме. Каждый из этих компонентов играет ключевую роль в обеспечении

пользователей всей необходимой информацией для эффективного контроля над приемом биологически активных добавок.

В модуле управления биологически активными добавками (БАДами) каждая запись о БАДе содержит несколько важных характеристик.

Название БАДа позволяет пользователю быстро идентифицировать конкретную добавку. Описание предоставляет дополнительную информацию о свойствах, преимуществах или особенностях данной БАДы. Помимо этого, каждая запись о БАДе также включает тип добавки, такой как порошок, таблетки, капсулы и другие формы, которые могут быть представлены в приложении.

Дозировка указывает, сколько БАДа следует принимать за один раз или за один прием, обеспечивая точное соблюдение рекомендаций. Время приема позволяет пользователю указать оптимальное время для употребления БАДа, что может быть важно для достижения максимальной эффективности и избежание негативных последствий от передозировки, которые могут привести к ухудшению здоровья. Периодичность приема определяет, насколько часто следует принимать данную добавку. Так как часто дозировка зависит от веса, а время и периодичность приема необходимо для создания напоминаний, то соответственно эти пункты будут указываться не у самой биологически активной добавки, а у напоминания, который связан с добавкой.

Заметки в модуле управления БАДами представляют собой инструмент для пользователей ведения своего дневника здоровья. Основные характеристики заметок включают время написания и, собственно, сам текст записи.

Пользователь может использовать заметки для фиксации своих биомедицинских характеристик, таких как уровень энергии, настроение, физическое состояние или любые другие параметры, которые считает важными для отслеживания своего здоровья и благополучия. Эти записи помогают пользователям следить за изменениями в своем организме, выявлять

тенденции и анализировать влияние употребления БАДов на их общее состояние.

Напоминания в модуле управления БАДами являются важным инструментом для пользователей, обеспечивая им регулярный прием необходимых добавок. Каждое напоминание привязано к определенному БАДу и включает в себя несколько ключевых характеристик. Во-первых, это время приема, которое указывает, когда необходимо принять добавку. Количество приемов определяет общее число приемов данной добавки, который он определил для себя в течение курса употребления БАДов. Периодичность напоминаний задает интервал между приемами, такой как каждый день, через день, каждые два дня и т. д. Наконец, дозировка добавки зависит от ее типа (грамм, количество таблеток и т. д.) и предоставляет информацию о количестве, которое необходимо принять в каждом приеме в соответствии с рекомендациями специалистов. Эти напоминания помогают пользователям следить за регулярным употреблением БАДов, что способствует достижению желаемых результатов и поддержанию здоровья.

2.3 Задачи модуля управления биологически активными добавками

В модуле управления приемом биологически активных добавок (БАДов) мы сталкиваемся с несколькими ключевыми задачами.

Первая задача — это реализация функционала напоминаний. Этот функционал должен предоставлять пользователям информацию о времени приема, количестве приемов в день и периодичности напоминаний для каждой добавки. Например, пользователь может установить напоминание принимать определенную добавку два раза в день, утром и вечером, каждый день или раз в неделю.

Вторая задача - создание системы отправки и принятия уведомлений о необходимости приёма биологически активной добавки. Пользователям необходимо получать уведомления в соответствии с установленными параметрами напоминаний. Это позволит им не забывать о приеме БАДов и следовать рекомендациям по их употреблению, что может повлиять на их физическую активность и улучшить спортивные результаты.

Третья задача – реализация дневника здоровья, которой представляет собой список заметок пользователя. В функционале дневника приема БАДов пользователь может фиксировать свои биомедицинские показатели после приема добавок. Это поможет пользователю анализировать изменения своего организма и оценивать эффективность употребления биологически активных добавок.

Последняя задача включает разработку удобного пользовательского интерфейса [12] для модуля управления приемом биологически активных добавок. Эта задача включает в себя создание понятного дизайна, который позволит пользователям легко взаимодействовать с функционалом приложения. Также важно обеспечить удобную и интуитивно понятную навигацию по приложению, которая позволит быстро находить нужные функции и информацию. Важным аспектом является адаптивность интерфейса под разные типы устройств и экраны, обеспечивая комфортное использование приложения на большинстве мобильных Android устройств.

2.4 Задачи для Backend-части приложения:

При разработке Backend-части приложения необходимо решить ряд ключевых задач, список которых представлен ниже.

Первая задача – Создание HTTP-обработчиков, которые необходимы для взаимодействия пользователя (в нашем случае через мобильное

приложения) с базой данных для создания, редактирования, удаления и получения всех компонентов, необходимых для контроля тренировочного процесса и приема БАДов, а также необходимые для аутентификации пользователя.

Вторая задача – Реализация механизмов безопасности и аутентификации. Реализация механизмов безопасности является критически важной для защиты данных пользователей, что является для многих критически важным пунктом для выбора продукта. Это включает в себя реализацию протоколов шифрования, а также механизмов аутентификации и авторизации пользователей [13]. Авторизация необходима для того, чтобы каждый пользователь имел права доступа для получения и редактирования компонентов, созданием которых занимался конкретно он и никакой другой пользователь не имел право изменять или удалять компоненты, не принадлежащие ему.

Третья задача – Разработка структуры базы данных. Необходимо разработать такую структуру, которая была бы удобна для хранения информации, не содержала избыточной информации. Для этого необходимо провести нормализацию базы данных и привести ее как минимум к третьей нормальной форме. Кроме того, необходимо обеспечить быстрый отклик на получение данных, чтобы пользователь мог получить комфортный и приятный опыт использования мобильного приложения без долгого ожидания получения информации о тренировках, БАДах и так далее.

Четвертая задача – Реализация системы уведомлений и напоминаний. Важно разработать механизмы для отправки уведомлений на мобильные устройства пользователя о приеме БАДов в указанное пользователем время используя push-уведомления.

Выводы

При анализе компонентов проекта мы выделили ключевые аспекты, которые определяют функционал и эффективность системы. В части управления приемом биологически активных добавок (БАДов) мы обсудили важность ведения дневника приема добавок, который позволяет пользователям систематизировать информацию о своем здоровье и контролировать употребление БАДов. Также мы выделили необходимость создания системы напоминаний, которая помогает пользователям следить за регулярным приемом добавок.

Важными аспектами Backend-части приложения являются безопасность и аутентификация, которые обеспечивают защиту данных пользователей. Мы подчеркнули значимость эффективной структуры базы данных для хранения информации о приеме БАДов, обеспечивая ее масштабируемость и надежность.

Исследование позволило выявить ключевые задачи и аспекты, которые необходимо учесть при дальнейшей разработке и реализации проекта. Ведение дневника приема добавок и система напоминаний являются важнейшими функциями для пользователей, обеспечивая систематизацию и контроль за приемом БАДов. Безопасность и аутентификация данных пользователей критически важны для защиты информации и повышения доверия к приложению. Эффективная структура базы данных обеспечивает надежность и масштабируемость системы. Интеграция с другими медицинскими приложениями и продуманный пользовательский интерфейс делают систему комплексной и удобной в использовании. Учитывая эти аспекты, мы можем создать эффективное и надежное приложение, которое будет удовлетворять потребности пользователей и способствовать улучшению их здоровья.

3 Реализация модуля управления приема БАДов и Backend-части мобильного приложения

3.1 Требования к функциональным характеристикам

Backend-часть мобильного приложения для контроля тренировочного процесса и приема биологических добавок должна обеспечивать возможность выполнения перечисленных ниже функций:

- обеспечить возможность регистрации спортсмена в приложении;
- аутентификации спортсмена в приложении;
- управление учетным записями спортсменов;
- безопасность данных пользователей;
- обеспечить возможность создания, редактирования, удаления упражнений;
- создание, изменение, удаление персонализированных тренировочных программ;
- возможность отправки уведомлений и напоминаний о предстоящих тренировках, приеме БАД.

Модуль БАДов во Frontend-части приложения мобильного для контроля тренировочного процесса и приема биологических добавок должна обеспечивать возможность выполнения перечисленных ниже функций:

- создание, изменение, удаление БАДов;
- создание и удаление напоминаний о приеме БАДов;
- прием и уведомления пользователя о том, что наступило время для приема биологически активных добавок;
- создание, изменение, удаление записей в дневнике здоровья.

3.2 Требования к нефункциональным характеристикам

Модуль БАДов во frontend-части приложения мобильного для контроля тренировочного процесса и приема биологических добавок должна обеспечивать возможность выполнения перечисленных ниже функций:

- процессор не ниже Snapdragon 450 8х Cortex-A53 1,8 ГГц;
- оперативная память не менее 2 Гб;
- операционная система Android 8.0 и выше (посмотреть в Anroid).

3.3 Варианты использования разрабатываемого сервиса

Диаграмма вариантов использования пользователем модуля для управления приема БАДов в мобильном приложении представлена в соответствии с рисунком 5.

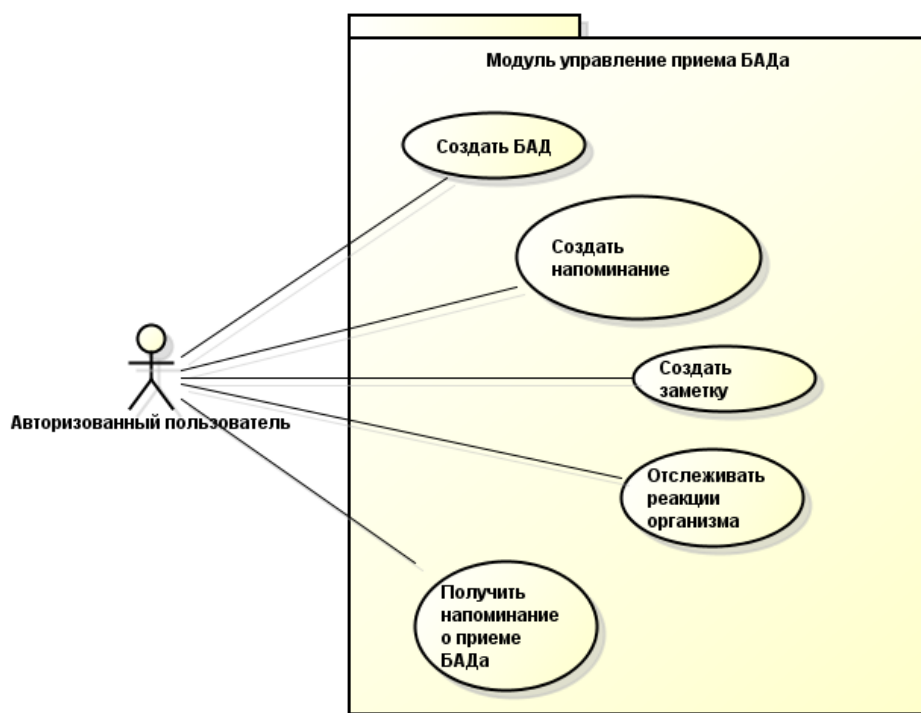


Рисунок 5 — Диаграмма вариантов использования навыка

Сценарий создания нового БАДа/напоминания/заметки:

- пользователь переходит в модуль управления приема БАДами в мобильном приложении;
- пользователь нажимает на кнопку добавления БАДа/напоминания/заметки;
- пользователь заполняет необходимые поля;
- пользователь нажимает на кнопку сохранения;
- мобильное приложение отправляет запрос на сервер на создание нового БАДа/напоминания/заметки;
- сервер проверяет корректность авторизацию пользователя и корректность введенных данных;
- сервер сохраняет БАД/напоминание/заметки в базе данных.

Сценарий получения напоминаний о приеме БАДов:

- сервер ищет все напоминания, которые должны быть выполнены в текущее время;
- сервер проходит по напоминаниям и отправляет пользователю, что их создал уведомления;
- мобильное приложение получает сообщение о сервере;
- мобильное приложение выводит уведомление пользователю на мобильный экран;
- пользователь принимает нужный ему БАД.

Сценарий отслеживания реакции организма пользователя:

- пользователь переходит в модуль управления приема БАДами в мобильном приложении;
- пользователь переходит в раздел заметок;
- пользователь читает последние заметки о своем организме;
- пользователь отслеживает изменения в своем организме по этим заметкам и делает некоторые выводы;

3.4 Общая структура устройства мобильного приложения

Структура устройства мобильного приложения представлена в соответствии с рисунком 6.

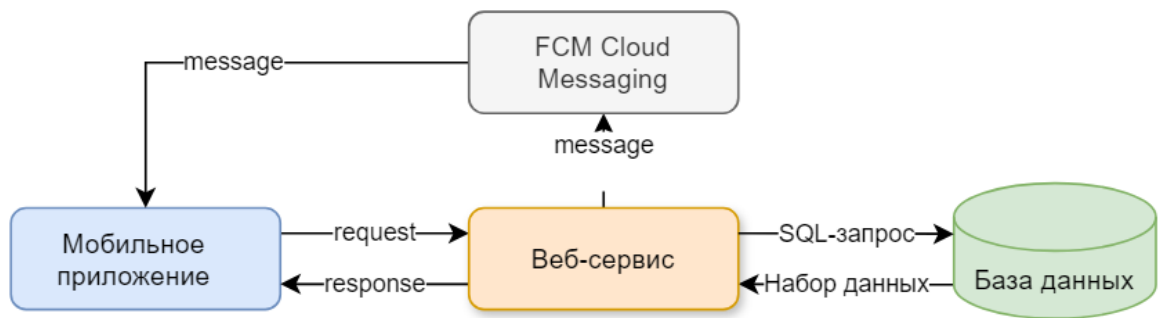


Рисунок 6 — Схема устройства мобильного приложения

Пользователь взаимодействует с мобильным приложением для создания, изменения, получения и удаления упражнений, тренировок, напоминаний и других компонентов, которые определяют контроль тренировочного процесса и приема биологически активных добавок. При добавлении новой записи или ее обновлении мобильное приложение отправляет серверу POST и PUT запрос, в теле которого содержится JSON-файл со всеми необходимыми полями для создания/обновления. Для получения данных с сервера мобильное приложение отправляет GET-запросы, которые могут содержать параметры запроса. Например, для получения упражнений, отвечающих за одну группу мышц. Для удаления мобильному приложению необходимо отправлять DELETE-запрос с указанием id элемента, который необходимо удалить.

Перед обработкой запроса пользователя веб-сервис проверяет, авторизован ли пользователь и имеет ли он право на осуществление запроса. Далее, веб-сервис осуществляет валидацию данных. Далее, если все предварительные проверки были пройдены, веб-сервис связывается с базой

данных через SQL-запросы. В случае, если база данных выполнила запрос, веб-сервис оповещает пользователя об этом с помощью HTTP-ответа.

Отдельно необходимо кратко описать процесс отправки уведомлений-напоминаний о приеме биологически активных добавок. На сервере необходимо установить хронологический демон-планировщик задач, который через определенный промежуток времени (например, каждую минуту) запускает на веб-сервисе задачу, которая получает список всех напоминаний, время которых совпадает с текущим. Далее, веб-сервис отправляет сообщение FCM Cloud Messaging, FCM Cloud Messaging идентифицирует устройство, на которое необходимо отправить push-уведомление. FCM Cloud Messaging проверяет доступность устройства. Если устройство доступно, то FCM Cloud Messaging отправляет сообщение мобильному устройству, которое обрабатывает это сообщение и выводит уведомление пользователю на экран о необходимости приёма биологически активной добавки. Если устройство недоступно, то FCM Cloud Messaging сохраняет уведомление в своем хранилище до момента, пока устройство не станет доступно для отправки уведомлений.

3.5 Проектирование Backend-части мобильного приложения

3.5.1 Проектирование схемы базы данных

Проектирование Backend-части приложения необходимо было начать с проектирования схемы базы данных. На рисунке 7 представлена схема базы данных мобильного приложения по контролю тренировочного процесса и приема БАДов.

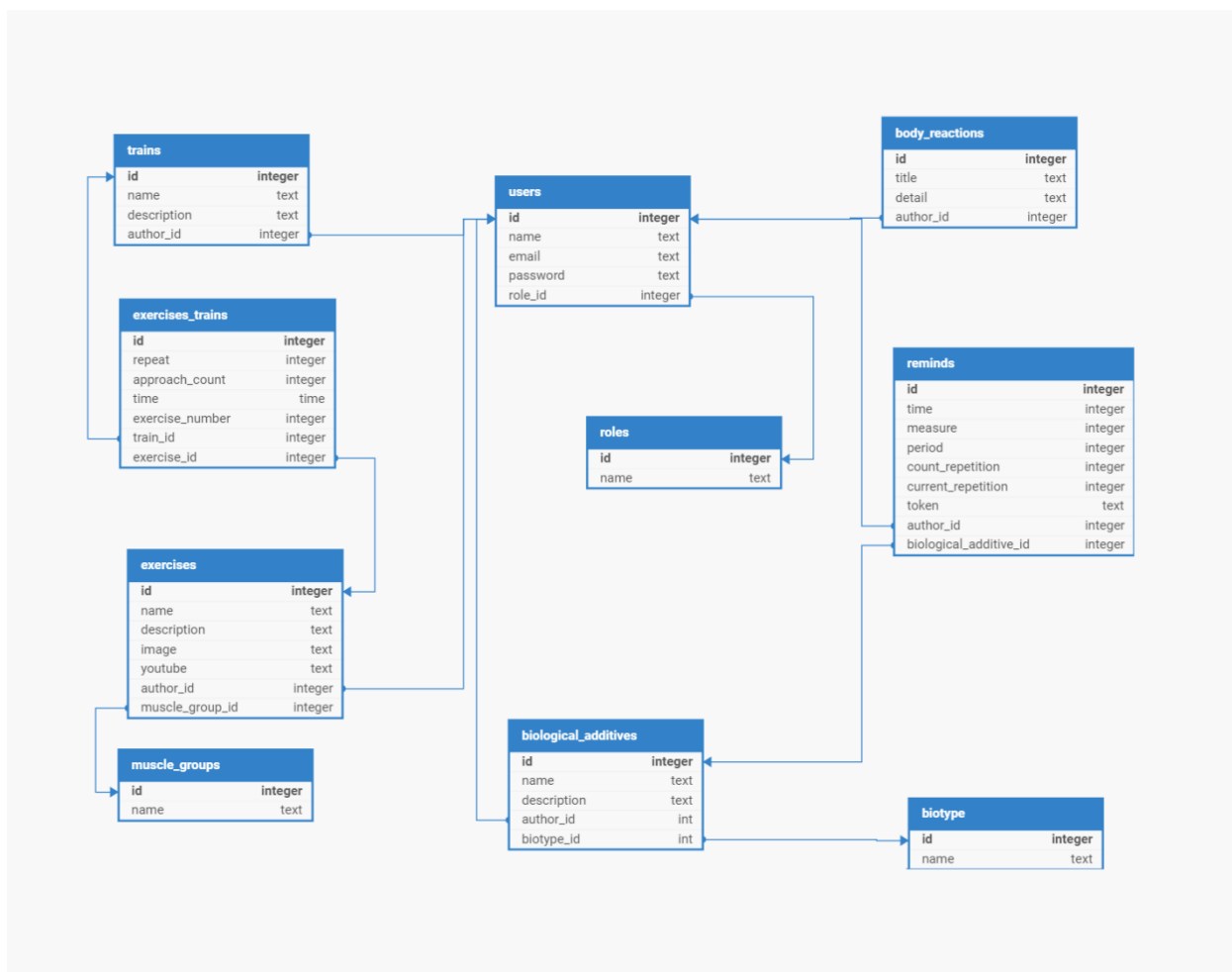


Рисунок 7 — Схема базы данных.

Как видно содержит несколько ключевых сущностей, а именно:

– Пользователь. Каждый элемент таблицы о пользователе должен содержать логин (должен быть уникальным), электронную почту (должна быть уникальной), пароль, а также роль пользователя, от которой может меняться возможности пользователя;

– Упражнение. Помимо названия, описания и создателя упражнения, необходимо хранить информацию о группе мышц, на развитие которых направлено упражнения, а также вспомогательные материалы для того, чтобы пользователю было проще понять правильное выполнение каждого упражнения, а именно ссылка на картинку, ссылка на видео, демонстрирующее выполнение упражнения;

– Тренировка. Сама тренировка содержит в себе только название и описание, но нам важна ее связь с упражнениями. Каждая связь тренировки и упражнения показывают, какой порядковый номер имеет упражнение в контексте тренировки, сколько времени необходимо на его выполнение, сколько повторов необходимо сделать, с какими весами работать;

– БАД. Содержит только информацию о названии и описании биологически активной добавки. В описание можно указать противопоказания врачей, основные компоненты и другие факты, связанные с конкретным БАДом, которые хочет отметить пользователь;

– Напоминания. Содержит информации о времени приема БАДа, о периоде (задается число дней, через которое будет отправляться напоминание о приеме БАДов), последняя дата приема БАДа, дозировка, текущее количество приемов и конечное число приемов;

– Дневник здоровья, который содержит описание изменений организма пользователя.

3.5.2 Проектирование веб-сервиса

В основе своего веб-приложения я решил использовать архитектуру REST API, так как это уже является устоявшимся стандартом для современных веб и мобильных приложений. Из основных ее плюсов можно отметить большое сообщество разработчиков, большой выбор инструментов, гибкость стандарта и простоту.

Написание Backend-части мобильной приложения было решено сделать на фреймворке NestJS [14, 15] и у этого решения есть несколько причин. Во-первых, модульная архитектура позволяет создавать структурированные приложения, использующие архитектуру MVC. Во-вторых, NestJs имеют большую базу различных библиотек, которые упрощают разработчику

создание веб-сервиса. Если же в самом NestJs отсутствует та или иная библиотека, то не составит труда найти иные библиотеки для Node.js и адаптировать их для работы в NestJs. Кроме того, так как NestJs является фреймворком для Node.js [16, 17], то необходимо отметить и наличие встроенного сервера, что дает возможность осуществлять быстрое и удобное развертывание веб-сервиса.

3.5.2.1 Взаимодействие веб-сервиса с базой данных

Для работы приложения с базами данных существует несколько подходов: использование ORM (Object-Relational Mapping) и прямые вызовы SQL-запросов из кода. И хотя с помощью прямых SQL-запросов мы имеем больший контроль и можем использовать специфичные функции операции SQL, что может быть полезно в специфичных случаях для улучшения производительности, но для нашего веб-сервиса было решено использовать ORM систему.

ORM – это технология, которая связывает базы данных с концепциями объектно-ориентированных языков программирования и получившая большую популярность. Большинство ORM имеют встроенные механизмы для предотвращения SQL-инъекций, чем помогает избежать несанкционированные получения информации, ее изменения и удаления. Многие ORM умеют работать с различными базами данных, таким образом мы имеем независимость от конкретной базы данных и можем в любой момент поменять базу данных. ORM дают более удобный способ взаимодействия веб-сервиса с базой данных через методы, а не через громоздкие SQL-запросы. Ну и наконец, многие ORM имеют удобные механизмы для миграции и обновления базы данных, что упрощает разработку и экономит время [18].

Для Node.js есть множество библиотек, которые предоставляют удобные ORM, но выбор наш остановился на Prisma [19]. Выбор обоснован тем, что у Prisma есть подробная документация по внедрению и использованию ее в проектах на Node.js. Prisma умеет работать с различными базами данных, как SQL (MySQL, PostgreSQL), так и NoSQL (MongoDB). Кроме того, Prisma имеет удобный инструмент для миграции базы данных (Prisma migrate), что позволяет быстро осуществить миграцию базы данных или обновить ее. Поскольку наш веб-сервер написан на NestJS + TypeScript, то бонусом для нашей реализации является то, что все результаты запросов Prisma полностью типизированы [20].

В качестве базы данных был выбран PostgreSQL. Все данные о подключении к базе данных были вынесены env-файл, что позволяет скрыть конфиденциальную информацию о логине, пароле, необходимых для подключения к базе данных, адрес расположения самой базы данных, а также делает код более чистым.

3.5.2.2 Проектирование аутентификации.

Для аутентификации в мобильном приложении будет использоваться JWT-токен. Выбор использования JWT-токена для реализации аутентификации обоснован рядом преимуществ. Прежде всего, использование криптографических алгоритмов обеспечивает защиту от подделки и несанкционированного доступа, что предоставляет высокий уровень безопасности. Не стоит забывать еще о том, что для работы JWT-токена нет необходимости хранить информацию о сессии, так как вся информация об аутентификации уже содержится в JWT-токене. Данное решение позволяет уменьшить нагрузку на сервер, а также может предоставить гибкость и масштабируемость.

Стоит отметить простоту использования JWT-токена, а также возможности его модификации. Например, можно расширить использование JWT, добавив токен обновления, который позволит обеспечить более долгосрочную аутентификацию пользователя.

Ограничения мобильных приложениях в использовании куков не позволит нам хранить JWT-токен в cookie-файлах и передавать их на сервер, поэтому необходимо использовать альтернативный подход. Чаще всего используется подход, когда JWT-токен передается в HTTP-запросах через заголовок `Authorization` с префиксом «Bearer». Этот способ удобен для мобильных приложений, поэтому именно он будет использоваться для аутентификации в мобильном приложении для контроля тренировочного процесса и приёма биологически активных добавок спортсменами.

Для реализации JWT-токена в приложении можно использовать библиотеку `Passport.js`. Она представляет собой стратегию аутентификации на основе `JSON Web Token`.

Библиотека `Passport.js` [21] предлагает лёгкость в установке и использовании, а также поддержку популярных стратегий аутентификации, включая JWT. Библиотека автоматически проверяет подпись и срок действия токена. Помимо этого, в `Passport.js` есть стратегии `OAuth` для внедрения аутентификации через сторонние приложения, что может помочь при дальнейшем развитии проекта. В `NestJs` присутствует пакет `@nestjs/passport` для работы с `Passport.js`.

3.5.2.3 Реализация обработки запросов.

Перед обработкой запроса веб-сервисом, веб-сервис должен сначала выполнить проверочные этапы: проверить, имеет ли пользователь право на этот запрос и провести валидацию переданных пользователем в запросе

данных. Только после этого запрос может быть выполнен. Рассмотрим каждый из этапов.

Для процесса регистрации и аутентификации нет необходимости осуществлять проверку аутентификации пользователя, так что это единственные запросы, на которых этап проверки прав пользователя не нужен. Для остальных запросов, например, для получения списка тренировок, необходима проверка аутентификации пользователя. Для этого мы воспользуемся Guard, которые включены в фреймворк NestJS. Guard — это механизм, который позволяет контролировать доступ к определённым частям приложения и ограничивает доступ к маршрутам или действиям только для определенных пользователей [22]. В нашем случае необходимо сделать ограничение, что только пользователи, которые успешно прошли аутентификацию, могут успешно получить данные о тренировках, упражнениях, БАДах, напоминаниях.

Следующий этап — валидация данных. Первое, что необходимо сделать — это проверка соответствия типов и корректность заполнения полей. Это необходимо, чтобы ограничить поступления на вход информационной системы заведомо ошибочных, неполных или неточных данных. Это помогает избежать ошибочных результатов работы системы, утраты данных и сбоев в работе систем. Для проверки корректности данных будет использоваться библиотека ClassValidator, которая поможет проверить типы данных, их корректность (например, соответствие пароля регулярному выражению) без написания лишнего кода, что позволяет уменьшить его количество и сделать его более чистым [23].

При валидации данных также необходимо проверить и соответствие данных базе данных. Так, если пользователь делает запрос на получение элемента, который он не создавал, система должна ему ответить отказом. Приведу еще один пример: при создании тренировки важно указать список упражнений, из которых тренировка состоит, так что здесь нужно осуществить

две проверки: наличие данных упражнений и имеет ли к ним доступ пользователь.

Если все проведенные проверки были пройдены, то веб-сервис выполняет запрос пользователя и возвращает ему HTTP-ответ.

3.5.3 Реализация напоминаний

Для того чтобы напоминания о приеме биологически активных добавок (БАДов) приходили пользователям, необходимо настроить возможность отправки сервером push-уведомлений клиенту. Push-уведомления являются важной функцией, которая помогает поддерживать регулярность приема добавок и обеспечивает пользователям своевременные напоминания. Для отправки push-уведомлений с сервера используется сервис Firebase Cloud Messaging (FCM), созданный компанией Google [24]. Firebase Cloud Messaging (FCM) – это мощный и удобный инструмент, который позволяет разработчикам отправлять уведомления на устройства, работающие под управлением операционных систем Android и iOS, а также на веб-приложения. FCM является частью экосистемы Firebase, которая включает в себя множество сервисов для разработки и поддержки мобильных и веб-приложений. FCM предоставляет широкие возможности для управления и доставки уведомлений. Он позволяет персонализировать уведомления, что означает, что каждое уведомление может быть настроено под конкретного пользователя или группу пользователей. Это позволяет отправлять уведомления, которые максимально соответствуют потребностям и ожиданиям пользователей.

Для работы с FCM (Firebase Cloud Messaging) необходимо сначала зарегистрироваться на сайте Firebase Console и создать новый проект. Далее, для серверной части нам необходимо получить приватный ключ и настроить

веб-сервис для работы с FCM по инструкции, что указана на странице генерации приватного ключа [25]. На рисунке 8 представлена страница генерации закрытого ключа, которая позволяет обращаться веб-сервису к Firebase Cloud Messaging с запросом на создание уведомления.

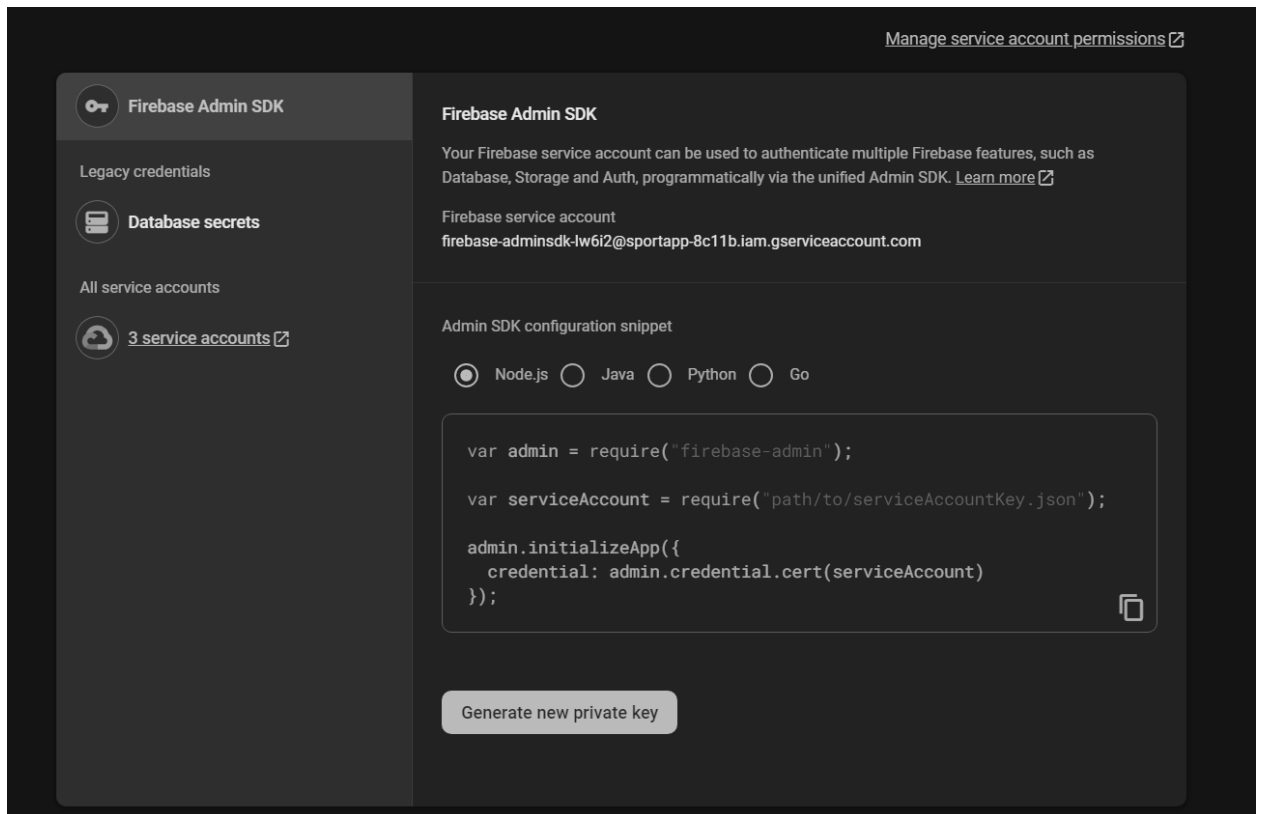


Рисунок 8 — генерация приватного ключа и пример настройки FCM admin для Node.js

После предварительной настройки FCM в веб-сервис, необходимо настроить веб-сервис так, чтобы через какой-то промежуток времени, он проверял наличие напоминаний, которые необходимо отправить пользователю и, собственно, отправить их. Для механизма планирования задач, которым в нашем веб-сервисе будет выступать cron [26]. Он позволяет выполнять определённые задачи или скрипты в заданные интервалы времени или в определённое время суток.

В NestJs для реализации cron используется пакет `@nestjs/schedule`. С его помощью можно создавать и настраивать задания cron, которые будут выполняться автоматически в заданное время.

3.5.4 Диаграмма классов

На рисунке 9 продемонстрирована диаграмма классов концептуального уровня Backend-части мобильного приложения.

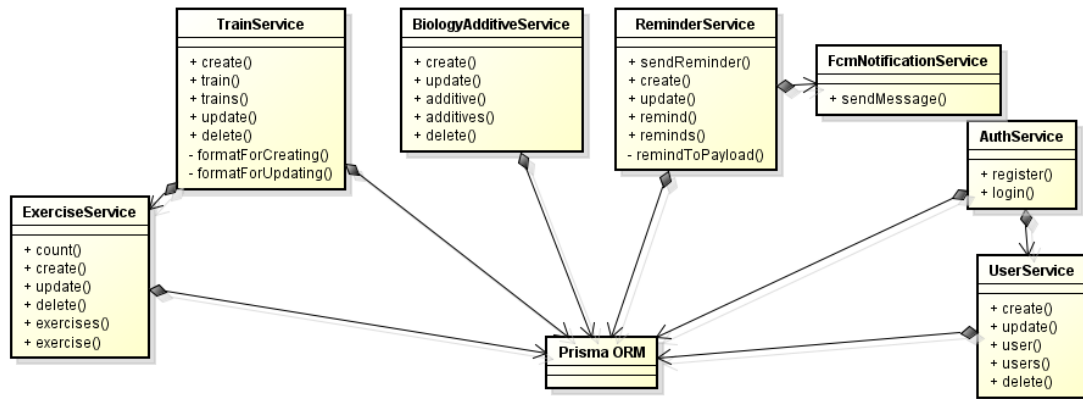


Рисунок 9 — Диаграмма классов концептуального уровня Backend-части приложения

Классы ExerciseService, TrainService, BiologyAdditiveService, ReminderService, UserService необходимы для создания (метод create), редактирования (метод update), получения (например, для класса UserService за получение одного экземпляра отвечает метод user, для получения нескольких экземпляров – users) и удаления (метод delete) элементов. Каждый из них содержит в себе экземпляр класса Prisma ORM, которая отвечает за взаимодействие с базой данных. AuthService отвечает за регистрацию (метод register) и вход пользователя (метод login).

За логику отправки уведомления пользователю отвечает FcmNotificationService, который используется классом ReminderService для отправки напоминаний о приеме биологически активных добавок.

3.6 Проектирование модуля управления приемом БАДов

Для создания модуля отвечающего за управления приема БАДов необходимо сначала создать интерфейс и настроить взаимодействие с сервером для создания, получения, изменения, удаления элементов. После этого необходимо настроить мобильное приложение на получение FCM сообщение и его вывод пользователю в виде push-уведомления.

Вся frontend-часть мобильного приложения написана на языке программирования Kotlin и модуль управления приемом БАДов не стал исключением. Kotlin был выбран потому, что этот язык был объявлен официальным языком разработки для Android, что гарантирует его поддержку и развитие. Кроме того, Kotlin полностью совместим с Java, что позволяет использовать существующие библиотеки и фреймворки, а также упрощает миграцию старого кода. Для написания UI-интерфейса использовался фреймворк Jetpack Compose [27], который упрощает разработку, сокращает количество шаблонов и обеспечивает адаптивность и автоматическое обновление пользовательского интерфейса, уменьшая зависимость от представлений.

Для реализации же приема сообщений необходимо в проекте в Firebase Console добавить Android приложение, после чего скачать и переместить в папку с проектом файл конфигурации и настроить все дополнительные зависимости. Далее, необходимо создать сервис для обработки этих сообщений в мобильном приложении, формировании из них уведомлений и отправки этих уведомлений пользователю в виде push-уведомлений. На рисунке 10 продемонстрированы основные классы модуля управления приемом биологически активных добавок.

Класс MainActivity – это главный класс мобильного приложения. Инициализация мобильного приложения происходит в методе onCreate. Для работы с уведомлениями MainActivity использует класс

NotificationMessagingService, который получает сообщения от FirebaseMessaging и выводит сообщение в виде push-уведомления. MainActivity содержит внутри себя экземпляры классов BioScreen, отвечающий за отрисовку информации о БАДах и напоминаниях, связанных с ними, а также BodyReactionScreen, отвечающий за отрисовку дневника здоровья. BioAdditiveController управляет отображением BioScreen, получает на вход пользовательский ввод, интерпретирует его и отправляет все изменения на сервер. Аналогично, BodyReactionController управляет отображением BodyReactionScreen, получает на вход пользовательский ввод, интерпретирует его и отправляет все изменения на сервер.

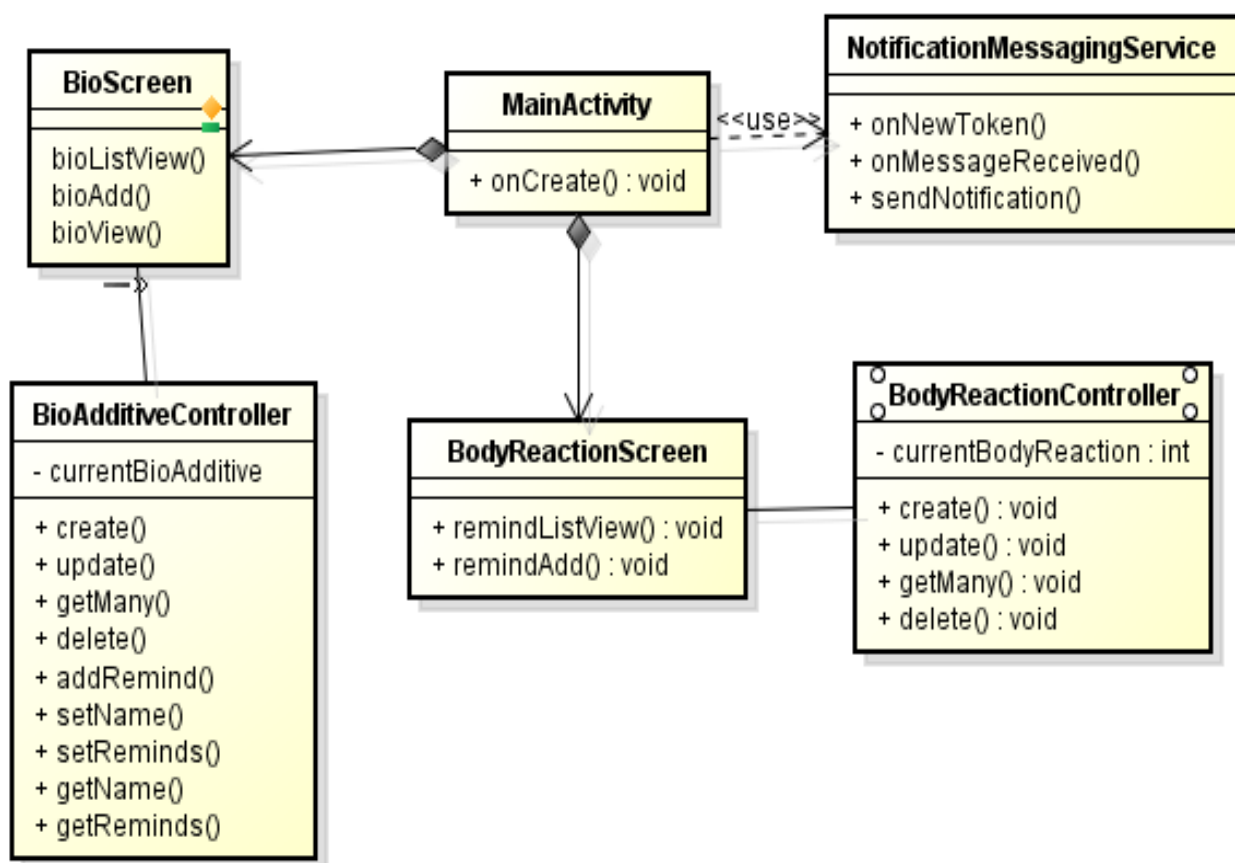


Рисунок 10 — Диаграмма классов концептуального уровня Backend-части приложения

3.7 Используемые языки разработки и программные средства

Backend-часть приложения сервис создана с применением TypeScript и основан на Node.js v20.11.1. Основным фреймворком для создания веб-сервиса был выбран фреймворк NestJS. Для работы с базой данных была выбрана Prisma ORM, а базой данных выступала PostgreSQL, для быстрого развертывания PostgreSQL использовался docker. Для написания Backend-части использовалась интегрированная среда разработки Visual Studio Code.

Fronted-часть приложения написана на языке программирования Kotlin с использованием фреймворка Jetpack Compose в интегрированной среде разработки IntelliJ IDEA 2022.3 и Android Studio.

Для реализации отправки и принятия уведомлений о приеме БАДов использовалось кроссплатформенный облачный сервис для сообщений и уведомлений FCM (Firebase Cloud Messaging).

Для распределенного контроля версий была использована система Git и веб-сервис для хостинга проектов GitHub.

Выводы

В данной главе были всесторонне рассмотрены функциональные и нефункциональные требования к разрабатываемому мобильному приложению для контроля тренировочного процесса и приема биологически активных добавок. Внимание было уделено различным аспектам проекта, включая проектирование, разработку и тестирование ключевых компонентов системы.

Одним из первых шагов в проектировании приложения стало создание диаграммы вариантов использования для модуля, отвечающего за прием биологически активных добавок (БАДов). Диаграмма позволяет

визуализировать взаимодействие пользователя с системой и помогает определить основные функции и сценарии использования модуля. В сценариях использования были подробно описаны действия пользователя, такие как добавление новых добавок, установка напоминаний, ведение дневника приема и анализ полученных данных. Эти сценарии помогают разработчикам и тестировщикам лучше понять требования к системе и разработать более точные и эффективные решения.

Проектирование схемы базы данных мобильного приложения является критически важным этапом разработки. В данной главе была спроектирована и разработана схема базы данных, включающая таблицы для хранения информации о пользователях, тренировках, добавках и напоминаниях. Особое внимание было уделено обеспечению целостности данных и оптимизации запросов для повышения производительности системы. Схема базы данных была разработана с учетом возможности масштабирования и дальнейшего расширения функционала приложения.

Была представлена подробная информация о реализации веб-сервиса, который служит основой для взаимодействия мобильного приложения с серверной частью системы. Веб-сервис обеспечивает выполнение основных функций, таких как аутентификация пользователей, работа с базой данных и выполнение предварительных проверок перед обработкой запросов. Реализация аутентификации включает использование современных методов защиты данных, таких как шифрование и многофакторная аутентификация, что позволяет обеспечить высокий уровень безопасности и конфиденциальности пользовательских данных.

Особое внимание было уделено реализации уведомлений о приеме БАДов, как со стороны сервера, так и со стороны клиента (мобильного Android приложения). Для этого использовался сервис Firebase Cloud Messaging (FCM), который позволяет отправлять push-уведомления на устройства пользователей. Подробно были рассмотрены все этапы настройки и использования FCM, включая регистрацию приложения в Firebase,

интеграцию SDK Firebase в мобильное приложение и настройку сервера для отправки уведомлений. Уведомления помогают пользователям не забывать о приеме добавок и следовать назначенному режиму, что способствует достижению лучших результатов в тренировочном процессе и поддержанию здоровья.

Также в данной главе были подробно описаны выбранные средства и языки программирования, которые использовались для проектирования и разработки Backend-части приложения и модуля для приема биологически активных добавок. Было объяснено, почему именно эти технологии были выбраны, и какие преимущества они предоставляют. Например, использование языка программирования Kotlin для разработки Android-приложения обеспечивает высокую производительность и удобство кодирования, а использование Node.js для серверной части позволяет эффективно обрабатывать большое количество запросов и легко масштабировать систему.

Таким образом, в данной главе были подробно рассмотрены все ключевые аспекты разработки мобильного приложения для контроля тренировочного процесса и приема биологически активных добавок. Определение функциональных и нефункциональных требований, создание диаграммы вариантов использования, проектирование схемы базы данных, реализация веб-сервиса и настройка уведомлений – все эти этапы обеспечили создание эффективного и надежного приложения. Проведенный анализ и выбранные технологии позволили создать продукт, отвечающий потребностям пользователей и обеспечивающий высокое качество и безопасность данных.

4 Тестирование модуля, отвечающего за управление приема БАДов и Backend-части мобильного приложения.

4.1 Ручное тестирование модуля для управления работы с БАДами

Для проверки работоспособности программы необходимо провести тестирование мобильного приложения. Для проверки работоспособности программы были выбраны такие виды тестирования как: ручное тестирование для проверки корректности взаимодействия мобильного приложения с сервером. Помимо этого, необходимо провести юзабилити тестирование, которое необходимо для проверки нашего приложения с точки зрения удобства и комфорта для конечного пользователя, а также нагрузочные тесты, необходимые для оценки производительности, надежности, предельных возможностей [28].

Для того, чтобы провести ручное тестирование необходимо написать сценарии тестирования, которые учитывают различные действия пользователя во время взаимодействия с мобильным приложением. Рассмотрим тест-кейсы, участвовавшие в ручном тестировании мобильного приложения для контроля тренировочного процесса и приема биологически активных добавок спортсмена, а конкретнее модуля, что отвечал за управление приёмом БАДов [29].

Тест-кейс №1. Получение списка напоминаний.

Ожидаемый результат: на экране мобильного телефона пользователя будет выведен список напоминаний.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь переходит на экран всех напоминаний.

Тест-кейс №2. Успешное создание нового БАДа и напоминаний для него.

Ожидаемый результат: создастся новый БАД и напоминания, связанные с ним

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь переходит на экран всех напоминаний.
- 3) Пользователь нажимает кнопку «добавить напоминание».
- 4) Пользователь выбирает тип БАДа.
- 5) Пользователь указывает название БАДа.
- 6) Пользователь нажимает на кнопку добавления приема БАДа. В нем он указывает корректный формат дозировки и времени приема.
- 7) Пользователь нажимает на кнопку сохранения.

Тест-кейс №3. Неудачное создание нового БАДа и напоминания для него при вводе некорректных данных

Ожидаемый результат: на экране пользователя будет выведена ошибка о некорректно введенных данных, БАД и напоминания для него не будут созданы.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь переходит на экран всех напоминаний.
- 3) Пользователь нажимает кнопку добавить напоминание.
- 4) Пользователь выбирает тип БАДа.
- 5) Пользователь указывает название БАДа.
- 6) Пользователь нажимает на кнопку добавления приема БАДа. В нем он указывает некорректный формат дозировки и времени приема.
- 7) Пользователь нажимает на кнопку сохранения.

Тест-кейс №4. Прерывание создания нового БАДа и напоминания для него.

Ожидаемый результат: БАД и напоминания для него не будут созданы.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь переходит на экран всех напоминаний.
- 3) Пользователь нажимает кнопку «добавить напоминание».
- 4) Пользователь выбирает тип БАДа.
- 5) Пользователь указывает название БАДа.
- 6) Пользователь нажимает на кнопку добавления приема БАДа. В нем он указывает дозировку время приема.
- 7) Пользователь нажимает на кнопку назад или любую кнопку из навигационного поля.

Тест-кейс №5. Успешное редактирование БАДа и напоминания для него.

Ожидаемый результат: БАД и напоминания, связанные с ним, обновятся.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь переходит на экран всех напоминаний.
- 3) Пользователь нажимает кнопку редактирования напоминания.
- 4) Пользователь выбирает тип БАДа.
- 5) Пользователь указывает название БАДа.
- 6) Пользователь нажимает на кнопку добавления приемов БАДа или редактирует уже ранее созданные.
- 7) Пользователь нажимает на кнопку сохранения.

Тест-кейс №6. Неудачное редактирование нового БАДа и напоминания для него при вводе некорректных данных.

Ожидаемый результат: на экране пользователя будет выведена ошибка о некорректно введенных данных, БАД и напоминания для него не будут обновлены.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь переходит на экран всех напоминаний.

3) Пользователь нажимает кнопку добавить напоминание.

4) Пользователь выбирает тип БАДа.

5) Пользователь указывает название БАДа.

6) Пользователь нажимает на кнопку добавления приемов БАДа или редактирует уже ранее созданные. Во время ввода данных пользователь указывает неверный формат времени приема или дозировки.

7) Пользователь нажимает на кнопку сохранения.

Тест-кейс №7. Прерывание редактирования нового БАДа и напоминания для него.

Ожидаемый результат: БАД и напоминания для него не будут обновлены.

Действия тестировщика:

1) Пользователь заходит в раздел управления приемом БАДов.

2) Пользователь переходит на экран всех напоминаний.

3) Пользователь нажимает кнопку добавить напоминание.

4) Пользователь выбирает тип БАДа (порошок таблетки)

5) Пользователь указывает название БАДа.

6) Пользователь нажимает на кнопку добавления приема БАДа. В нем он указывает дозировку и корректный формат времени приема. Пользователь может создать сразу несколько приемов, связанных с БАДом.

7) Пользователь нажимает на кнопку назад или любую кнопку из навигационного поля.

Тест-кейс №8. Удаления БАДа и напоминания для него.

Ожидаемый результат: БАД и напоминания, связанные с ним, удалятся.

Действия тестировщика:

1) Пользователь заходит в раздел управления приемом БАДов.

2) Пользователь переходит на экран всех напоминаний.

3) Пользователь нажимает кнопку удаления напоминания.

Тест-кейс №9 Получение списка заметок.

Ожидаемый результат: на экране мобильного телефона пользователя будет выведен список записок.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь нажимает на кнопку перехода к списку заметок.

Тест-кейс №10. Успешное создание новой заметки.

Ожидаемый результат: создается новая заметка в списке заметок.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь нажимает на кнопку перехода к списку заметок.
- 3) Пользователь нажимает кнопку добавления заметок.
- 4) Пользователь указывает заголовок и описание заметки.
- 5) Пользователь нажимает на кнопку сохранения.

Тест-кейс №11. Успешное редактирование заметки.

Ожидаемый результат: заметка будет обновлена.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь нажимает на кнопку перехода к списку заметок.
- 3) Пользователь нажимает кнопку редактирования заметок.
- 4) Пользователь указывает заголовок и описание заметки.
- 5) Пользователь нажимает на кнопку сохранения.

Тест-кейс №12. Прерывание создания новой заметки.

Ожидаемый результат: создается новая заметка в списке заметок.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь нажимает на кнопку перехода к списку заметок.
- 3) Пользователь нажимает кнопку добавления заметок.
- 4) Пользователь указывает заголовок и описание заметки.
- 5) Пользователь нажимает на кнопку назад или любую кнопку из

навигационного поля.

Тест-кейс №13. Прерывание редактирования заметки.

Ожидаемый результат: заметка будет обновлена.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь нажимает на кнопку перехода к списку заметок.
- 3) Пользователь нажимает кнопку редактирования заметок.
- 4) Пользователь указывает заголовок и описание заметки.
- 5) Пользователь нажимает на кнопку назад или любую кнопку из

навигационного поля.

Тест-кейс №14. Удаление заметки.

Ожидаемый результат: заметка удалится.

Действия тестировщика:

- 1) Пользователь заходит в раздел управления приемом БАДов.
- 2) Пользователь нажимает на кнопку перехода к списку заметок.
- 3) Пользователь нажимает кнопку удаления заметки.

Тест-кейс №15. Получение уведомления.

Ожидаемый результат: пользователь получит уведомление на мобильный телефон.

Действия тестировщика:

- 1) Пользователь выполняет действия 1–7 из тест-кейса №2
- 2) Пользователь выходит из мобильного приложения и ждет наступление времени, которое он указал в напоминании.

В процессе ручного тестирования все тест-кейсы были пройдены успешно, не было обнаружено каких-либо ошибок. Это говорит о высоком качестве разработки и тщательной проработке всех компонентов системы. Ручное тестирование включало проверку всех основных функций мобильного приложения. Каждый из этих тестов проводился в различных условиях и сценариях, чтобы убедиться в стабильности и надежности работы системы.

4.2 Юзабилити-тестирование

Юзабилити тестирование – это вид тестирования, при котором оценивается удобство использования продукта.

Целью юзабилити тестирования является создание продукта, который максимально удобен и интуитивно понятен для пользователей. Это тестирование помогает разработчикам понять, как пользователи взаимодействуют с продуктом в реальных условиях, выявить слабые места интерфейса и разработать улучшения, которые сделают использование продукта более простым и приятным.

В качестве инструмента юзабилити тестирования было решено использовать анкетирование. Анкетирование проходило 24 независимых пользователя после достаточно продолжительного использования мобильного приложения для контроля тренировочного процесса и приема биологически активных добавок. В частности, они оценили модуль управления приема БАДов.

В процессе анкетирования были заданы вопросы:

- Является ли дизайн приложения интуитивно понятным и привлекательным для вас?
- Какова ваша общая оценка удобства использования мобильного приложения по шкале от 1 до 5?
- Как вы оцениваете скорость работы приложения при выполнении различных действий?
- Насколько легко для вас было освоиться с функциями работы модуля, отвечающим за управление приемом БАДов?
- Вас устраивает система уведомлений о приеме биологически активных добавок?
- Есть ли необходимость в функции дневника здоровья (заметок)?

На рисунках 11–16 представлены результаты анкетирования.



Рисунок 11 — Результаты анкетирования независимых пользователей



Рисунок 12 — Результаты анкетирования независимых пользователей



Рисунок 13 — Результаты анкетирования независимых пользователей



Рисунок 14 — Результаты анкетирования независимых пользователей

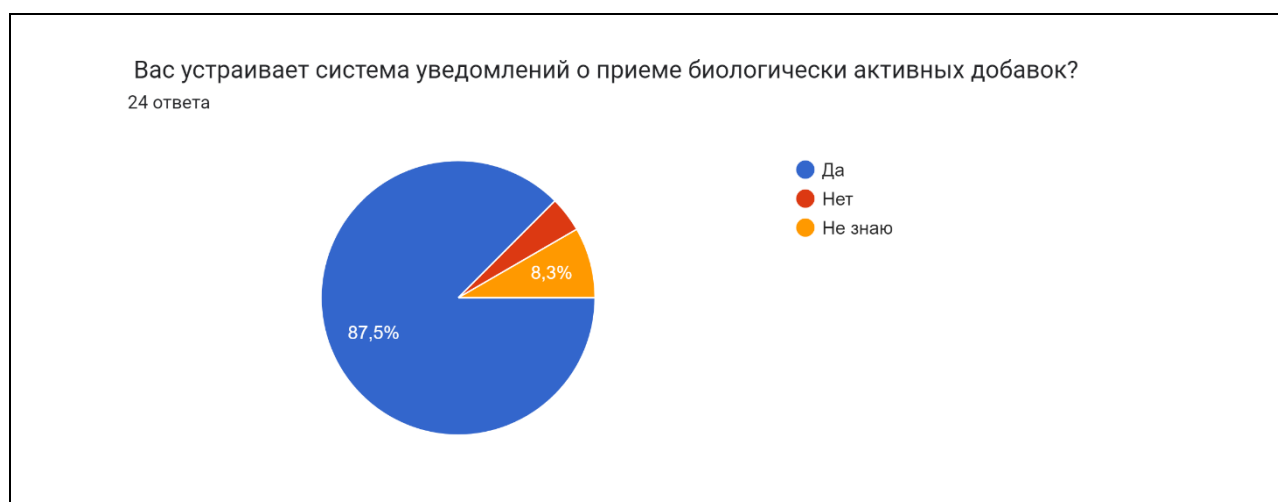


Рисунок 15 — Результаты анкетирования независимых пользователей



Рисунок 16 — Результаты анкетирования независимых пользователей

Результаты опроса показали, что большинство участников опроса считают, что мобильное приложение по контролю тренировочного процесса и приему биологически активных добавок обладает интуитивно понятным и привлекательным интерфейсом.

75% опрошенных поставили приложению наивысшую оценку. Около 21% пользователей оценили приложение, как хорошее. Оставшиеся опрошенные пользователи поставили оценку «Удовлетворительно».

Следующий вопрос касался сложности освоения модуля о приеме биологически активных добавках. Три четверти опрошенных посчитали, что освоить функции, за которые отвечает модуль, легко. Одна пятая опрошенных пользователей оценили сложность освоения, как удовлетворительную. И около 4% пользователей посчитали, что довольно трудно освоить функционал модуля, контролирующей приём биологически активных добавок.

Пользователи при анкетировании высоко оценили скорость работы приложения. Большая часть опрошенных, поставила наивысшую оценку скорости работы. Четверть опрошенных считают, что скорость работы приложения является хорошей.

Система уведомлений о приёме биологически активных добавок устраивает большинство пользователей.

На вопрос о необходимости функции дневника здоровья большинство пользователей подтвердило, что эта функция нужна внутри приложения и поможет лучше контролировать процесс спортивной активности пользователя.

4.3 Нагрузочное тестирование

Нагрузочное тестирование — это вид тестирования, направленный на проверку производительности и надежности приложения под нагрузкой. Оно

позволяет выявить узкие места и потенциальные проблемы, которые могут возникнуть при увеличении числа пользователей или объема данных. Тестирование помогает гарантировать, что серверная часть приложения сможет обрабатывать большие объемы запросов без задержек и сбоев, обеспечивая пользователям плавный и стабильный опыт взаимодействия. Кроме того, оно помогает оценить, насколько эффективно система масштабируется и как она справляется с экстремальными условиями эксплуатации [30].

Для создания и выполнения нагрузочных тестов была использована утилита Artillery, которая является инструментом командной строки, созданная для проведения тестов производительности и нагрузки веб-приложений. Для ее работы необходимо составить конфигурационный файл. В конфигурационном файле необходимо указать ссылку, на котором работает веб-сервис, время тестирования, количество запросов в секунду и указать, собственно, HTTP-request, по которым будет осуществляться тестирование. Для тестирования веб-сервиса были выбрано время тестирования одна минута и десять запросов в секунду. Тестирование осуществлялось на локальном компьютере. Результаты нагрузочного тестирования Backend-части мобильного приложения для контроля и приема биологически активных добавок представлены в таблицах 2 и 3.

Таблица 2 — время отклика на HTTP-request

Минимальное время (мс)	Среднее время (мс)	Максимальное время
29	33	108
30	56	134
23	52	75
24	49	77
24	50	1072
23	44	141
23	15	165

Таблица 3 — время отклика на HTTP-request для n-го процентиля

50-й процентиль	75-й процентиль	95-й процентиль	99-й процентиль
46.1	50.9	64.7	66
43.4	59.7	71.5	106.7
43.4	58.6	67.4	73
45.2	63.4	71.5	74.4
61	82.3	788.5	1022.7
71.5	89.1	100.5	135.7
71.5	74.4	133	133

Из приведенных данных можно сделать вывод, что Backend-часть нашего мобильного приложения по контролю тренировок и приему биологически активных добавок успешно справляется с нагрузками. Анализ времени отклика HTTP-запросов показал, что система обеспечивает стабильную и быструю работу даже при высоких нагрузках. Минимальное время отклика колеблется около 25 миллисекунд, что свидетельствует о быстрой реакции на запросы. Среднее время отклика на уровне 50 миллисекунд также указывает на высокую производительность и эффективность серверной части приложения. Кроме того, даже при анализе верхних процентилей, таких как 95-й и 99-й, система демонстрирует устойчивость и надежность, что подтверждает её способность обрабатывать большие объемы данных без существенного снижения скорости.

Выводы

В данном разделе был рассмотрен процесс тестирования Backend-части и модуля управления приемом биологически активных добавок. Был проведено ручное тестирование, юзабилити-тестирование и нагрузочное тестирование.

В ходе проведения ручного тестирования были написаны тест-кейсы, по которым была осуществлена проверка в результате которых не было выявлено каких-либо ошибок в работе модуля по приему биологически активных добавок.

Далее, было проделана работа по юзабилити-тестированию, в процессе которой было проведено анкетирование независимых пользователей, которые высоко оценили мобильное приложение, его скорость работы и интуитивность пользовательского интерфейса.

Последним тестированием оказалось нагрузочное тестирование, необходимое для проверки работы веб-сервиса под нагрузкой. В ходе нагрузочного тестирования не было выявлено каких-либо серьезных проблем. Приложение показало стабильную работу в процессе проведения нагрузочного тестирования.

Заключение

В процессе выполнения исследования и работы были решены следующие задачи и достигнуты соответствующие результаты:

- Проведен анализ предметной области, в ходе которого были рассмотрены теоретические основы контроля тренировочного процесса, а также теоретическая обоснованность управления приемов биологически активных добавок (БАДов), были выявлены тенденции на увеличение интереса людей занятием и цифровизацию этого процесса и запрос на новые решения в сфере мобильных приложений, предназначенных для контроля спортивной деятельности пользователя;
- Проведен анализ технологических аспектов, необходимых для создания Backend-части мобильного приложения для контроля тренировок и приема биологически активных добавок;

- Проведен анализ и были выделены основные задачи, которые необходимо решить Backend-частью приложения. Из них необходимо выделить создания структуры базы данных, обеспечение безопасности и аутентификации пользователей, создания обработчиков запросов, а также создание и отправка уведомлений о необходимости приема биологически активной добавки в указанное пользователем время;

- Проведен анализ и были выделены основные задачи модуля управления приемом БАДов, такие как создание, получение, редактирование и удаление уведомлений о необходимости приема биологически активных добавок;

- Реализован Backend-часть мобильного приложения с использованием NestJS, которая позволяет создавать упражнения, тренировки, напоминания о приеме БАДов, а также в нужное время отправляет пользователю на мобильное устройство напоминание о приеме БАДов;

- Реализован модуль, отвечающий за управления приемом биологически активных добавок во Frontend-части приложения, которая позволяет пользователю создавать заметки, а также создавать напоминания и получать push-уведомления о необходимости приема БАДов;

- Проведено тестирование разработанной программы с помощью ручных тестов по описанным сценариям взаимодействия пользователя с навыком, проведено юзабилити-тестирование двадцати четырех пользователей, которые положительно оценили работу мобильного приложения, в частности модуля управления приемом БАДов;

В дальнейшем планируется улучшение текущих функций и добавление новых. Планируется добавить историю упражнений, чтобы можно было проследить прогресс пользователя за время тренировок. Например, увеличение повторений, времени, весов, с которыми пользователь работает. Планируется сделать настройку уведомлений более гибкой с указанием. Например, прием БАДов только по определенным дням недели, через день, через два дня и так далее.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мусаева, Н. М. Пищевые и биологически активные добавки : учебно-методическое пособие / Н. М. Мусаева. – Махачкала : ДагГАУ имени М.М.Джамбулатова, 2019. – 91 с. – URL: <https://e.lanbook.com/book/159407> (дата обращения: 12.05.2024). – Режим доступа: ЭБС «Лань», для авториз. пользователей.
2. Чем грозит избыток БАД здоровому человеку? // MedAboutMe : сайт. – URL: https://medaboutme.ru/articles/chem_grozit_izbytok_bad_zdorovomu_cheloveku/ (дата обращения: 28.05.2024).
3. Спортсмен в России занимаются 76 млн человек (2 млн – у компа), из фигурки ушли сотни тысяч – арифметика от Минспорта // Sports.ru : сайт. – URL: <https://www.sports.ru/tribuna/blogs/allresp/3231402.html> (дата обращения: 28.05.2024)
4. Попова, Н. Н. Пищевые и биологически активные добавки : учебное пособие / Н. Н. Попова, Е. С. Попов, И. П. Щетилина. – Воронеж : ВГУИТ, 2016. – 67 с. – ISBN 978-5-00032-220-8. – URL: <https://e.lanbook.com/book/92220> (дата обращения: 28.05.2024). – Режим доступа: ЭБС «Лань», для авториз. пользователей.
5. Баланов, А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов. – Санкт-Петербург : Лань, 2024. – 312 с. – ISBN 978-5-507-48818-6. – URL: <https://e.lanbook.com/book/394556> (дата обращения: 12.05.2024). – Режим доступа: ЭБС «Лань», для авториз. пользователей.
6. Новиков, Б. А. Основы технологий баз данных : учебное пособие / Б. А. Новиков, Е. А. Горшкова, Н. Г. Графеева ; под ред. Е. В. Рогова. — 2-е изд. — Москва : ДМК Пресс, 2020. - 582 с. - ISBN 978-5-97060-841-8.
7. Джуба, С. Изучаем PostgreSQL 10 / С. Джуба, А. Волков. — Москва : ДМК Пресс, 2019. — 400 с. — ISBN 978-5-97060-643-8.

8. Требования ACID на простом языке // Habr : сайт. – URL: <https://habr.com/ru/articles/555920/> (дата обращения: 28.05.2024).
9. MyTherapy // MyTherapy : официальный сайт. – URL: <https://www.mytherapyapp.com/ru> (дата обращения: 28.05.2024).
10. MediSafe // MediSafe : официальный сайт. – URL: <https://www.medisafe.com/> (дата обращения: 28.05.2024).
11. Мои таблетки // Google Play : официальный сайт. – URL: <https://play.google.com/store/apps/details?id=com.devsoldiers.calendar.pills.limit> (дата обращения: 28.05.2024).
12. Малышев, К. В. Построение пользовательских интерфейсов / К. В. Малышев. – Москва : ДМК Пресс, 2021. – 268 с. – ISBN 978-5-97060-962-0.
13. Идентификация, Аутентификация, Авторизация. В чем же разница? // Habr : сайт. – URL: <https://habr.com/ru/articles/720842/> (дата обращения: 28.05.2024).
14. NestJS // NestJS : официальный сайт. – URL: <https://nestjs.com/> (дата обращения: 12.05.2024).
15. Руководство по NestJS. Часть 1 // Habr : сайт. – URL: <https://habr.com/ru/companies/timeweb/articles/663234/> (дата обращения: 28.05.2024).
16. Node.js // Node.js : официальный сайт. – URL: <https://nodejs.org/> (дата обращения: 12.05.2024).
17. Заяц, А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js / А. М. Заяц, Н. П. Васильев. – 3-е изд., стер. – Санкт-Петербург : Лань, 2023. – 120 с. – ISBN 978-5-507-45423-5.
18. ORM или как забыть о проектировании БД // Habr : сайт. – URL: <https://habr.com/ru/articles/237889/> (дата обращения: 28.05.2024).
19. Prisma // Prisma : официальный сайт. – URL: <https://nodejs.org/> (дата обращения: 25.05.2024).
20. Обзор Prisma ORM: как забыть об SQL и сосредоточиться на данных // Habr : сайт. – URL: <https://habr.com/ru/companies/macloud/articles/554952/> (дата обращения: 28.05.2024).

21. Passport (authentication) // NestJs : официальный сайт. – URL: <https://docs.nestjs.com/recipes/passport> (дата обращения: 28.05.2024).
22. Guards (authentication) // NestJs : официальный сайт. – URL: <https://docs-nestjs.netlify.app/guards> (дата обращения: 28.05.2024).
23. How to use class-validator and generate custom error object in nest.js // Dev : сайт. – URL: <https://dev.to/nithinkjoy/how-to-use-class-validator-and-return-custom-error-object-in-nestjs-562h> (дата обращения: 28.05.2024).
24. Firebase Cloud Messaging // Microsoft : сайт. – URL: <https://learn.microsoft.com/ru-ru/previous-versions/xamarin/android/data-cloud/google-messaging/firebase-cloud-messaging?cid=kerryherger> (дата обращения: 28.05.2024).
25. How to send push notifications with Firebase, React and NestJS // Mohima : сайт. – URL: <https://mohima24.hashnode.dev/how-to-send-push-notifications-with-firebase-react-and-nestjs> (дата обращения: 28.05.2024).
26. Бобровский, В. И. Расширенное администрирование сетевой операционной системы GNU/Linux. Локальное системное администрирование : учебное пособие / В. И. Бобровский, А. В. Дагаев, Е. П. Журавель. – Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2022. – 138 с.
27. Введение в Jetpack Compose // Habr : сайт. – URL: <https://habr.com/ru/companies/rncb/articles/669374/> (дата обращения: 28.05.2024).
28. Игнатъев, А. В. Тестирование программного обеспечения / А. В. Игнатъев. – 3-е изд., стер. – Санкт-Петербург : Лань, 2023. – 56 с. – ISBN 978-5-507-45425-9. – URL: <https://e.lanbook.com/book/269873> (дата обращения: 29.05.2024). – Режим доступа: ЭБС «Лань», для авториз. пользователей.
29. Гид по ручному тестированию приложений: преимущества, этапы и методологии // Habr : сайт. – URL: <https://habr.com/ru/companies/skillbox/articles/418889/> (дата обращения: 28.05.2024).

30. Поговорим о нагрузочном тестировании // Habr : сайт. – URL:
<https://habr.com/ru/companies/veeam/articles/578942/> (дата обращения:
28.05.2024).

Приложение А

Справка о результатах проверки выпускной квалификационной работы на
наличие заимствований

Приложение Б
Техническое задание

Приложение В
Руководство системного программиста