Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ТЕХНИЧЕСКИЙ ПРОЕКТ**

«Информационная система для учета фитнес-активности и составления тренировочных планов»

Киров,

2024

**СОДЕРЖАНИЕ**

[**1. Бизнес-процессы**. 3](#_Toc182789728)

[**2. Варианты использования** 5](#_Toc182789729)

[**4**. **Передача и хранение данных** 10](#_Toc182789730)

[**4.1 Передача данных** 10](#_Toc182789731)

[**4.2 Хранение данных** 10](#_Toc182789732)

[**4.3 Выбор целевой СУБД** 12](#_Toc182789733)

[**5**. **Технологический стек** 14](#_Toc182789734)

## **1. Бизнес-процессы**.

На рисунке 1 представлена контекстная диаграмма IDEF0 для пользователя.

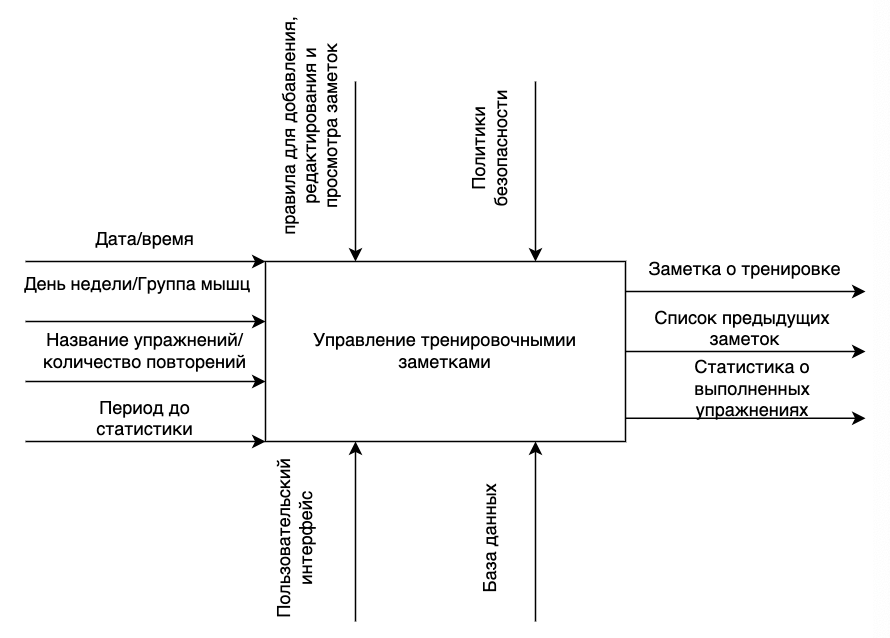


Рисунок 1 – контекстная диаграмма мессенджера

**Входы:**

* Пользовательский интерфейс
* База данных
* Правила для добавления, редактирования и просмотра заметок
* Политики безопасности

**Выходы:**

* Заметка о тренировке
* Список предыдущих заметок
* Статистика о выполненных упражнениях

**Механизмы:**

* Управление тренировочными заметками

**Правила:**

* Правила для добавления, редактирования и просмотра заметок
* Политики безопасности

Для разбиения сложного процесса на составляющие подфункции применяется декомпозиция.

Диаграмма декомпозиции, представляет контекстную функцию, разложенную на 4 составляющих более простых функций.

1. Создание заметки.
2. Редактирование/просмотр заметок.
3. Запрос на получение статистики.
4. Запись в бд.

На рисунке 2 показан результат декомпозиции Информационной системы для учета фитнес-активности.

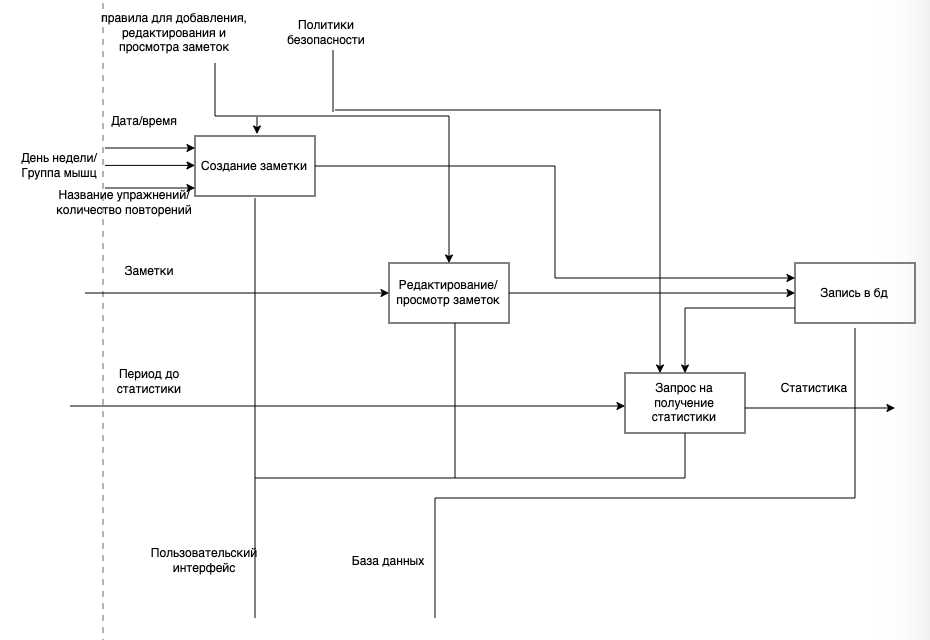


Рисунок 2 – результат декомпозиции Информационной системы для учета фитнес-активности.

## **2. Варианты использования**

На рисунке 3 представлена диаграмма прецедентов для мессенджера.

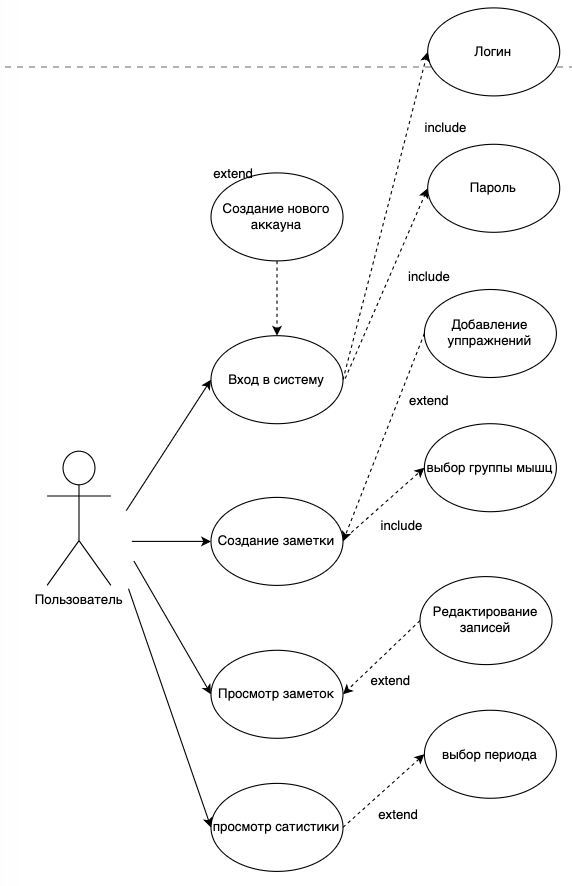


Рисунок 3 диаграмма прецендентов для мессенджера

Исходя из данной диаграммы возможны 5 вариантов использования:

* Вход в систему
* Создание заметки
* Просмотр заметок
* Просмотр статистики

Далее представлены отдельные диаграммы для ранее перечисленных вариантов использования.

На рисунке 4 представлена диаграмма прецендента «Вход в систему».

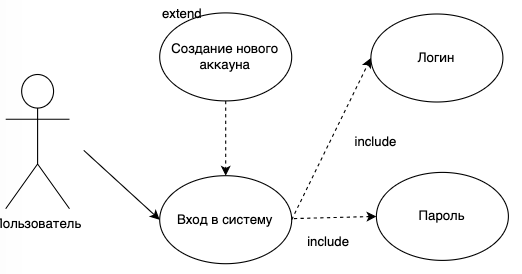


Рисунок 4 диаграмма прецедента «Вход в систему»

На рисунке 5 представлена диаграмма прецендента «Вход в систему».

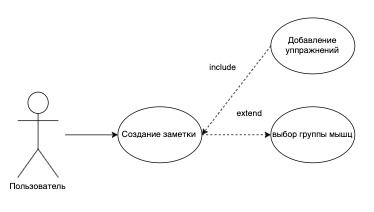


Рисунок 5 диаграмма прецедента «Создание заметки»

На рисунке 6 представлена диаграмма прецендента «Просмотр сообщений».

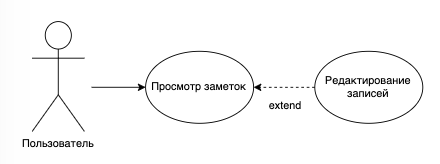


Рисунок 6 диаграмма прецедента «Просмотр сообщений»

На рисунке 7 представлена диаграмма прецендента «Просмотр статистики».

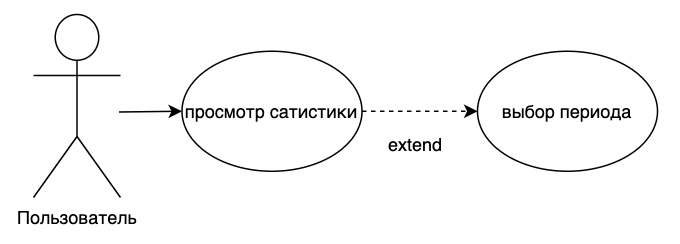


Рисунок 7 диаграмма прецедента «Просмотр статистики»

**3. Пользовательский интерфейс**

На данной экранной форме изображено окно, в котором вы сможете создать новую заметку см.рис 4. Прежде чем начать записывать упражнения вам необходимо выбрать группу(ы) мышц. Также, на каждом экране будет расположено меню в нижней части экрана с тремя возможными вкладками «Создание тренировки» (это уже выбранная страница), «Все тренировки», «Статистика».

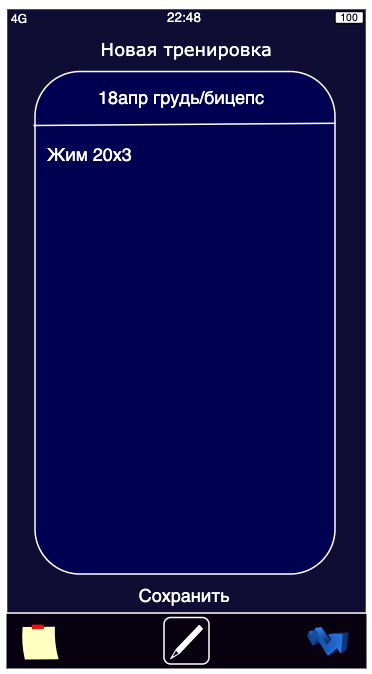


Рисунок 8 – Экранная форма входа

На вкладке «Тренировки» см.рис 9 вы сможете просмотреть весь список прошедших тренировок. При нажатии на карандашик рядом с заметкой откроется точно такое же окно, как и при создании заметок, где вы сможете редактировать заметки.

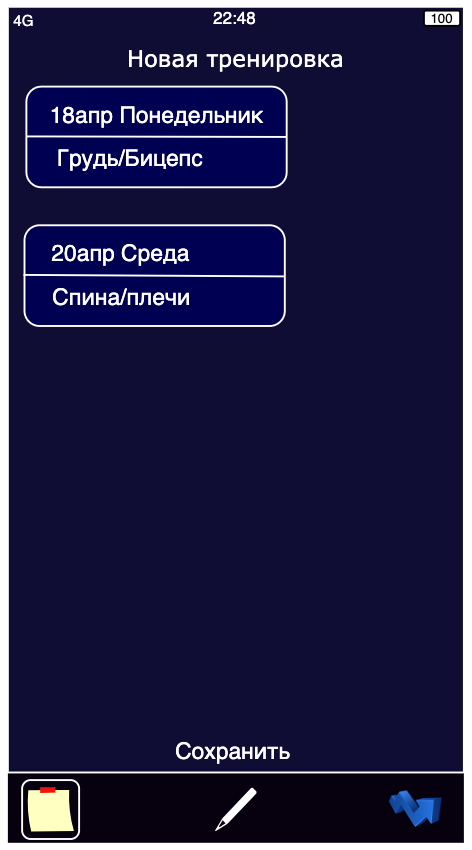


Рисунок 9 – Экранная форма создания аккаунта

На экранной форме для отображения статистики см.рис 10 вам потребуется выбрать 2 месяца, после чего высветятся результаты всех упражнений до/после за выбранный период.

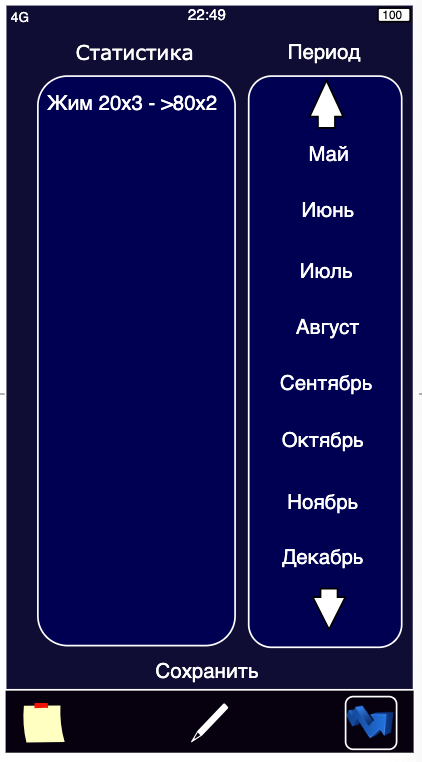


Рисунок 10 – Экранная форма отображения статистики

## **4**. **Передача и хранение данных**

### **4.1 Передача данных**

На рисунке 13 представлена диаграмма DFD для Информационной системы для учета фитнес-активности.

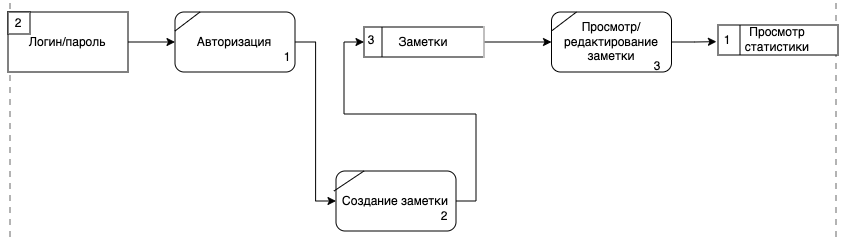


Рисунок 13 – диаграмма DFD для мессенджера

Данная DFD диаграмма описывает процесс работы с заметками в системе. Пользователь сначала проходит авторизацию, предоставляя правильный логин/пароль. После успешной авторизации, пользователь может создавать новые заметки, просматривать и редактировать уже существующие. При просмотре заметки, информация о ней записывается в сущность "Просмотренные Заметки". Пользователь также имеет возможность просмотреть статистику по количеству созданных и просмотренных заметок, что позволяет ему анализировать свою активность. Все данные хранятся в реляционной базе данных, что обеспечивает их целостность и доступность. Таким образом, DFD диаграмма наглядно демонстрирует взаимодействие пользователя с системой и потоки данных между различными сущностями и процессами.

### **4.2 Хранение данных**

В таблице 1 выделен набор сущностей с их свойствами для мессенджера.

Таблица 1 – Набор сущностей и их свойств

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сущность | Свойство | Тип | Уникальность | Обязательность заполнения | Ограничения |
| Пользователи | пользователь\_id | number | Да | Да | Первичный ключ, только числа |
| имя | VARCHAR(100 | Нет | Да | Имя пользователя |
| email | VARCHAR(100) | Да | Да | Электронная почта |
| пароль | varchar(1000) | Нет | Да | Хешированный пароль |
| Воркаут | воркаут\_id | Number | Да | Да | Уникальный идентификатор воркаута |
| пользователь\_id | number | нет | Да | Ссылка на пользователя |
| дата | DATE | Нет | Да | Дата проведения воркаута |
| записи | TEXT | Нет | Нет | Описание или заметки |
| Упражнения | упражнение\_id | number | Нет | Да | Уникальный идентификатор упражнения |
| имя | number | Нет | Да | Название упражнения |
| группа\_мышц | VARCHAR(100) | Нет | Нет | Целевая группа мышц |
| Детали | детали\_id | number | Да | Да | Уникальный идентификатор записи |
| воркаут\_id | number | Нет | Да | Ссылка на воркаут |
| упражнение\_id | number | Нет | Да | Ссылка на упражнение |
| повторения | number | Нет | Нет | Количество повторений |
| наборы | number | Нет | Нет | Количество подходов |

Связи между таблицами:

1. **User → Workout** 
   * Один пользователь может иметь несколько тренировок.
   * Связь реализуется с помощью внешнего ключа user\_id в таблице **Workout**, который ссылается на user\_id в таблице **User**.
2. **Workout → WorkoutDetail** 
   * Одна тренировка может включать несколько записей деталей тренировки.
   * Связь реализуется с помощью внешнего ключа workout\_id в таблице **WorkoutDetail**, который ссылается на workout\_id в таблице **Workout**.
3. **Exercise → WorkoutDetail** 
   * Одно упражнение может быть использовано в нескольких деталях тренировок.
   * Связь реализуется с помощью внешнего ключа exercise\_id в таблице **WorkoutDetail**, который ссылается на exercise\_id в таблице **Exercise**.

Логическая модель данных представлена на рисунке 14.



Рисунок 14 – Логическая диаграмма

### **4.3 Выбор целевой СУБД**

В качестве целевой СУБД был выбран SQlite, он обладает высокой производительностью и масштабируемостью. Он способен обрабатывать большие объемы данных и поддерживать высокий уровень нагрузки.

Описание полей созданных таблиц представлены в таблице 2.

Таблица 2 – Описание полей таблиц

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Атрибут | Тип | Размер |
| User | user\_id | INTEGER | 10000 символов |
| name | TEXT | 64 символов |
| Email | TEXT | 64 символов |
| password | TEXT | 64 символов |
| Workout | workout\_id | INTEGER | 10000 символов |
| user\_id | INTEGER | 10000 символов |
| date | TEXT | 1000 символов |
| notes | TEXT | 1000 символов |
| Exercise | exercise\_id | INTEGER | 10000 символов |
| name | TEXT | 10000 символов |
| muscle\_group | TEXT | 10000 символов |
| WorkoutDetail | detail\_id | INTEGER | 10000 символов |
| workout\_id | INTEGER | 10000 символов |
| exercise\_id | INTEGER | 10000 символов |
| reps | INTEGER | 10000 символов |
| sets | INTEGER | 10000 символов |

Физическая модель данных представлена на рисунке 15.

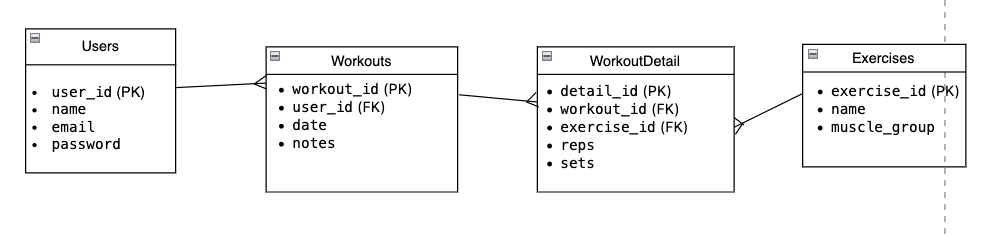


Рисунок 15 – Физическая диаграмма

## **5**. **Технологический стек**

В качестве языка программирования при написании программы использовался язык программирования Python. Python – это мультипарадигменный язык программирования высокого уровня.

Для реализации пользовательского интерфейса использовалась библиотека kivy, позволяющая реализовывать инструменты в разработке на языке python.

В качестве СУБД использовалась СУБД SQlite.

6 Алгоритмы решения задач

**6.1 Регистрация аккаунта**

Приложение проверяет корректность заполнения полей регистрации:

Поля считаются некорректными, если:

Указан некорректный формат электронной почты.

Пароль слишком короткий.

Электронная почта уже зарегистрирована.

Если данные корректны, при нажатии кнопки «Создать аккаунт» создаётся запись в таблице Пользователи с уникальным идентификатором, именем, email и хэшированным паролем. В случае ошибок отображается сообщение.

**6.2 Вход в систему**

Приложение проверяет корректность заполнения полей:

Поля считаются некорректными, если:

Указан неверный формат электронной почты.

Неправильный пароль.

Если данные корректны, выполняется проверка введённых данных с записями в таблице Пользователи. При успешной аутентификации происходит переход на основную экранную форму, иначе отображается уведомление об ошибке.

**6.3 Создание и редактирование тренировки**

На вкладке «Создание тренировки» отображается форма для создания новой записи. Пользователь:

Выбирает группы мышц.

Добавляет упражнения, указывая:

Название упражнения.

Количество повторений и подходов.

При сохранении создаётся запись в таблице Воркаут с привязкой к текущему пользователю. Редактирование тренировки выполняется аналогично: пользователь выбирает запись из списка и изменяет данные.

**6.4 Отображение списка тренировок**

На вкладке «Все тренировки» отображается список завершённых тренировок.

При выборе записи отображается подробная информация.

Нажатие на «карандаш» открывает окно редактирования.

**6.5 Просмотр статистики**

На вкладке «Статистика» пользователь выбирает два временных периода для анализа прогресса. Приложение выводит:

Общее количество тренировок за выбранный период.

Изменения по целевым группам мышц (до/после).

Данные берутся из таблиц Воркаут и Детали.

**6.6 Хранение данных**

Хранение данных:

Информация хранится в реляционной базе данных SQLite, с описанными таблицами:

Пользователи — для данных пользователей.

Воркаут — записи тренировок.

Упражнения — справочник упражнений.

Детали — подробности каждой тренировки.