**ПРИЛОЖЕНИЕ Б**

|  |  |  |  |
| --- | --- | --- | --- |
| Продольный бланк_ВятГУ_распорядительный акт+ | | | |
|  | |  | |
|  |  |  |  |
|  |  |  |  |
|  | | | |
| **ТЕХНИЧЕСКИЙ ПРОЕКТ**  Мобильное приложение “фитнес тренер”  2023 | | | |
|  | | | |

**Содержание**

[1 Описание проекта 3](#_gjdgxs)

[2. Основные функциональные возможности 4](#_30j0zll)

[3. Пользовательский сценарий и интерфейс 6](#_1fob9te)

[4. Организация хранения данных 7](#_3znysh7)

[5. Бизнес процессы приложения 11](#_cpjxoe27gamc)

[6. Разработка алгоритма решения задачи 12](#_z3xnsj501x5)

[7. Структура программы 13](#_tyjcwt)

# **Описание проекта**

Предметом разработки является мобильное приложение “Фитнес Тренер” для фитнеса, которое помогает людям следить за своим здоровьем и физической формой.

Назначение платформы:

* предоставление пользователю полезной информации и инструмента для поддержания здорового образа жизни;
* контроль над питанием и потреблением жидкости;

Цель создания проекта: разработка удобного инструмента для пользователей, помогающего им улучшить свои физические показатели и поддерживать здоровый образ жизни.

## **Основные функциональные возможности**

Мобильное приложение “Фитнес тренер” должно обеспечивать следующие возможности:

* Возможность выбора тренировок;
* добавление собственных напитков;
* добавление собственных блюд ;
* Отслеживание уровня потребления воды и калорий;
* Уведомления пользователю для поддержания водного баланса в организме;

ввод данных для расчета калорий и водного баланса для пользователя;

Бизнес-логика и ее декомпозиция представлены на рисунках 1 и 2.

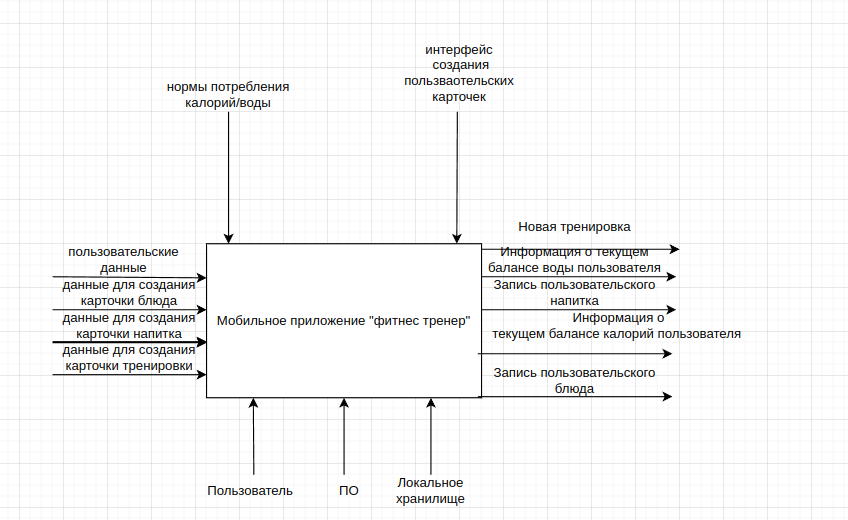


Рисунок 1 - функциональная диаграмма IDEF0

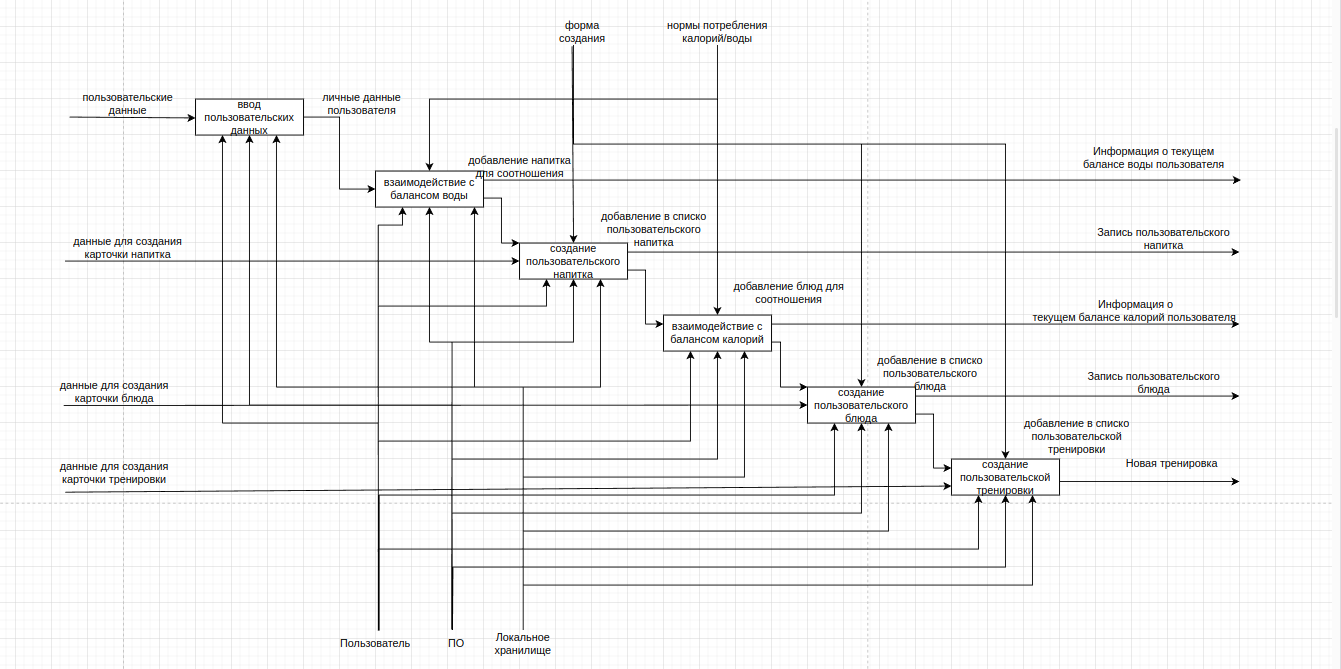


Рисунок 2 - декомпозиция IDEF0

## **Пользовательский сценарий и интерфейс**

Для роли пользователя предполагаются следующий сценарий, представленного на рисунке 3.

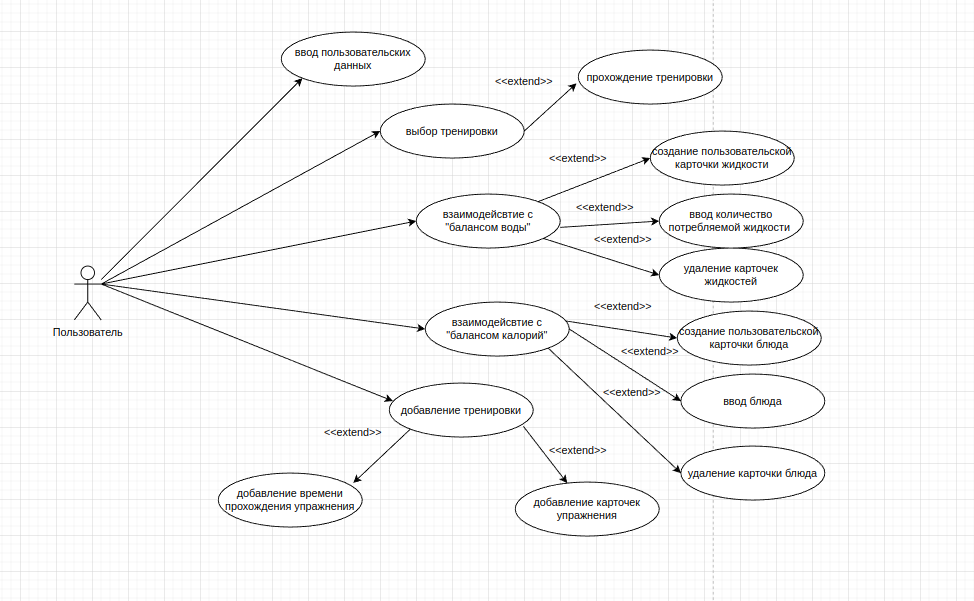


Рисунок 3 – use case диаграмма

Пользовательский сценарий предполагает, что пользователь после ввода своих данных будет иметь возможность к следующим функциям приложения:

* Взаимодействие с выбором тренировки (выбор из списка тренировки)
* Взаимодействие с «балансом воды» (информация о текущем балансе воды пользователя и исходящие от него функции)
* Взаимодействие с «балансом калорий» (информация о текущем балансе калорий пользователя и исходящие от него функции)
* Добавление тренировки (составление самим пользователем списка упражнений и времени на прохождение данной тренировки)

## **Организация хранения данных**

Все временные (постоянно изменяющиеся) данные хранятся в файлах типа JSON.

Каждый файл – является самостоятельным объектом, почти не зависящим от остальных файлов, все зависимости, правила и проверки прописываются в логике приложения. Структура файлов, для наглядного понимания, представлена в формате: <Наименование файла>.<Наименование массива данных>.Файл – это локальное хранилище данных. Структура – это строгий вид хранения данных в файле. Логика управления локальным хранилищем данных – это правила написания, хранения, удаления и изменения данных внутри файлов. Файл JSON позволяет записывать и хранить данные в строгом виде (ключ – значение). Для того чтобы программа смогла взаимодействовать с файлами и понимала то, что в них записано, необходимо создать модели со структурой (интерфейсы), которые преобразовывали бы данные в файлах в понятные для программы объекты, с которыми впоследствии можно было удобно работать. Все данные в файлах хранятся в виде массива объектов. Каждый объект – это отдельная запись, операция. У каждый операции есть отличительный признак – идентификатор.

На рисунке 4 показано описание потоков данных в виде диаграммы DFD

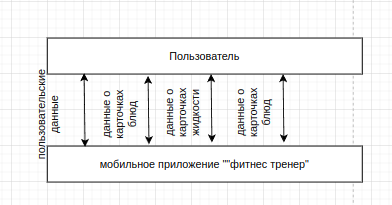


Рисунок 4 – диаграмма DFD

На рисунке 5 показано декомпозиция диаграммы DFD.

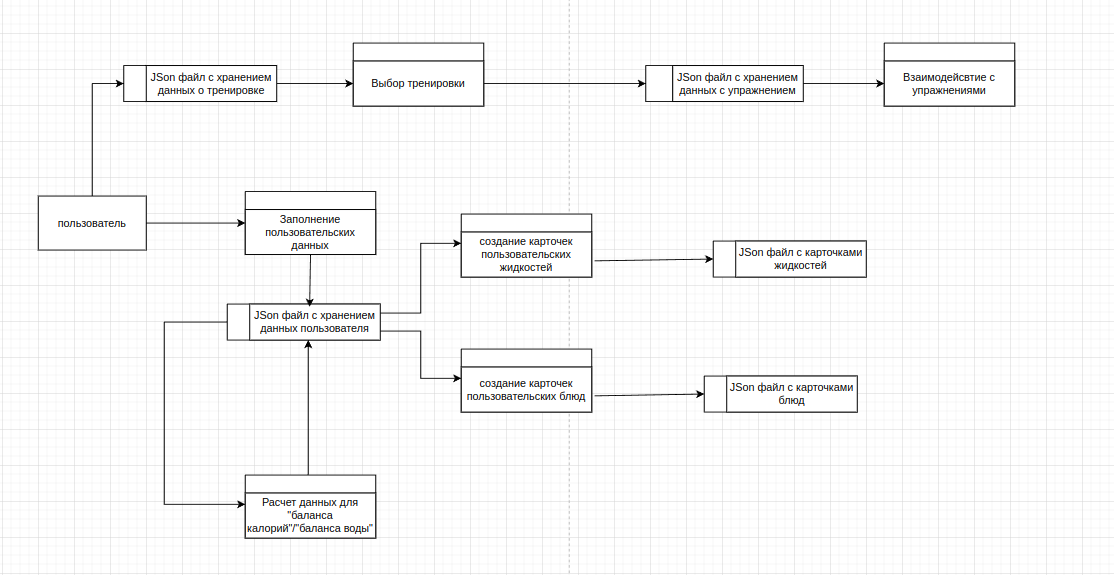


Рисунок 5 – декомпозиция DFD

Структуры файлов представлены в таблице 1.

Таблица 1 – Структура файлов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Структура | Ключ | Тип | Уникальность | Обязательность заполнения | Органичения |
| User\_data | возраст | Числовой | - | Обяз | Not null |
| вес | Числовой | - | Обяз | Not null |
| физ. активность | Числовой | - | Обяз | Not null |
| норм. воды | Числовой | - | Обяз | Not null |
| норм. калорий | Числовой | - | Обяз | Not null |
| training | идентификатор | Числовой | Уник | Обяз | Not null,  Unique,  Primary key |
| Название | Строчный | - | Обяз | Not null |
| количество тренировок | Числовой | - | Обяз | Not null |
| exercise | идентификатор | Числовой | Уник | Обяз | Not null,  Unique,  Primary key |
| номер видео | Числовой | Уник | Обяз | Not null,  Unique,  Primary key |
| время | Числовой | - | Обяз | Not null |
| Текст | Строчный | - | - | - |
| Идентификатор тренировки | Числовой | - | Обяз | Not null,  Foreigh key |
| карточка воды | идентификатор | Числовой | Уник | Обяз | Not null,  Unique,  Primary key |
| Цвет | Строчный | - | Обяз | Not null |
| количество(мл) | Числовой | - | Обяз | Not null |
|  | Числовой | - | Обяз | Not null |
| карточка воды | идентификатор | Числовой | Уник | Обяз | Not null,  Unique,  Primary key |
|  | Цвет | Строчный | - | Обяз | Not null |
|  | количество(гр) | Числовой | - | Обяз | Not null |
|  | количество калорий | Числовой | - | Обяз | Not null |

## **Бизнес процессы приложения**

при первом включении приложения, пользователь должен ввести персональные данные: рост, вес, физическая активность. После этого приложение сохраняет данные и также рассчитывает норму калорий и количество воды в день.

Пользователю показывается главный экран, на котором ему показывается 2 диаграммы , которые характеризуют отношение нормы калорий и воды в день с добавленными калориями и воды на данный момент. при нажатии на одну из диаграмм будет открываться соответствующее окно с блюдами или напитками. В данном окне будет находиться список блюд/напитков, которые при нажатии добавляются для отношение с нормой в день. Еще в нижнем экране будет кнопка для добавления пользовательского блюда/напитка. У данных элементов должно храниться следующие( пользователь при создании своего тоже должен вводить):

* Количество употребления( гр/мл)
* количество калорий/коэффициент воды
* настройка дизайна.

В верхней части имеется список тренировок, при нажатии ,будет открываться список упражнений с кнопкой начать. После чего будет выходить окно с фотографией и описанием упражнения,таймером и кнопками начать, после выполнения открывается следующее окно с упражнением. После окончания последнего упражнения данное окно закрывается.

все данные хранятся в json файлах. Логика и таблицы описаны выше.

## **Разработка алгоритма решения задачи**

Входная форма при вводе данных рассчитывает следующие математические формулы :

1. Для женщин

1.1 dci = (вес\*10 + рост\*6,25-возраст\*5- 165)\*коэффициент активности

1.2 суточная норма воды = вес \* 31

1. Для мужчин

2.1 dci = (вес\*10 + рост\*6,25-возраст\*5+5)\*коэффициент активности

2.2 суточная норма воды = вес \* 35

После этого создается json файл с личными данными и выводами по данным расчетам.

основные форма для ввода информации сделаны с помощью разбиения json файла на массив, последующее добавление и обратное сохранение. вывод данных осуществляется с помощью контроллера , который временно сохраняет данные в массив и дает доступ к каждому элементу. Также можно осуществлять обновление и удаление по элементу в массив, так как он будет совпадать как и с визуальными элементами, так и с id данной записи.

Навигация и осуществления каких-либо нажатий приводит к открытию нового окна, который визуализирован выше главного экрана.

Для отображения пользователю какого количества воды нужно для нормы и нормы калорий в сутки, будет проверяться каждый раз при добавлении суммарное количество с выводящим числом , которое рассчитывается с первого запуска приложения. При выполнении нормы, Приложения не будет выводить ошибку при последующих добавлений, а будет и дальше добавляться только без проверки и также после досягания нормы, пользователю выводится сообщение, что он ее выполнил.ы

## **Структура программы**

Для данного мобильного приложения был выбран язык программирования Kotlin с компонентами Java. Для работы с Json файлами был выбран механизм SharedPreferences.