## Аннотация

Мобильное приложение «Pocket Trainer» предназначено для ведения дневника тренировок спортсмена, отслеживания прогресса тренировочного процесса.

В приложении доступны на выбор более 100 упражнений, которыми может воспользоваться спортсмен, а также обширное количество программ тренировок для разных целей (Набор мышечной массы, похудение, сохранение мышечной массы и т.д.)

## 

**Содержание**

[Введение 4](#_Toc73531374)

[1. Назначение и область применения 5](#_Toc73531375)

[2. Постановка задачи 6](#_Toc73531376)

[3. Описание программы 7](#_Toc73531377)

[3.1 Общие сведения 7](#_Toc73531378)

[3.2 Функциональное назначение 7](#_Toc73531379)

[3.3 Описание логической структуры 8](#_Toc73531380)

[3.4 Используемые технические средства 20](#_Toc73531381)

[3.5 Вызов и загрузка 20](#_Toc73531382)

[3.6 Входные и выходные данные 21](#_Toc73531383)

[4. Программа и методика испытаний 22](#_Toc73531384)

[4.1 Объекты испытаний 22](#_Toc73531385)

[4.2 Цель испытаний 22](#_Toc73531386)

[4.3 Требования к программе 22](#_Toc73531387)

[4.4 Методы испытаний 22](#_Toc73531388)

[4.5 Тестовый пример 23](#_Toc73531389)

[5. Руководство оператора 38](#_Toc73531390)

[5.1 Выполнение программы 38](#_Toc73531391)

[5.2 Сообщение оператору 53](#_Toc73531392)

[6. Мероприятия по информационной безопасности 56](#_Toc73531393)

[Заключение 60](#_Toc73531394)

[Приложения 61](#_Toc73531395)

[Источники, использованные при разработке 65](#_Toc73531396)

**Введение**

## 

Приложение Pocket Trainer является простым и наглядным дневником тренировок и включает в себя множество полезных инструментов для занятий в тренажерном зале.

Разработка мобильного приложения «Pocket Trainer» проводилось на основе следующих документов:

1) Дипломное задание;

2) Техническое задание.

**1. Назначение и область применения**

Назначением мобильного приложения является улучшение условий ведения тренировочного дневника спортсмена. Мобильное приложение будет содержать в себе все необходимые опции по ведению дневника. Пользователь сможет по обычному нажатию узнать всю необходимую информацию о выбранном упражнении.

Существующими аналогами приложения являются:

* Фитнес тренер Fitness Online упражнения тренировки - <https://play.google.com/store/apps/details?id=fitness.online.app&hl=ru&gl=US>
* Твой Тренер: фитнес в зале и дома - <https://play.google.com/store/apps/details?id=com.tvoytrener.timkoilia&hl=ru&gl=US>

**2. Постановка задачи**

Необходимо разработать мобильное приложение для ведения тренировочного процесса спортсмена, которое должно удовлетворять следующим требованиям:

* В мобильном приложении должна присутствовать возможность выбора упражнения или программы тренировок;
* Должен присутствовать календарь тренировок, с помощью которого пользователь сможет ориентироваться по датам;
* На календаре должны быть выделены тренировочные дни для удобства выбора дня;
* Главная страница приложения должна выводить пользователю всю основную информацию о выбранном тренировочном дне. В случае если же пользователь еще не выбрал тренировочный день, выводить информацию о текущем дне;
* Должны присутствовать информационные страницы, на которых будут отображаться данные о выбранном пользователе упражнении или тренировочной программе;
* Должна присутствовать страница со статистикой, в которой можно будет узнать общее количество тренировок;
* Дополнительной возможностью должна быть реализована страница с расчётом индекса массы тела человека.

Индекс массы тела (англ. body mass index (BMI), ИМТ) — величина, позволяющая оценить степень соответствия массы человека и его роста и тем самым косвенно судить о том, является ли масса недостаточной, нормальной или избыточной.

Индекс массы тела рассчитывается по формуле:

Где m – масса тела в килограммах

h – рост в метрах.

**3. Описание программы**

**3.1 Общие сведения**

Наименование приложения «Pocket Trainer».

Мобильное приложение «Pocket Trainer» написано на программном языке C# с помощью фреймворка кроссплатформенной разработки Xamarin Forms.

Локальная БД разработана с помощью SQLite.

Серверная БД разработана с помощью MySQL.

REST API для связи мобильного приложения и серверной БД было написано с помощью языка программирования PHP.

Для полной работоспособности приложения необходим доступ в интернет.

**3.2 Функциональное назначение**

Программа предназначена для решения задач по хранению, обработке и редактированию информации пользователя касающейся тренировочного процесса.

Программа решает задачи по выводу информации пользователю на экран мобильного устройства.

**3.3 Описание логической структуры**

Ниже предоставлен алгоритм взаимодействия мобильного приложения и сервера.



Рисунок 1 - Взаимодействие с сервером

Диаграмма вариантов использования мобильного приложения:



Диаграмма деятельности при добавлении тренировочной рутины в выбранный день:



Рисунок 2 - Диаграмма деятельности добавления тренировочной рутины

Для расчёта индекса массы тела также была создана диаграмма деятельности:



Рисунок 3 - Диаграмма деятельности расчёта ИМТ

Ниже представлена диаграммы деятельности просмотра информации об упражнении/программе тренировок.



Рисунок 4 - Диаграммы деятельности вкладок "Список упражнений" и "Список программ тренировок"

В приложении реализованы два интерфейса:



Рисунок 5 - Диаграмма классов (реализация интерфейсов)

Интерфейс IFile – разработан для работы с файловой системой внутри мобильного устройства. Это нужно для хранения заметок на внутреннем хранилище телефона.

FileManager – класс, реализующий методы интерфейса IFile.

Интерфейс IMessage – разработан для отображения Toast сообщений пользователю в реальном времени. В Xamarin Forms нет собственной реализации данных сообщений поэтому для работы сообщений необходим данный интерфейс.

MessageAndroid – класс, реализующий методы интерфейса IMessage. Находится в Android проекте.

Ниже приведена диаграмма класса RestService.



Рисунок 6 - Диаграмма классов(RestService)

Класс RestService – это класс, предназначенный для связи мобильного приложения и REST API созданного на сервере.

Класс ProcessInfo зависит от класса RestService и использует свойство client класса RestService для отправки запросов на сервер.

В классе ProcessInfo присутствует 8 методов, каждый из которых выбирает какую то конкретную информацию и десериализует ее в сущности, находящиеся в мобильном приложении.

К примеру метод LoadOneExercise – метод, принимающий в параметры id упражнения и возвращающий с сервера всю информацию по выбранному упражнению.



Рисунок 7 - Класс LocalService

Выше показан класс LocalService отвечающий за работу с локальной базой данных на стороне пользователя.

Главным свойством класса является свойство db. Данное свойство отвечает за настройку соединения с локальной базой данных.

Инициализация подключения происходит в методе Init().

А дальше по мере пользования приложением и вызова пользователем команд происходит обращение к соответствующим методам в данном классе.

Например, когда пользователь хочет передвинуть упражнение в тренировочном дне на строчку выше будет вызываться метод MoveExerciseUp и передаваться в качестве параметров номер места упражнения в тренировочном дне (place) а также ID текущего тренировочного дня.

Ниже приведена диаграмма серверной базы данных:

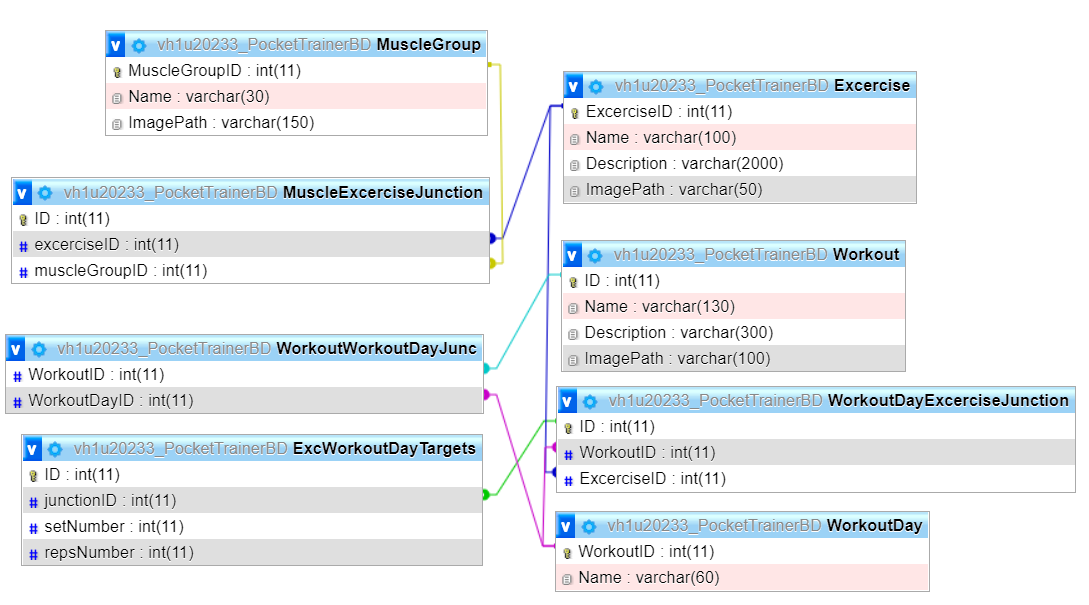


Рисунок 8 - Серверная БД

Серверная бд состоит из 8 таблиц:

* MuscleGroup – таблица с информацией о каждой мышечной группе в базе данных;
* Exсercise – таблица с информацией о каждом упражнении в базе данных;
* MuscleExcerciseJunction – таблица, связывающая упражнение и группу мышц между собой;
* Workout – таблица с информацией о каждой программе тренировок в базе данных;
* WorkoutDay – таблица с информацией о тренировочных днях;
* WorkoutWorkoutDayJunc – таблица, связывающая тренировочные дни и программы тренировок;
* WorkoutDayExcerciseJunction – таблица, связывающая упражнение и тренировочный день.
* ExcWorkoutDayTargets – таблица с информацией о количестве подходов и повторений определенного упражнения в тренировочном дне.

Каждая таблица (за исключением WorkoutWorkoutDayJunc) имеет первичные ключи с автоинкрементируемым значением.

В мобильном приложении присутствует 4 сущности, которые с помощью SQLite преобразуются в таблицы в локальной базе данных.

Ниже представлены таблицы локальной БД:

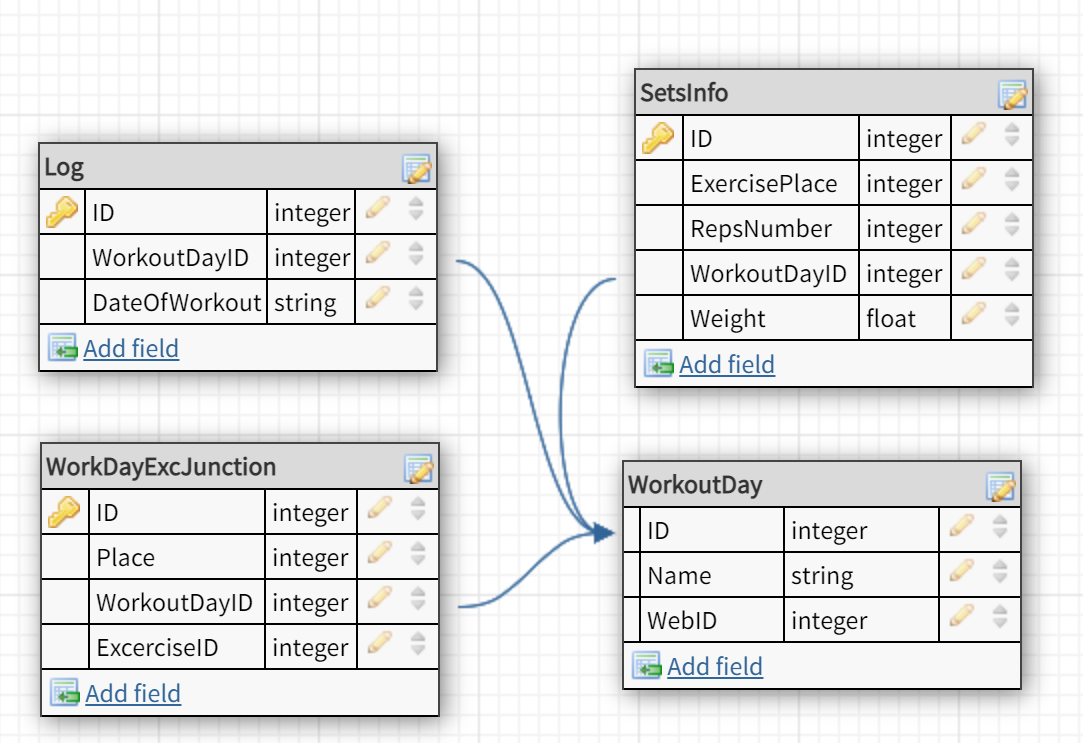


Рисунок 9 - Локальная БД

Таблицы:

* Log – таблица, отвечающая за то, какой тренировочный день на какой дате стоит;
* SetsInfo – таблица, в которой хранится информация о тренировочных сетах;
* WorkoutDay – таблица, в которой содержится информация о тренировочных днях пользователя;
* WorkoutDayExcJunction – таблица, связывающая WorkoutDay и серверную таблицу Exercise с помощью поля ExcerciseID.

Связь между мобильным приложением и сервером осуществляется с помощью REST API.

Representational State Transfer (REST) в переводе — это передача состояния представления. Технология позволяет получать и модифицировать данные и состояния удаленных приложений, передавая HTTP-вызовы через интернет или любую другую сеть.

Если проще, то REST API — это когда серверное приложение дает доступ к своим данным клиентскому приложению по определенному URL.

Файлы REST API расположены на сервере и распределены по папкам в следующем порядке:



Рисунок 10 - REST API

Файл dbconnect.php – это файл для подключения к серверной базе данных.

В папке objects располагаются все объекты, на основе которых осуществляется выборка данных.

В обособленных папках под каждый объект созданы необходимые операции.

Информация с серверной базы данных передается в JSON-формате:

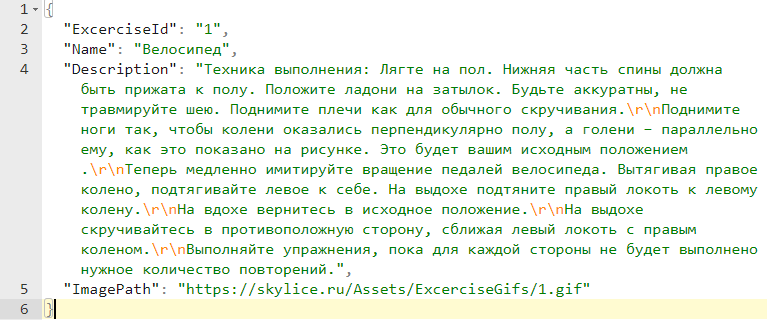


Рисунок 11. JSON-информация

Таким образом информацию получает мобильное приложение. Далее с помощью библиотеки NewtonSoft.Json данные полученные в виде json файла десериализуются по классам внутри программы.

**3.4 Используемые технические средства**

Для использования мобильного приложения необходимо иметь мобильное устройство с операционной системой Android не менее версии 5.0.

Для корректной работоспособности приложения необходим доступ в интернет.

**3.5 Вызов и загрузка**

Для запуска мобильного приложения необходимо нажать на иконку на рабочем столе мобильного устройства.

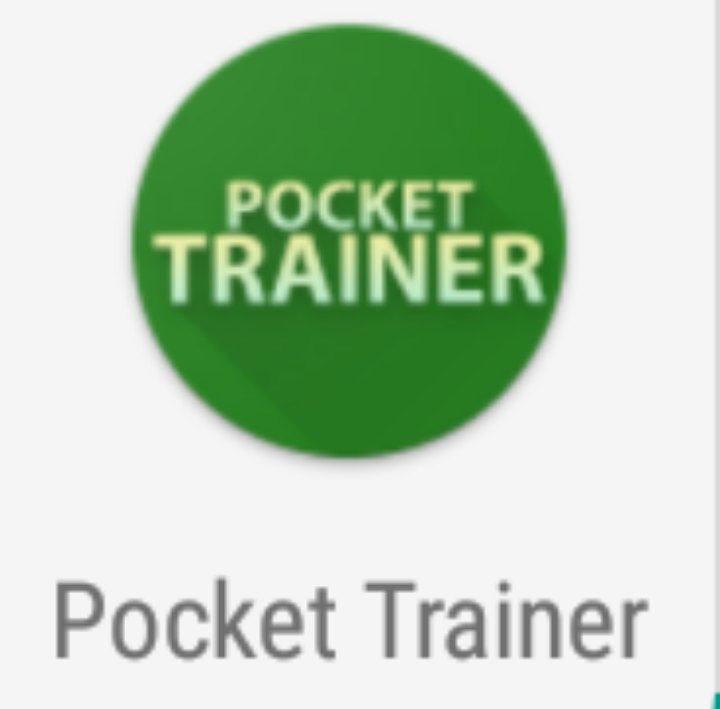


Рисунок 12 - Иконка приложения в телефоне

**3.6 Входные и выходные данные**

Входными данными в мобильном приложении являются:

* Добавление пользователем упражнения, тренировочного дня или прошлой тренировки;
* Изменение значения тренировочного подхода (вес и количество);
* Добавление нового подхода;
* Удаление подхода;
* Удаление упражнения из тренировочного дня;
* Перемещение упражнения в тренировочном дне;
* Смена выбранной даты на календаре;
* Команды пользователя по нажатию на кнопки в боковом меню;
* Выбор выборки на странице статистики;
* Ввод пользователем значений на странице с расчётом индекса массы тела;
* Значения, вводимые пользователем в редакторе записей на странице записок;

Выходные данные:

* Календарь;
* Списки упражнений пользователя в конкретный день или их отсутствие;
* Списки существующих программ тренировок;
* Списки упражнений;
* Информация по выбранному упражнению;
* Индекс массы тела и результат расчётов;
* Созданные пользователем записки внутри приложения;
* Список мышечных групп;

**4. Программа и методика испытаний**

**4.1 Объекты испытаний**

Объектом для испытаний является мобильное приложение «Pocket Trainer» и его функционал.

**4.2 Цель испытаний**

Испытания проводятся с целью проверки соответствия проекта требованиям, указанным в техническом задании.

**4.3 Требования к программе**

Соответствие программного продукта требованиям, указанным в техническом задании.

**4.4 Методы испытаний**

Испытания проводятся в следующем порядке:

* Тестирование главной страницы – Главная страница должна быть исправна, все элементы, расположенные на ней должны корректно отвечать на пользовательские команды;
* Внешний вид главного меню – Главное меню должно быть легко читаемым и интуитивно понятным;
* Наличие и работоспособность информационных страниц – Информационные страницы об упражнениях и программах тренировки должны присутствовать в приложении и быть доступны для пользователя;
* Наличие и работоспособность страницы с заметками – Страница с заметками должна присутствовать в приложении и пользователь должен иметь возможность воспользоваться этой функцией;
* Проверка правильности расчёта индекса массы тела – Индекс массы тела должен рассчитываться на отдельной страницы, ввод данных должен происходить корректно и результат в виде заключения и самого индекса должны рассчитываться правильно и предоставляться пользователю;
* Наличие и работоспособность страницы со статистикой – Страница со статистикой должна присутствовать в мобильном приложении, а также быть доступной для пользователя и корректно работать;
* Добавление упражнения, тренировки в текущий день – Выбранное упражнение или тренировка должны корректно добавляться в указанный пользователем день;
* Удаление упражнения, изменение порядка упражнений – Выбранное упражнение должно удаляться из тренировочного дня по команде пользователя. Перестановка порядка упражнений должна выполняться и не заходить за границы тренировочного дня;
* Добавление и редактирование подходов к упражнению – При желании у пользователя должна быть возможность добавить подход и отредактировать его;

**4.5 Тестовый пример**

Тестирование главной страницы.

Главная страница открывается при запуске приложения выглядит следующим образом. В верхней части страницы расположен календарь который позволяет пользователю выбрать необходимый день для тренировки. Для удобства просмотра упражнений в тренировочный день была сделана кнопка сокрытия календаря.

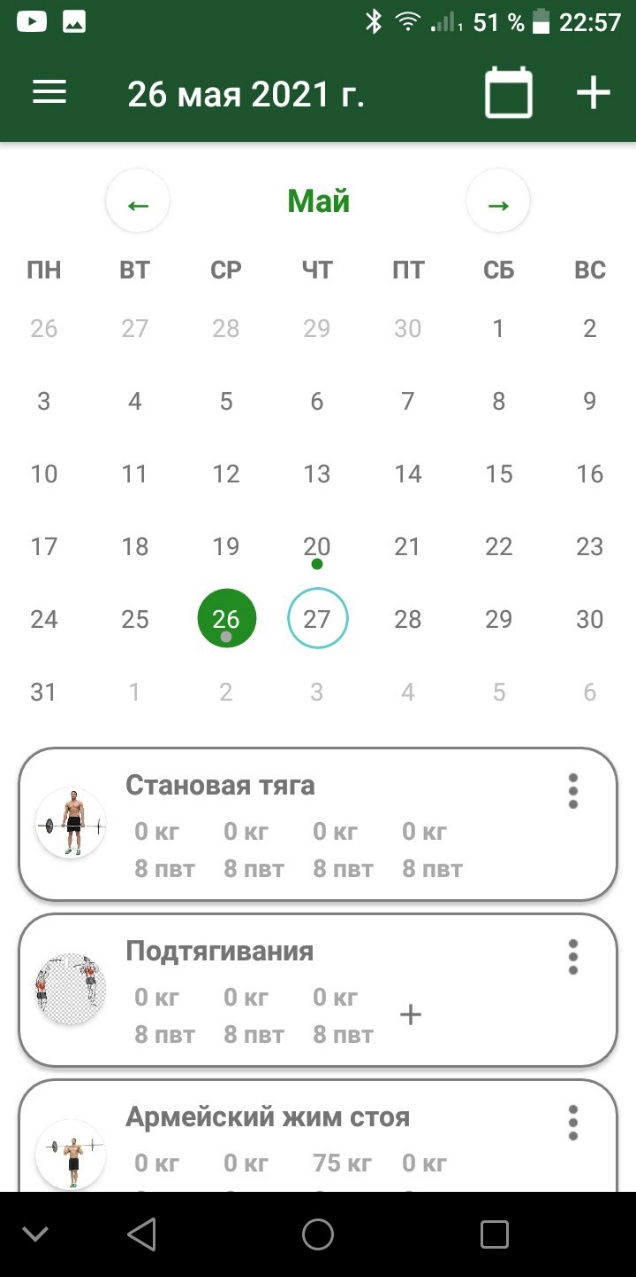
 

Рисунок 13 - Главная страница с календарем/без

Главное меню.

Для запуска главного меню пользователю необходимо нажать на соответствующую кнопку в левом верхнем углу экрана мобильного устройства или же провести пальцем с левого края экрана в правый. Главное меню приложения выполнено в виде бокового меню, на котором расположены все доступные для пользователя опции.

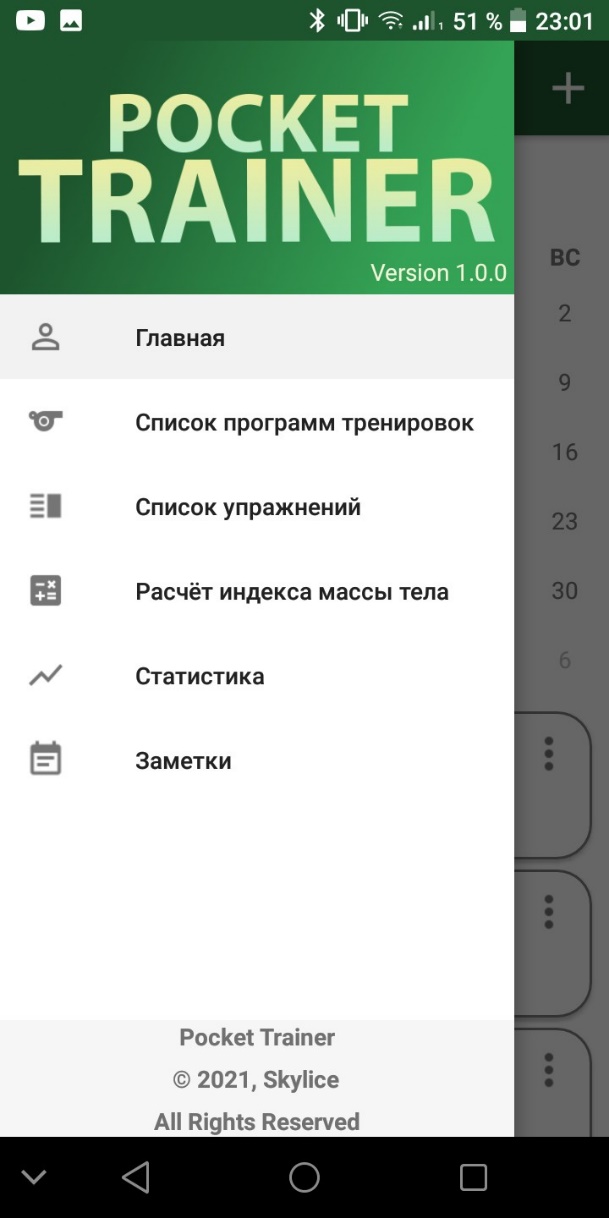


Рисунок 14 - Главное меню

Главное меню выглядит просто и при этом понятно. Отчётливо видна каждая доступная пользователю вкладка. В верхней части меню расположен логотип приложения, а также его текущая версия.

Информационные страницы.

Чтобы найти страницу с упражнениями необходимо нажать на соответствующую кнопку в главном меню приложения (Рисунок 14.).

После выбора вкладки приложение перенесет пользователя на окно выбора группы мышц. Все упражнения разбиты по группам мышц и чтобы найти нужное необходимо сначала выбрать группу мышц. Для тестирования выберем панель «Грудь».

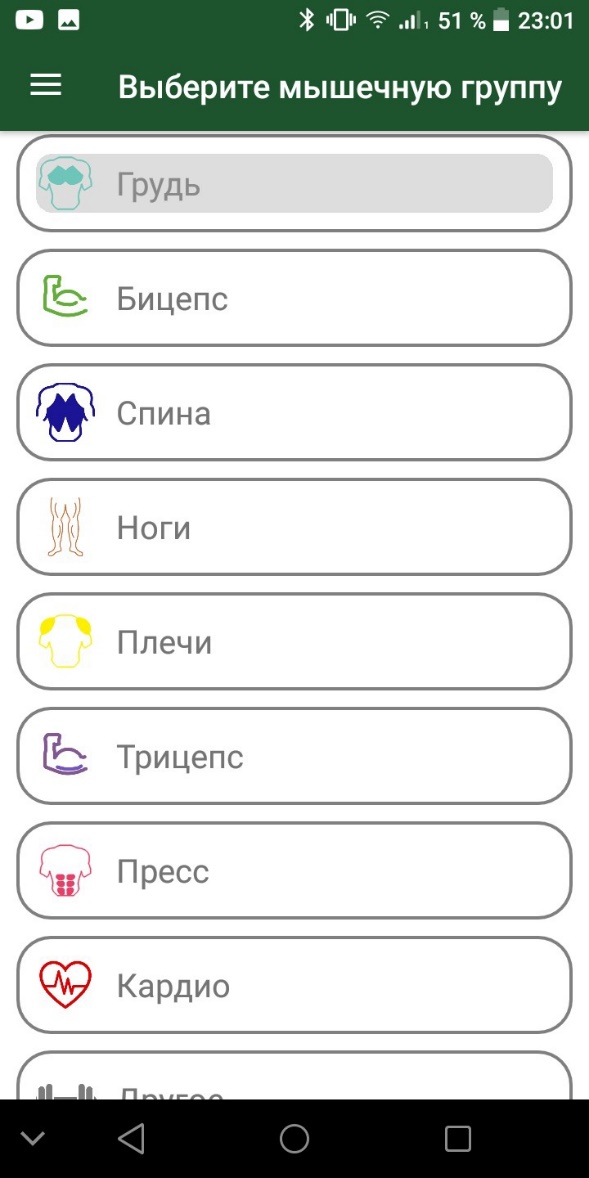


Рисунок 15 - Список мышечных групп

После выбора группы мышц окно сменяется на окно выбора упражнения. Так как была выбрана Грудь в качестве группы мышц, вывелись все упражнения задействующие данную мышечную группу (Рисунок 13. рисунок слева).

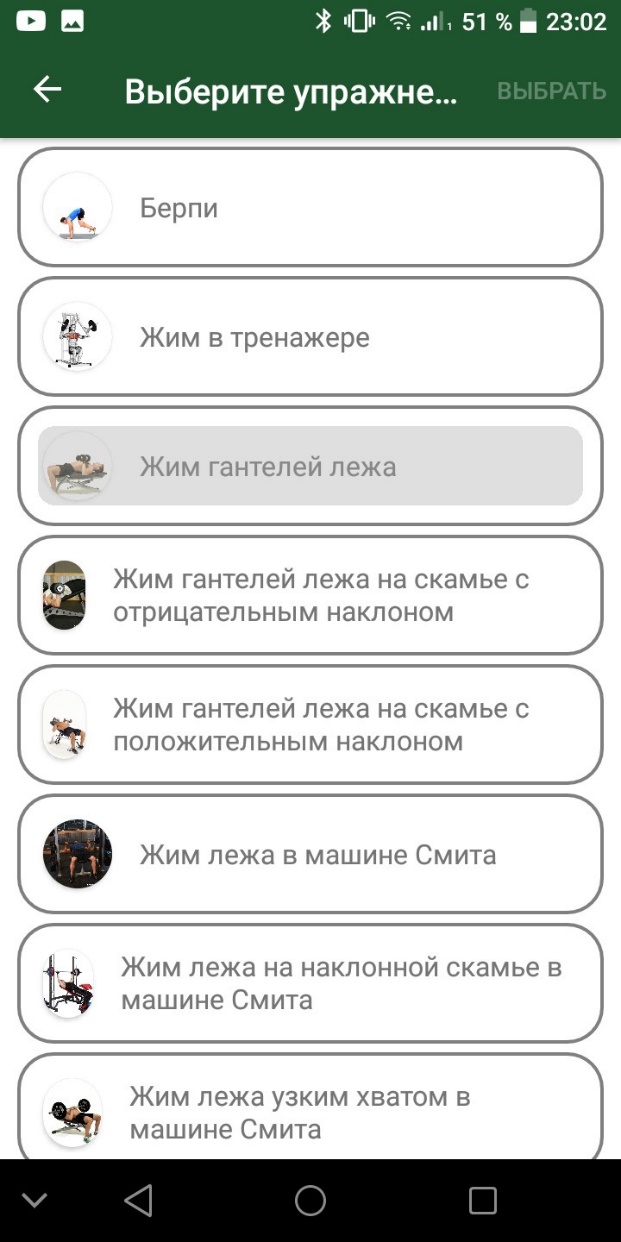
 

Рисунок 16 - Список упражнений/ Информация об упражнении

После того, как все упражнения загрузятся выбираем упражнение «Жим гантелей лежа» с помощью нажатия пальцем. После нажатия страница успешно сменяется страницей с информацией о выбранном упражнении. Вся информация на странице успешно загружается и отображается для пользователя (Рисунок 16. рисунок справа).

Страница с заметками.

Для перехода на страницу с заметками необходимо перейти в боковое меню и выбрать соответствующую вкладку.

После перехода на страницу будет предоставлен список всех заметок, сделанных пользователем за время работы с приложением. Если же записок нет, то будет выведено соответствующее сообщение (Рисунок 17. Рисунок слева) Для добавления новой записи необходимо нажать на кнопку в верхнем правом углу страницы.

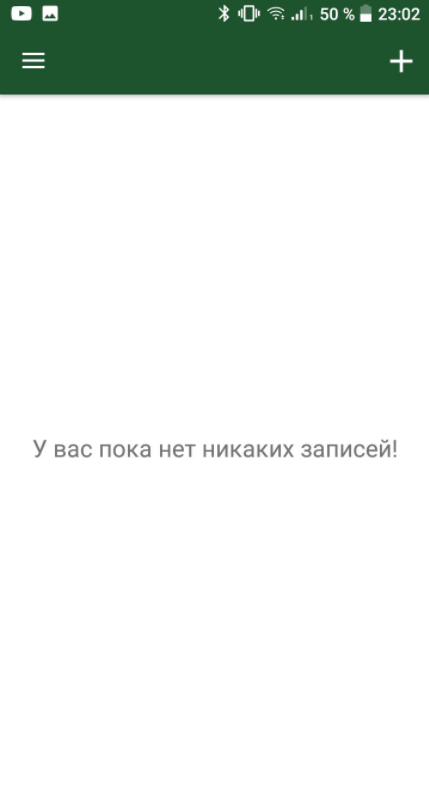
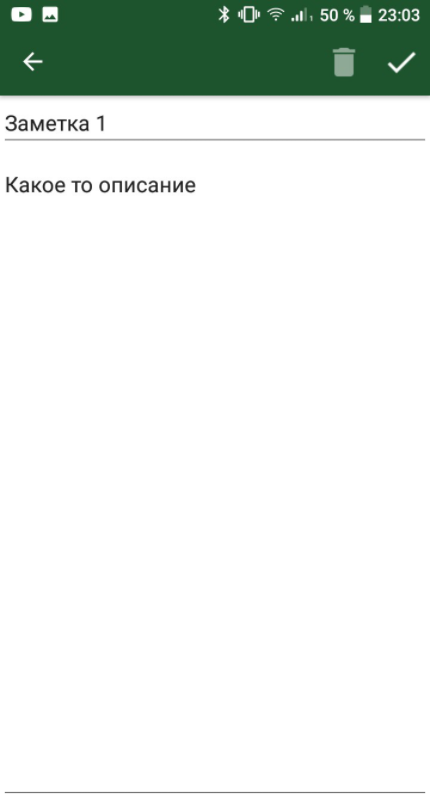
 

Рисунок 17 - Страница Заметки

На странице создания заметки можно редактировать название заметки, а также ее содержимое. После того как заметка была отредактирована ее можно сохранить при помощи кнопки в правом верхнем углу экрана. Для удаления присутствует кнопка в виде мусорного бака, которая позволяет удалить редактируемую заметку.

Расчёт ИМТ (Индекс массы тела).

Для того чтобы перейти в раздел расчёта ИМТ необходимо выбрать нужную вкладку в боковом меню мобильного приложения. После откроется страница с расчётом индекса массы тела. (Рисунок 18. Рисунок слева) В правом верхнем углу расположена кнопка справки. После нажатия на нее успешно открывается окошко, рассказывающее о том, что такое ИМТ. (Рисунок 18. Рисунок справа)

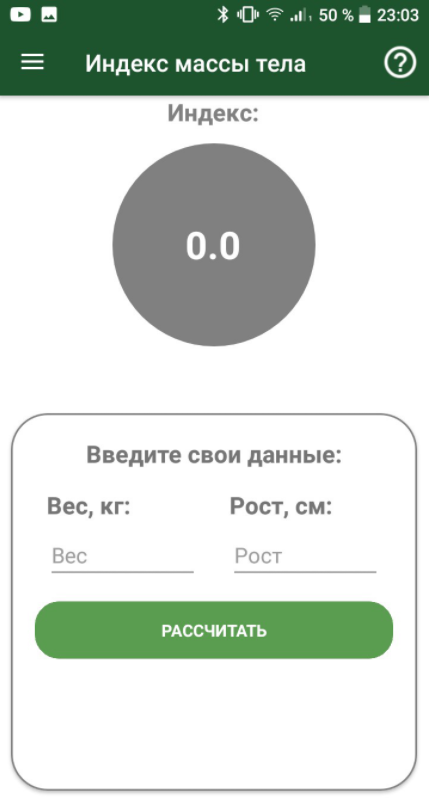
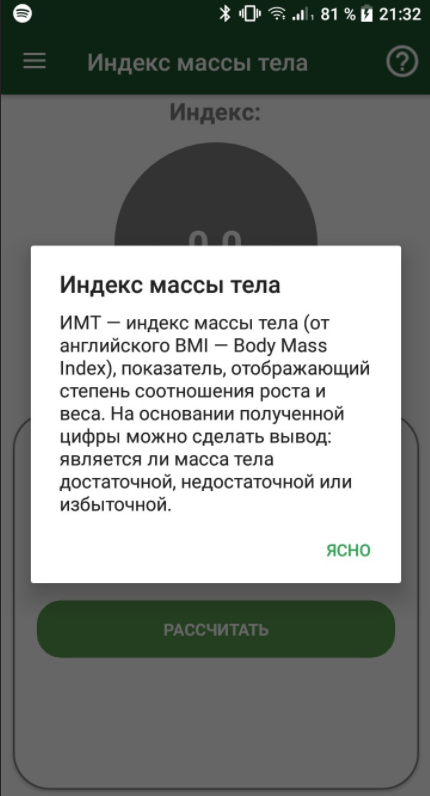
 

Рисунок 18 - Индекс массы тела

Для начала расчёта индекса массы тела необходимо ввести данные о весе и росте пользователя. При некорректно введенных данных пользователю выдаст сообщение, оповещающее его об этом. (Рисунок 19. Рисунок слева)

После ввода данных и нажатия на кнопку «Рассчитать» пользователю выведется результат. (Рисунок 19. Рисунок справа)

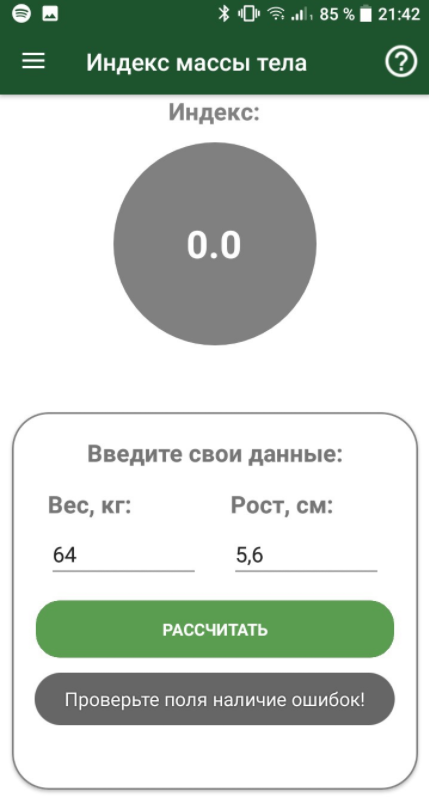
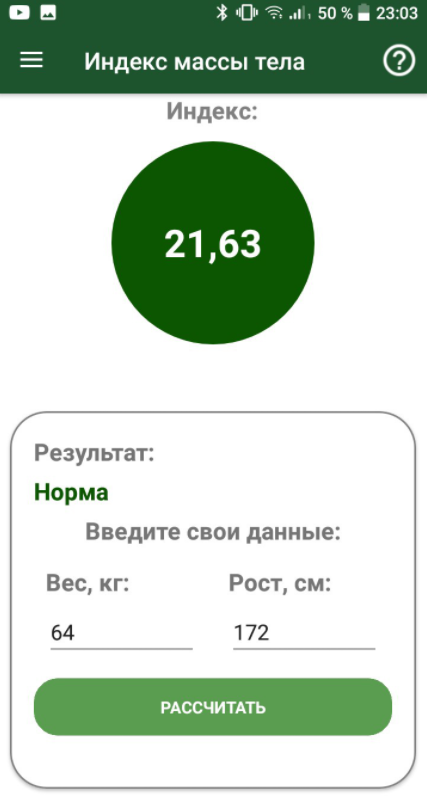
 

Рисунок 19 - Расчёт

Результат рассчитан корректно следовательно тестирование прошло успешно.

Статистика.

Для перехода в окно со статистикой необходимо нажать на соответствующую вкладку в боковом меню мобильного приложения.

После пользователя перенесет в окно со статистикой.

Здесь отображается вся основная информация, касающаяся тренировочного процесса пользователя за все время. В верхней части страницы отображена общая информация о количестве выполненных тренировок, упражнений и тренировочных подходов. (Рисунок 17. Основная информация)

В нижней же части находится график, с помощью которого пользователь может отследить количество проведенных тренировок в месяц или в год. (Рисунок 20. График)

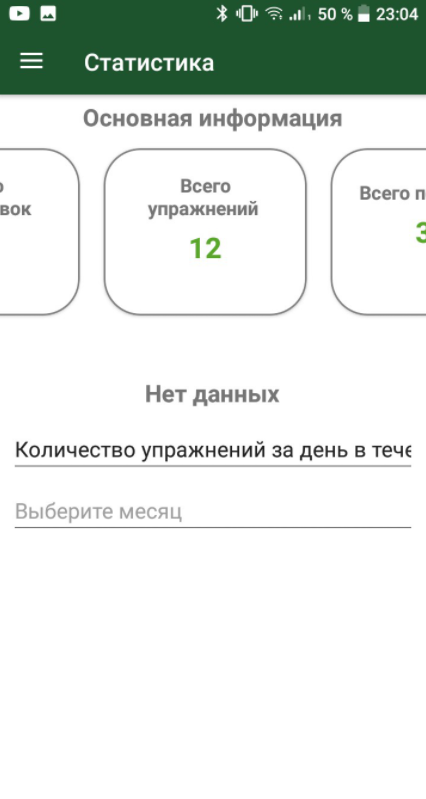
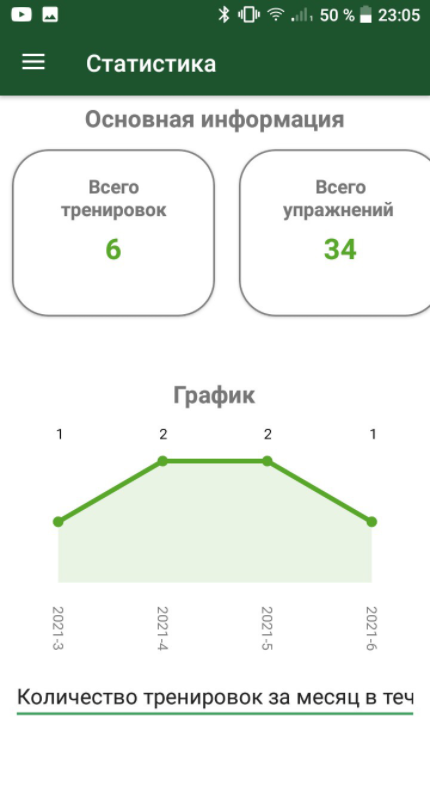
 

Рисунок 20 - Страница со статистикой

Добавление упражнения в тренировочный день.

Для добавления упражнения необходимо перейти на главную страницу мобильного приложения, выбрать тренировочный день и нажать на кнопку в форме плюса в верхней панели управления.

После нажатия пользователю предложат выбрать один из нескольких вариантов: Добавить упражнение, программу или прошлую тренировку.

Для тестирования выберем вариант добавления упражнения.

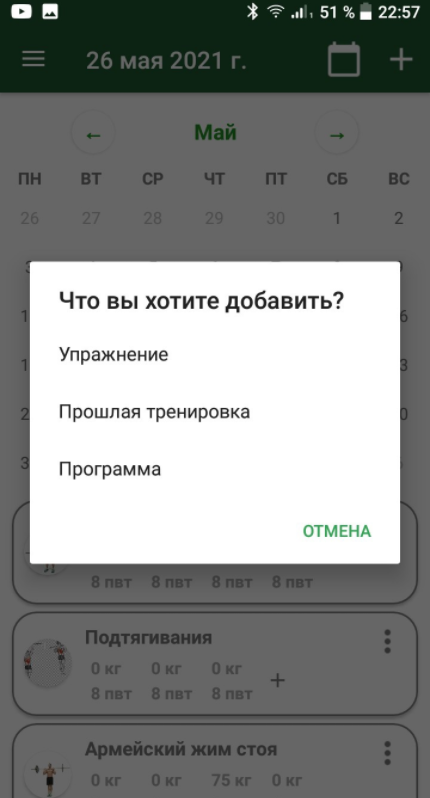


Рисунок 21 - Добавление

После выбора добавления упражнения пользователя перенесет на страницу с выбором мышечной группы. После выбора пользователем группы мышц его перенесет на страницу выбора упражнения. Для выбора упражнения необходимо одним нажатием выбрать нужное. После выбора упражнение выделится, а кнопка «выбрать» в правом верхнем углу экрана станет доступной.

По нажатию на кнопку упражнение успешно добавится в тренировочный процесс.

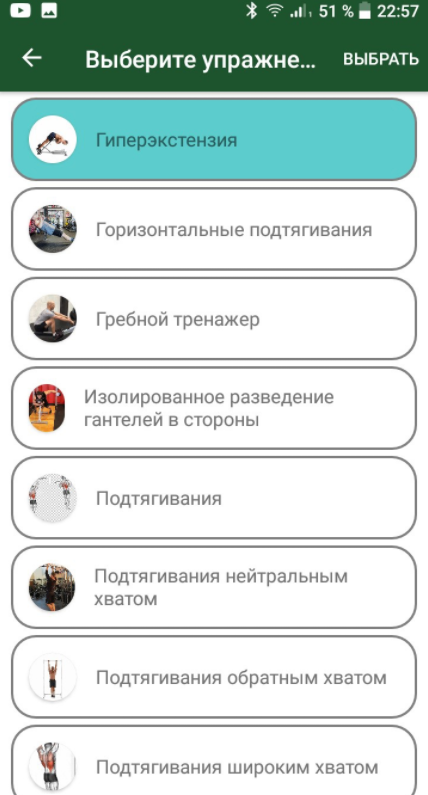
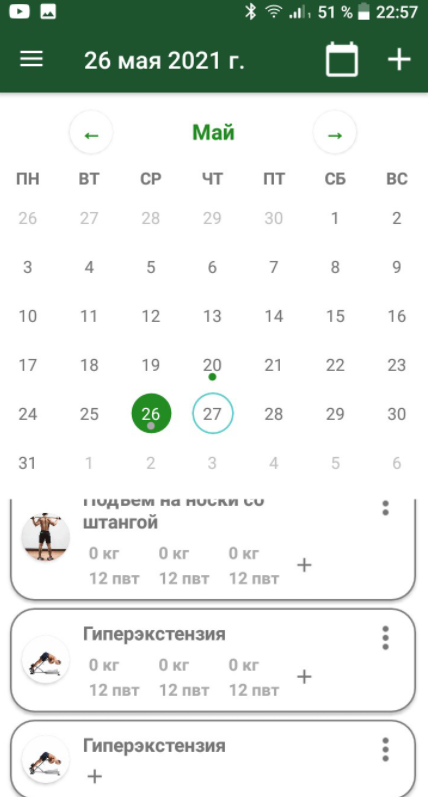
 

Рисунок 22 - Добавление упражнения

Удаление упражнения, изменение порядка упражнений.

Для удаления или перемещения упражнения в списке необходимо нажать на кнопку, находящуюся в правом верхнем углу панели с нужным упражнением. Она выполнена в виде трех вертикальных точек.

Впоследствии нажатия на кнопку откроется меню, в котором пользователю будут предоставлены все возможные действия с упражнением.

Для удаления выбираем вариант «Удалить». После этого упражнение успешно удаляется из тренировочного дня.

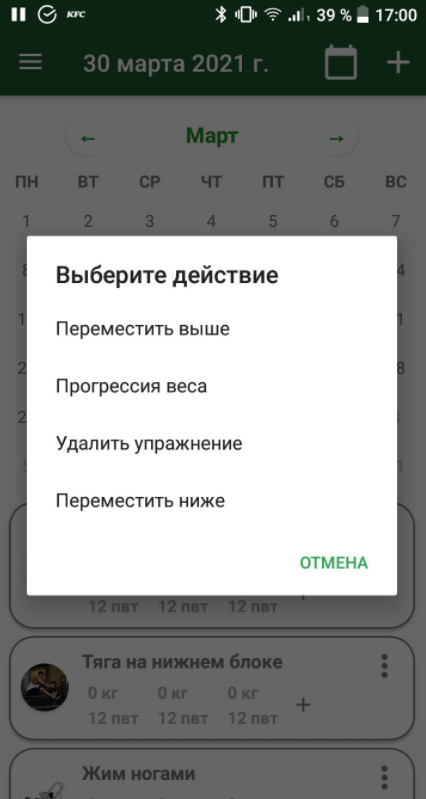
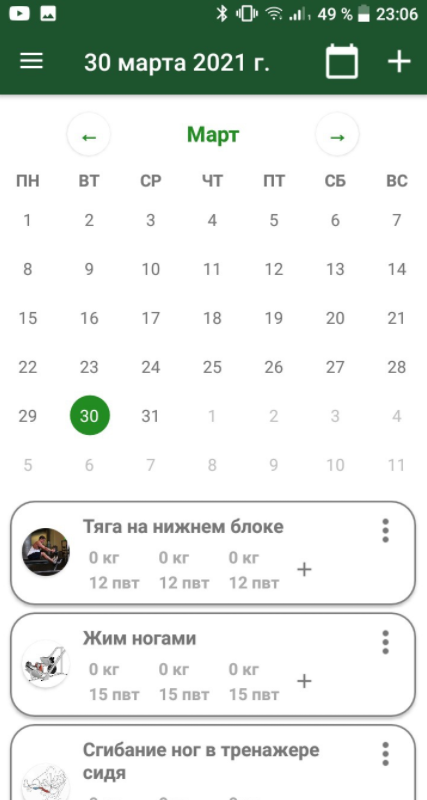
 

Рисунок 23 - Удаление

Удаление производится успешно, а все упражнения которые были ниже смещаются на один элемент вверх.

Для перемещения по списку выбираем необходимое упражнение, нажимаем на вспомогательную кнопку на панели упражнения, выбираем нужное направление (Вверх-вниз). После этого в зависимости от местоположения упражнения в тренировочном дне оно либо переместится, либо нет. (Рисунок 24-25.)

К примеру если упражнение будет стоять на первом месте в списке и пользователь захочет его поставить еще выше, ему будет выведено сообщение, оповещающее его о невозможности данного действия. (Рисунок 25. Рисунок справа)

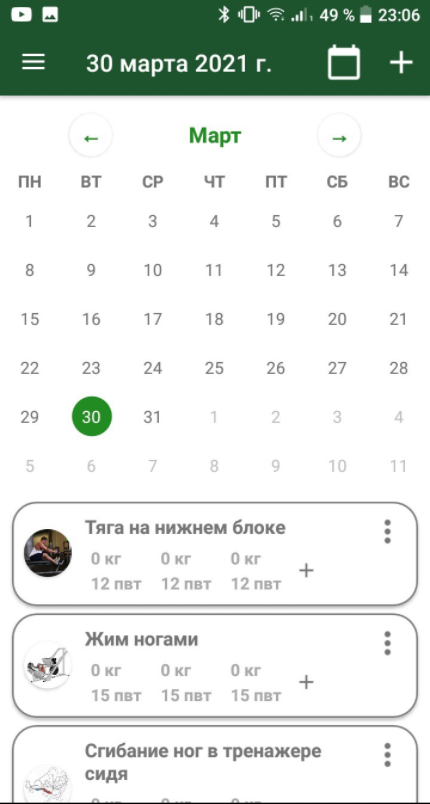
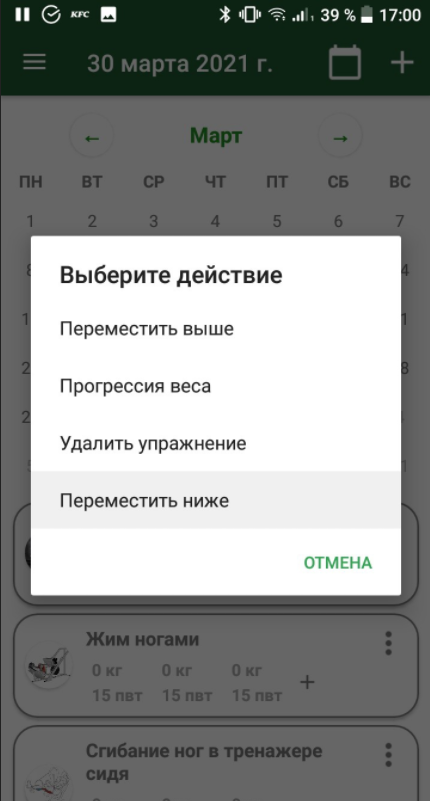
 

Рисунок 24 - Перемещение (Часть 1)

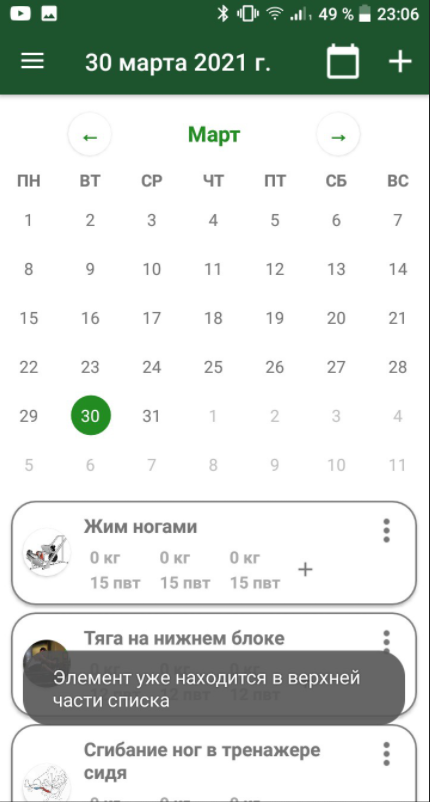
 

Рисунок 25 - Перемещение (Часть 2)

Добавление и редактирование подходов к упражнению.

Для добавления подхода необходимо нажать на кнопку в виде плюса на панели с необходимым упражнением. После нажатия в панель с упражнением успешно добавится новый подход с нулевыми значениями.

Для редактирования подхода необходимо нажать по нужному подходу пальцем. После этого пользователю на экран всплывет меню, в котором он сможет поменять значения у выбранного подхода.

При вводе неправильных значений кнопка «ОК» не будет доступна для нажатия, что защищает приложение от ввода некорректных данных.

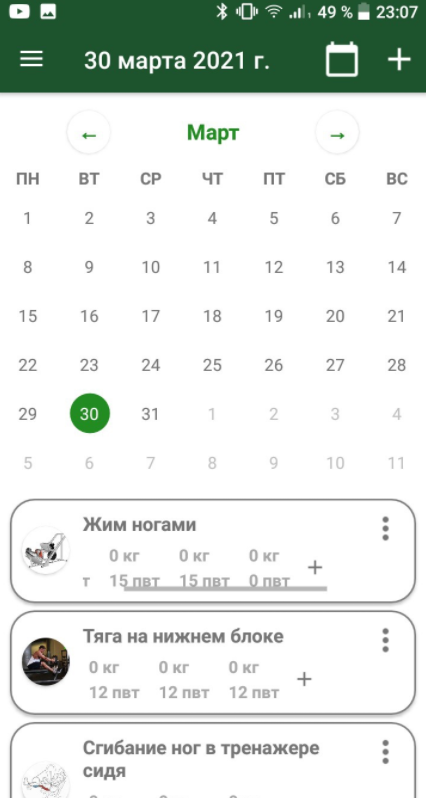


Рисунок 26 - Добавление подхода

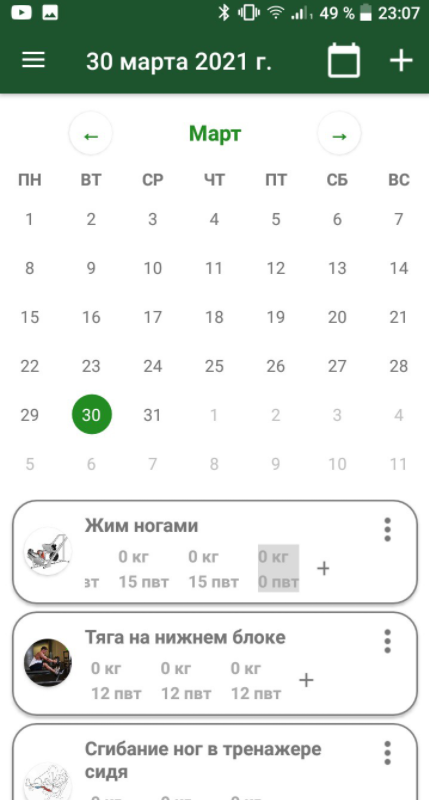
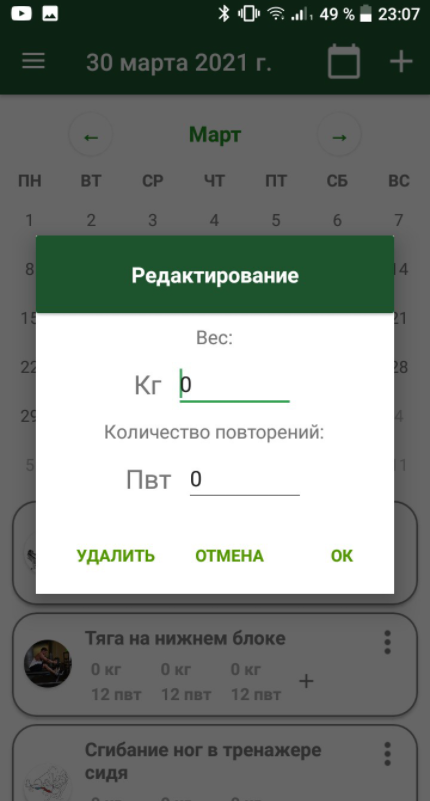
 

Рисунок 27 - Редактирование подхода

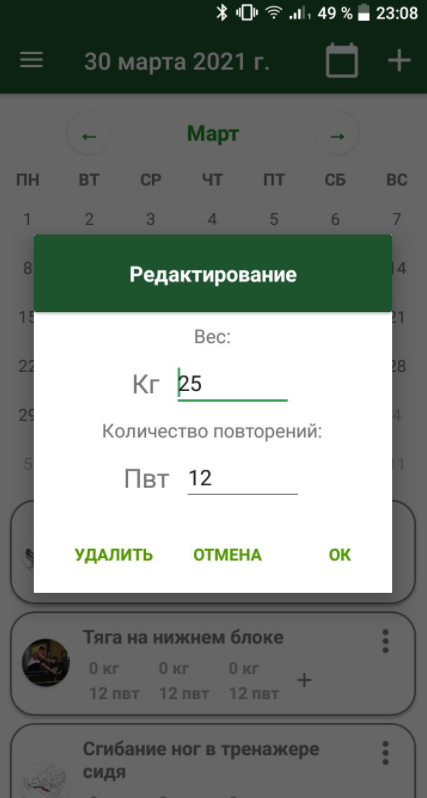
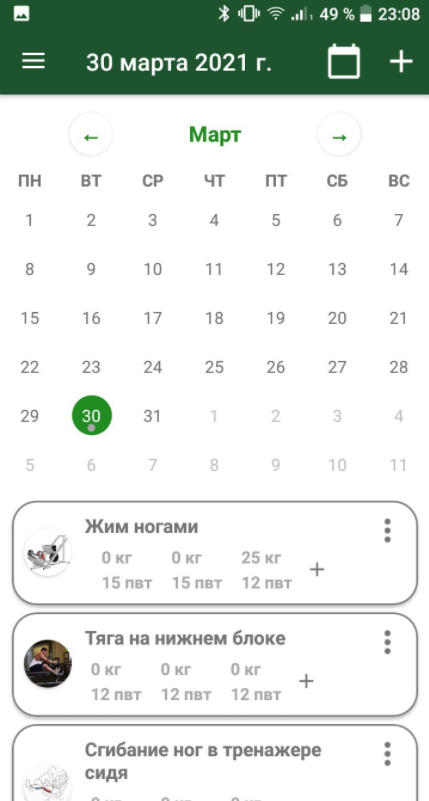
 

Рисунок 28 - Редактирование подхода (Часть 2)

Заключение тестирования.

В ходе прохождения тестирования никаких неисправностей выявлено не было. Были протестированы все основные функции мобильного приложения. В ходе тестирования все элементы работали исправно и никаких ошибок не происходило.

**5. Руководство оператора**

**5.1 Выполнение программы**

Для начала работы с приложением необходимо найти его иконку в меню Android или же на рабочем столе. Для запуска приложения необходимо нажать на него пальцем.

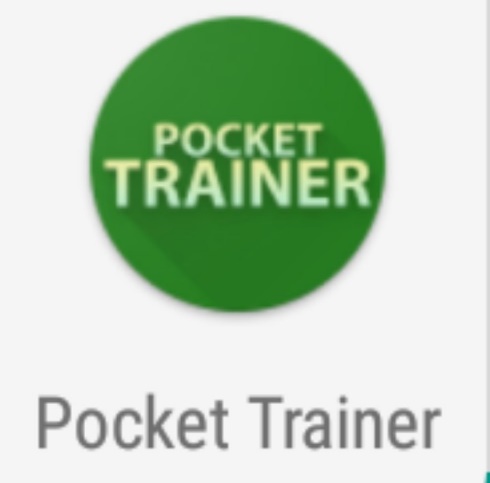


Рисунок 29 - Иконка

После нажатия приложение начнет запускаться. После загрузки откроется главная страница приложения. На этой странице расположены следующие элементы:

* Панель инструментов – находится в верхней части окна приложения, на главной странице на ней расположены кнопка вызова меню, текущая дата, а также две вспомогательные кнопки: кнопка в виде календаря, позволяющая скрыть или показать календарь, а также кнопка в виде плюса, позволяющая добавить упражнение или список упражнений в тренировочный день;
* Календарь – является одним из главных элементов на странице и позволяет выбирать необходимую дату для создания тренировки;
* Список упражнений – панель, находящаяся под календарем на главной странице. На ней собраны все упражнения, которые были добавлены на этот день. В случае если упражнения отсутствуют – выводится соответствующее сообщение.

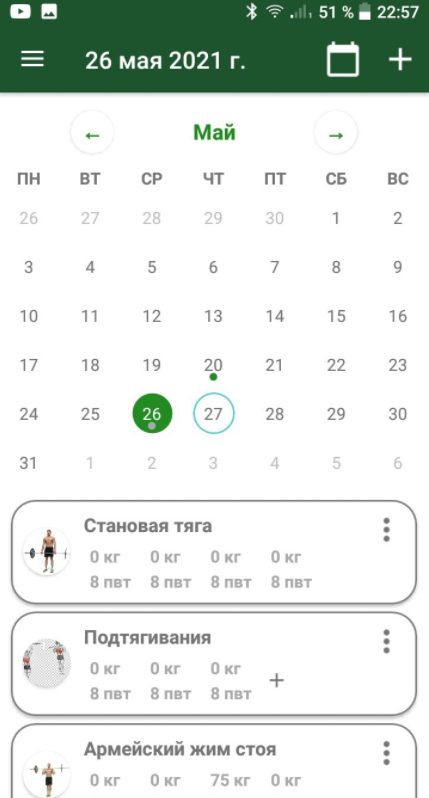
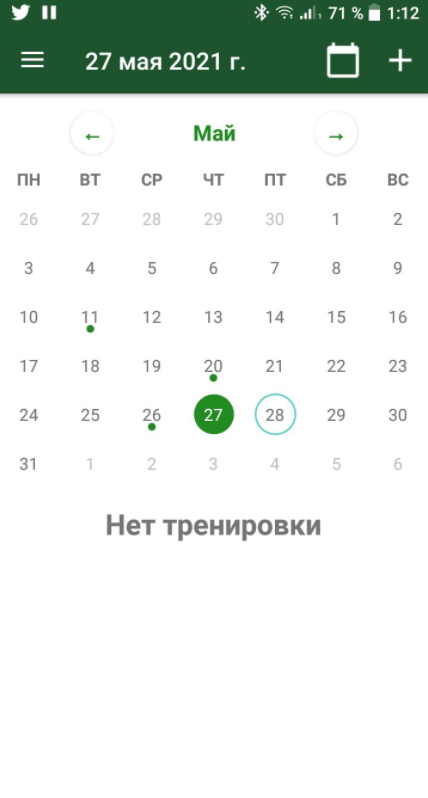
 

Рисунок 30 - Главная страница с тренировкой/без

Для выбора нужного дня необходимо нажать по нему пальцем. Чтобы листать месяцы в календаре необходимо либо использовать кнопки «стрелки» в верхней части календаря, либо листать месяцы с помощью пальца проводя от одного края экрана в другой.

Точки на календаре показывают дни, в которых уже присутствует тренировочная программа.

По желанию есть возможность скрыть календарь нажав на кнопку в виде календаря в правой верхней части экрана.

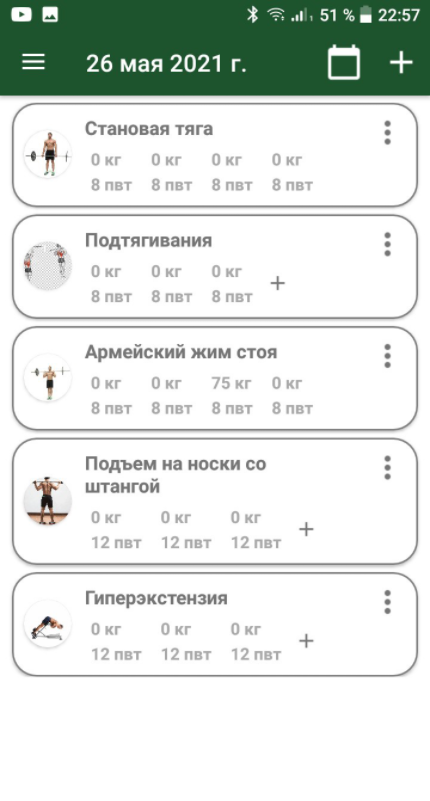


Рисунок 31 - Скрытый календарь

Чтобы добавить упражнение или программу тренировок в тренировочный день необходимо выбрать нужный день, а затем нажать на кнопку в виде плюса в правой верхней части страницы.

Далее откроется меню из которого предоставляется возможность добавления чего-то более конкретного: упражнения, прошлой тренировки или же программы.

Чтобы добавить упражнение выбираем пункт меню «Упражнение» и переходим в окно выбора группы мышц.

Здесь необходимо выбрать группу мышц, на которую будет выполняться упражнение. Выбираем необходимую посредством нажатия пальцем.

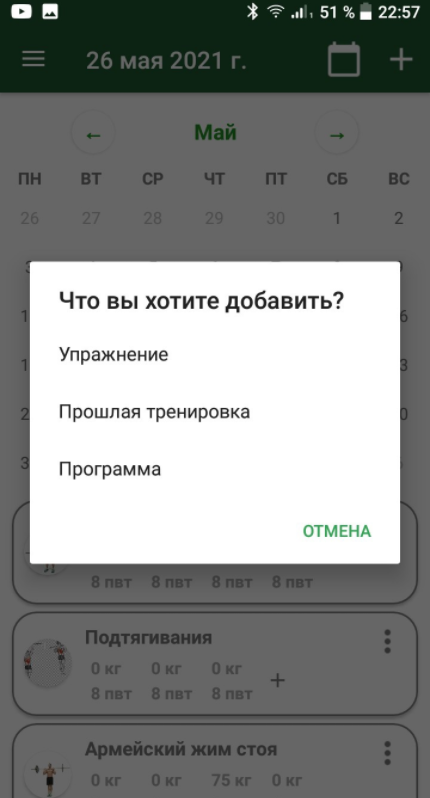
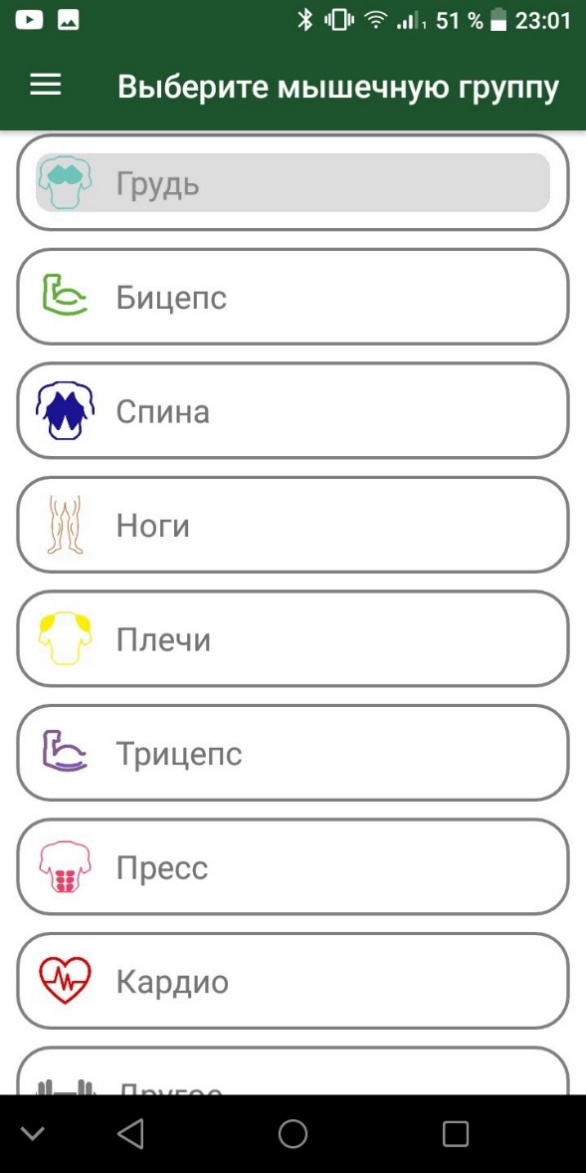
 

Рисунок 32 - Добавление упражнения

После выбора группы мышц предоставляется список всех упражнений, которые задействуют выбранную мышечную группу.

Для выбора необходимо нажать пальцем по нужному упражнению. После этого оно выделится светло голубым цветом и появится возможность нажать на кнопку «Выбрать» в правом верхнем углу страницы.

После нажатия на кнопку приложение перенесет обратно на главную страницу, где уже будет добавлено выбранное ранее упражнение.

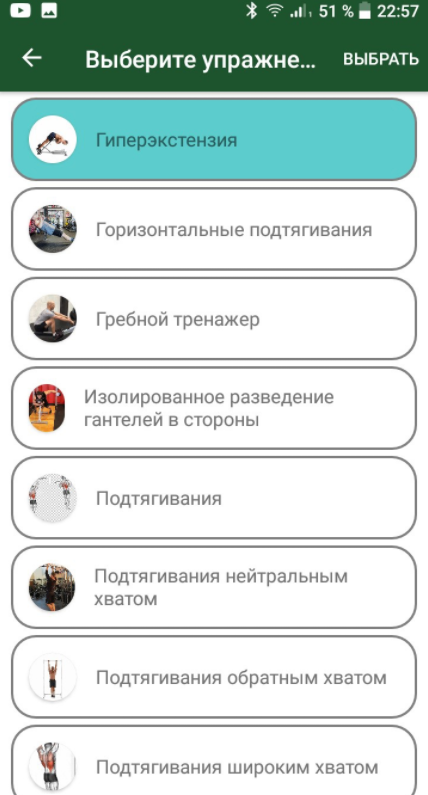
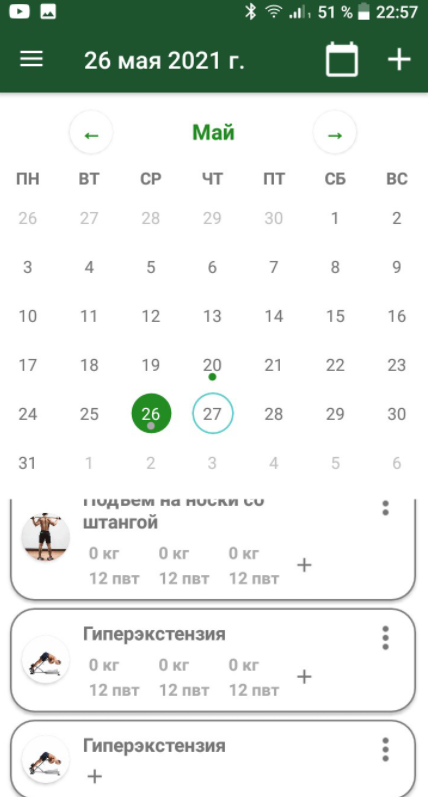
 

Рисунок 33 - Добавление упражнения (Часть 2)

Аналогичным способо работают и добавление программы и прошлой тренировки.

Чтобы удалить ненужное упражнение необходимо нажать на кнопку, находящуюся на панели упражнения в правом верхнем углу. Выполнена она в стиле трех вертикально расположенных точек.

После нажатия на кнопку выведется вспомогательное меню, где расположены следующие поля:

* Переместить выше;
* Прогрессия веса;
* Удалить;
* Переместить ниже.

Для удаления необходимо выбрать кнопку «Удалить». После этого упражнение успешно удалится из тренировочного дня.

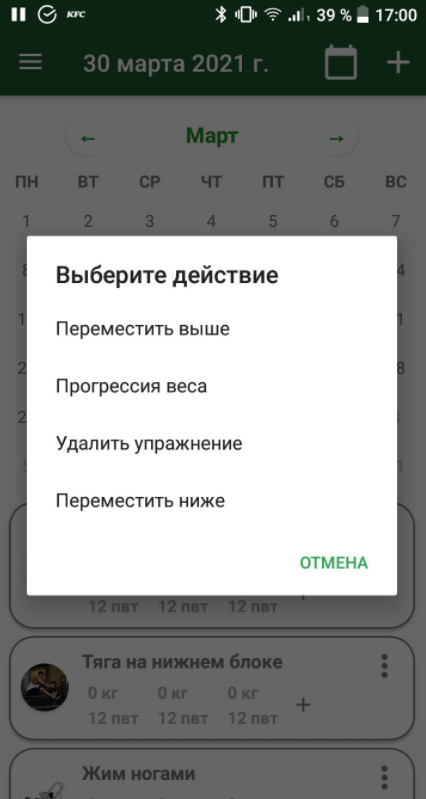
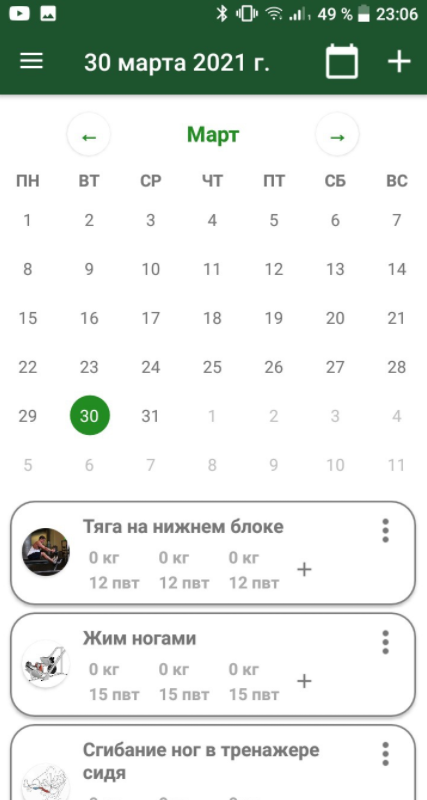
 

Рисунок 34 - Удаление упражнения

Для перемещения упражнения по списку в тренировочной программе необходимо все также нажать на кнопку с тремя точками и выбрать необходимый пункт. (Рисунок 34. Рисунок слева)

После чего упражнение успешно переместится, при условии, что оно не находится на краю тренировочной программы, и вы не пытаетесь переместить его за пределы тренировки.

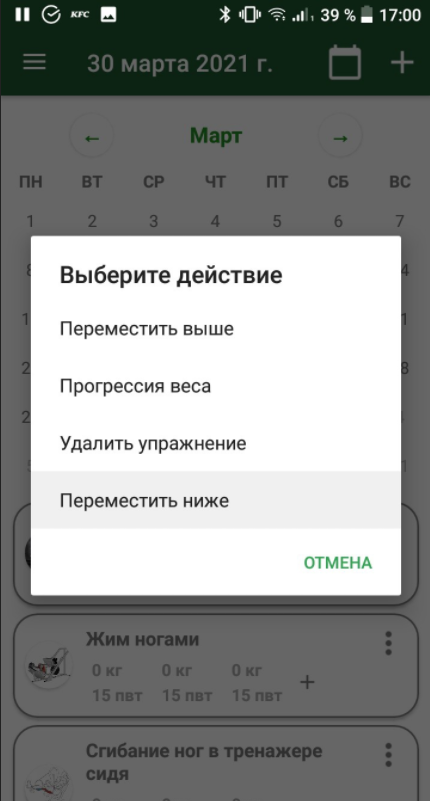
 

Рисунок 35 - Перемещение упражнения

Для того, чтобы отследить прогрессию веса определенного упражнения необходимо нажать на «Прогрессия веса» (Рис. 36) и далее необходимое окно, на котором будет предоставлена вся информация по конкретному упражнению.

Пользователь может узнать максимальный вес, с которым делалось упражнение, общее число повторений, а также сколько раз в тренировках оно было выбрано за всё время.

С помощью графика можно наглядно увидеть как росла или падала прогрессия веса в определенный промежуток времени. Для выбора дано отображение по месяцам, а также за целый год.

Рисунок 36 - Прогрессия веса

Для добавления нового подхода в упражнение необходимо нажать на плюс, находящийся на панели выбранного упражнения под названием.

После этого упражнению добавится новый подход. Изначально и вес и количество у нового подхода будет равным нулю. Для изменения параметров подхода необходимо нажать по нему пальцем. После этого выведется окошко, в котором можно сменить количество и вес выбранного подхода.

В случае неправильного ввода данных кнопка «ОК» будет заблокирована.

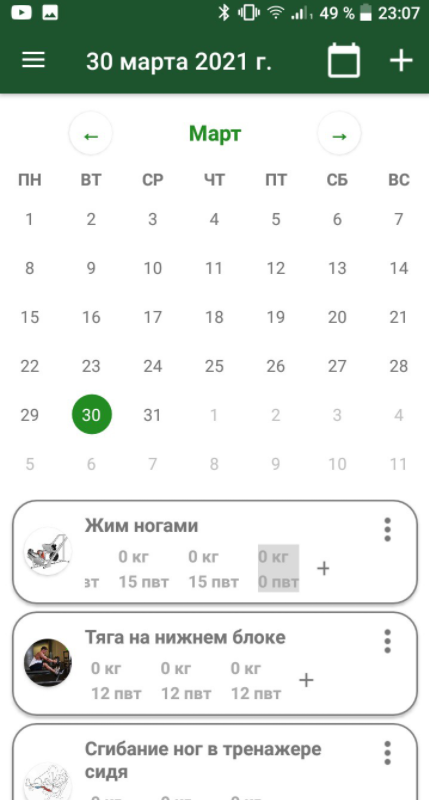
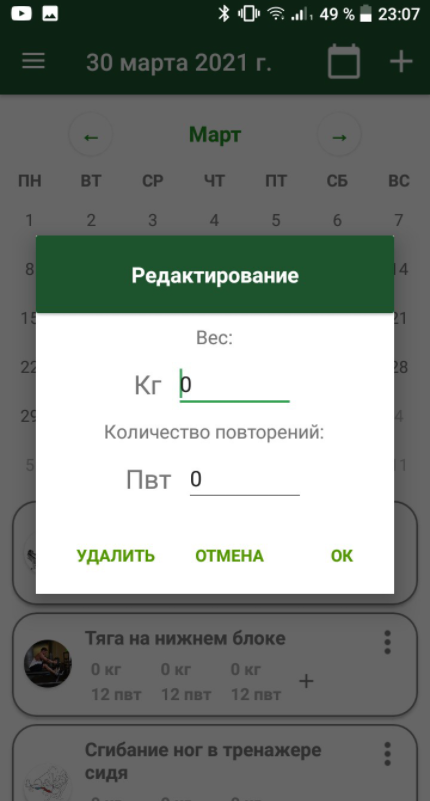
 

Рисунок 37 - Редактирование подхода (Часть 1)

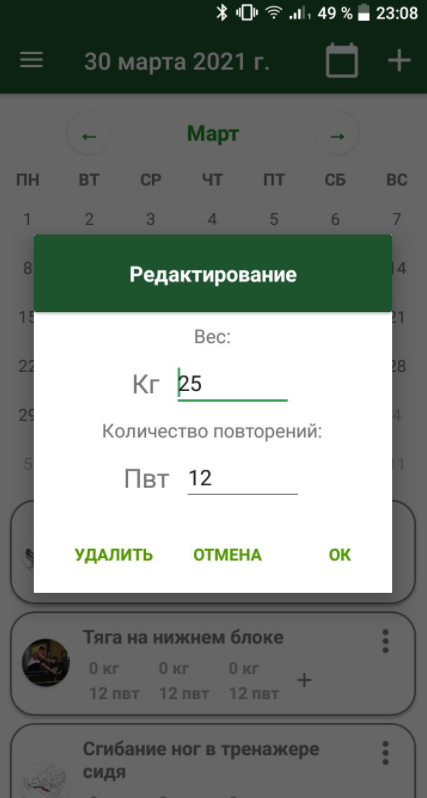
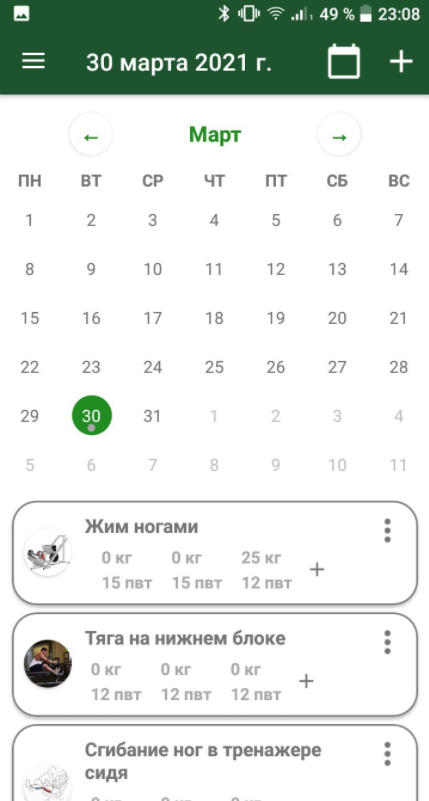
 

Рисунок 38 - Редактирование подхода (Часть 2)

Для открытия бокового меню необходимо нажать на кнопку в левом верхнем углу главной страницы. Также можно просто провести пальцем с левой части экрана в правую.

Откроется меню где будет предоставлена возможность выбрать одну из нескольких вкладок.

Для того чтобы посмотреть общую статистику необходимо зайти во вкладку «Статистика» в главном меню приложения.

Страница со статистикой выглядит следующим образом:

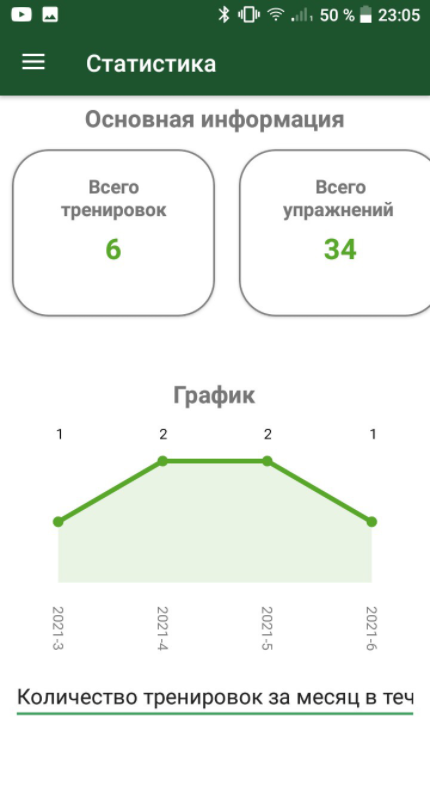


Рисунок 39 - Статистика

На данной странице есть возможность посмотреть на общую статистику пользования приложением. Здесь можно увидеть общее количество выполненных тренировок, количество выполненных упражнений и подходов.

Также здесь можно отобразить на графике одну из нескольких выборок. К примеру, можно отобразить общее количество тренировок в месяц в течение текущего года.

В приложении присутствует страница для создания личных заметок. Для перехода на страницу с заметками необходимо открыть главное меню и выбрать страницу «Заметки».

После чего откроется страница с заметками.

Здесь можно просмотреть все заметки, когда-либо созданные через это приложение, а также отредактировать или добавить новую.

Есть возможность удаления заметок во время редактирования.

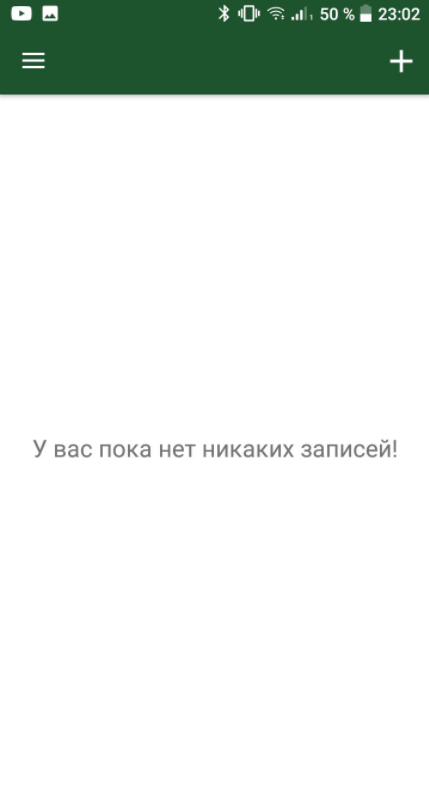
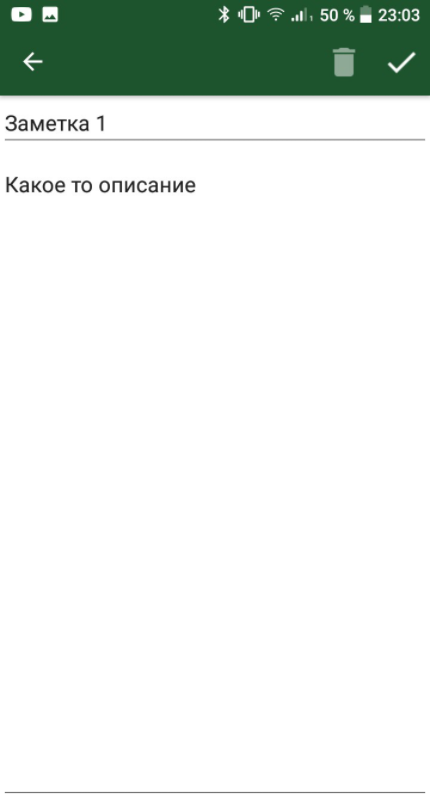
 

Рисунок 40 - Заметки

Для создания новой заметки необходимо нажать на «Плюс» на панели инструментов. После чего откроется пустая заметка, в которой можно писать необходимый текст. Для создания необходимо нажать на галку на панели инструментов.

Чтобы удалить заметку необходимо нажать на иконку в стиле мусорного бака.

Также в приложении можно расчитать индекс массы тела человека.

Для этого перейдите в раздел «Индекс массы тела» в главном меню мобильного приложения.

Откроется страница, на которой можно расчитать ИМТ.

Чтобы поподробнее узнать о том, что такое индекс массы тела можно нажать на справку, которая находится в правом верхнем углу.

Для расчёта необходимо ввести рост и вес. После этого нажать на кнопку расчитать и в скором времени результат будет обработан и предоставлен на экране.

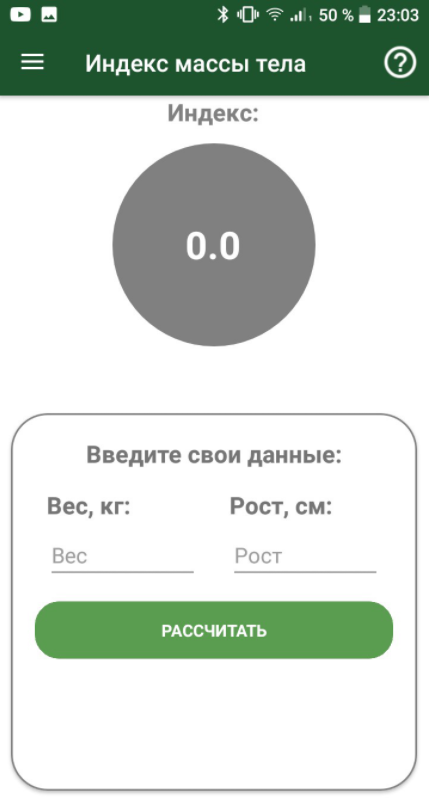
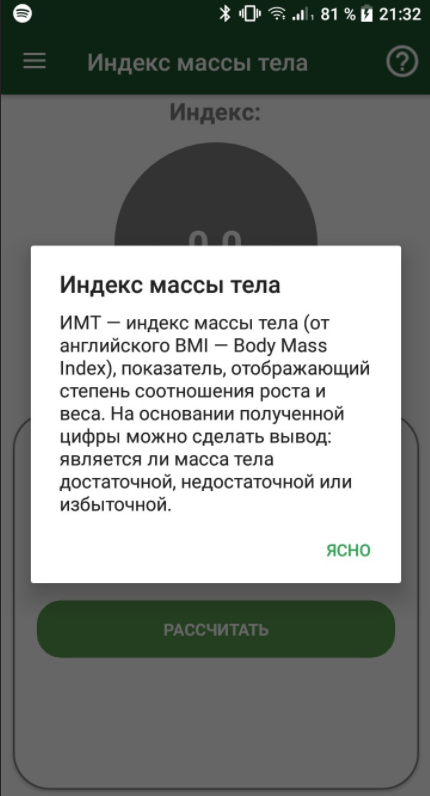
 

Рисунок 41 - Индекс массы тела (Часть 1)

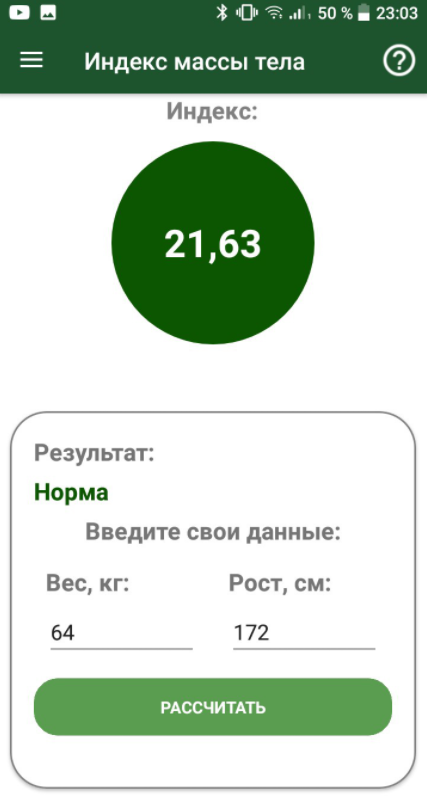


Рисунок 42 - Индекс массы тела. Результат

Если необходимо узнать информацию о каком-либо упражнении, то можно воспользоваться вкладкой «Список упражнений» в главном меню «Pocket Trainer».

После чего откроется вкладка с выбором мышечной группы, как было показано ранее на рисунке 32. Далее необходимо нажать на нужную группу мышц и выбрать необходимое упражнение после.

В результате откроется информационная страница о выбранном упражнении.

Здесь отображена вся информация: GIF-изображение, показывающее технику выполнения, задействованные в упражнении группы мышц и описание упражнения.



Рисунок 43 - Информационная страница (Упражнение)

Также можно посмотреть все доступные в приложении программы тренировок, а также из каких тренировочных дней состоит каждая программа.

Для этого необходимо зайти во вкладку «Список программ тренировок».

Откроется список всех доступных тренировочных программ. Для выбора какой-либо необходимо нажать по ней с помощью пальца. Откроется список тренировочных дней, которые присутствуют в выбранной программе.

Если хочется узнать какие упражнение находятся в тренировочном дне, достаточно просто нажать по нужному тренировочному дню.

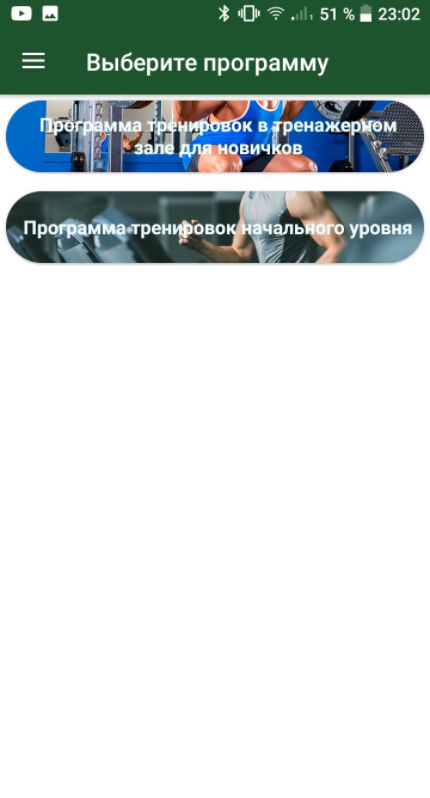
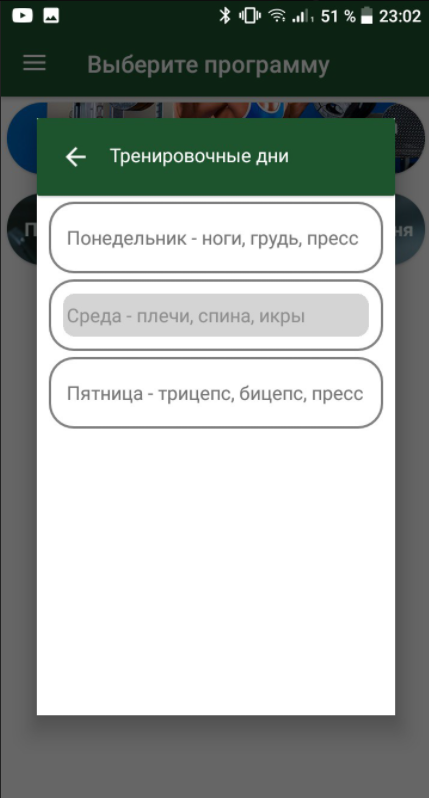
 

Рисунок 44 - Программы тренировок

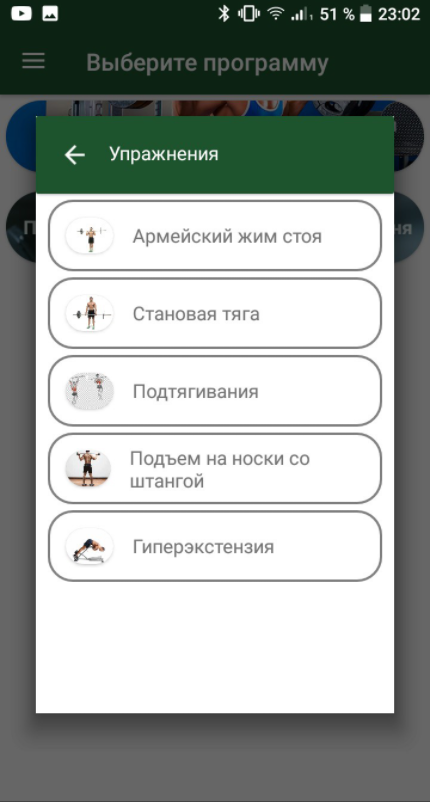


Рисунок 45 - Список упражнений в тренировочный день

**5.2 Сообщение оператору**

В случае отсутствия интернета пользователю выводится сообщение, оповещающее его об этом:

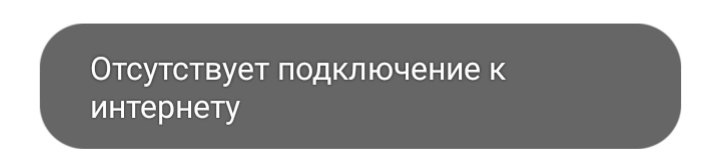


Рисунок 46 - Отсутствие подключения

При неправильном вводе данных для расчёта индекса массы тела, после нажатия на кнопку «Рассчитать» выведется следующее сообщение:

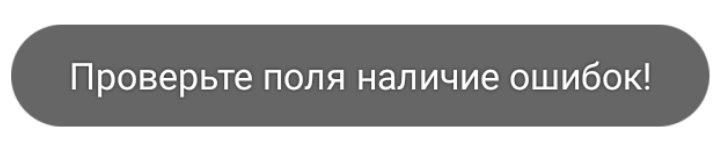


Рисунок 47 - Ошибка в расчёте ИМТ

Заметки.

При попытке сохранить заметку без названия выведется следующее сообщение:

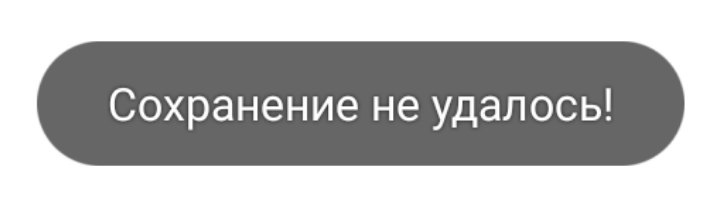


Рисунок 48 - Неудачное сохранение

Для решения этой проблемы необходимо ввести название для заметки.

В случае удачного сохранения пользователю будет выведено следующее сообщение:

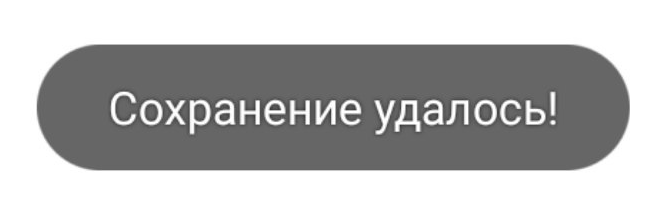


Рисунок 49 - Успешное сохранение

При удалении заметки выведется окно предупреждения. Для того чтобы закрыть его необходимо нажать на один из предложенных вариантов.

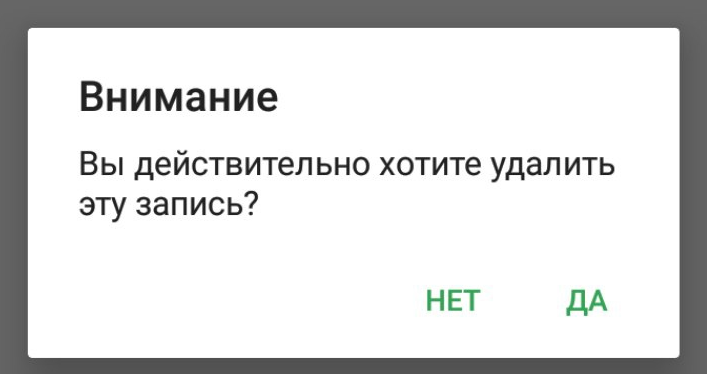


Рисунок 50 - Предупреждение

Если пользователь нажмет «Да» заметка удалится, и пользователь будет оповещен об этом с помощью сообщения:



Рисунок 51 - Успешное удаление

При попытке пользователем переместить упражнение выше, когда выбранное упражнение уже в верхней части списка будет выведено следующее сообщение:

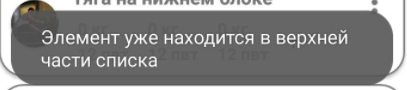


Рисунок 52 - Верхняя часть списка

В обратном же случае выведется обратное сообщение:

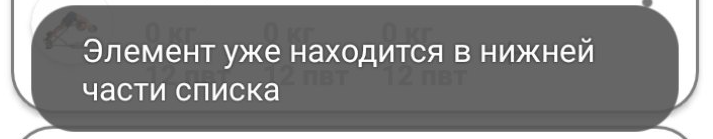


Рисунок 53 - Нижняя часть списка

**6. Мероприятия по информационной безопасности**

Угрозами информационной безопасности для мобильного приложения «Pocket Trainer» являются:

* Угрозы конфиденциальности;
* Угрозы целостности;
* Угрозы доступности;

Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней.

К примеру, пользователь мобильного приложения получит доступ к информации другого пользователя без наличия разрешения.

Данная угроза нейтрализуется с помощью разделения базы данных на серверную и клиентскую части. Так, на стороне сервера расположена только та информация, которая является общедоступной для всех пользователей (Программы тренировок, информация об упражнениях), а вся личная информация пользователя (история тренировок) хранится на локальной базе данных в памяти мобильного телефона пользователя.

Если же пользователь захочет ограничить доступ к данным в приложении, то он может это сделать с помощью встроенной защиты Андроид.

Угрозы нарушения целостности — это угрозы, связанные с вероятностью модификации той или иной информации, хранящейся в информационной системе.

Вероятной угрозой нарушения целостности информации является изменение данных серверной БД (общедоступной информации) пользователем.

Для обеспечения целостности информации мобильного приложения доступ к базе данных и другим файлам на сервере защищен с помощью пароля.

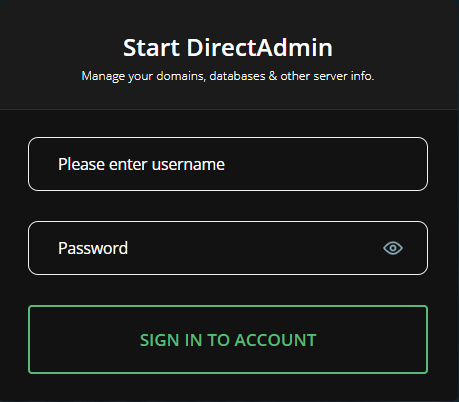


Рисунок 54 - Авторизация для администрирования

Без знания логина и пароля человек просто не сможет зайти в панель управления, где позволено заниматься администрированием выбранного домена и БД.

Сам домен защищен с помощью криптографического протокола SSL.

SSL (англ. Secure Sockets Layer — уровень защищённых сокетов) — криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.

Когда пользователь заходит на веб-сайт, браузер запрашивает информацию о сертификате у сервера, который высылает копию SSL-сертификата с открытым ключом. Далее, браузер проверяет сертификат, название которого должно совпадать с именем веб-сайта.

Кроме того, проверяется дата действия сертификата и наличие корневого сертификата, выданного надежным центром сертификации. Если браузер доверяет сертификату, то он генерирует предварительный секрет (pre-master secret) сессии на основе открытого ключа, используя максимально высокий уровень шифрования, который поддерживают обе стороны.

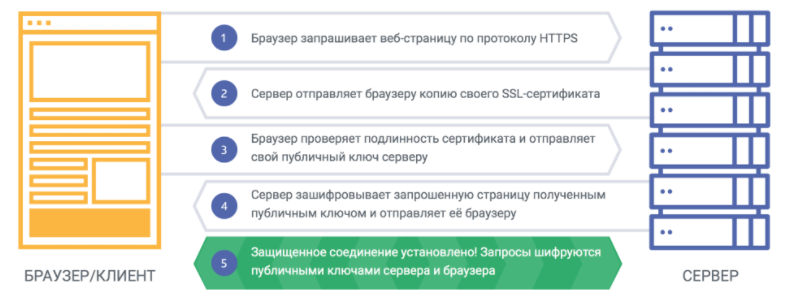


Рисунок 55 - Защита

Сервер расшифровывает предварительный секрет с помощью своего закрытого ключа, соглашается продолжить коммуникацию и создать общий секрет (master secret), используя определенный вид шифрования. Теперь обе стороны используют симметричный ключ, который действителен только для данной сессии. После ее завершения ключ уничтожается, а при следующем посещении сайта процесс рукопожатия запускается сначала.

Нарушение доступности представляет собой создание таких условий, при которых доступ к услуге или информации будет заблокирован.

Данная угроза нейтрализуется благодаря размещению базы данных на хостинге. Так, с помощью выделенного общедоступного домена пользователь сможет получить доступ к информации в любое время.

На стороне клиента (мобильного приложения) необходимо обеспечить целостность и доступность данных для пользователя.

Целостность – свойство, при наличии которого информация сохраняет заранее определенные вид и качество.

Обеспечение целостности данных реализовано с помощью указанных ранее пунктов (сертификата SSL и аутентификации пользователя). Эти два пункта обеспечивают защиту информации от изменений, подмены или уничтожения в процессе их передачи пользователю.

Доступность – такое состояние информации, когда она находится в том виде, месте и времени, которые необходимы пользователю, и в то время, когда она ему необходима.

Доступность данных для пользователя обеспечена с помощью интерфейса мобильного приложения. Так, необходимая информация предоставляется пользователю в нужный момент времени, когда он об этом просит.

Вся информация (или оповещение об ее отсутствии) выводится пользователю на экран после вызова определенных команд в мобильном приложении «Pocket Trainer».

**Заключение**

Программный продукт удовлетворяет всем требованиям, изложенным в техническом задании. Таким образом, задача, поставленная при проектировании мобильного приложения «Pocket Trainer», выполнена.

Реализация данного программного обеспечения была произведена с помощью интегрированной среды разработки Rider от компании JetBrains.

Мобильное приложение написано на программном языке C#. Серверная база данных была реализована с помощью среды PHPMyAdmin на специальном хостинге. REST API написано с помощью языка программирования PHP и находится на сервере. Локальная база данных была выполнена с помощью SQLite.

Данное мобильное приложение является актуальным в нынешнее время т.к. все большее и большее количество людей начинает интересоваться фитнесс тематикой и тренировками для улучшения своей внешности ли физических показателей, а для грамотной тренировки всегда необходимо вести тренировки в дневнике.

**Приложения**

**Текст программы**

Ниже приведен метод сбора информации об упражнениях на главной странице мобильного приложения:

async Task GatherCurrentExc()  
{  
 \_currentExcercises = new List<Excercise>();  
 \_repsStck = new List<StackLayout>();  
 var rc = new ProcessInfo();  
 *// Запрос на выборку ID дня тренировки* \_resultWorkoutId = await LocalService.Db.QueryAsync<Log>($"Select WorkoutDayID from Log where DateOfWorkout = '{Calendar.SelectedDate.Year}-{Calendar.SelectedDate.Month}-{Calendar.SelectedDate.Day}'");  
 if (!\_resultWorkoutId.Any())  
 {  
 LocalService.WaitHandle.Set();  
 return;  
 }  
 *// После выбираем все айдишники упражнений из локальной таблицы* var resultExercises = await LocalService.Db.QueryAsync<WorkDayExcJunction>(  
 $"Select ExcerciseID,Place from WorkDayExcJunction where WorkoutDayID={\_resultWorkoutId.Last().WorkoutDayID} order by Place");  
 *// Далее отправляем запросы на сервер и добавляем в лист с упражнениями возвращаемые значения* foreach (var item in resultExercises)  
 {  
 var exc = await rc.LoadOneExcercise(item.ExcerciseID);  
 \_currentExcercises.Add(exc);  
 }  
 foreach (var item in \_currentExcercises)  
 {  
 item.Sets = new List<SetsInfo>();  
 item.Sets = await LocalService.Db.QueryAsync<SetsInfo>(  
 $"Select \* from SetsInfo where WorkoutDayID={\_resultWorkoutId.Last().WorkoutDayID} and ExercisePlace={\_currentExcercises.IndexOf(item)+1}");  
 }  
 LocalService.WaitHandle.Set();  
}

}

Далее показан метод, выводящий собранные упражнения на экран:

async Task DisplayCurrentRoutine()  
{  
 \_exerciseCount = 0;  
 *//Выбор текущей тренировки* \_currentWorkout = await LocalService.Db.QueryAsync<Log>($"Select \* from Log where DateOfWorkout='{\_curdate}'");  
 if(Calendar.SelectedDate==null)  
 Calendar.SelectedDate=DateTime.Now;  
 Indicator.IsRunning = true;  
 Indicator.IsVisible = true;  
 NoRoutineLabel.IsVisible = false;  
 *// Проверка на то, есть ли уже упражнения на экране* if (!\_currentExcercises.Any() || \_currentExcercises == null)  
 {  
 await GatherCurrentExc();  
 await Task.Run((() => LocalService.WaitHandle.WaitOne()));  
 }  
 *// Проверка на наличие упражнений после сбора упражнений* if (!\_currentExcercises.Any() || \_currentExcercises == null)  
 {  
 NoRoutineLabel.IsVisible = true;  
 Indicator.IsRunning = false;  
 Indicator.IsVisible = false;  
 return;  
 }  
 *// Оформление пользовательского интерфейса под каждое упражнение* foreach (var item in \_currentExcercises)  
 {  
 var muscleImg = new Image()  
 {  
 Source = item.ImagePath,  
 HeightRequest = 30,  
 WidthRequest = 30,  
 Aspect = Aspect.**AspectFill** };  
 var recognizer = new TapGestureRecognizer  
 {  
 NumberOfTapsRequired = 1  
 };  
 recognizer.Tapped += OpenExerciseInfo;  
 var imgfrm = new Frame()  
 {  
 Content = muscleImg,  
 BorderColor = Color.Gray,  
 CornerRadius = 100,  
 HeightRequest = 40,  
 WidthRequest = 40,  
 VerticalOptions = LayoutOptions.Center,  
 Padding = 0,  
 AutomationId = \_exerciseCount.ToString(),  
 GestureRecognizers = { recognizer}  
 };  
  
 var excerciseName = new Label()  
 {  
 Text = item.Name,  
 FontSize = 16,  
 VerticalOptions = LayoutOptions.Center,  
 FontAttributes = FontAttributes.**Bold** };  
 var Horstck = new StackLayout()  
 {  
 Orientation = StackOrientation.**Horizontal** };  
 var counter = 0;  
 var repsRecognizer = new TapGestureRecognizer()  
 {  
 NumberOfTapsRequired = 1,  
 };  
 repsRecognizer.Tapped+= RepsRecognizerOnTapped;  
 var scrl = new ScrollView()  
 {  
 Orientation = ScrollOrientation.**Horizontal** };  
 foreach (var set in item.Sets)  
 {  
 var reps = new Label()  
 {  
 Text = set.RepsNumber.ToString() + " пвт",  
 FontSize = 14,  
 FontAttributes = FontAttributes.**Bold**,  
 TextColor = Color.DarkGray  
 };  
 var weight = new Label()  
 {  
 Text = set.Weight.ToString() + " кг",  
 FontSize = 14,  
 FontAttributes = FontAttributes.**Bold**,  
 TextColor = Color.DarkGray  
 };  
 var stck = new StackLayout()  
 {  
 Orientation = StackOrientation.**Vertical**,  
 Children = { weight,reps},  
 Margin = new Thickness(5,0),  
 Spacing = 2,  
 AutomationId = (counter+1).ToString() +"|"+set.ExercisePlace,  
 };  
 TouchEffect.SetColor(stck,Color.DarkGray);  
 Commands.SetTap(stck,new Command((() => RepsRecognizerOnTapped(stck,new EventArgs()))));  
 counter++;  
 Horstck.Children.Add(stck);  
 }  
 var imageButtonAddSet = new ImageButton()  
 {  
 Source = "ic\_action\_add\_black.png",  
 BackgroundColor = Color.Transparent,  
 HeightRequest = 20,  
 WidthRequest = 20,  
 Opacity = 70,  
 AutomationId = (\_exerciseCount).ToString()  
 };  
 imageButtonAddSet.Clicked+= imageButtonAddSetOnClicked;  
 Horstck.Children.Add(imageButtonAddSet);  
 scrl.Content = Horstck;  
 \_repsStck.Add(Horstck);  
 var Vertstck = new StackLayout()  
 {  
 Orientation = StackOrientation.**Vertical**,  
 VerticalOptions = LayoutOptions.Center,  
 Children = { excerciseName,scrl},  
 Margin = new Thickness(5,0)  
 };  
   
 var morebtn = new ImageButton()  
 {  
 Source = "ic\_action\_more\_vert.png",  
 HorizontalOptions = LayoutOptions.End,  
 HeightRequest = 30,  
 WidthRequest = 30,  
 VerticalOptions = LayoutOptions.Start,  
 BackgroundColor = Color.Transparent,  
 AutomationId = (\_exerciseCount + 1).ToString()  
 };  
 morebtn.Clicked+= MorebtnOnClicked;  
 var gr = new Grid()  
 {  
 ColumnDefinitions =  
 {  
 new ColumnDefinition{Width = GridLength.Auto},  
 new ColumnDefinition{Width = new GridLength(4,GridUnitType.**Star**)},  
 new ColumnDefinition{Width = GridLength.Star}  
 },  
 };  
 gr.Children.Add(imgfrm,0,0);  
 gr.Children.Add(morebtn,2,0);  
 gr.Children.Add(Vertstck,1,0);  
   
 var frm = new Frame()  
 {  
 BorderColor = Color.Gray,  
 CornerRadius = 20,  
 Margin = new Thickness(10,0),  
 Content = gr,  
 Padding = 10  
 };  
 ElementsStck.Children.Add(frm);  
 \_exerciseCount++;  
 }  
 Indicator.IsRunning = false;  
 Indicator.IsVisible = false;  
}

**Источники, использованные при разработке**

1. ГОСТ Р ИСО\_МЭК 25051-2017 Требования к качеству готового к использованию программного продукта (RUSP) и инструкции по тестированию.
2. ЕСПД Единая система программной документации.
3. Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов, Дэвид Скляр, 2017

**Internet – ресурсы**

1. PHP: Hypertext preprocessor [Электронный ресурс] – [http://php.net](http://php.net/)
2. MySQL [Электронный ресурс] - <https://www.mysql.com/>
3. Простой REST API в PHP - <https://only-to-top.ru/blog/programming/2019-11-06-rest-api-php.html>
4. Руководство по C# - <https://metanit.com/sharp/>
5. Руководство по программированию Xamarin Forms - <https://metanit.com/sharp/xamarin/>