КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное

бюджетное профессиональное образовательное учреждение

«Колледж информационных технологий»

**ОТЧЁТ**

**по модулю МДК 01.03 «Разработка мобильных приложений»**

**Специальность 09.02.07**

**«Информационные системы и программирование»**

**Специализация:**

**«Программист»**

**Отчет по учебной практике**

Студент группы 493:

Лукьянов И. А.

Преподаватель: Фомин А.В.

Санкт-Петербург 2023

СОДЕРЖАНИЕ

[1. Интерфейс пользователя 4](#_Toc128475368)

[1.1. Авторизация пользователя 4](#_Toc128475369)

[1.2. Регистрация нового пользователя 5](#_Toc128475370)

[1.3. Список мест 6](#_Toc128475371)

[1.4. Место 7](#_Toc128475372)

[1.5. Список счетчиков 8](#_Toc128475373)

[1.6. Счетчик 8](#_Toc128475374)

[1.7. Список тарифов 9](#_Toc128475375)

[1.8. Тариф 9](#_Toc128475376)

[1.9. Список измерений 10](#_Toc128475377)

[1.10. Измерение 11](#_Toc128475378)

[2. Структура базы данных 12](#_Toc128475379)

[2.1. Таблица Settings 12](#_Toc128475380)

[2.2. Таблица APIEndPoint 12](#_Toc128475381)

[2.3. Таблица Session 13](#_Toc128475382)

[3. Описание протокола взаимодействия 14](#_Toc128475383)

[3.1. Контроль сессий 15](#_Toc128475384)

[3.1.1. Функция sign\_in 15](#_Toc128475385)

[3.1.2. Функция sign\_out 16](#_Toc128475386)

[3.1.3. Функция register\_account 17](#_Toc128475387)

[3.2. Управление местами 18](#_Toc128475388)

[3.2.1. Функция add\_location 18](#_Toc128475389)

[3.2.2. Функция get\_locations 19](#_Toc128475390)

[3.2.3. Функция update\_location 20](#_Toc128475391)

[3.2.4. Функция delete\_location 21](#_Toc128475392)

[3.3. Управление счетчиками 22](#_Toc128475393)

[3.3.1. Функция add\_counter 22](#_Toc128475394)

[3.3.2. Функция get\_counters 23](#_Toc128475395)

[3.3.3. Функция update\_counter 24](#_Toc128475396)

[3.3.4. Функция delete\_counter 25](#_Toc128475397)

[3.4. Управление тарифами 26](#_Toc128475398)

[3.4.1. Функция add\_rate 26](#_Toc128475399)

[3.4.2. Функция get\_rates 27](#_Toc128475400)

[3.4.3. Функция update\_rate 28](#_Toc128475401)

[3.4.4. Функция delete\_rate 29](#_Toc128475402)

[3.5. Управление измерениями 30](#_Toc128475403)

[3.5.1. Функция add\_measurement 30](#_Toc128475404)

[3.5.2. Функция get\_measurements 31](#_Toc128475405)

[3.5.3. Функция update\_measurement 32](#_Toc128475406)

[3.5.4. Функция delete\_measurement 33](#_Toc128475407)

[Работа приложения 35](#_Toc128475408)

# Интерфейс пользователя

Приложение состоит из основных Activity:

* Authorization (авторизация) – стартовая Activity, которая служит для ввода данных учетной записи пользователя;
* Registration (регистрация) – можно запустить из Activity авторизации, служит для регистрации учетной записи пользователя;
* Menu (меню) – запускается после успешной авторизации, служит для выбора соответствующего списка.
* Locations (список мест) – служит для отображения списка мест пользователя, с возможностью добавления нового места;
* Location (место) – запускается после выбора места из списк, служит для отображения данных одного места с возможностью редактирования и удаления.
* Counters (список счетчиков) – служит для отображения списка мест пользователя, с возможностью добавления нового счетчика;
* Counter (счетчик) – запускается после выбора счетчика из списка, служит для отображения данных одного счетчика с возможностью редактирования и удаления.
* Rates (список тарифов) – служит для отображения списка тарифов пользователя, с возможностью добавления нового тарифа;
* Rate (тариф) – запускается после выбора тарифа из списка, служит для отображения данных одного тарифа с возможностью редактирования и удаления.
* Measurements (список измерений) – служит для отображения списка измерений пользователя, с возможностью добавления нового измерения;
* Measurement (измерение) – запускается после выбора измерения из списка, служит для отображения данных одного измерения с возможностью редактирования и удаления.

## Авторизация пользователя

На рисунке 1 показан макет внешнего вида формы авторизации пользователя.



Рисунок 1 – Activity для авторизации пользователя

На рисунке 2 показан внешний вид формы авторизации в приложении.

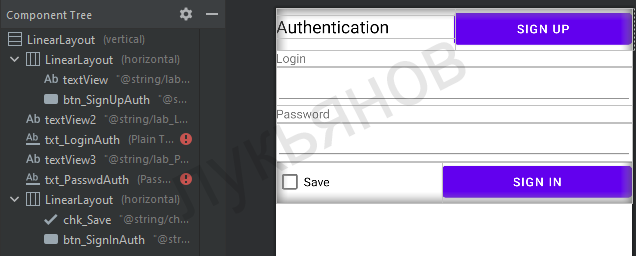


Рисунок 2 – Activity для авторизации пользователя в приложении

На экране присутствуют компоненты EditText для ввода имени пользователя и пароля, компонент CheckBox для возможности сохранения данных учетной записи в локальной базе данных SQLite3 и кнопки входа и регистрации.

## Регистрация нового пользователя

На рисунке 3 показан макет внешнего вида формы регистрации пользователя.

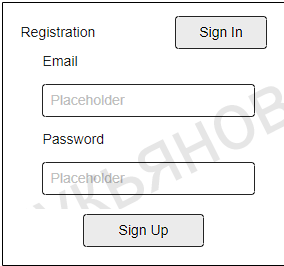


Рисунок 3 – Activity для регистрации пользователя

На рисунке 4 показ­­­­­­­­ан внешний вид формы регистрации в приложении.

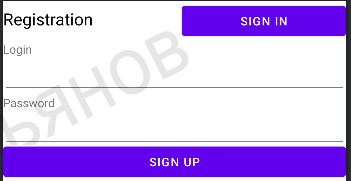


Рисунок 4 – Activity для регистрации пользователя в приложении

На экране присутствуют компоненты EditText для ввода имени пользователя, пароля, а также кнопки регистрации и входа.

## Список мест

На рисунке 5 показан макет внешнего вида формы списка мест.

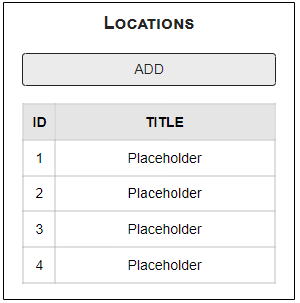


Рисунок 5 – Activity для списка мест

На рисунке 6 показан внешний вид формы списка мест в приложении.



Рисунок 6 – Activity для списка мест в приложении

На экране присутствуют компонент ListView для отображения списка мест и выбора места, а также кнопки добавления нового места.

## Место

На рисунке 7 показан макет внешнего вида формы места.



Рисунок 7 – Activity места

На рисунке 8 показан внешний вид формы места в приложении.



Рисунок 8 – Activity места в приложении

На экране присутствуют компоненты EditText для ввода названия места, а также кнопки для закрытия, сохранения и удаления места.

## Список счетчиков

На рисунке 9 показан макет внешнего вида формы списка счетчиков.

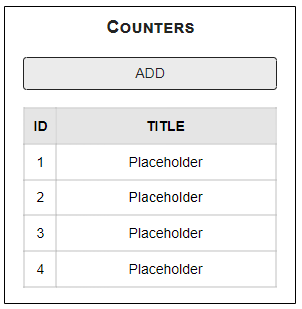


Рисунок 9 – Activity для списка счетчиков

На рисунке 10 показан внешний вид формы списка счетчиков в приложении.



Рисунок 10 – Activity для списка счетчиков в приложении

На экране присутствуют компонент ListView для отображения списка счетчиков и выбора счетчика, а также кнопки добавления нового счетчика

## Счетчик

На рисунке 11 показан макет внешнего вида формы счетчика.



Рисунок 11 – Activity счетчика

На рисунке 12 показан внешний вид формы заметки в приложении.



Рисунок 12 – Activity счетчика в приложении

На экране присутствуют компоненты для редактирования счетчика, а также кнопки для закрытия, сохранения и удаления счетчика.

## Список тарифов

На рисунке 13 показан макет внешнего вида формы списка тарифов.

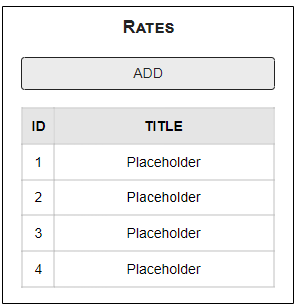


Рисунок 13 – Activity для тарифов заметок

На рисунке 14 показан внешний вид формы списка заметок в приложении.



Рисунок 14 – Activity для списка тарифов в приложении

На экране присутствуют компонент ListView для отображения списка тарифов и выбора тарифа, а также кнопки добавления нового тарифа.

## Тариф

На рисунке 15 показан макет внешнего вида формы тарифа.



Рисунок 15 – Activity тарифа

На рисунке 16 показан внешний вид формы тарифа в приложении.



Рисунок 16 – Activity тарифа в приложении

На экране присутствуют компоненты для редактирования тарифа, а также кнопки для закрытия, сохранения и удаления тарифа.

## Список измерений

На рисунке 17 показан макет внешнего вида формы списка измерений.

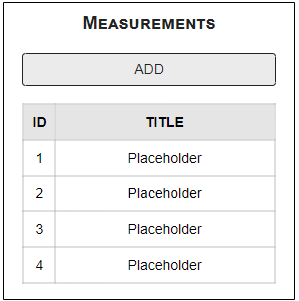


Рисунок 17 – Activity для списка измерений

На рисунке 18 показан внешний вид формы списка измерений в приложении.



Рисунок 19 – Activity для списка измерений в приложении

На экране присутствуют компонент ListView для отображения списка измерений и выбора измерений, а также кнопки добавления нового измерения.

## Измерение

На рисунке 20 показан макет внешнего вида формы измерения.



Рисунок 20 – Activity измерения

На рисунке 21 показан внешний вид формы измерения в приложении.



Рисунок 21 – Activity измерения в приложении

На экране присутствуют компоненты для редактирования измерения, а также кнопки для закрытия, сохранения и удаления измерения.

# Структура базы данных

База данных состоит из 3 таблиц:

1. Settings – сохраненные логин и пароль пользователя.
2. Session – сохраненная сессия.
3. APIEndPoint – сохраненная конечная точки.

ER-диаграмма базы данных представлена на рисунке 22.

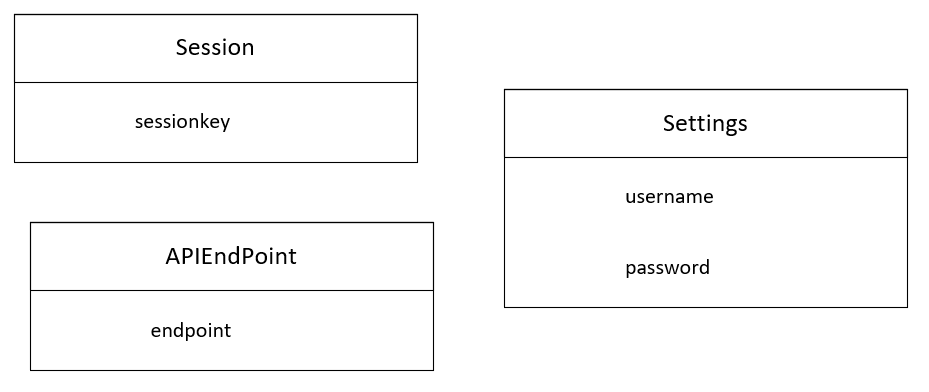


Рисунок 22 – ER-диаграмма базы данных

## Таблица Settings

Содержит сведения о сохраненных учетных данных приложения. Таблица состоит из 2 столбцов:

1. username – логин пользователя.
2. password – пароль пользователь.

Подробное описание столбцов представлено на рисунке 23.

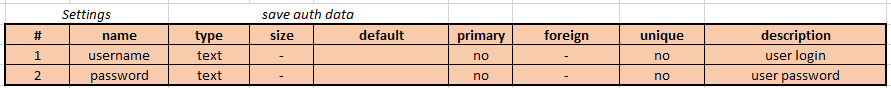


Рисунок 23 – Описание столбцов таблицы Settings

## Таблица APIEndPoint

Содержит сведения о сохраненной конечной точке приложения. Таблица состоит из 1 столбца:

1. endpoint – логин пользователя.

Подробное описание столбцов представлено на рисунке 24.

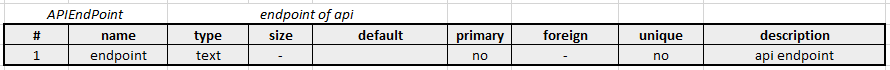


Рисунок 24 – Описание столбцов таблицы APIEndPoint

## Таблица Session

Содержит сведения о сохраненном ключе сессии приложения. Таблица состоит из 1 столбца:

1. sessionkey – ключ сессии.

Подробное описание столбцов представлено на рисунке 25.

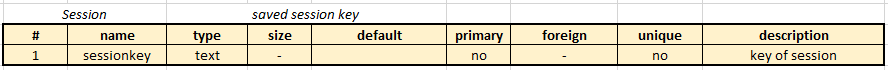


Рисунок 25 – Описание столбцов таблицы Settings­­­­

# Описание протокола взаимодействия

Для взаимодействия с базой данных мобильное приложение использует API на основе хранимых процедур СУБД Postgres и сервера PostgREST, который организует вызов процедур по протоколу HTTP и обмен данными в формате JSON.

Всего доступно 19 функций, список которых показан на рисунке 26:

1. sign\_in – авторизация пользователя.
2. sign\_out – выход из сессии.
3. register\_account – регистрация пользователя.
4. add\_location– добавление.
5. get\_locations – получение списка мест пользователя.
6. update\_location – обновление места.
7. delete\_location – удаление места.
8. add\_counter – добавление счетчика.
9. get\_ counters – получение списка счетчиков пользователя.
10. update\_ counter – обновление счетчика.
11. delete\_ counter – удаление счетчика.
12. add\_rate – добавление тарифа.
13. get\_ rates – получение списка тарифов пользователя.
14. update\_ rate – обновление тарифа.
15. delete\_ rate – удаление тарифа.
16. add\_measurement– добавление измерения.
17. get\_ measurements – получение списка измерений пользователя.
18. update\_ measurement – обновление измерения.
19. delete\_ measurement – удаление измерения.

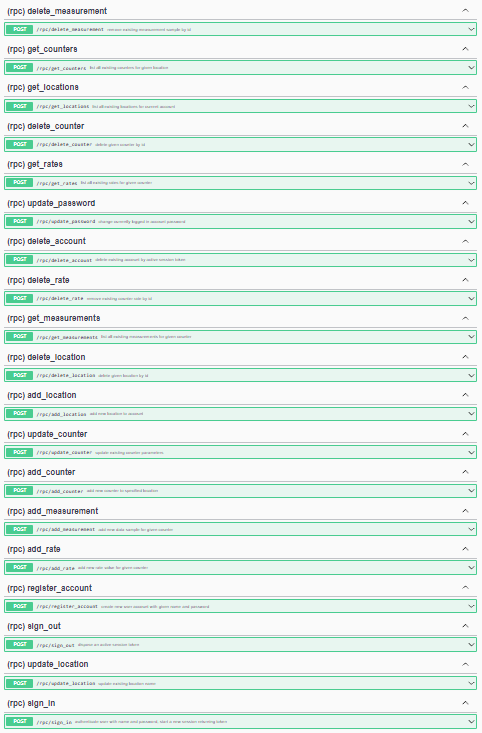


Рисунок 26 – Список хранимых процедур

## Контроль сессий

Данный набор функций позволяет управлять сессией пользователя.

### Функция sign\_in

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция sign\_out

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция register\_account

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

## Управление местами

Данный набор функций позволяет управлять местами пользователя.

### Функция add\_location

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция get\_locations

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция update\_location

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция delete\_location

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

## Управление счетчиками

Данный набор функций позволяет управлять счетчиками пользователя.

### Функция add\_counter

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция get\_counters

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция update\_counter

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция delete\_counter

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

## Управление тарифами

Данный набор функций позволяет управлять тарифами пользователя.

### Функция add\_rate

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция get\_rates

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция update\_rate

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция delete\_rate

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

## Управление измерениями

Данный набор функций позволяет управлять измерениями пользователя.

### Функция add\_measurement

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция get\_measurements

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция update\_measurement

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

### Функция delete\_measurement

Pattern.

*Входные параметры:*

* Pattern – Pattern.

*Выходные парамеры:*

* Pattern – Pattern.

Пример вызова функции показаны на рисунках 27 и 28.



Рисунок 27 – Пример работы функции patter



Рисунок 28 – Вызов функции pattern с неверными данными

# Работа приложения

Pattern.



Рисунок 61 – Pattern