КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное

бюджетное профессиональное образование учреждение

«Колледж информационных технологий»

**ОТЧЕТ**

**по практической работе №15**

**«Приложение для заметок с использованием REST API»**

**по МДК01.03.**

Работу выполнил студент 393 гр.:

Кашицын Артем Андреевич

Преподаватель:

Фомин Александр Валерьевич

Санкт-Петербург 2021

**ХОД РАБОТЫ**

1. Интерфейс пользователя

Диаграмма переходов между Activity на Рисунке 1

Приложение состоит из 4 Activity.

Вход в аккаунт (Main Activity) – начальная Activity, на которой

пользователь вводит данные для входа в систему по кнопке ВХОД, и может перейти на Activity регистрации по кнопке РЕГИСТРАЦИЯ.

Регистрация (Registration Activity) – Activity, на которой пользователь вводит данные для регистрации в системе. По кнопке ОК, если регистрация прошла успешно, пользователь входит в систему. По кнопке ОТМЕНА пользователь возвращается на Main Activity

Список заметок (List Acitvity) – Activity со списком заметок, созданных пользователем. Сверху экрана находится меню. По значку плюса осуществляется переход на Activity с редактированием созданной заметки. По значку стрелки, пользователь возвращается на Activity с входом в систему. По нажатию на элемент списка – переход на редактирование данной заметки

Редактирование заметки (EditNote Activity) – Activity для редактирования названия и содержимого заметки. В меню по нажатию значка сохранения, заметка сохраняется. Значок креста – удаление заметки. Значок стрелки – возвращение на Activity со списком заметок. Все пункты меню возвращают пользователя на List Activity



Рисунок 1 – Схема переходов между Activity

* 1. Авторизация пользователя

На рисунке 2 показана форма авторизации пользователя.

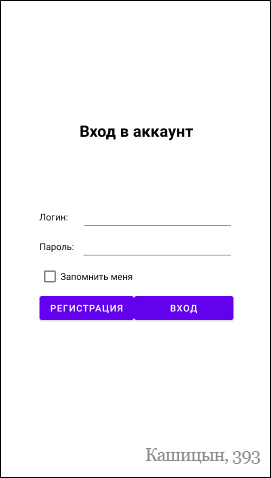


Рисунок 2 – Форма авторизации

На форме расположены 3 TextView с надписями: «Вход в аккаунт», «Логин», «Пароль». 2 EditText поля для ввода логина и пароля соответственно. 1 ComboBox с надписью «Запомнить меня» для автоматического ввода предыдущих EditText данными от последнего успешного входа в систему. 2 кнопки Регистрация – переход на форму с регистрацией пользователя. Кнопка Вход – перход на список заметок, если ключ сессии был получен (если логин и пароль есть на сервере)

1.2 Регистрация пользователя

На рисунке 3 показана форма регистрации пользователя.

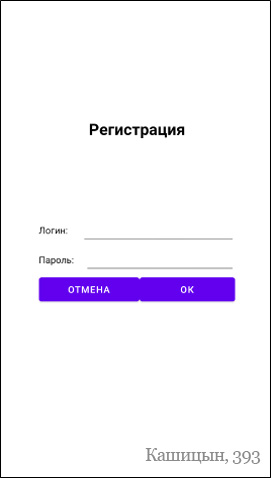


Рисунок 3 – Форма регистрации

На форме расположены 3 TextView с надписями: «Регистрация», «Логин», «Пароль». 2 EditText поля для ввода логина и пароля соответственно. 2 кнопки ОК – переход на форму со списком заметок пользователя, если регистрация прошла успешно. Кнопка Отмена – переход на форму авторизации.

* 1. Список заметок

На рисунке 4 показана форма списка заметок пользователя.

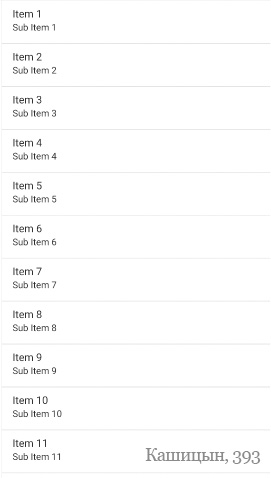


Рисунок 4 – Форма списка заметок пользователя

На форме расположен ListView, адаптер которого будет содержать список типа заметки (с названием, содержимым и идентификатором). По нажатию на элемент открывается форма с редактированием выбранной заметки. К данной форме прикреплено меню, изображенное на Рисунке 5

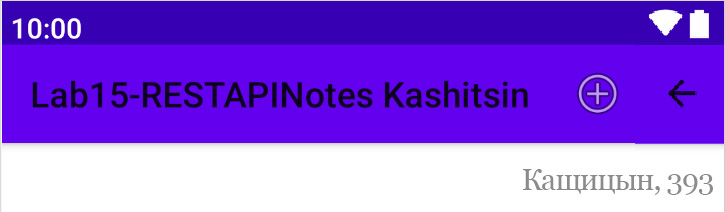


Рисунок 5 – Меню к форме списка заметок пользователя

На меню расположены 2 элемента. По нажатию на элемент с изображением плюса осуществляется переход на Activity с редактированием созданной заметки. По нажатию на элемент с изображением стрелки, пользователь возвращается на Activity с входом в систему.

* 1. Редактирование заметки

На рисунке 6 показана форма редактирования заметки



Рисунок 6 – Форма редактирования заметки

На форме расположены 2 TextView с надписями «Название заметки» и «Текст заметки» соответственно. 2 EditText для ввода названия заметки и содержимого заметки (тип MultiLine). К данной форме прикреплено меню, изображенное на Рисунке 7

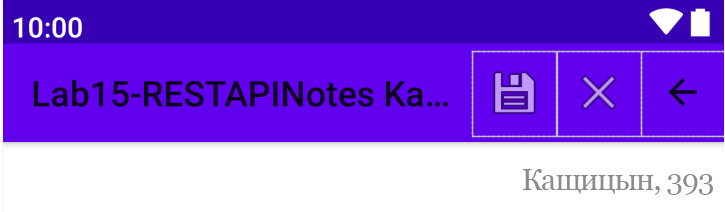


Рисунок 7 – Меню к форме редактирования заметки

На меню расположены 3 элемента. По нажатию на элемент с изображением дискеты, заметка сохраняет изменённые поля и осуществляется переход на Activity с списком заметок. По нажатию на элемент с изображением креста, пользователь удаляет заметку и возвращается на Activity со списком заметок. По нажатию на элемент с изображением стрелки, осуществляется переход на Activity с списком заметок.

1. Структура базы данных

ER-диаграмма базы данных показана на Рисунке 8

База данных состоит из 3 таблиц:

Account – список учетных записей

Session – список токенов для вошедших в систему пользователей

Note – список заметок

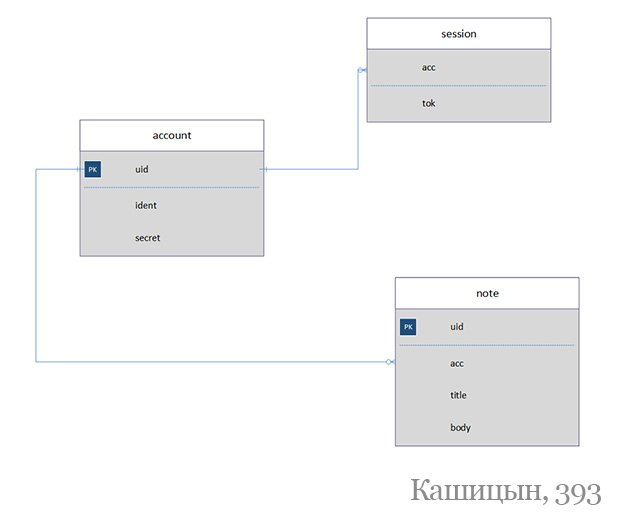


Рисунок 8 – ER-диаграмма базы данных

* 1. Таблица account

Таблица состоит из следующих столбцов:

Uid – идентификатор пользователя

Ident – логин пользователя

Secret – пароль пользователя

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя** | **Тип** | **Размер** | **Уникальное** | **Пустое** | **Ключ** |
| uid | integer | - | Да | Нет | pk |
| ident | varchar | 30 | Да | Нет | - |
| secret | varchar | 30 | - | Да | - |

* 1. Таблица session

Таблица состоит из следующих столбцов:

Acc – вторичный ключ от поля Uid пользователя (идентификатор пользователя)

Tok – токен сессии

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя** | **Тип** | **Размер** | **Уникальное** | **Пустое** | **Ключ** |
| acc | integer | - | Нет | Нет | fk |
| tok | char | 16 | Да | Нет | - |

* 1. Таблица note

Таблица состоит из следующих столбцов:

Uid – идентификатор заметки

Acc – вторичный ключ от поля Uid пользователя (идентификатор пользователя)

Title – название заметки

Body – содержимое заметки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя** | **Тип** | **Размер** | **Уникальное** | **Пустое** | **Ключ** |
| uid | integer | - | Да | Нет | pk |
| acc | integer | - | Нет | Нет | fk |
| title | varchar | 100 | Нет | Да | - |
| body | varchar | 100 | Нет | Да | - |

1. Описание протокола взаимодействия

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

Всего 8 функций. Входные данные в виде полей могут быть:

Usr – логин

Pass – пароль

Skey – токен сессии

Nid – идентификатор заметки

Nbody – содержимое заметки

Ntitle – название заметки

* 1. Контроль сессий

Данный набор функций позволяет управлять сессией пользователя

* 1. Функция reg\_account

Регистрирует пользователя в систему

Пример вызова показан на рисунке 9

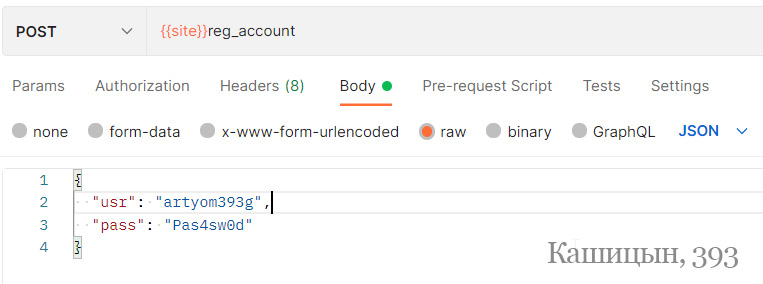


Рисунок 9 – Пример вызова функции reg\_account

* 1. Функция open\_session

Получение токена сессии пользователя с указанным логином и паролем

Пример вызова показан на рисунке 10

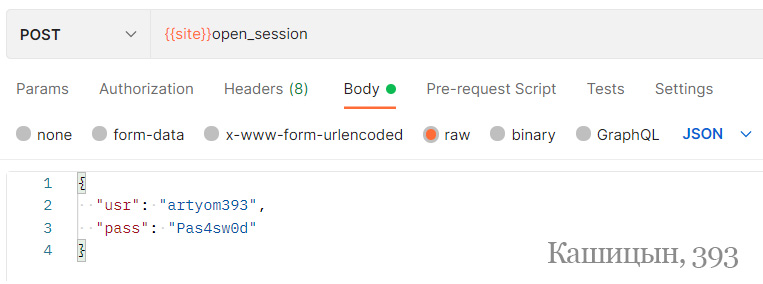


Рисунок 10 – Пример вызова функции open\_session

* 1. Функция add\_note

Создание новой пустой заметки пользователя по токену сессии.

Пример вызова показан на рисунке 11

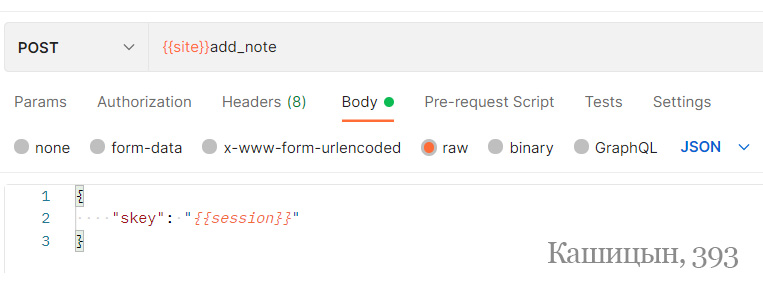


Рисунок 11 – Пример вызова функции add\_note

* 1. Функция update\_note

Обновление текста, названия заметки по её идентификатору и токену сессии

Пример вызова показан на рисунке 12

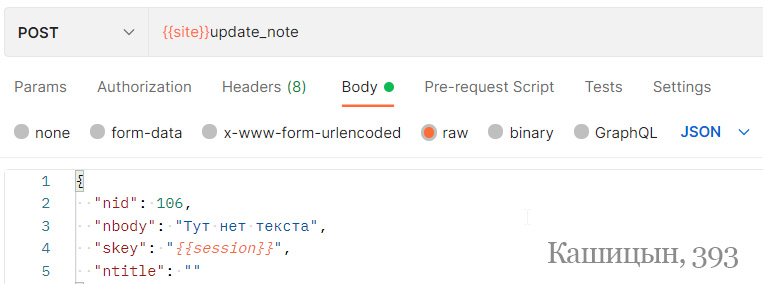


Рисунок 12 – Пример вызова функции update\_note

3.6 Функция list\_note

Получение массива всех заметок пользователя по токену сессии

Пример вызова показан на рисунке 13

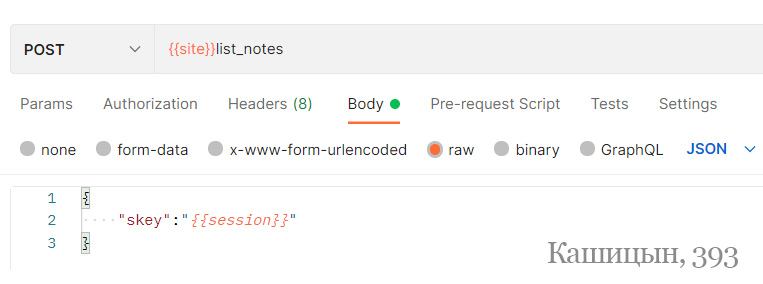


Рисунок 13 – Пример вызова функции list\_notes

3.7 Функция get\_contents

Получение содержимого заметки по её идентификатору и токену сессии

Пример вызова показан на рисунке 14

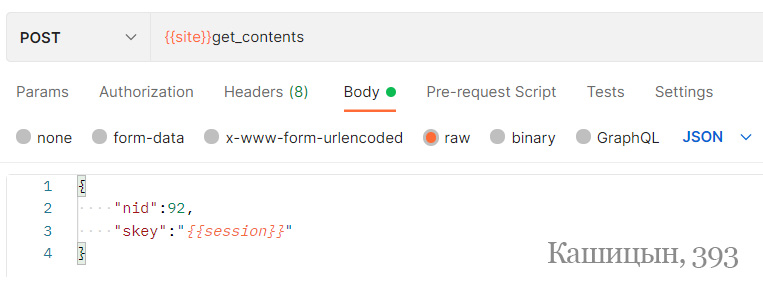


Рисунок 14 – Пример вызова функции get\_contents

3.8 Функция delete\_note

Удаление заметки по её идентификатору и токену сессии

Пример вызова показан на рисунке 15



Рисунок 15 – Пример вызова функции delete\_note

3.9 Функция close\_session

Удаление токена сессии по указанному токену сессии

Пример вызова показан на рисунке 16

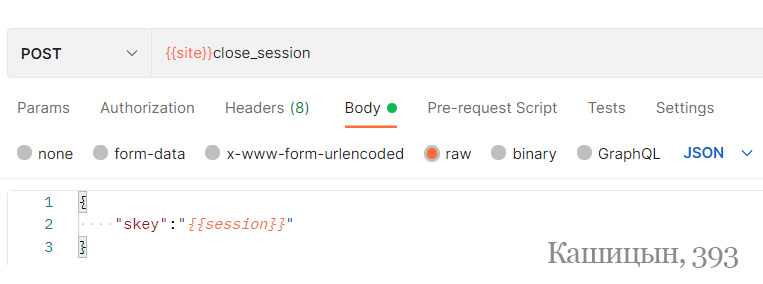
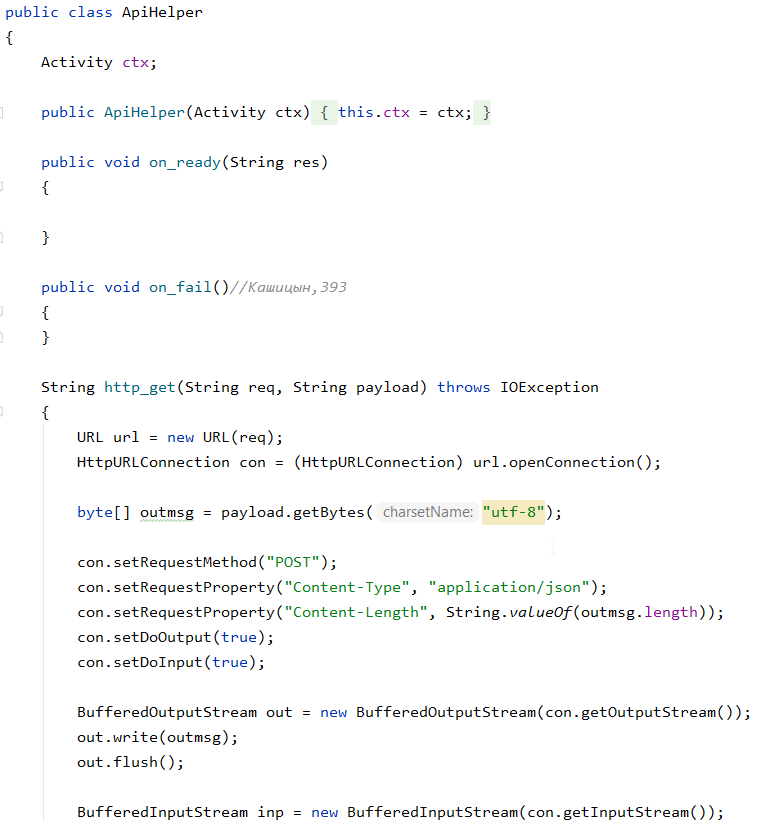
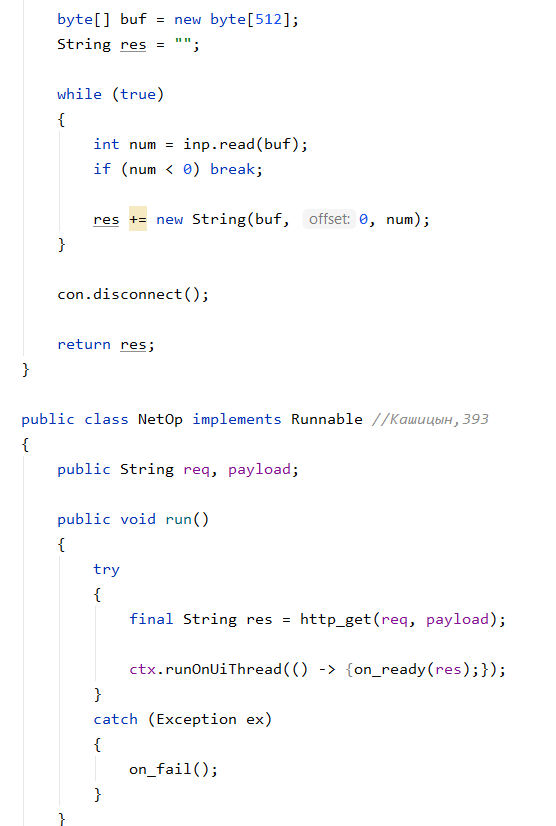


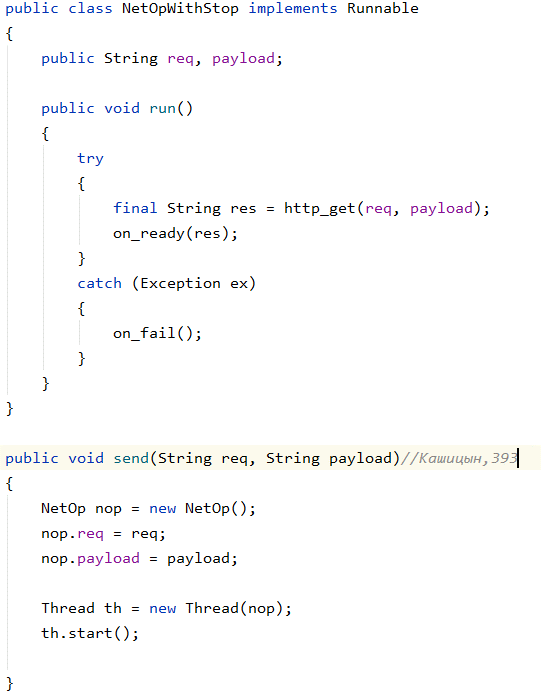
Рисунок 16 – Пример вызова функции close\_session

4.Код приложения

Код файла ApiHelper.java изображен на рисунке 17







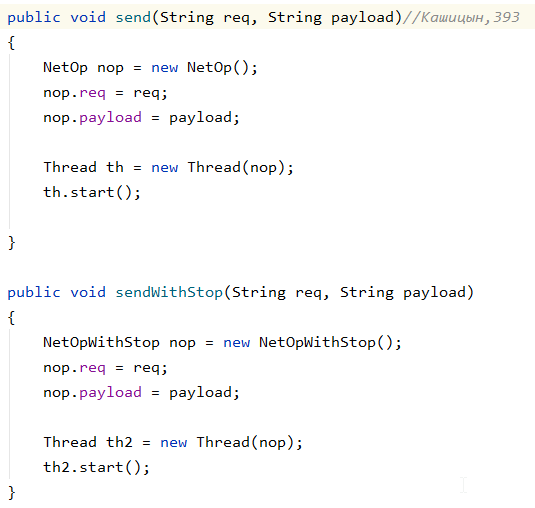


Рисунок 17 – файл ApiHelper.java

Функция send и sendWithStop выполняет http запрос в фоне и дожидаясь результата в отдельных потоках соответственно. Если http запрос выполнился без ошибок, то выполняется процедура on\_ready, напротив – on\_fail. Функция http\_get составляет http\_get с заголовками и нужным телом. После получения строки от буферизованного ответа сервера, соединение разрывается и возвращается строка

Код файла Note.java показана на рисунке 18

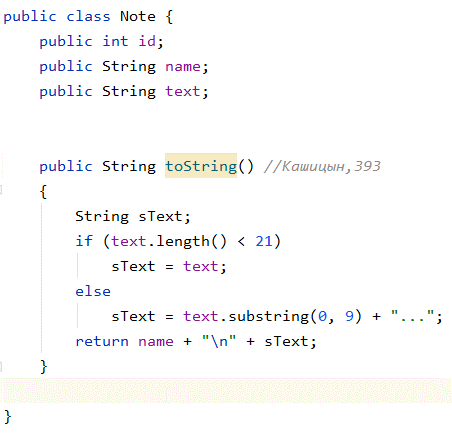


Рисунок 18 – файл Node.js

Класс Note содержит поля id – идентификатор заметки, name – название заметки, text – содержимое. toString() возвращает название заметки и 20 символов с начала содержимого

Код файла DB.java изображен на рисунке 19



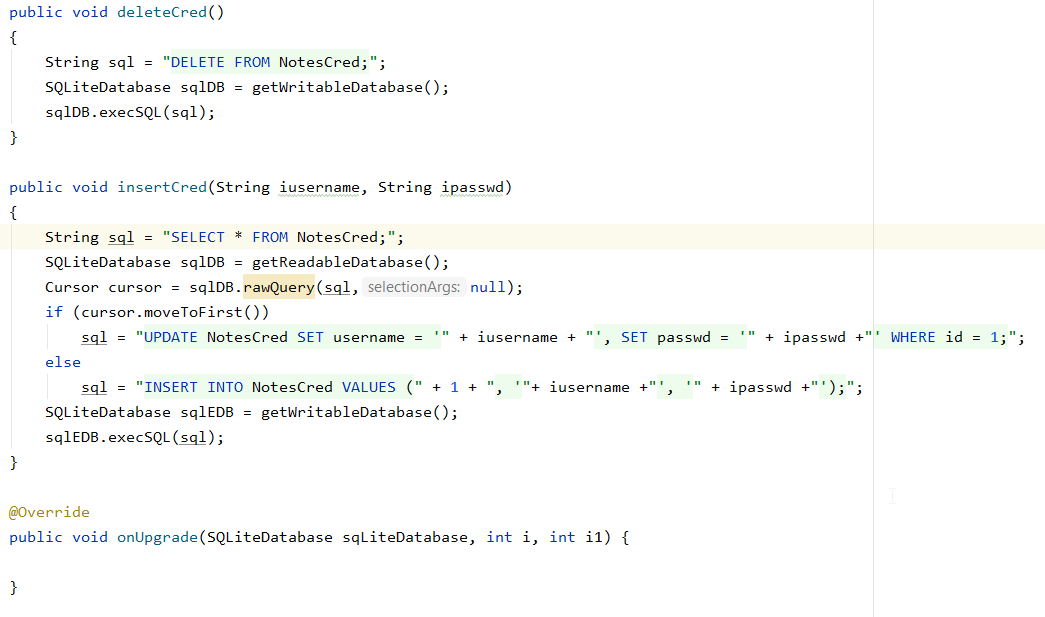


Рисунок 19 – файл DB.java

Создается БД с полями id – идентификатор логина и пароля, username, passwd – логин и пароль соответственно.

getCredUser и getCredPass – возвращают логин и пароль с id 1

deleteCred – удаление всех логинов и паролей

insertCred – добавляет или изменяет запись с логином и паролем под id 1

Код MainActivity:



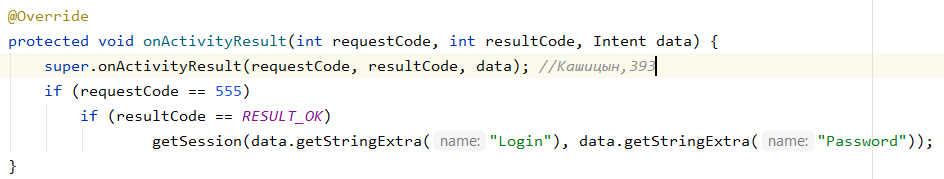
При появлении формы вводятся в поля логина и пароля – сохраненный в БД пользователь

По нажатию на кнопку Войти запускается функция getSession с переданными логин и паролем



Если на запрос получения сессии по логину и паролю, сервер вернул не null, то передаем ключ сессии на форму с списком заметок

Если пользователь нажал на кнопку Регистрация, то переход осуществляется на форму с регистрацией



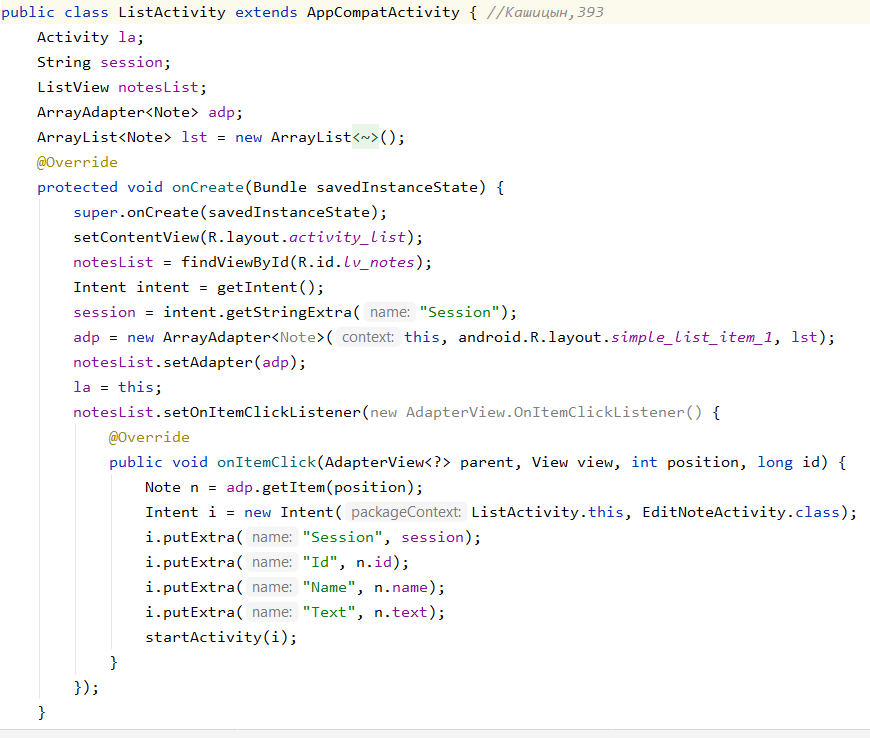
Если пользователь вернулся с формы регистрации, нажав на ОК (и при этом регистрация прошла успешно, то вызов getSession с полученным от формы регистрации логин паролем)

Код RegistrationActivity.java:



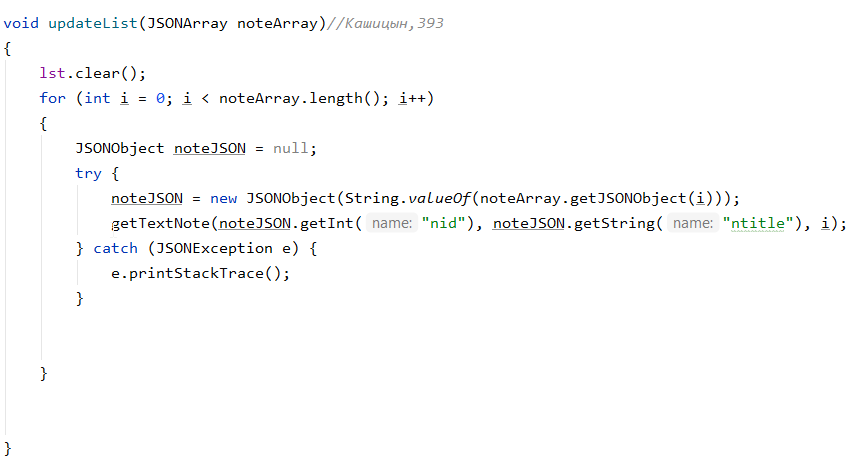
По нажатию на Ок – попытка зарегистрироваться, если удачная, сообщаем об этом форме входа

Код ListActivity:



Получаем ключ сессии и получаем список заметок, используя requestGetList(). По выбору заметки, переходим на форму редактирования, передав необходимые поля





Получаем от сервера (Если получится) JSON ответ, и пытаемся его распознать при помощи getTextNote(id название и номер полученной заметки)



Получаем от сервера содержание заметки по id, и составляем экземпляр Note



Если хотим добавить новую – создаем на сервере и переходим на форму редактирования

EditNoteActivity:

Получаем содержимое заметки и вставляем их в нужные поля



Сохраняем заметку на сервере – обновив её



Если просто закрыть заметку – finish()

Удаление заметки на сервере по кнопке с крестиком