**Государственное бюджетное профессиональное образовательное учреждение**

**«Нижегородский Губернский колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите:

преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_ Л.В. Мухина

\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Мухин

\_\_\_\_\_\_\_\_\_\_\_\_ Ю.С. Мамшева

«21» марта 2023г.

**ОТЧЕТ** **ПО**

**УЧЕБНОЙ ПРАКТИКЕ**

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КС

Руководитель: \_\_\_\_\_\_\_\_\_Мамшева Ю.С. 21.03.2023 г.

\_\_\_\_\_\_\_\_\_Мухина Л.В. 21.03.2023 г.

\_\_\_\_\_\_\_\_\_Мухин Н.А. 21.03.2023 г.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_Дрей И.А. 21.03.2023 г.

Специальность, группа: 09.02.07, 31П

Нижний Новгород

2023 г.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc129929428)

[МДК 01.01 4](#_Toc129929429)

[МДК 01.01 И МДК 01.02 11](#_Toc129929430)

[МДК 01.03 12](#_Toc129929431)

# **ВВЕДЕНИЕ**

Важным элементом процесса подготовки программиста является применение полученных знаний на практике и углубление полученных знаний, для расширения профессионального кругозора. Актуальность учебной практики заключается в получении практических навыков.

Целью учебной практики является изучение и закрепление теоретических и практических знаний по дисциплинам, изученным в ходе обучения.

Задачи учебной практики МДК 01.01:

* Построение ER диаграммы и проектирование базы данных;
* Разработать десктопное приложение WPF для реализации модуля «Учет и паспортизация активов»;
* Разработать десктопное приложение WPF для реализации модуля «Работа с абонентами»;

Задачи учебной практики по MДК 01.01 и 01.02:

* Привести базу данных к третьей нормальной форме;
* Привести данные к нужному формату и заполнить базу данных;
* Разработать десктопное приложение WPF для фирмы, продающей канцелярские товары;
* Реализовать DLL-библиотеку;
* Разработать юнит-тесты для написанной библиотеки;
* Составить документ с тест-кейсами.

Задачи учебной практики по МДК 01.03:

* Создание приложения в Android Studio в соответствие с макетом
* Реализовать логику работы приложения в соответствие с прототипом.

# **РАЗРАБОТКА WPF ПРИЛОЖЕНИЯ**

В рамках прохождения учебной практики по данной дисциплине нужно было разработать программное решение для ООО «Телеком Нева Связь».

Проект располагается в 2х ветвях в данном репозитории: <https://github.com/dreyvania999/UPM>

Задание было разделено на две сессии, в каждой из которых нужно было разработать соответствующие модули.

В рамках первой сессии были поставлены следующие задачи:

* проектирование базы данных (ERD);
* заполнение DataDictionary\_Template.xlsx;
* разработка окна Авторизации пользователей системы.

На первом этапе разработки нужно было спроектировать базу данных. Так как вся информация по обеим сессиям была выдана единовременно было принято решение создать все таблицы. Все таблицы нужно было привести к третьей нормальной форме, определить типы данных для полей таблицы и связи между таблицами.

ER-диаграмма базы данных представлена на рисунке 1.



Рисунок 1 – ER-диаграмма

После создания таблицы заполняются данными из .xlsx таблиц, представленных в папках с заданиями. Все данные приводятся к нужному виду и экспортируются в базу данных.

Далее заполняется DataDictionary\_Template.xlsx для таблиц, отображенных на ER-диаграмме. Пример заполнения для таблиц с сотрудниками и ролями представлен на рисунке 2.

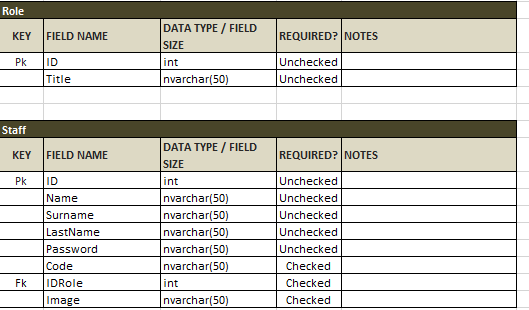


Рисунок 2 – Заполнение DataDictionary\_Template.xlsx

Следующим этапом было создание окна авторизации. Оно должно быть создано в соответствие с представленным макетом (рис. 3 - 4): Проект с окном авторизации расположен на ветке master.

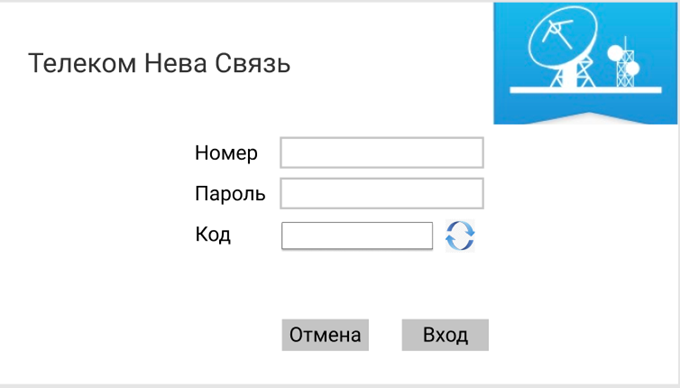


Рисунок 3 – Макет для окна авторизации

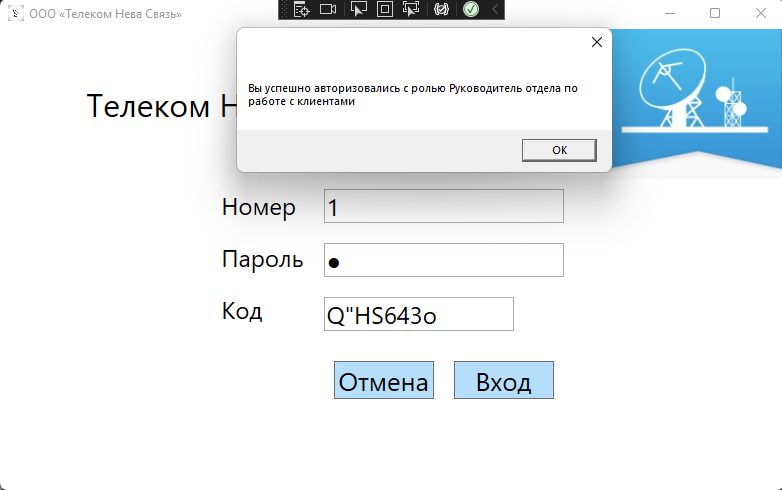


Рисунок 4 – Окно авторизации

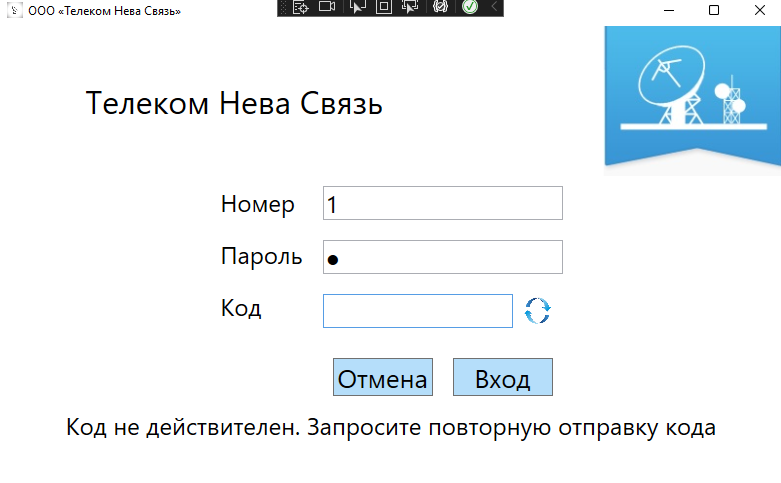


Рисунок 5 – Окно авторизации если пользователь не успел ввести код

Авторизация осуществляется по следующему алгоритму:

* сотрудник вводит номер и пароль;
* при вводе пароля сотрудником и нажатии клавиши Enter на служебный телефон отправляется СМС с единоразовым кодом доступа;
* сотрудник вводит код и далее получает доступ к необходимому функционалу.

Вместо создания эмулятор СМС сообщений пользователю выводится MessageBox с кодом авторизации.

Если в течение 10 секунд после закрытия окна с кодом пользователь не успевает ввести код авторизации, то код необходимо получить заново нажатием на изображение около поля «Код», которое выведет новый код для пользователя.

После успешной авторизации сотруднику должно быть выведено сообщение с названием его роли.

В рамках второй сессии были поставлены следующие задачи;

* разработка модуля работы с абонентами;
* разработка модуля CRM.

Разработка осуществлялась в соответствие с макетом (рис. 6 -7). Для просмотра проекта с данными модулями необходимо переключиться на ветку Second. При открытии осуществляется переход на страницу с абонентами, к которой есть доступ у всех ролей пользователей. Также отображаемые элементы меню изменяются в зависимости от доступа роли текущего пользователя. Также у каждого пользователя изменяются события в правой части экрана.

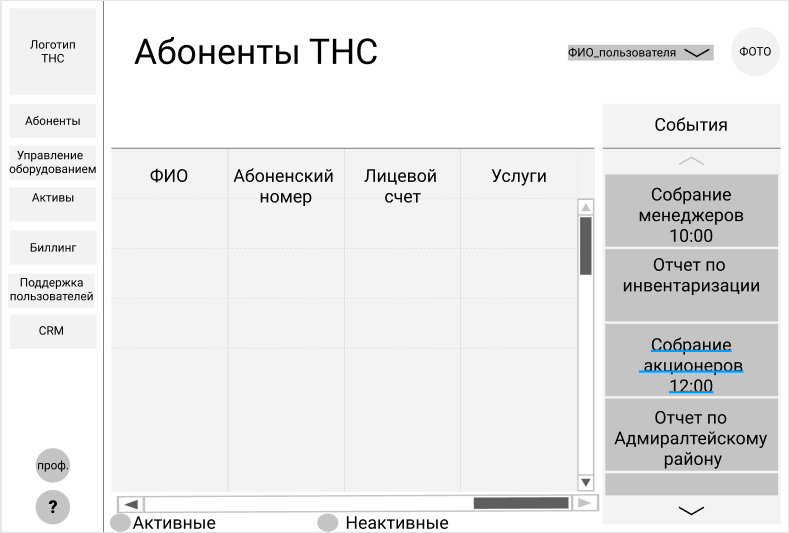


Рисунок 6 – Макет работы с абонентами

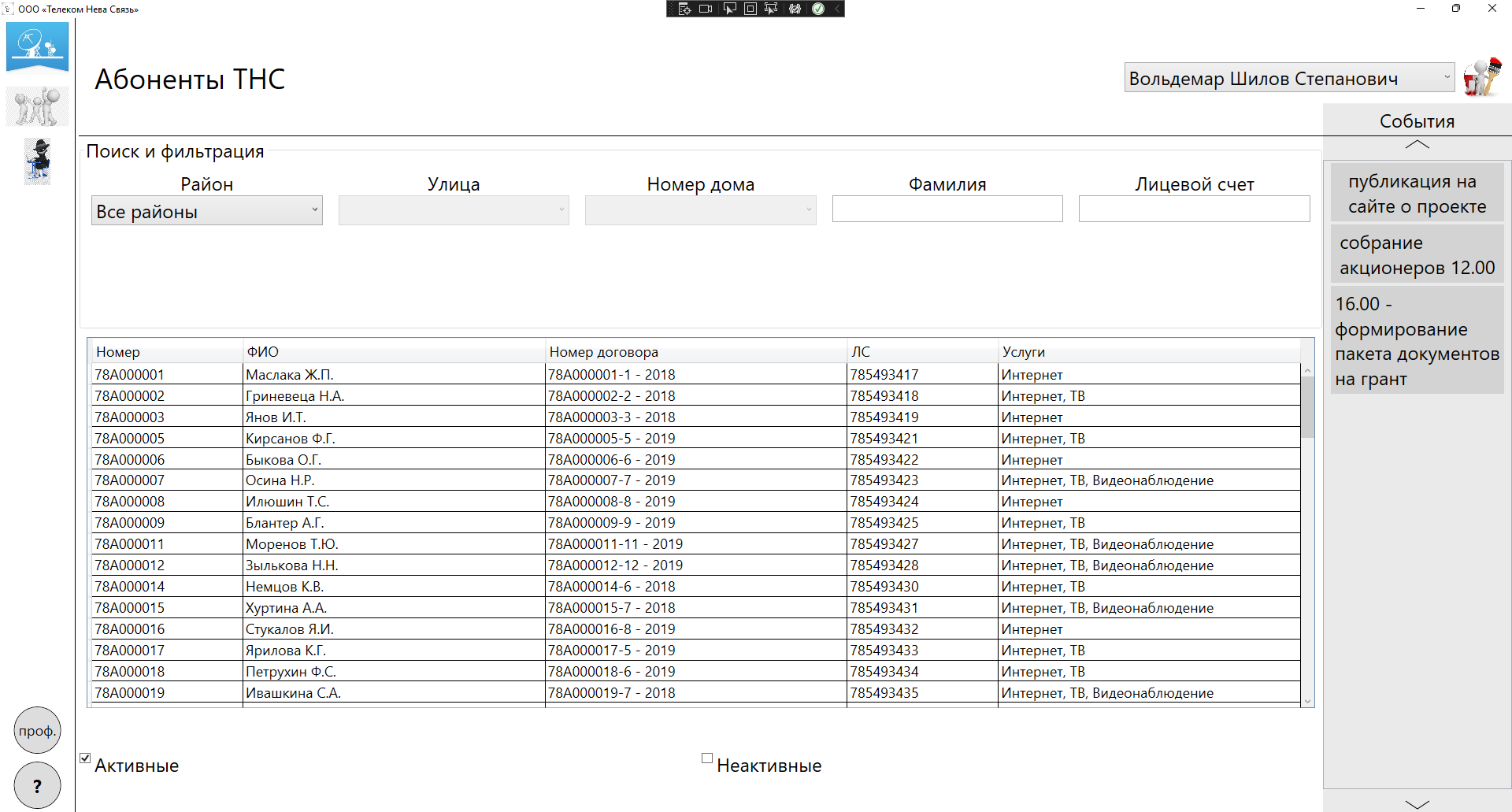


Рисунок 7 – Окно работы с абонентами

В рамках задания нужно было реализовать смену кнопок с надписью на пиктограммы (рис. 8).

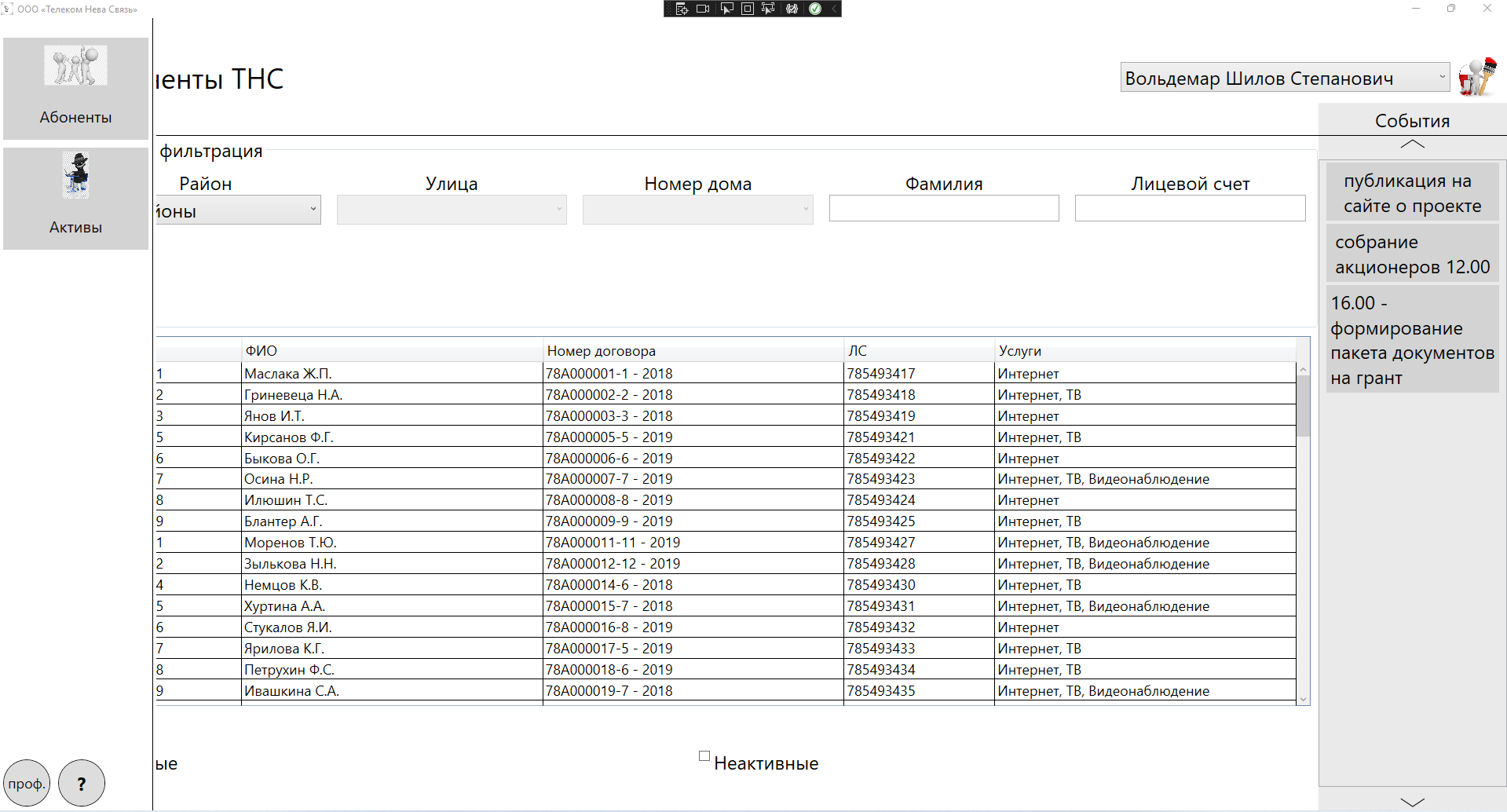


Рисунок 7 – Дополнительный формат меню окна

Также на вкладке Абоненты предусмотрена сортировка и поиск нужных абонентов. Фильтрация по районам города (рис. 9), поиск по фамилии абонента (рис. 10) и поиск по лицевому счету (рис. 11). Также поиск и фильтрация работают одновременно (рис. 12).

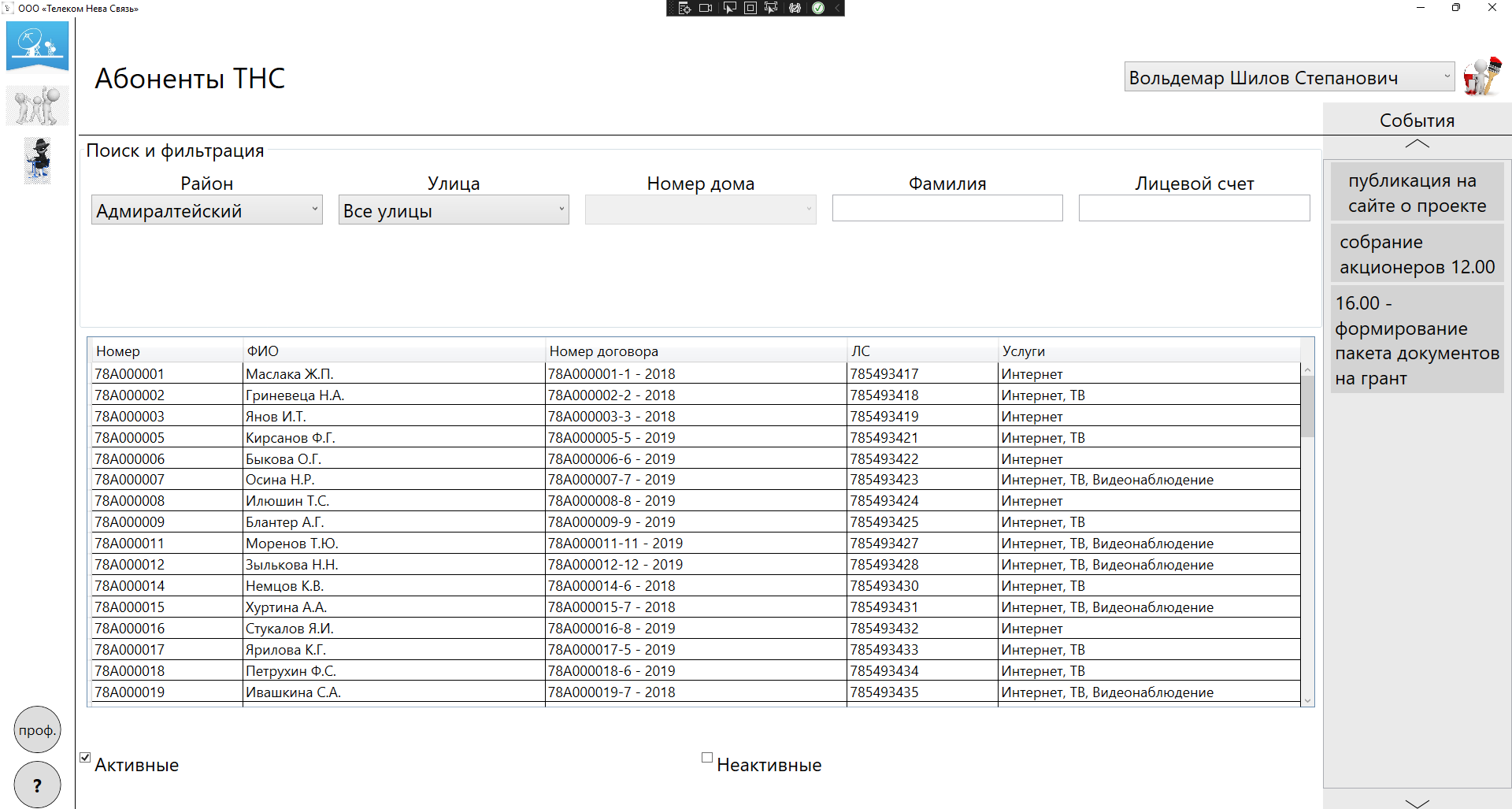


Рисунок 9 – Фильтрация по району города

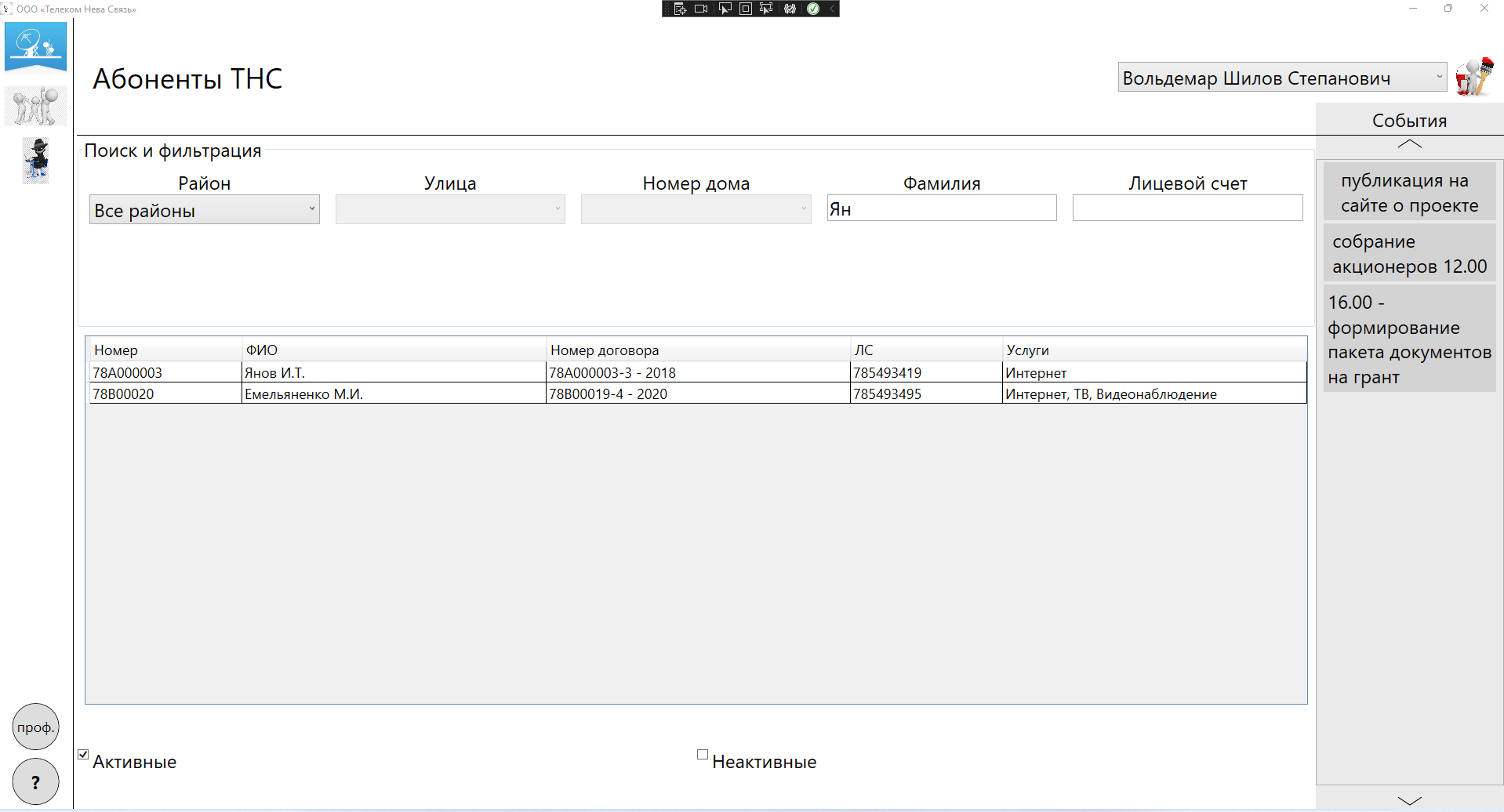


Рисунок 10 – Поиск абонентов по фамилии

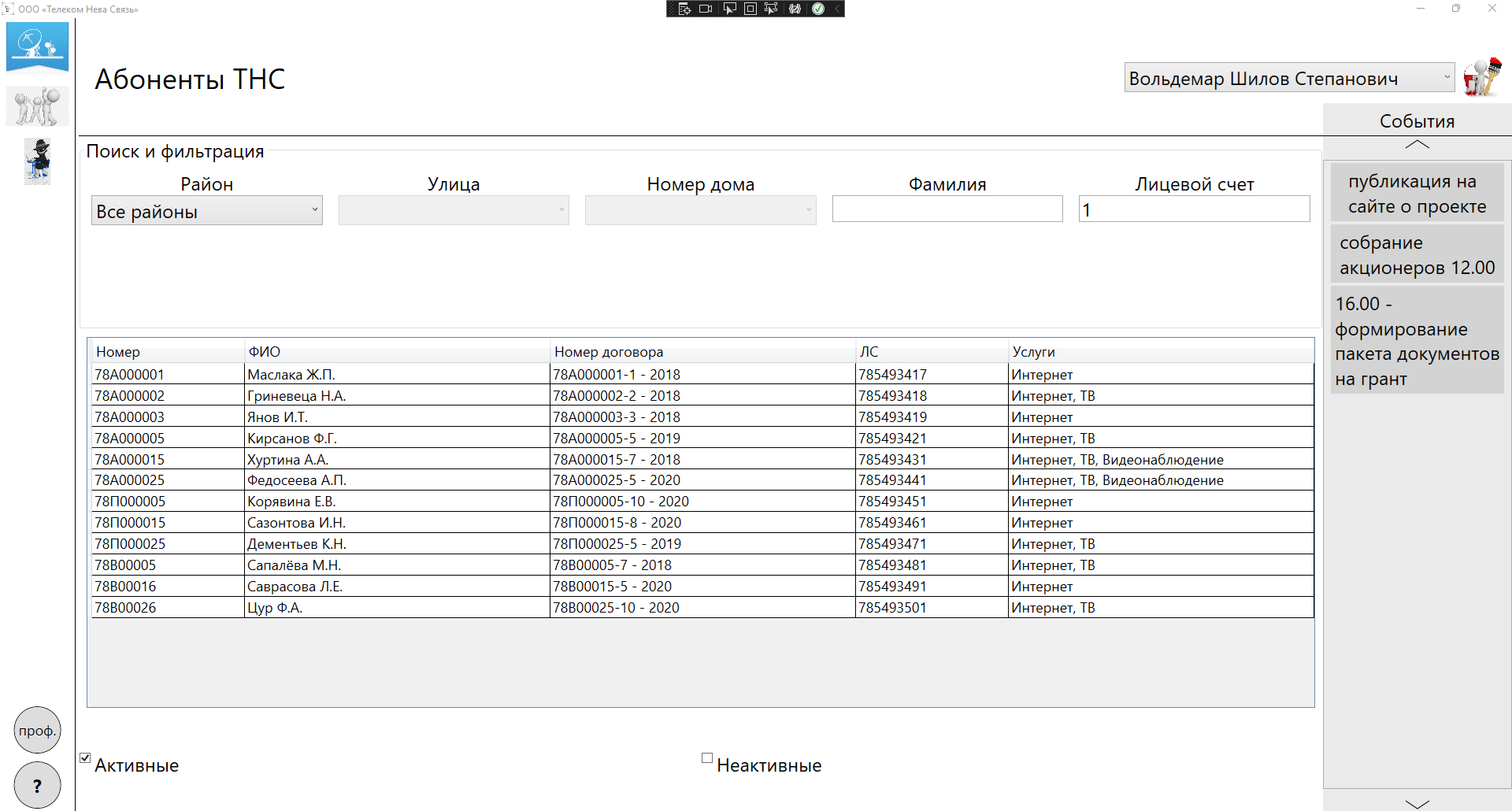


Рисунок 11 – Поиск абонентов по лицевому счету

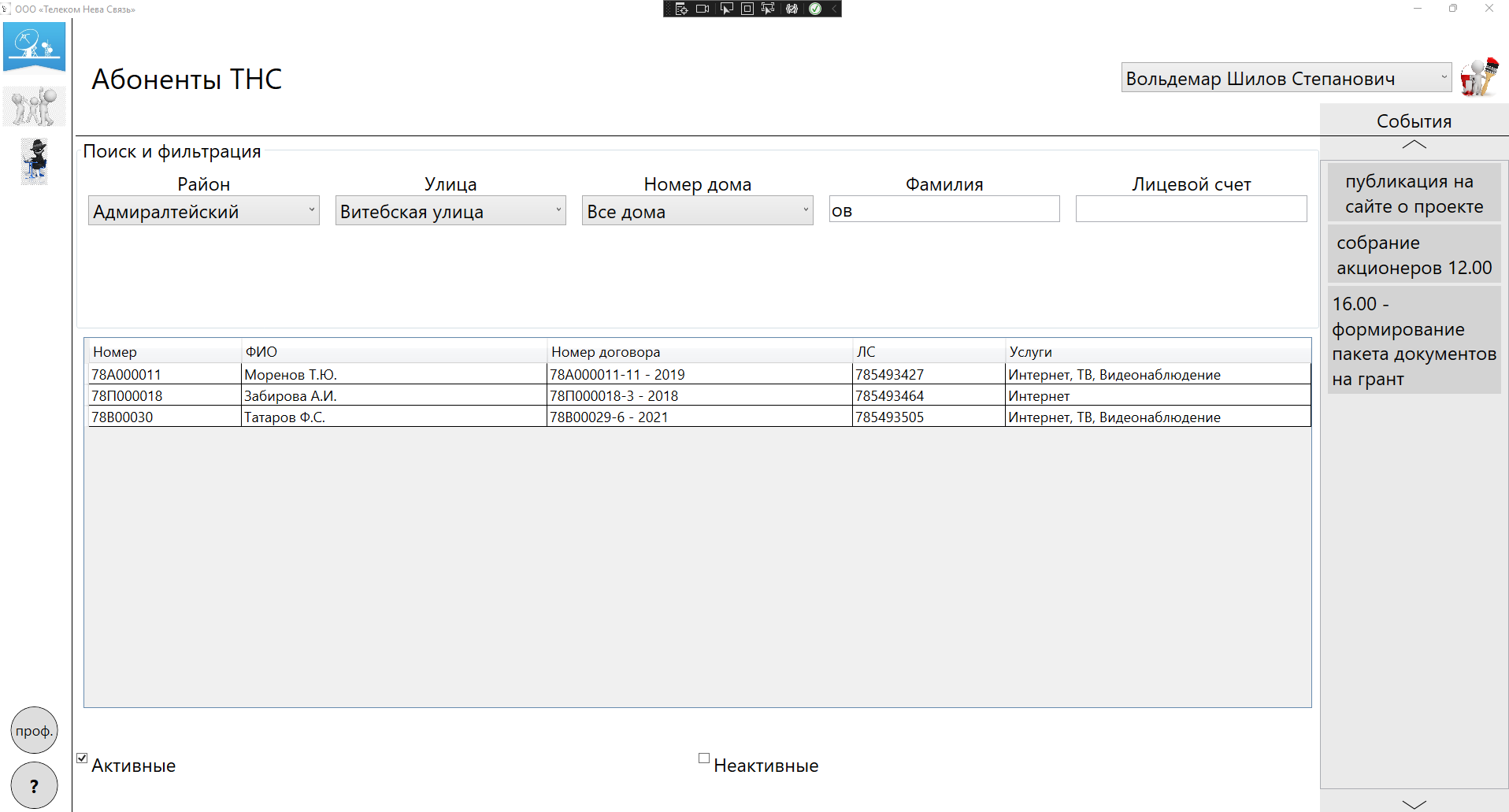


Рисунок 12 – Одновременная работа поиска и фильтрации

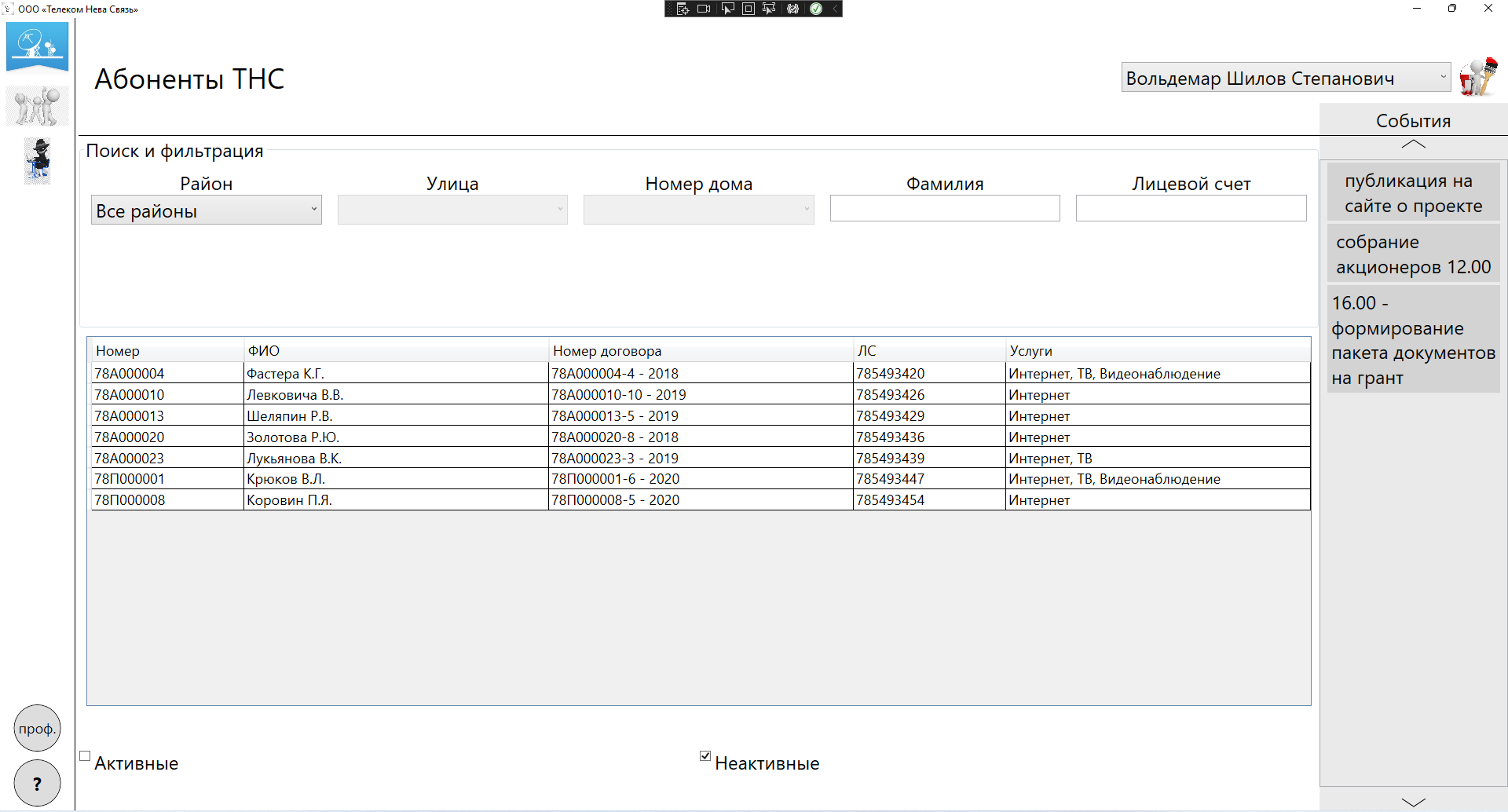


Рисунок 13 – Отображение неактивных пользователей

Вкладка CRM включает в себя заполнение заявки пользователя при обращении к оператору (рис. 14). После нажатия на кнопку «Сформировать заявку» открывается окно с карточкой заявки и подробным описанием запроса от пользователя (рис. 15).

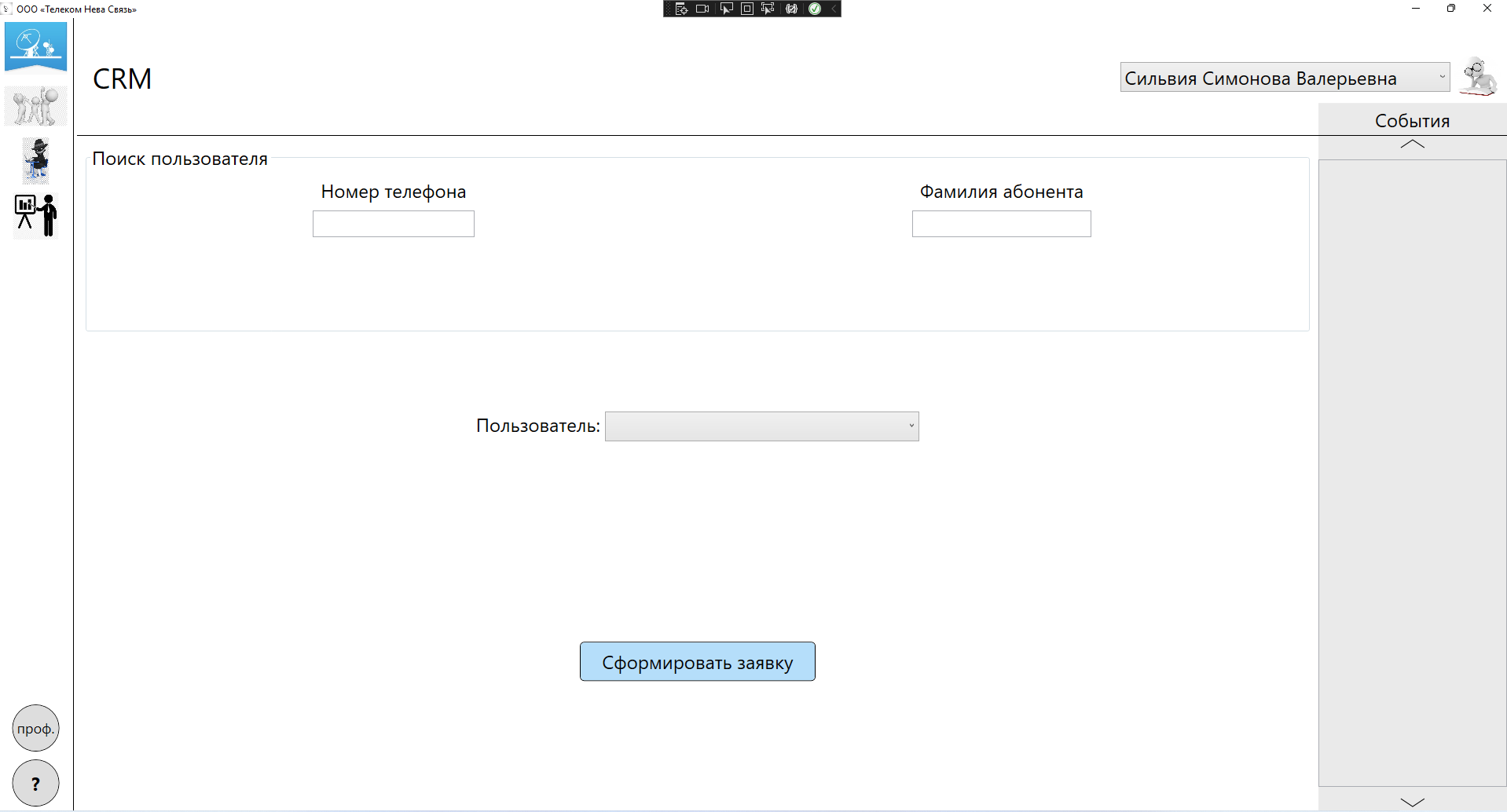


Рисунок 14 – Вкладка CRM

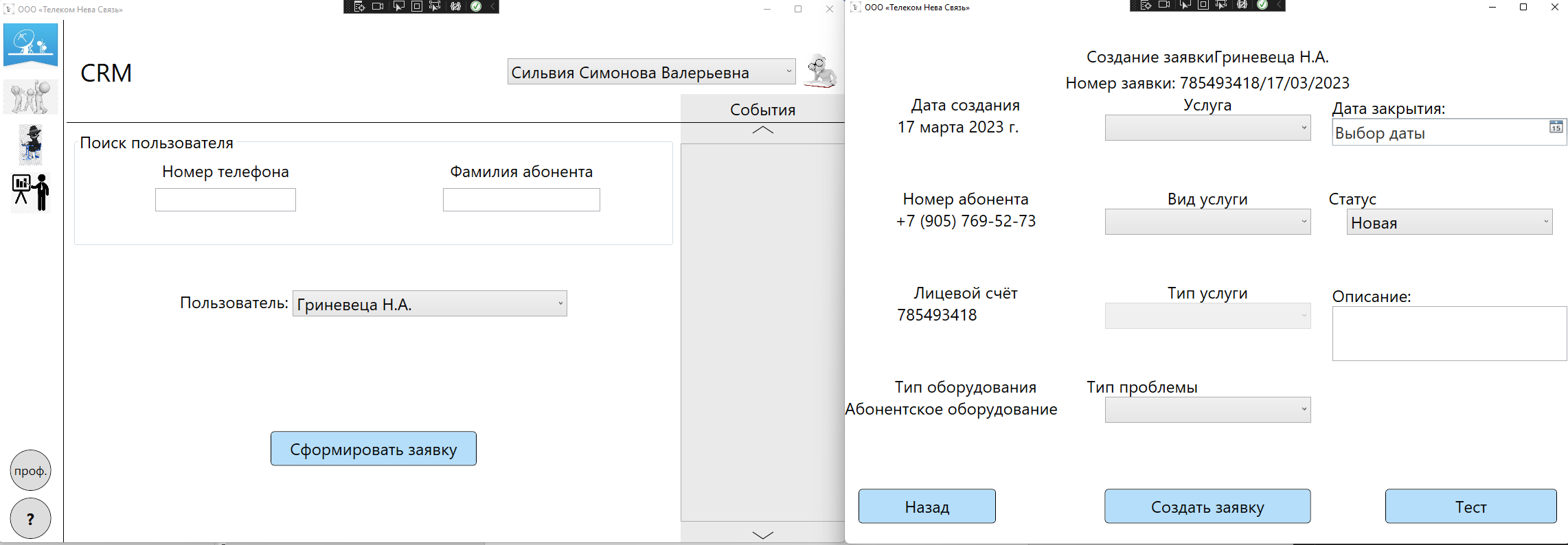


Рисунок 15 – Формирование заявки

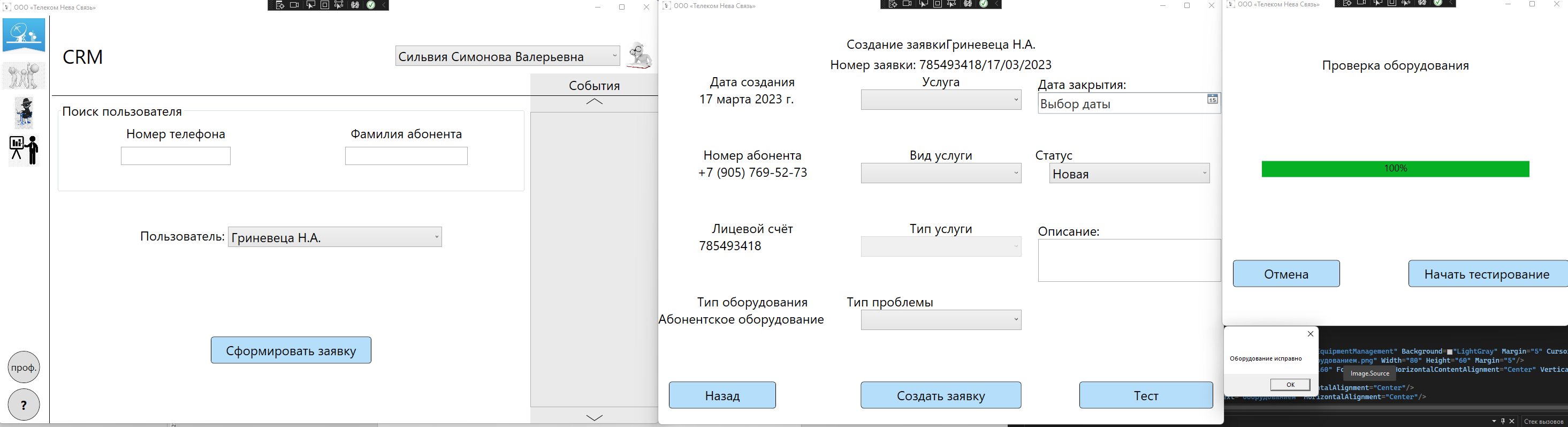


Рисунок 16 – Проверка оборудования клиента по заявке

В ходе работы были разработаны и спроектированы модули для авторизации и работы с клиентами. Так же была разработана и заполнена база данных. В ходе работы также был добавлен readme.md файл для начала работы с проектом. В ветке мастер добавлены все необходимые файлы для проекта

# **РАЗРАБОТКА WPF ПРИЛОЖЕНИЯ И ТЕСТИРОВАНИЕ**

Перед началом разработки приложения нужно было привести базу данных к необходимому формату.

Предоставленный в материалах скрипт был выгружен в MS SQL Server. Далее были доработаны имеющиеся таблицы, а именно добавлены новые таблицы для приведения к третьей нормальной форме, связи и поля в имеющихся таблицах. Результат приведения базы к нужному виду представлен на рисунке 1.

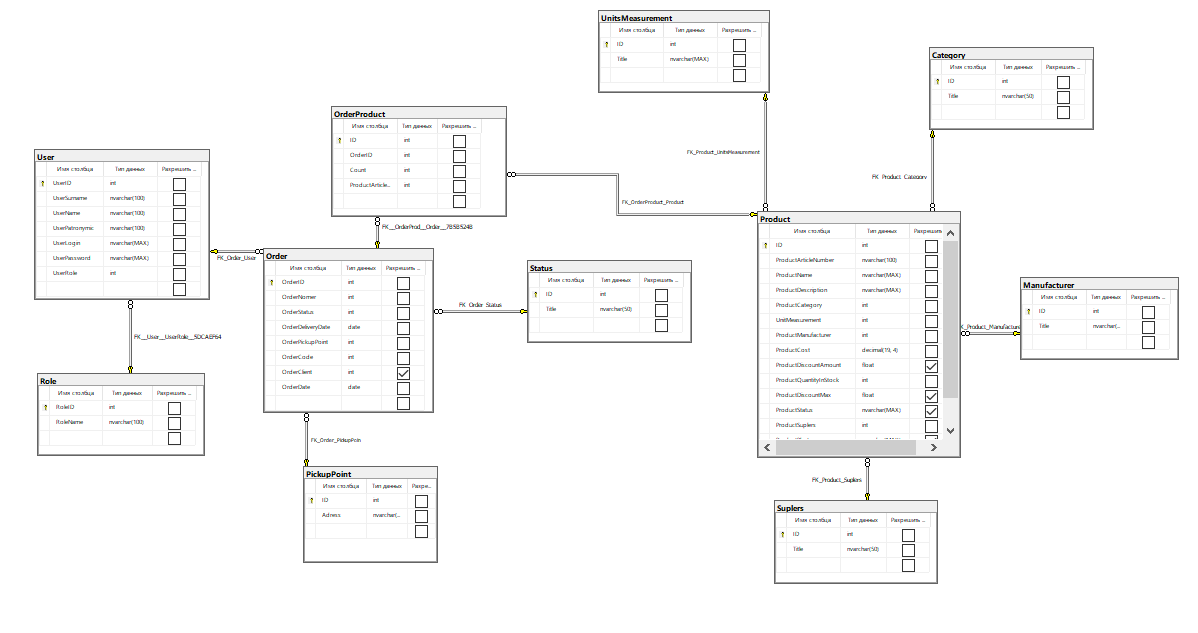


Рисунок 1 – Диаграмма базы данных 41practic6

Далее для заполнения базы были приведены к нужному формату данные для импорта из предоставленных в материалах excel таблицах. Пример заполненной таблицы представлен на рисунке 2.

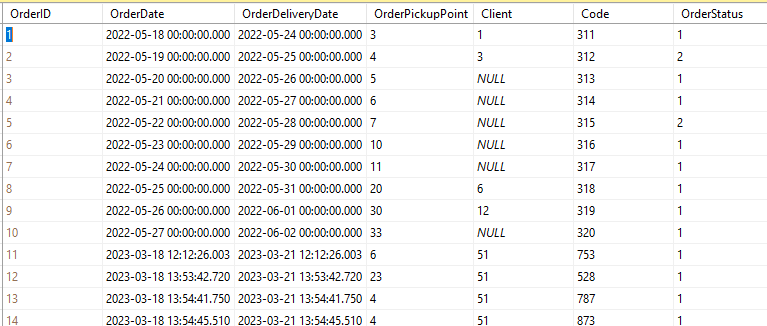


Рисунок 2 – Таблица, заполненная данными для импорта

После завершения работы с данными переходим к разработке WPF приложения.

В первую очередь была реализована страница авторизации с дополнительной проверкой captcha в случае неверной авторизации (рис. 3).

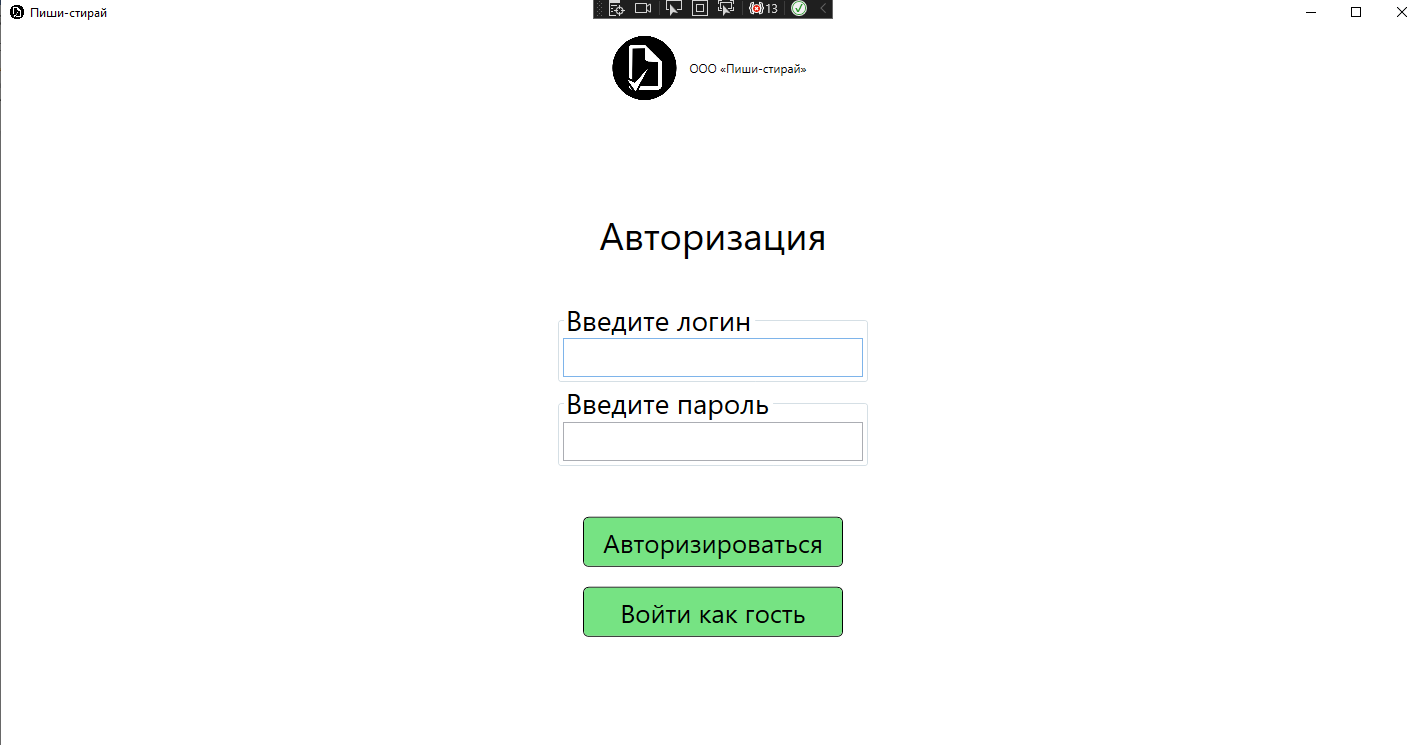


Рисунок 3 – Страница авторизации

В случае неуспешной авторизации появляется captcha. В случае если пользователь не вводит проверочный код с первого раза, то кнопки блокируются на 10 секунд и появляется таймер, оповещающий пользователя об оставшемся времени до следующей попытки авторизации (рис. 4).

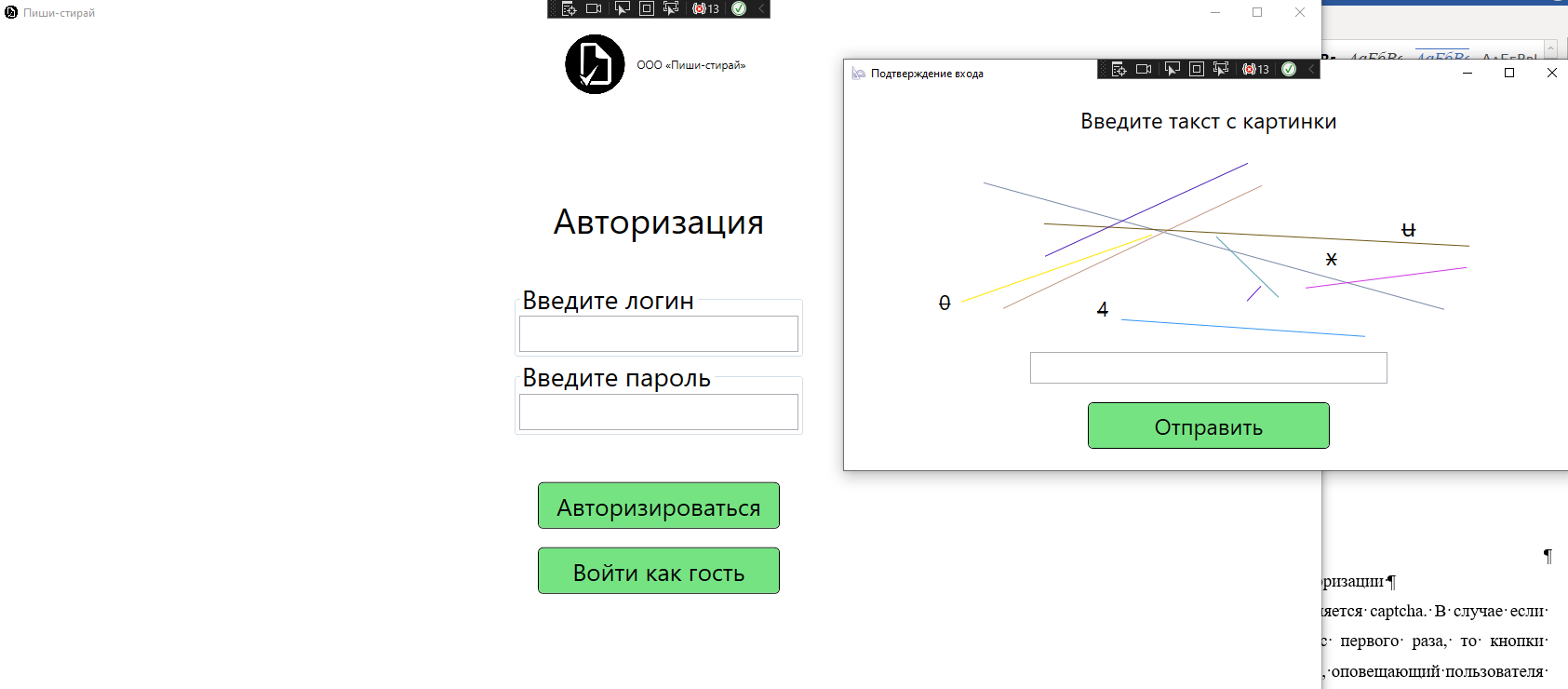


Рисунок 4 – Неверный ввод captcha

После авторизации осуществляется переход на страницу товаров. По умолчанию выбрана сортировка по возрастанию стоимости и весь диапазон скидок. Для гостей и пользователя становится недоступна возможность удаления товара и отображается она только при авторизации администратором и менеджером (Рис. 5-6).



Рисунок 5 – Окно товаров для пользователя и гостя

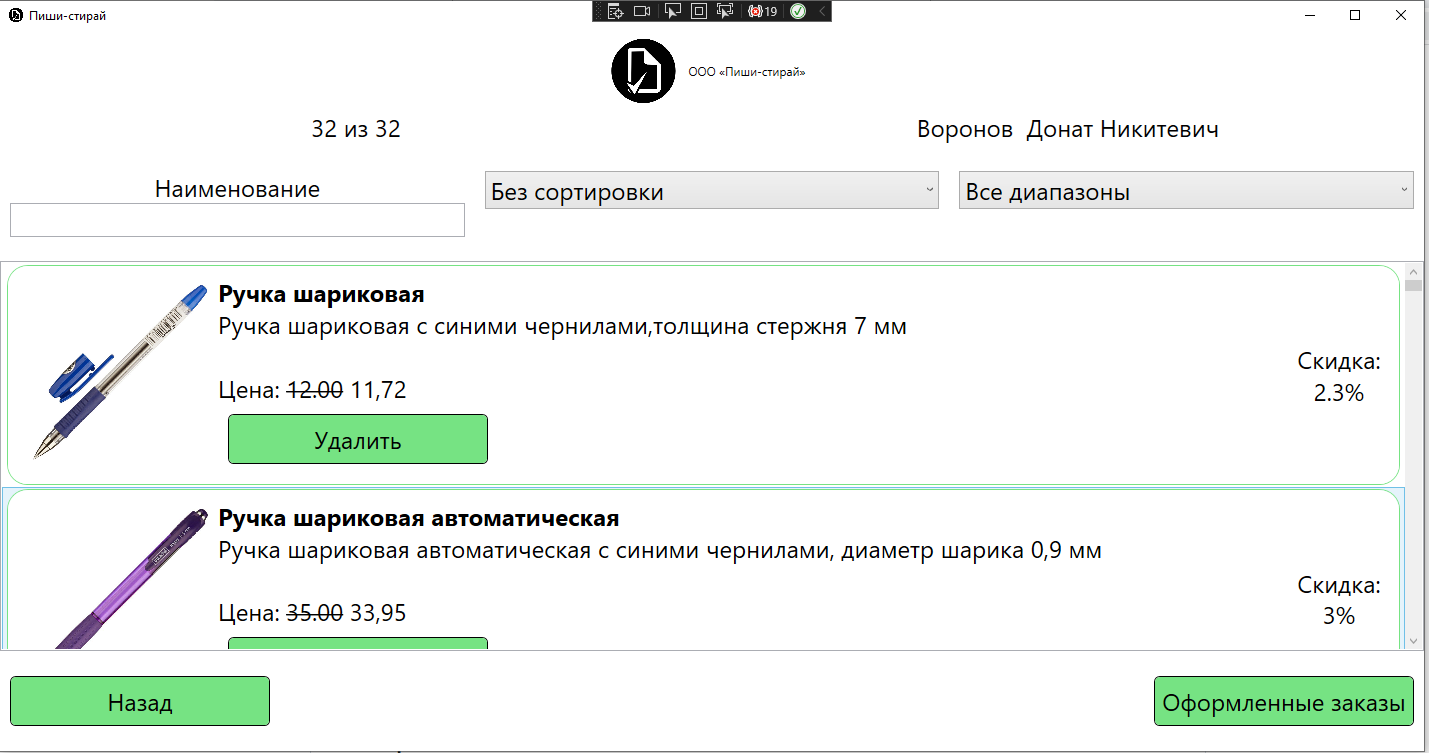


Рисунок 6 – Окно товаров для администратора и менеджера

При нажатии на элемент правой кнопкой мыши появляется контекстное меню, в котором есть пункт «добавить в корзину», при нажатии на него выбранный товар добавится в корзину. Работы данной функции продемонстрирована на рисунках 7-8.

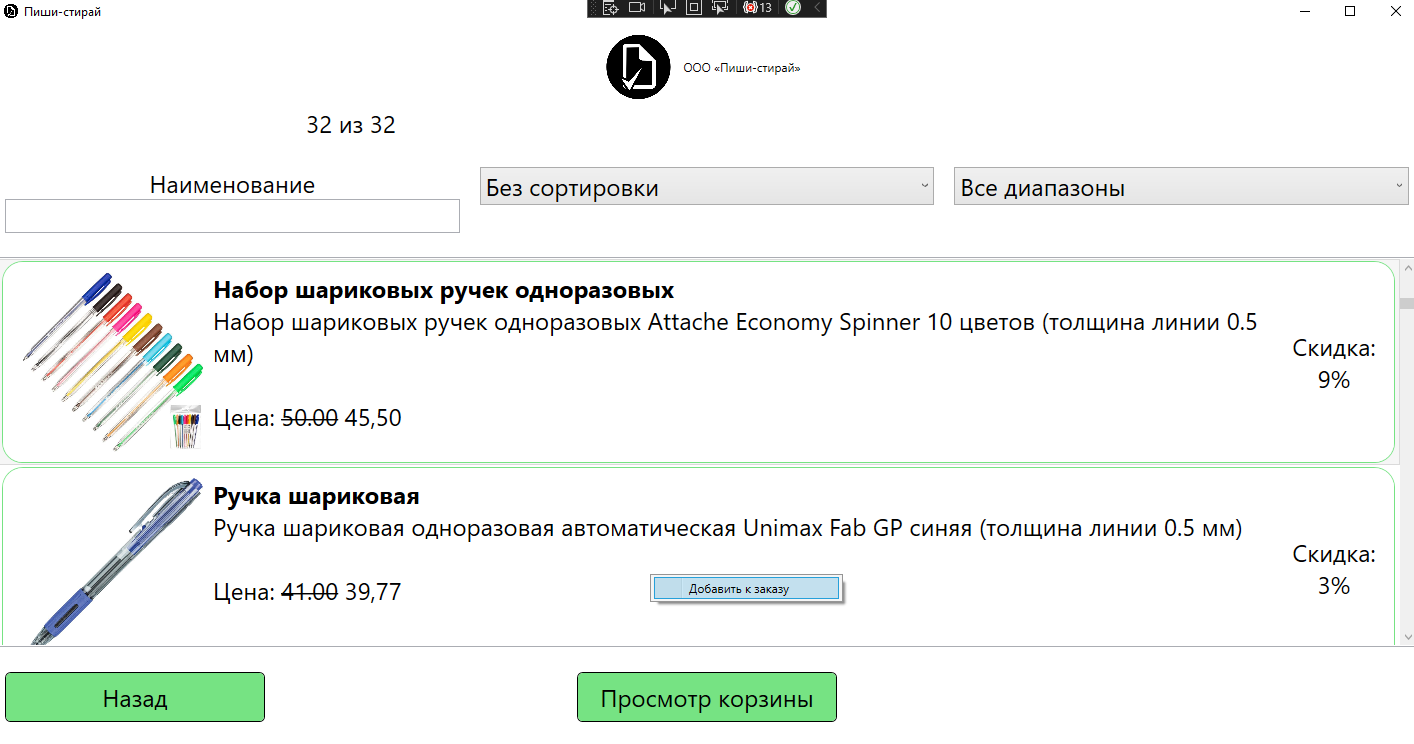


Рисунок 7 – Использование контекстного меню

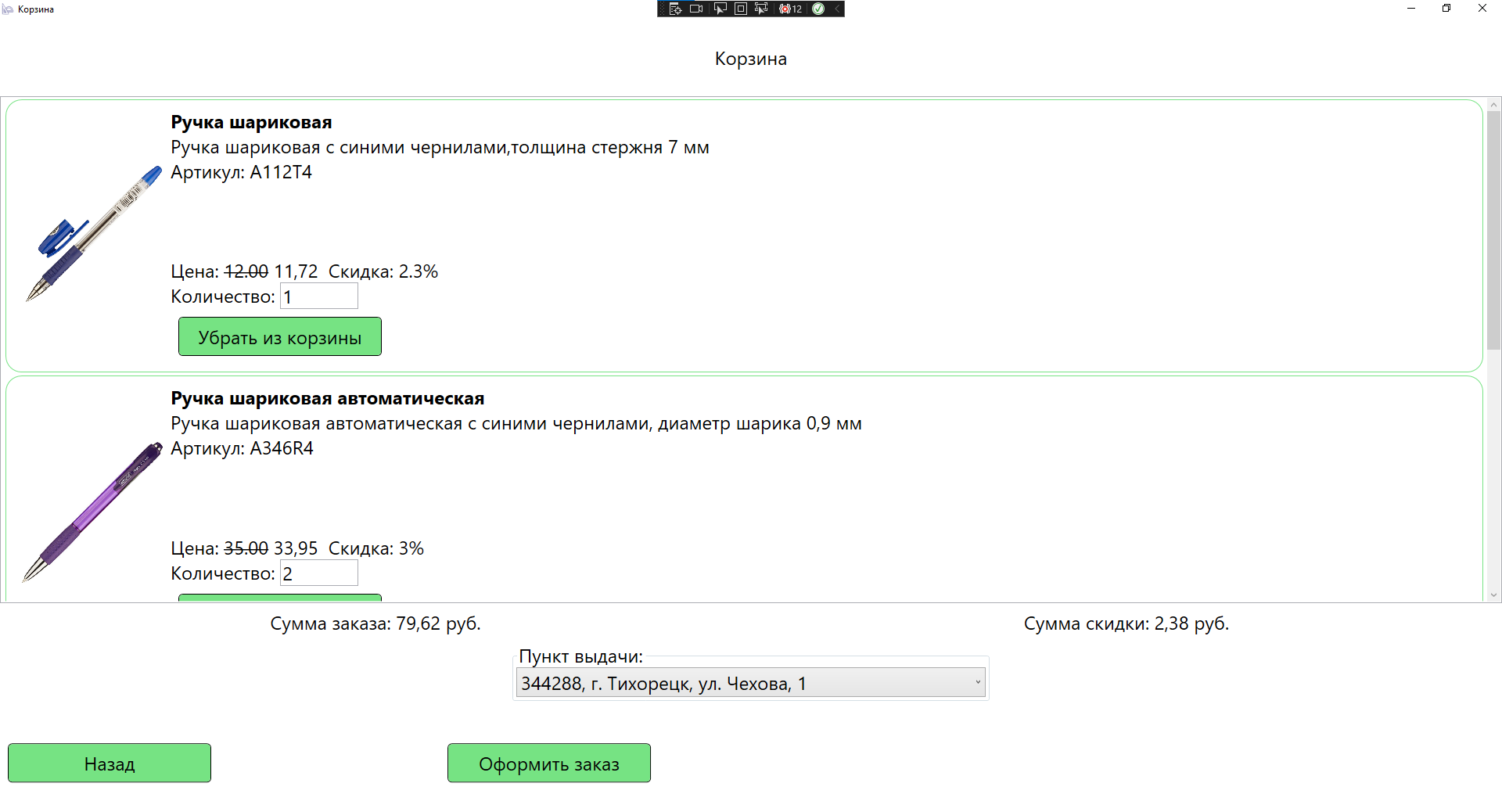


Рисунок 8 – Товары, добавленные через контекстное меню

Товары из корзины можно удалить, введя значение «0» в поле количество или с помощью кнопки удалить (рис. 9).

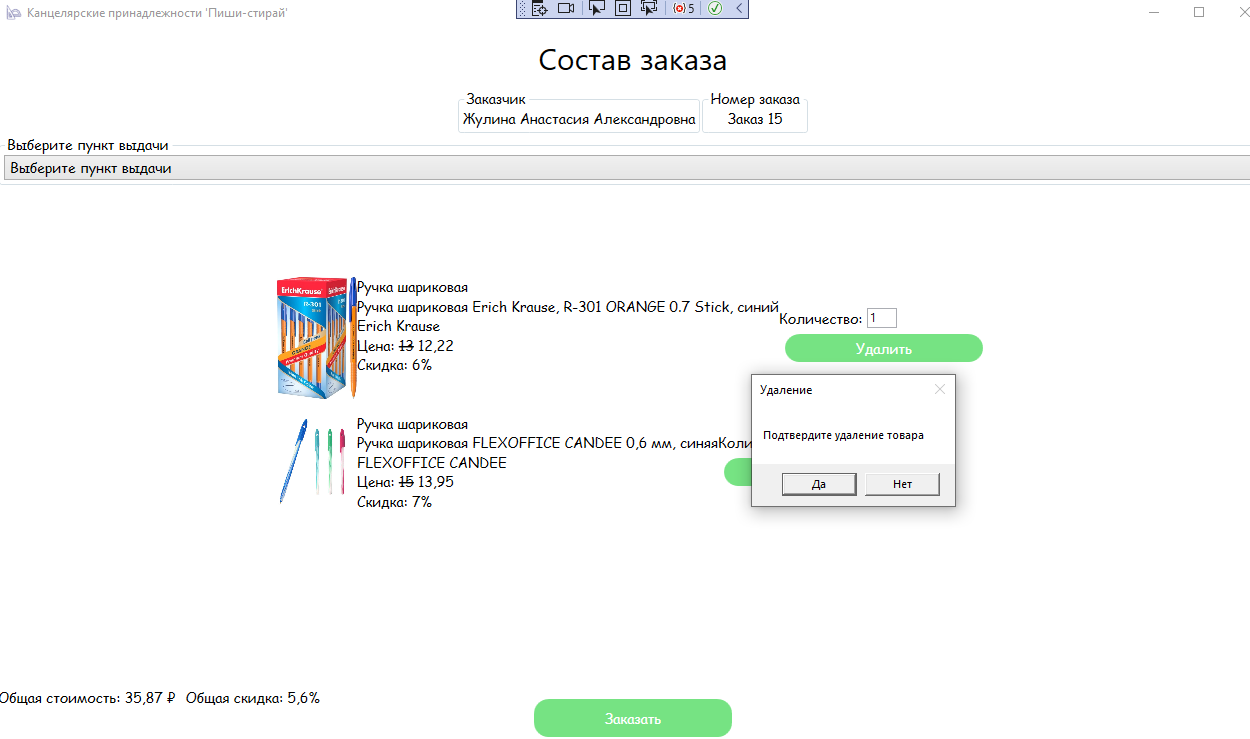


Рисунок 9 – Удаление товара из корзины, нажатием на элемент интерфейса

После оформления заказа в БД появится соответствующая запись.

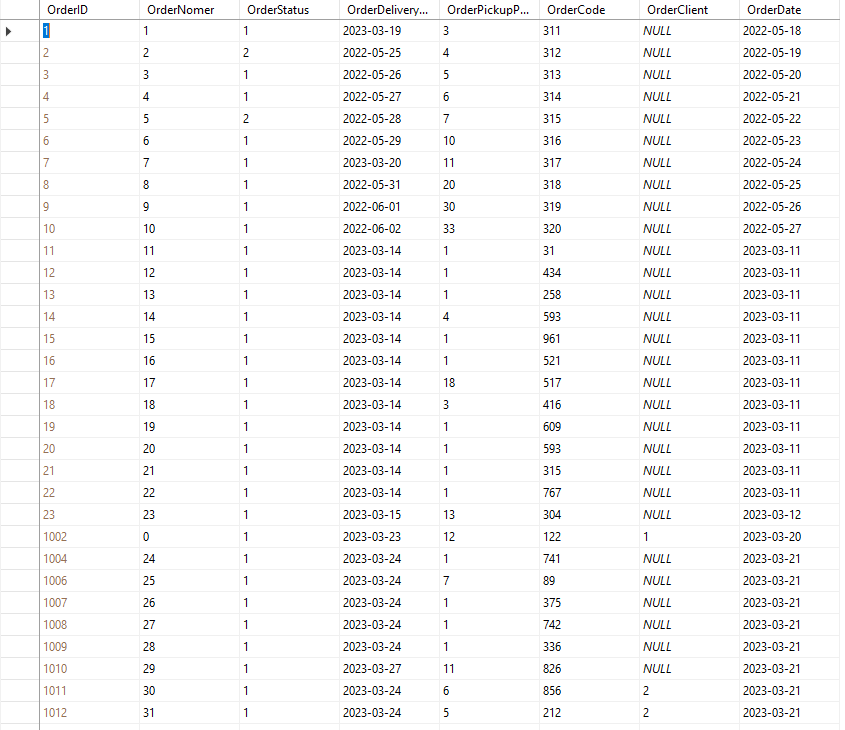


Рисунок 10 – Запись о заказе

В рамках второй сессии нужно было реализовать проект DLL-библиотеки и unit-тестов к ней. Библиотека должна возвращать список свободных временных интервалов. Был реализован метод, сравнивающий время начала и окончания консультации и длительность консультации, в случае если время начала больше времени окончания или время консультации меньше или равное 0, то программа возвращает null. После данной проверки формируется лист с рабочим временем, в котором идет проверка объектом TimeSpan по времени начала, если появляются консультации, которые указаны раньше рабочего времени, то далее они учитываться не будут.

Далее в цикле осуществляется проверка вхождения времени консультации в промежуток рабочего времени менеджера и проверка на незанятое время.

После окончания прохождения цикла список переформировывается в строковый массив, который возвращает список временных интервалов.

Для этой библиотеки были разработаны юнит-тесты. Прохождение юнит тестов представлено на рисунке 11.

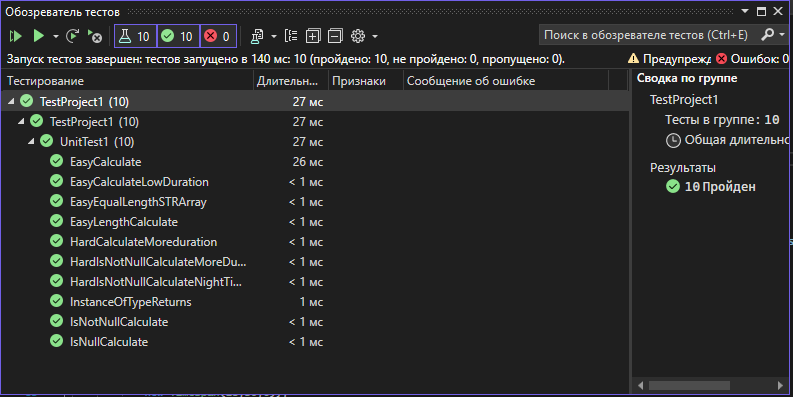


Рисунок 11 – Прохождение юнит-тестов

# **РАЗРАБОТКА МОБИЛНОГО ПРИЛОЖЕНИЯ**

В рамках учебной практики нужно было разработать мобильное приложение в соответствие с макетом (рис. 1 – 2). А также реализовать логику переходов в соответствие с прототипом. Ссылка на проект: <https://github.com/dreyvania999/UP_Mob>



Рисунок 1 – Макет приложения

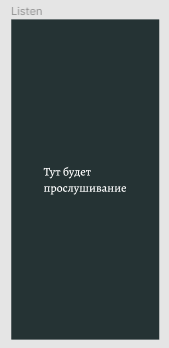
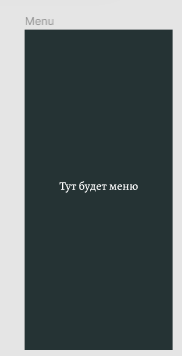


Рисунок 2 – Страницы заглушки

Структура проекта представлена на рисунках 3-4. Было создано три адаптера и три маски для работы с Api. AdapterElement и MaskElement для работы со списком цитат, AdapterFeeling и MaskFeeling работы со списком ощущений, а AdapterImageProfile и AMaskImageProfile для работы с изображением профиля пользователя.

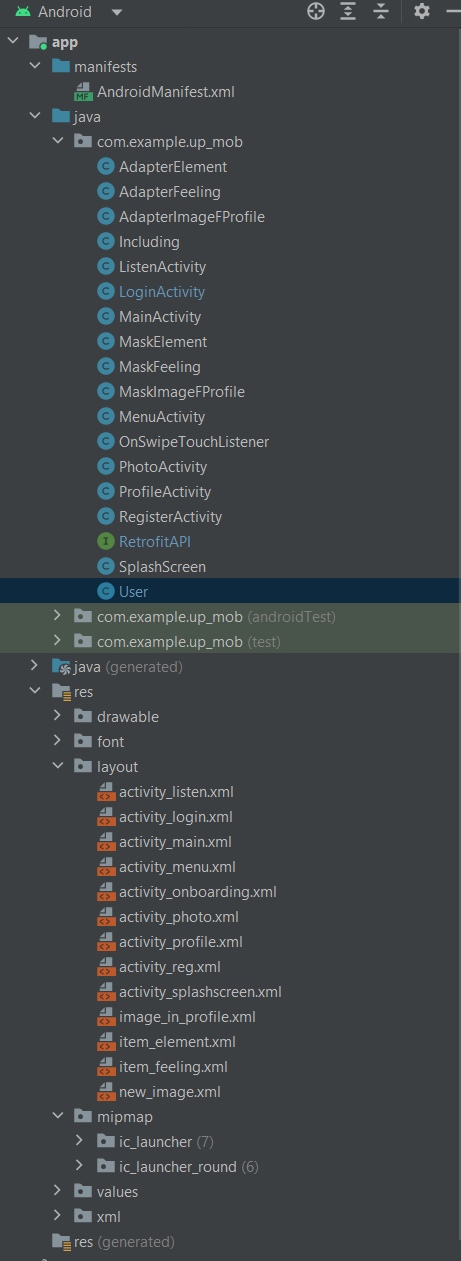


Рисунок 3 – Классы и интерфейсы

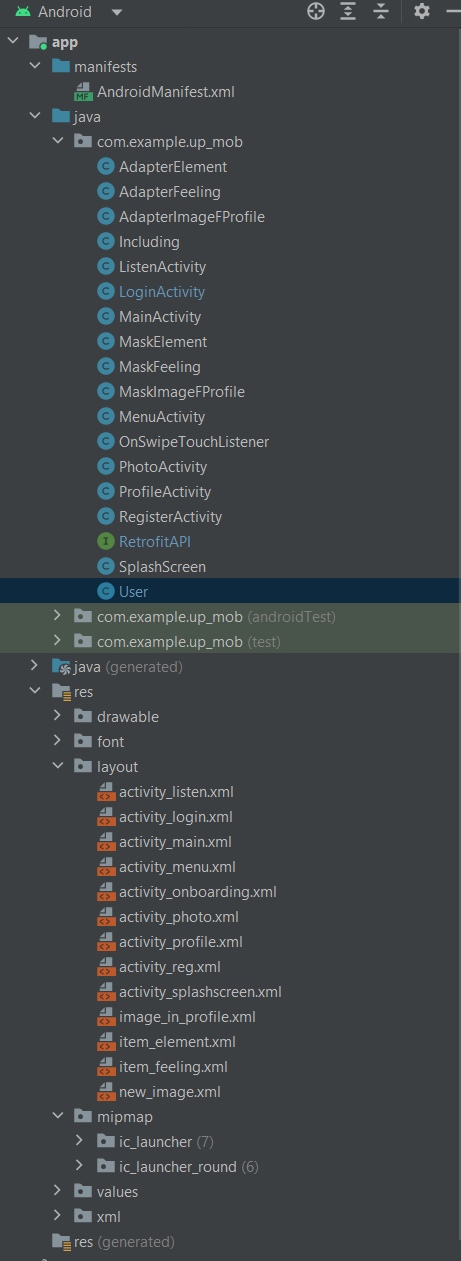


Рисунок 4 – Layout

Экран загрузки, представленный на рисунке 5, отображается при каждом запуске приложения.



Рисунок 5 – Экран загрузки

При входе в приложение пользователю представляется возможность войти в профиль или зарегистрироваться (рис. 6).

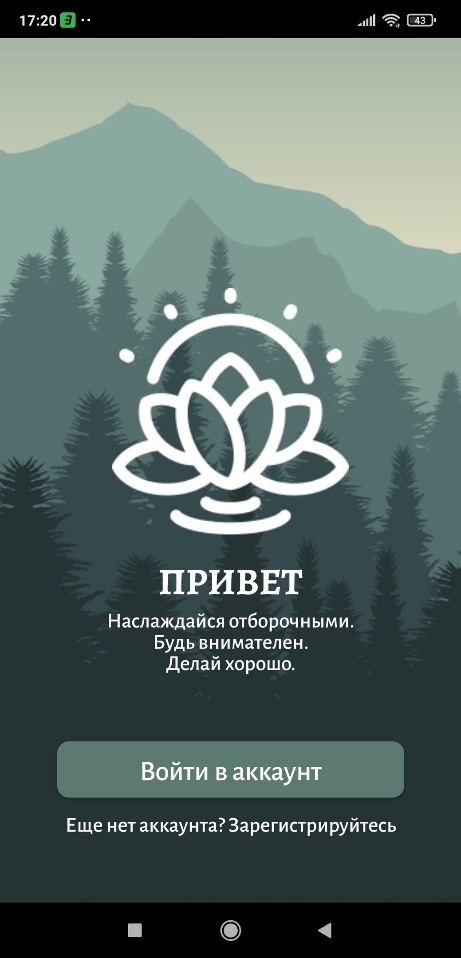


Рисунок 6 – Окно входа и регистрации

После соответствующего выбора пользователь перейдет либо на страницу авторизации, либо на страницу заглушку для регистрации (рис. 7).

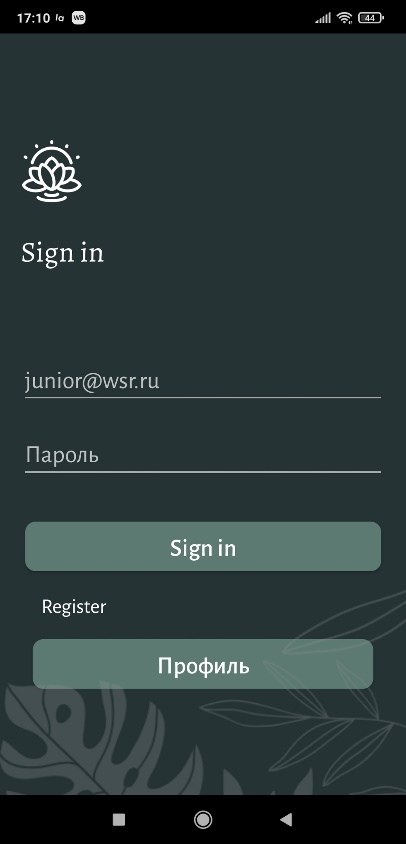


Рисунок 7 – Страница авторизации

При обращении к Api после ввода данных пользователь получает токен авторизации.

Был реализован метод для последующего запуска приложения с последней учетной записью, чтобы пользователь пропускал этап выбора авторизации и входа в приложение при каждом запуске.

После успешной авторизации пользователь переходит на страницу со списком цитат и ощущений, возвращаемых из Api (рис. 8). Реализована возможность листать список цитат и ощущений свайпом по экрану.



Рисунок 8 – Окно Main

После нажатия «exit» происходит переход на страницу авторизации, на которой отображается почта ранее авторизованного пользователя (рис. 9). E-mail сохраняется и после перезагрузки приложения.

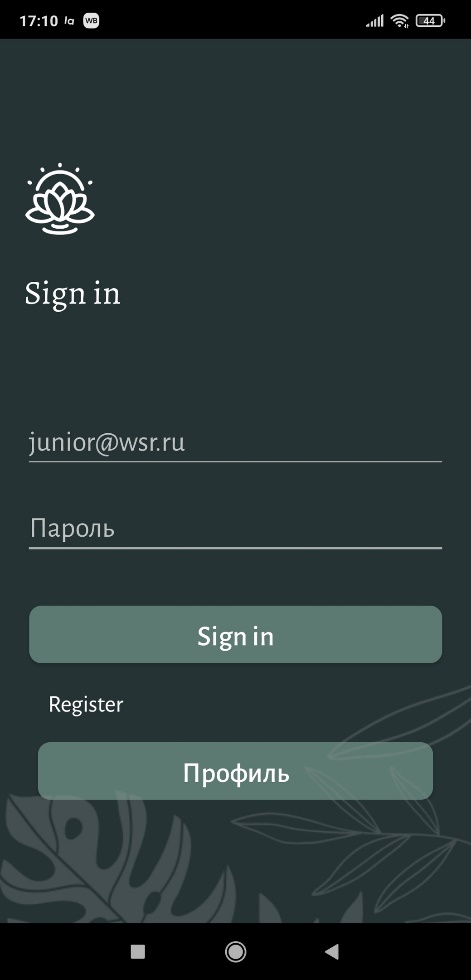


Рисунок 9 – Страница авторизации после выхода пользователя из аккаунта

Внешний вид профиля пользователя представлен на рисунке 10.

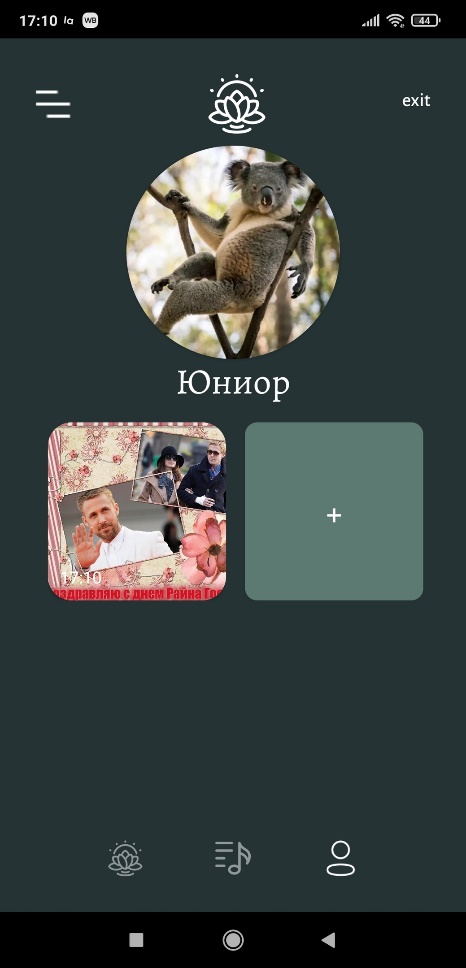


Рисунок 10 – Профиль пользователя

При открытии фотографии отображается само фото и кнопки «удалить» и «закрыть» (рис. 11). Для реализации функционала кнопок используются жесты, при проведении по экрану справа налево фотография будет удалена, если пользователь проводит слепа направо, то фотография закроется и откроется страница профиля с галереей. Также фотографию можно приближать и возвращать к изначальному размеру двойным нажатием по экрану.



Рисунок 11 – Страница фотографии