**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 КОМПЬЮТЕРНЫХ СИСТЕМ

Руководитель: \_\_\_\_\_\_\_\_\_\_ Мухина Л.В., Мухин Н.А., Мамшева Ю.С. 21.03.2023 г.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Быкова А.А. 21.03.2023 г.

Специальность, группа: 09.02.07, 41П

Нижний Новгород

2023 г.

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc130045988)

[**Разработка программных модулей** 4](#_Toc130045989)

[**Разработка программных модулей** 17](#_Toc130045990)

[**Разработка мобильных приложений** 26](#_Toc130045991)

[**ЗАКЛЮЧЕНИЕ** 36](#_Toc130045992)

[**ПРИЛОЖЕНИЯ** 37](#_Toc130045993)

# **ВВЕДЕНИЕ**

Важным элементом процесса подготовки специалиста в области программирования является практика. Актуальность учебной практики заключается в получении практических навыков, без которых невозможно обойтись специалисту.

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

Задачи учебной практики по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем:

• Разработать десктопное приложение для магазина канцелярии «Пиши-Стирай»;

• Разработать динамическую библиотеку, которая позволит вернуть список свободных временных интервалов в графике сотрудников для формирования оптимального графика работы и реализовать 10 unit-тестов для данной библиотеки;

• Написать тестовую документацию на добавление товара администратором;

• Спроектировать базу данных на основе предметной области;

• Создать словарь данных для ER-диаграммы;

• Разработать авторизацию для ООО «Телеком Нева Связь»;

• Разработать приложение для работы с абонентами;

• Разработать мобильное приложение в соответствии с дизайном

# **Разработка программных модулей**

В рамках выполнения задания необходимо разработать основные модули информационной системы для ООО «Пиши-стирай» - магазин по продаже канцелярских товаров:

• неавторизованный клиент и авторизованный клиент может просматривать товары и формировать заказ;

• менеджер может просматривать товары, формировать и редактировать заказы;

• администратор может добавлять/редактировать/удалять товары, просматривать и редактировать заказы.

Кроме того, разрабатываемая информационная система предполагает установку на терминалах при входе в торговые центры города. На терминале клиент (авторизованный и неавторизованный) может просмотреть товары, сформировать заказ и выбрать удобный для него пункт выдачи.

Предусмотреть следующие страницы:

* Авторизация;
* Список товаров;
* Список оформленных заказов;
* Просмотр формируемого заказа (корзина).

Изначально необходимо восстановить базу данных из скрипта. Для восстановления таблиц в созданную базу данных воспользуемся предоставленным скриптом (ms.sql). Необходимо доработать эту базу данных, обеспечив хранение всех данных, которые были предоставлены. Для этого нужно будет добавить необходимые сущности, атрибуты и связи.

В результате доработки база данных выглядит соответствующим образом (рис.1).

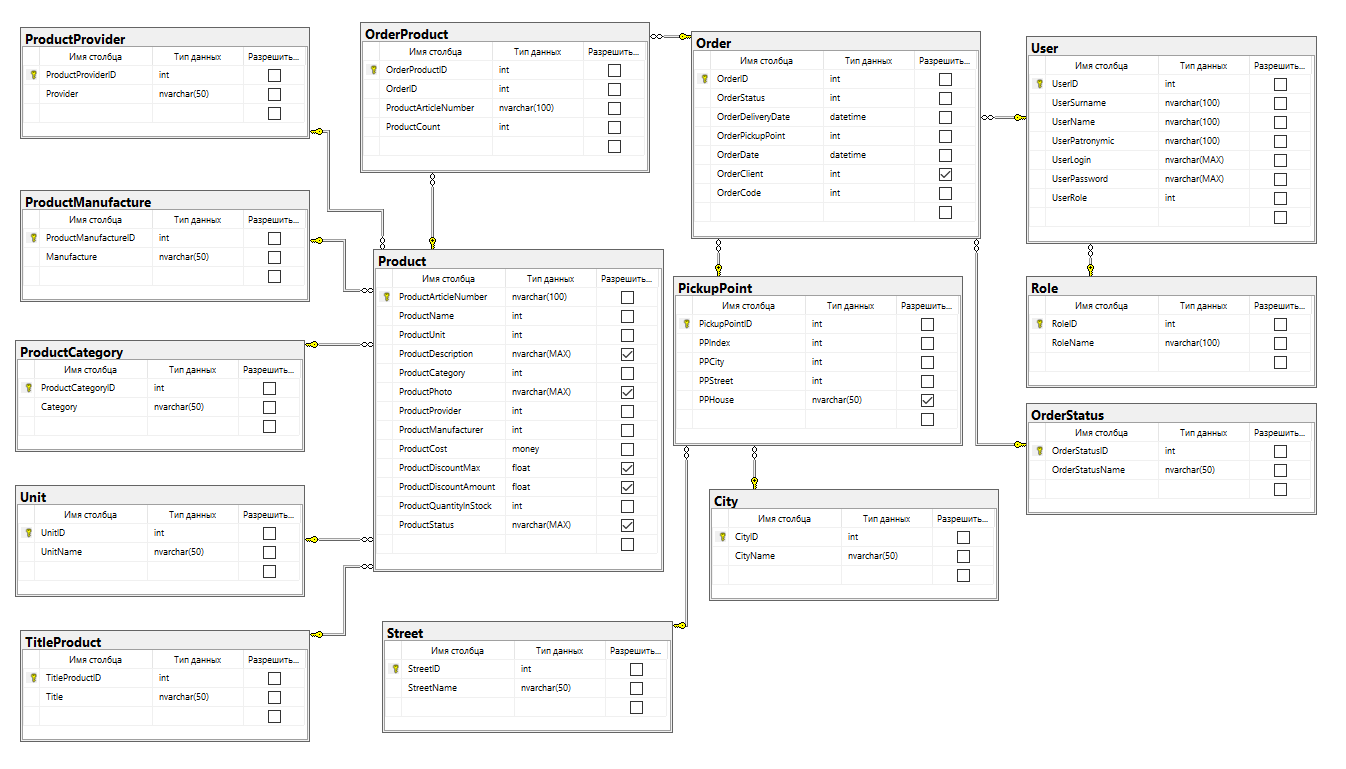


Рисунок 1 – База данных

Также необходимо импортировать предоставленные данные из файлов в базу (рис.2-3).

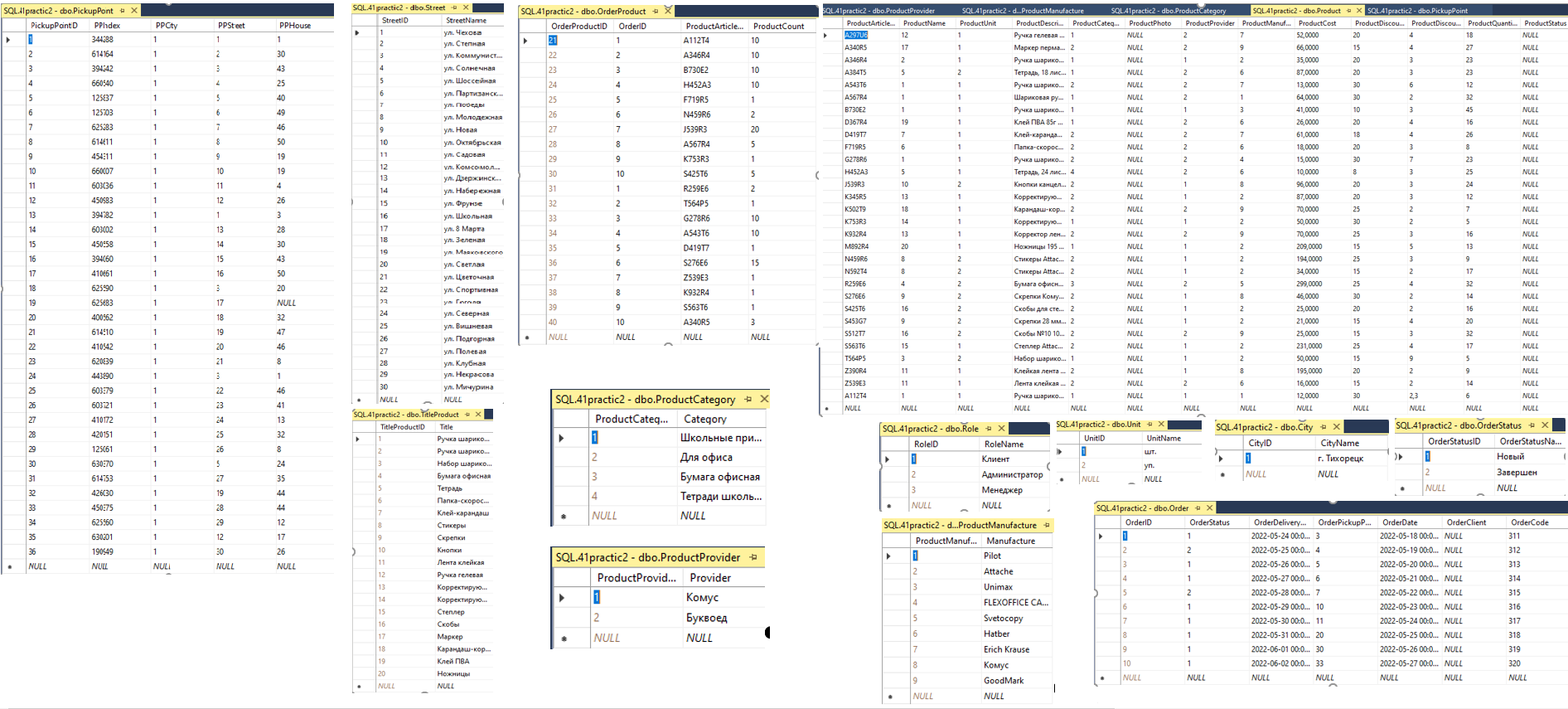
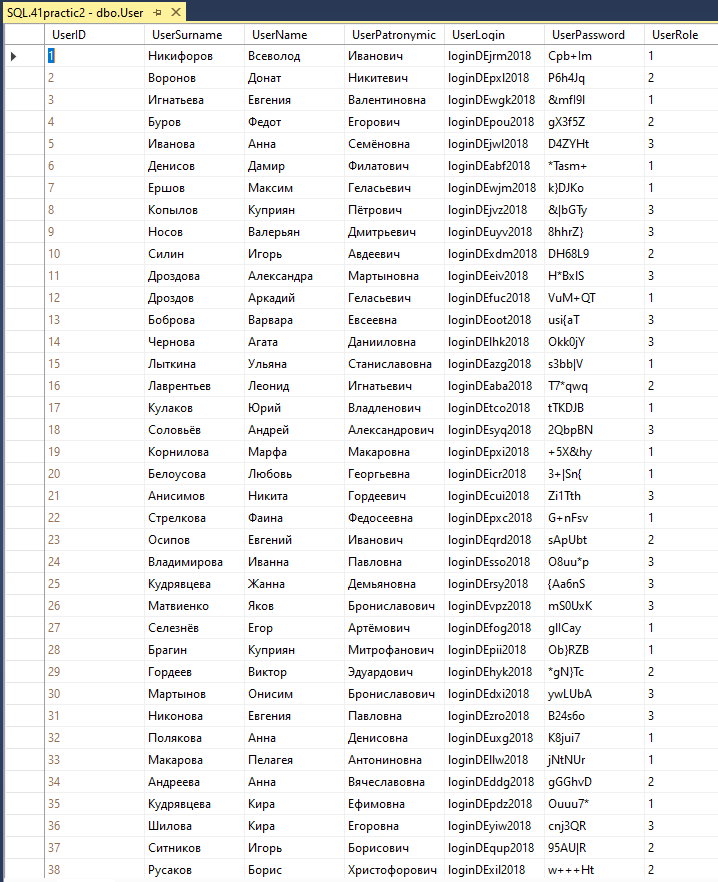


Рисунок 2 – Импорт данных



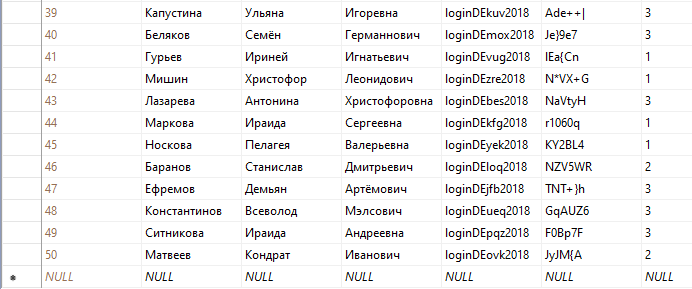


Рисунок 3 – Импорт данных

**Авторизация**

При запуске приложения окно входа – первое, что видит пользователь. На ней пользователю предлагается ввести свой логин и пароль или есть возможность перейти на экран просмотра товаров в роли гостя (рис.4).

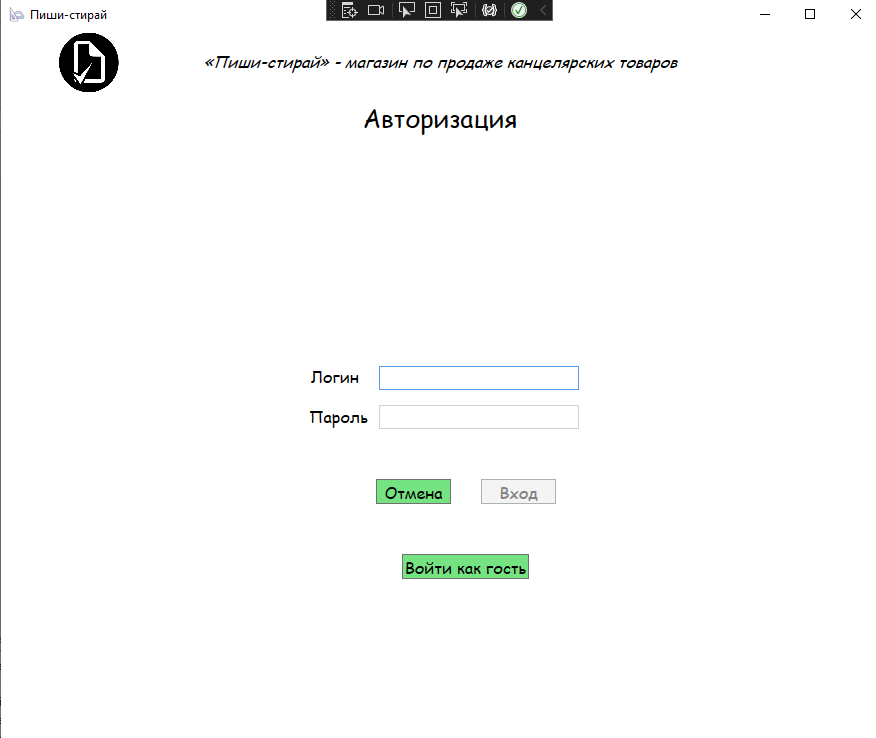


Рисунок 4 – Авторизация

После входа в любую учетную запись должна быть реализована возможность выхода на главный экран – окно входа. При переходе в любую учетную запись в интерфейсе (правый верхний угол) должны отображаться ФИО пользователя.



Рисунок 5 – Список товаров

После первой попытки неуспешной авторизации система выдает сообщение о неуспешной авторизации.

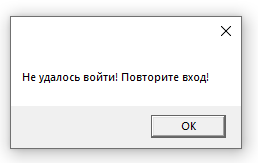


Рисунок 6 – Попытка неправильного входа

Затем помимо ввода логина и пароля просит ввести captcha, состоящую из 4 символов (цифры и буквы латинского алфавита) и графического шума.

CAPTCHA - должна содержать минимум 4 символа (буква или цифра), которые выведены не в одной линии. Символы должны быть либо перечеркнуты, либо наложены друг на друга.

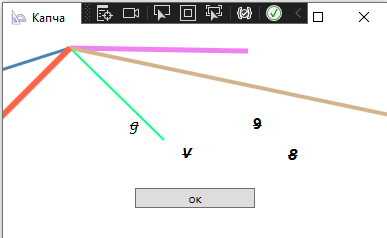


Рисунок 7 – Каптча

После попытки неудачной авторизации с вводом captcha, система блокирует возможность входа на 10 секунд.

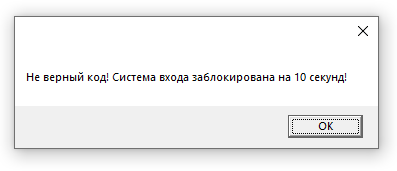


Рисунок 7 – Блокировка системы на 10 секунд

**Список товаров**

Очень важно постоянно получать актуальную информацию об остатках товаров на складах, поэтому необходимо реализовать вывод товаров, которые хранятся в базе данных, согласно предоставленному макету.

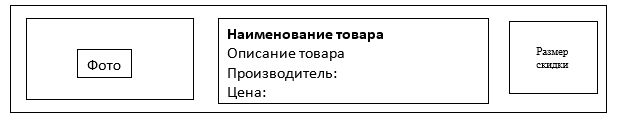


Рисунок 8 – Макет

При отсутствии изображения выводится картинка-заглушка из ресурсов.

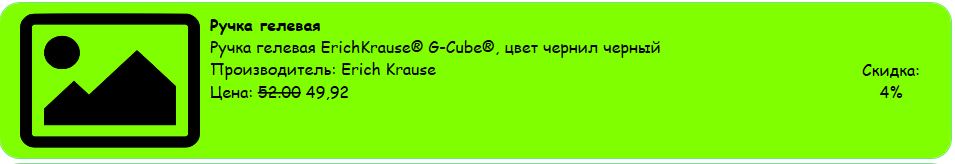


Рисунок 9 – Заглушка

Вывод осуществляется списком. Подсвечиваются товары у которых имеется скидка.



Рисунок 10 – Список товаров

Если у товара снижена стоимость, то основная цена должна быть перечеркнута, и рядом с ней указана итоговая стоимость.

Есть возможность отсортировать товары по стоимости, отфильтровать данные по скидке и с помощью поисковой строки сможет найти любой товар. Также присутствует возможность сбросить сортировку и фильтр, выбрав первый элемент выпадающего списка. Поиск, сортировка и фильтрация работают в реальном времени, без использования каких-либо кнопок. Если информации по данным нет, то для пользователя выводится информация об отсутствии результатов поиска.

В верхней части окна списка товаров представлено количество выведенных записей и общее количество записей в базе данных.

Также если ролью пользователя является администратор или менеджер то он может удалить товар.



Рисунок 11 – Удаление товара

Если товар участвует в заказе, то товар удалить нельзя.

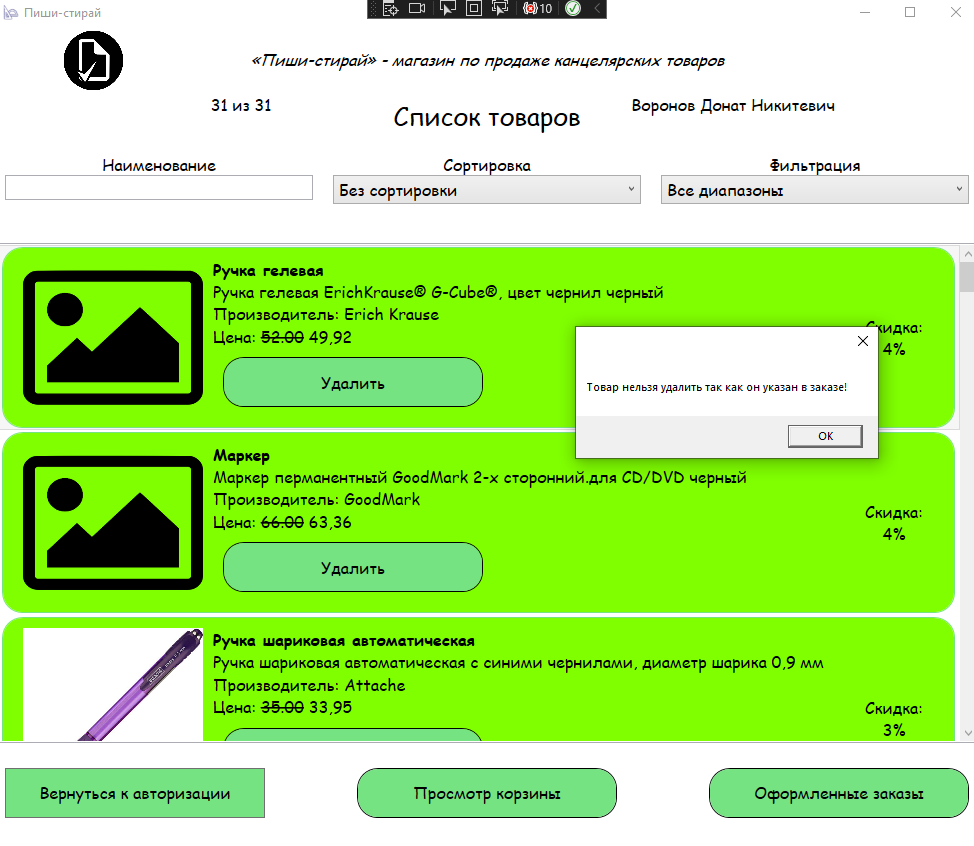


Рисунок 12 – Запрет на удаление

**Формирование заказа (терминал)**

В модуле формирования заказа реализована страница для авторизированных пользователей и гостя. При просмотре списка товаров пользователь может выбрать товар, по нажатию правой кнопки мыши вызвать контекстное меню и добавить данный товар в корзину по команде «Добавить корзину». При нажатии на команду происходит формирование заказа и в корзину добавляется выбранный товар в количестве 1 единицы.

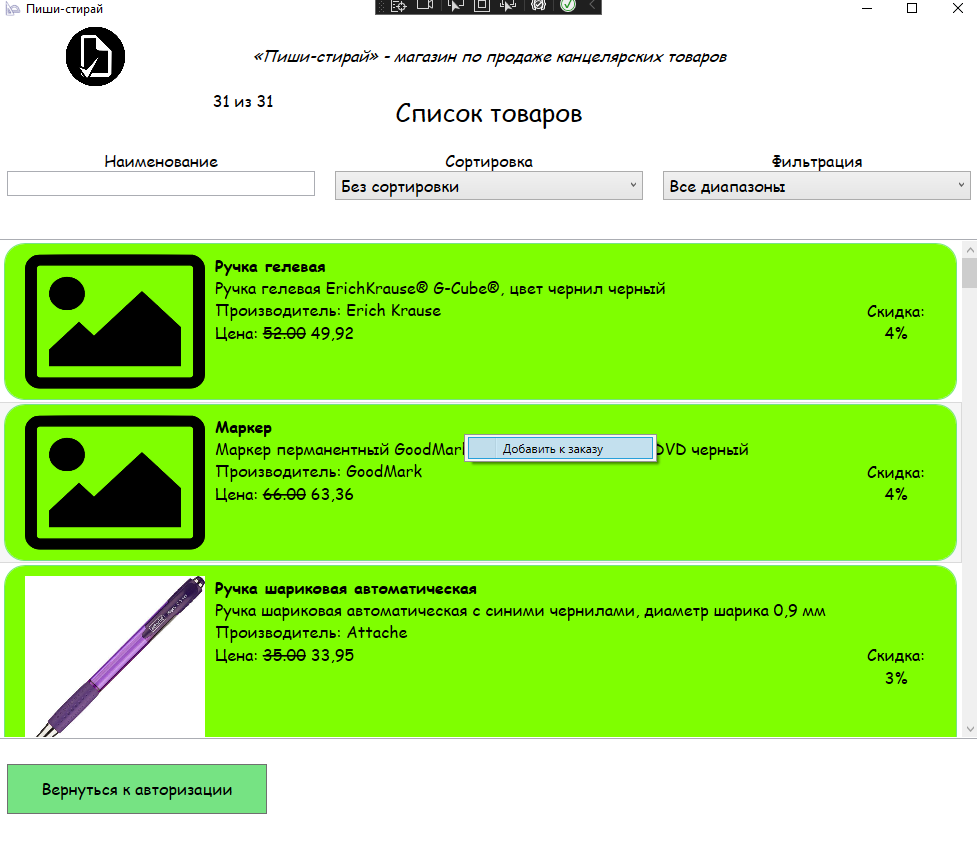


Рисунок 13 – Контекстное меню

Если хотя бы один товар присутствует в корзине, то для пользователя появляется кнопка для просмотра заказа. Если же ни один товар не добавлен к заказу, то кнопка просмотра корзины не доступна.

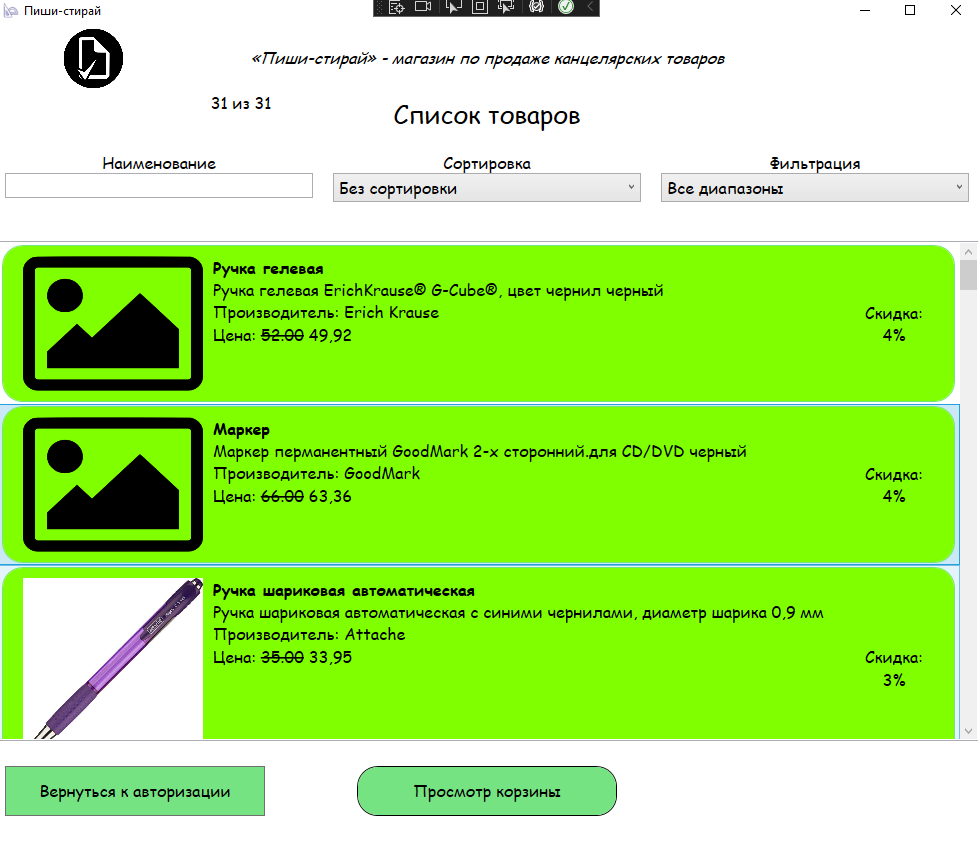


Рисунок 14 – Кнопка просмотра корзины

Просмотр корзины реализован в отдельном модальном окне. При просмотре корзины отображается вся необходимая информация о товаре, в том числе и изображение.

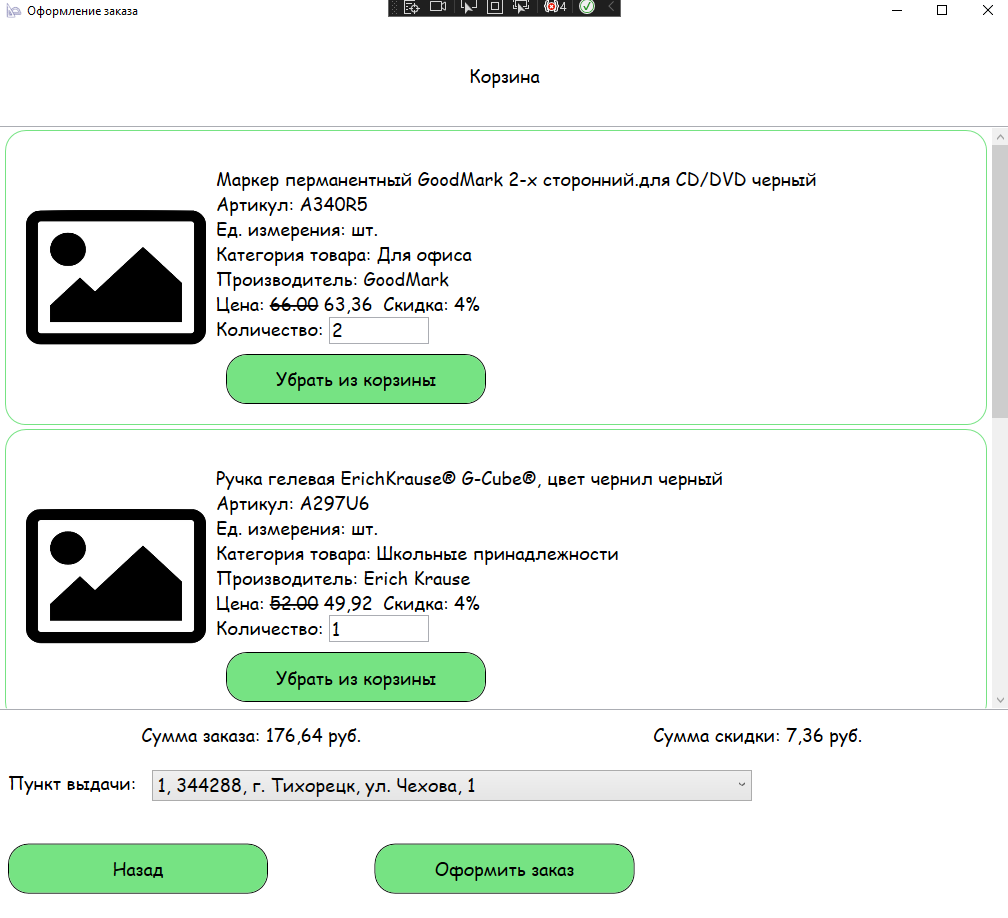


Рисунок 15 – Корзина

Присутствует возможность удаления товара из корзины, для этого необходимо в поле количества указать 0 или при нажатии на кнопку «Удалить».

Если заказ формирует авторизированный пользователь, то в окне просмотра корзины отображается ФИО клиента.

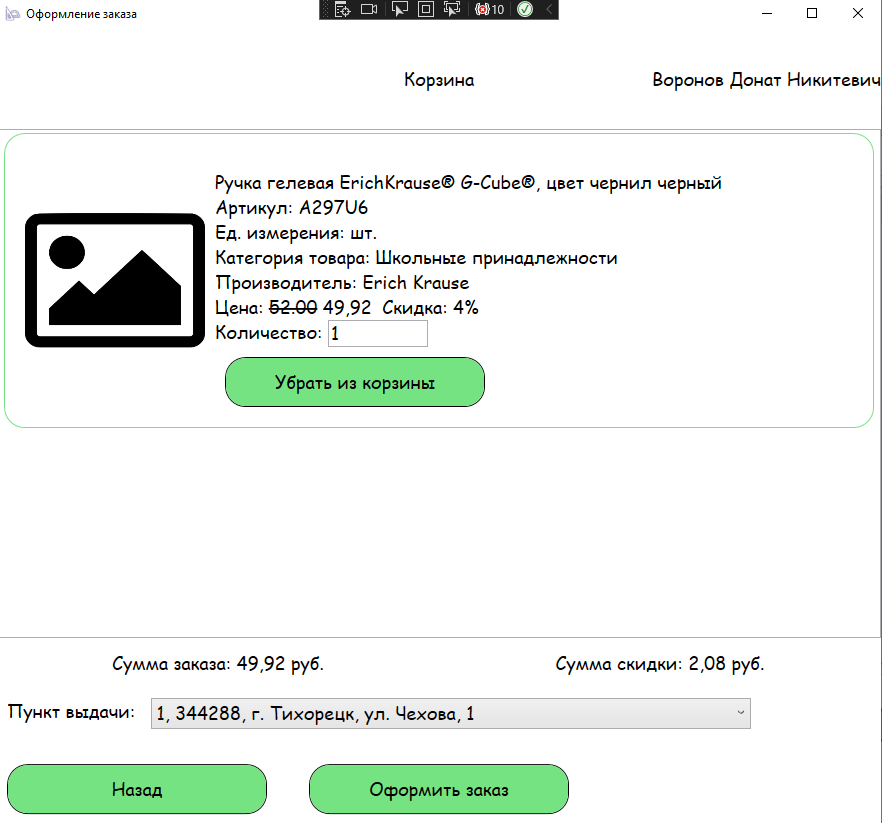


Рисунок 16 – Корзина

Также при изменении количества товара или удаления одного из товаров ведется пересчет суммы и скидки в реальном времени.

На кнопку «Оформить заказ» переходить формирование заказа со всей необходимой информацией.

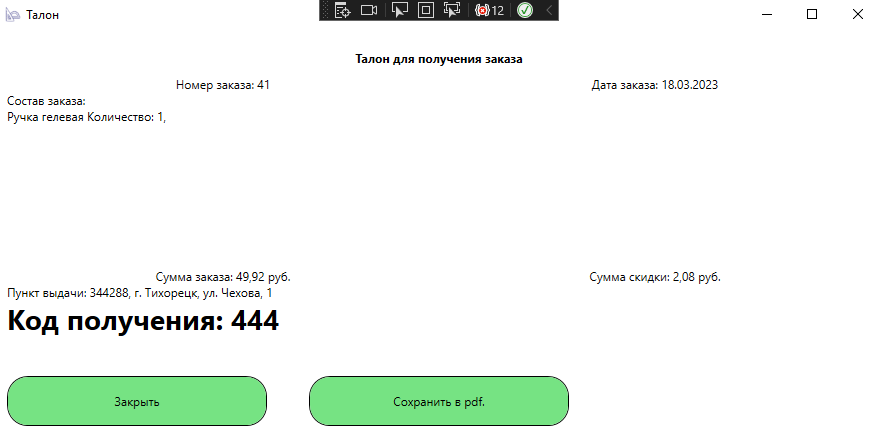


Рисунок 17 – Сформированный товар

Также можно сохранить талон в pdf.

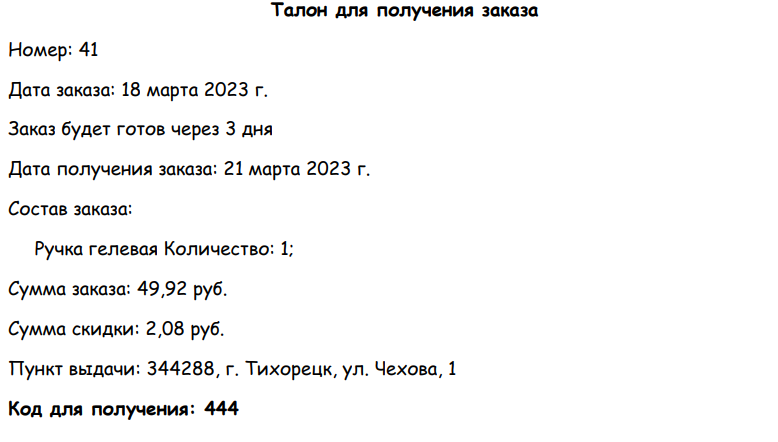


Рисунок 18 – Талон pdf

**Работа с заказами**

Был реализован интерфейс для работы с заказами, который доступен администратору или менеджеру. Для просмотра данных заказа представлена вся необходимая информация: номер заказа, состав заказа, дата заказа, дата доставки, общая сумма заказа и скидка всего заказа, а также ФИО клиента, если заказ был оформлен авторизированным пользователем.

Если товара на складе достаточно позиций, то строка с заказом высвечивалась с определенным цветом (#20b2aa), если хотя бы одного товара нет на складе, то строка была выделена цветом (#ff8c00).

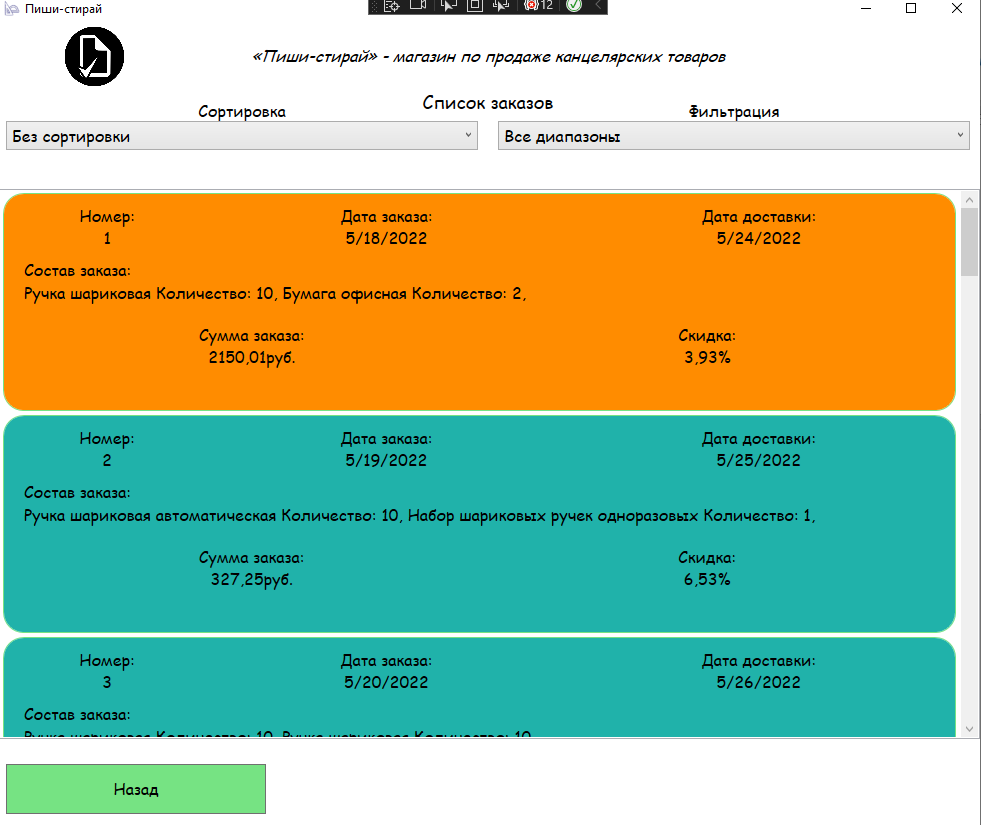


Рисунок 19 – Список заказов

У пользователя есть возможность отсортировать заказы по стоимости и отфильтровать данные по размеру суммарной скидке. Также присутствует возможность скинуть фильтр и сортировку, выбрав в выпадающем списке первый элемент. Сортировка и фильтрация работают в реальном времени, без использования каких-либо кнопок. Если информации по данным нет, то для пользователя выводится информация об отсутствии результатов поиска.

**Библиотека**

Также была разработана библиотека и метод с соответствующими названиями.

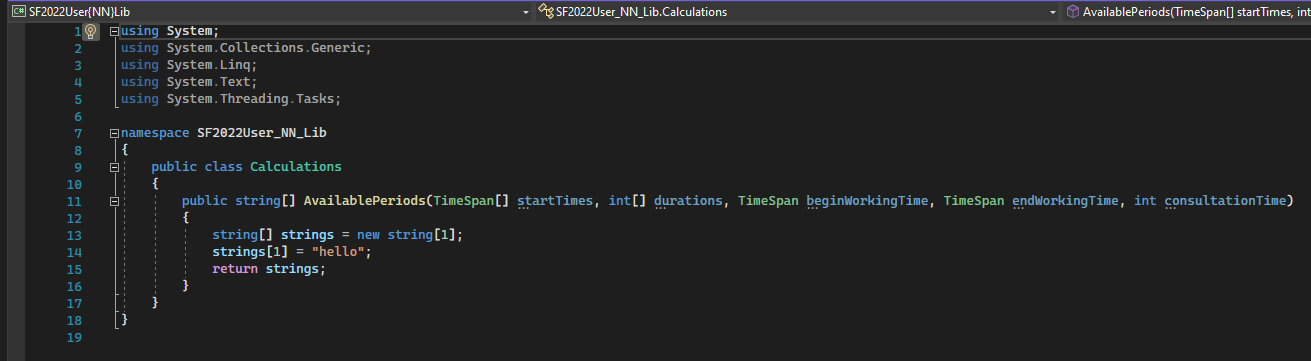


Рисунок 20 – Библиотека

**Тестирование**

Был создан проект для тестирования.

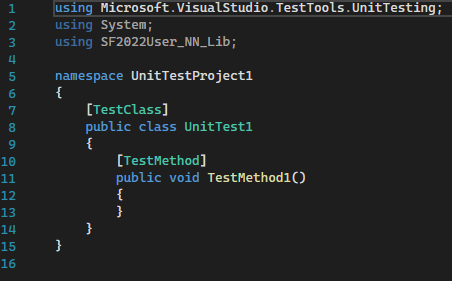


Рисунок 21 – Тестирование

# **Разработка программных модулей**

В рамках выполнения задания необходимо разработать основные модули информационной системы для ООО «Телеком Нева Связь» - провайдер последней мили:

* Модуль авторизации;
* Модуль работы с абонентами;
* Модуль «CRM» для работы с заявками.

Изначально необходимо спроектировать базу данных. На основе данного описания предметной области создадим ER-диаграмму будущей базы данных. Обязательно реализуем 3 нормальную форму с обеспечением ссылочной целостности.

Рисунок 1 – ER-диаграмма

Для диаграммы ER необходимо создать словарь данных – набор информации, описывающий, какой тип данных хранится в базе данных, их формат, структуру и способы использования данных.

Рисунок 2 – Data Dictionary

**Авторизация**

Разрабатываемая OSS/BSS система будет иметь модульную структуру, доступ к модулям будет зависеть от роли пользователя. Доступ возможен только для авторизованных пользователей. В связи с этим при запуске системы первым окном реализуем окном авторизации.

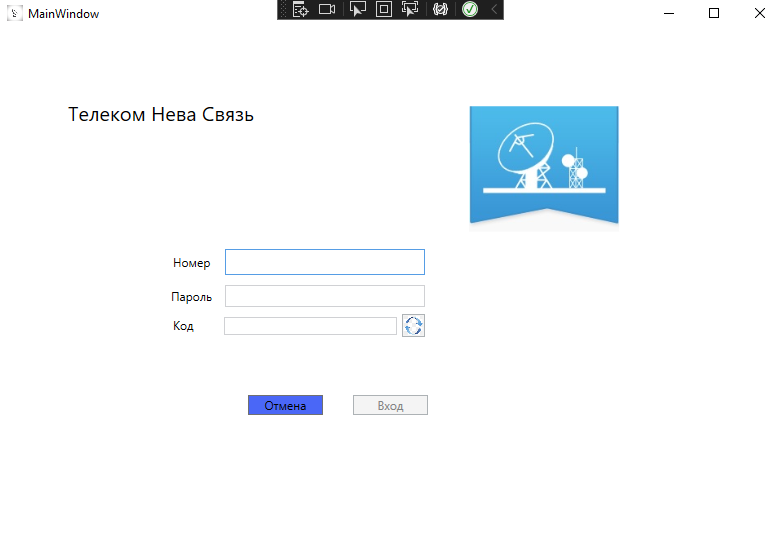
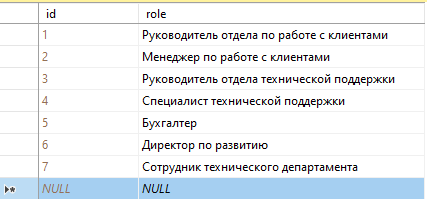
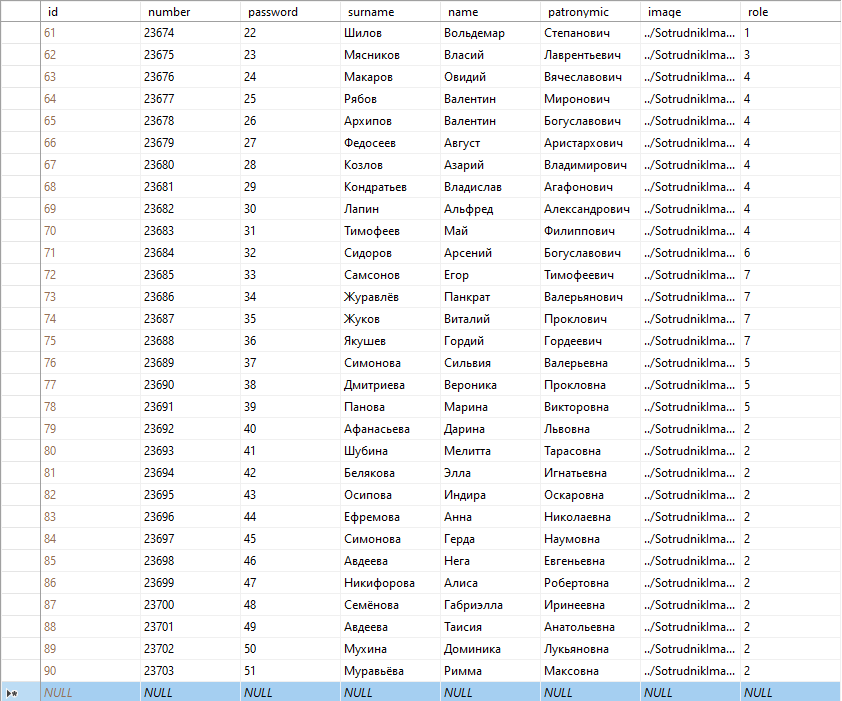


Рисунок 3 – Авторизация

Для реализации авторизации в базе данных создадим необходимые таблицы и заполним их тестовыми данными.





Доступ к системе имеют только сотрудники. У каждого сотрудника есть номер, по которому и разграничиваются права доступа.

Алгоритм авторизации:

● 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.

Реализуем данный функционал с условием эмуляции работы с СМС:

При открытии окна активны только поле для номера ввода сотрудника и кнопка “Отмена”.

При вводе номера сотрудника и нажатию Enter происходит проверка номера сотрудника. Если номер сотрудника есть в базе данных, то поле для ввода пароля становится активным и в нем установлен курсор.

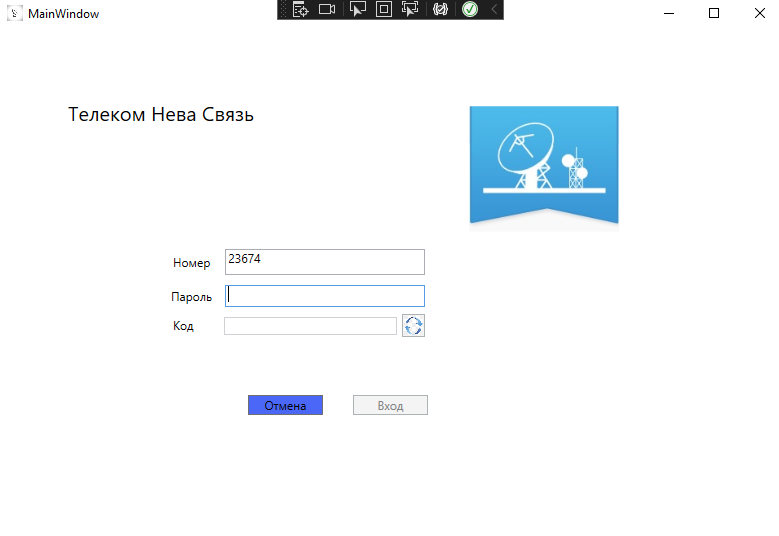


Рисунок 3 – Правильный ввод номера сотрудника

Если номер сотрудника в базе отсутствует, появляется сообщение об ошибке.

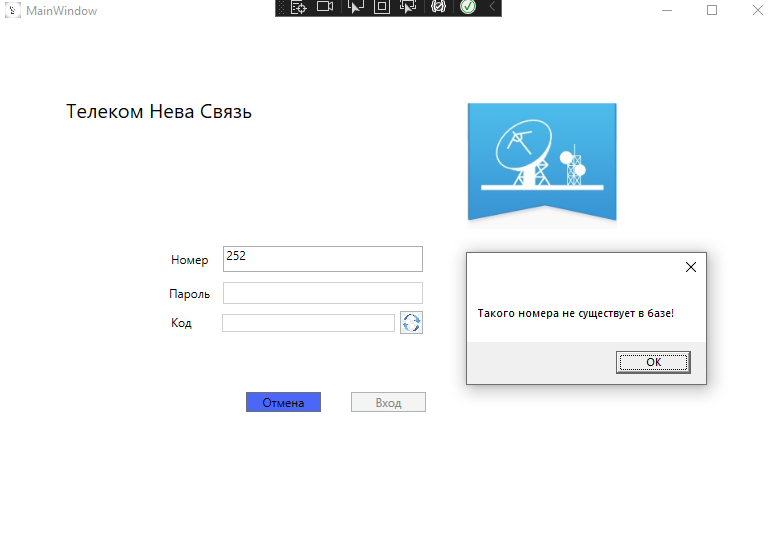


Рисунок 3 – Неправильный ввод номера сотрудника

После ввода пароля по нажатию на Enter открывается модальное окно со сгенерированным кодом доступа (8 символов, латиница, верхний и нижний регистр, спецсимвол, цифра).

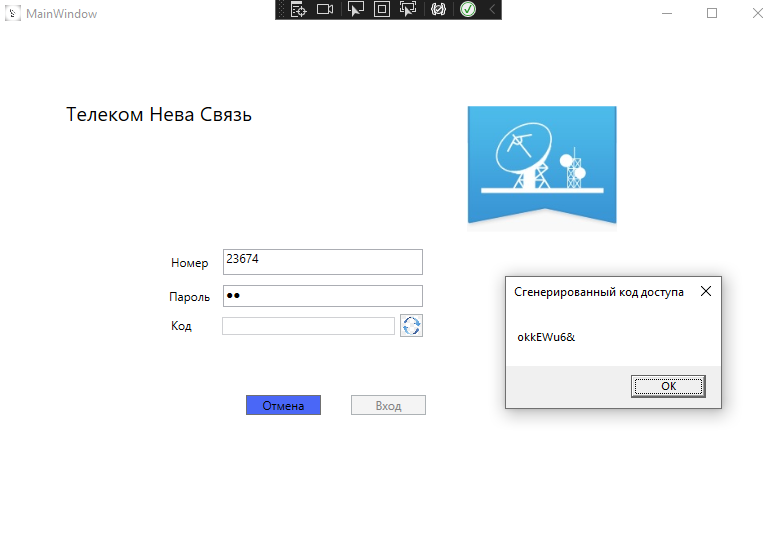


Рисунок 3 – Вывод сгенерированного кода доступа

В течение 10 секунд после закрытия окна с кодом пользователь должен ввести код и авторизоваться (по Enter и “Вход”). Если в течение 10 секунд код не введен, для повторной “отправки” кода необходимо нажать .

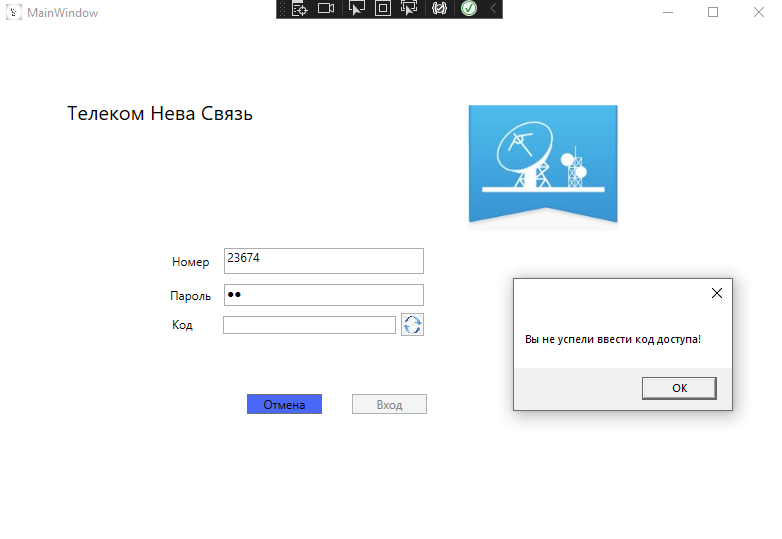


Рисунок 3 – Истечение таймера на ввод кода доступа

При вводе неправильного пароля код не генерируется, и система сообщает пользователю о неверном пароле.

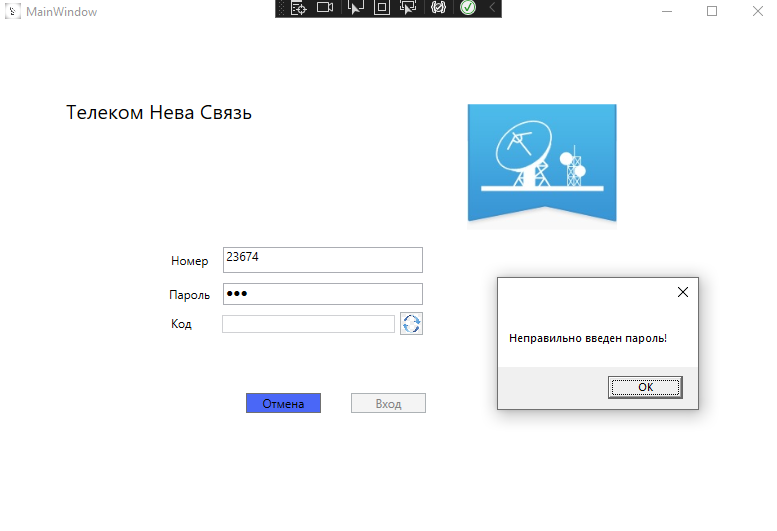


Рисунок 3 – Неправильный ввод пароля

У пользователя должна быть возможность очистить все поля ввода нажатием на кнопку «Отмена».

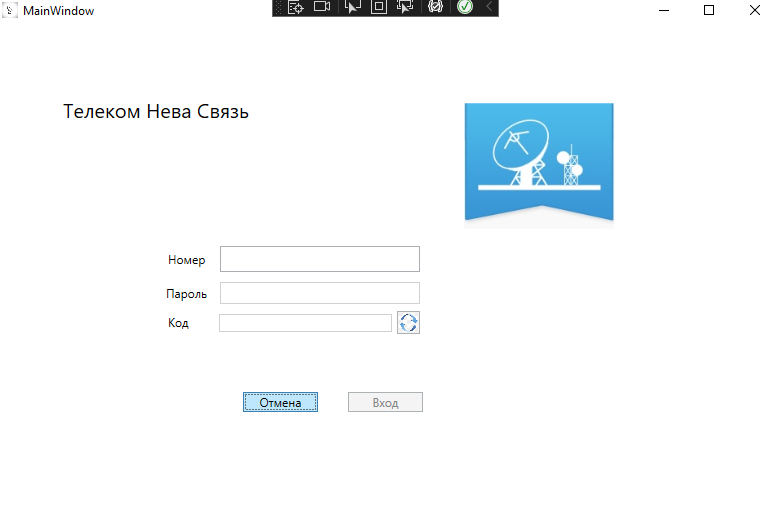


Рисунок 3 – Очистка полей по кнопке «Отмена»

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

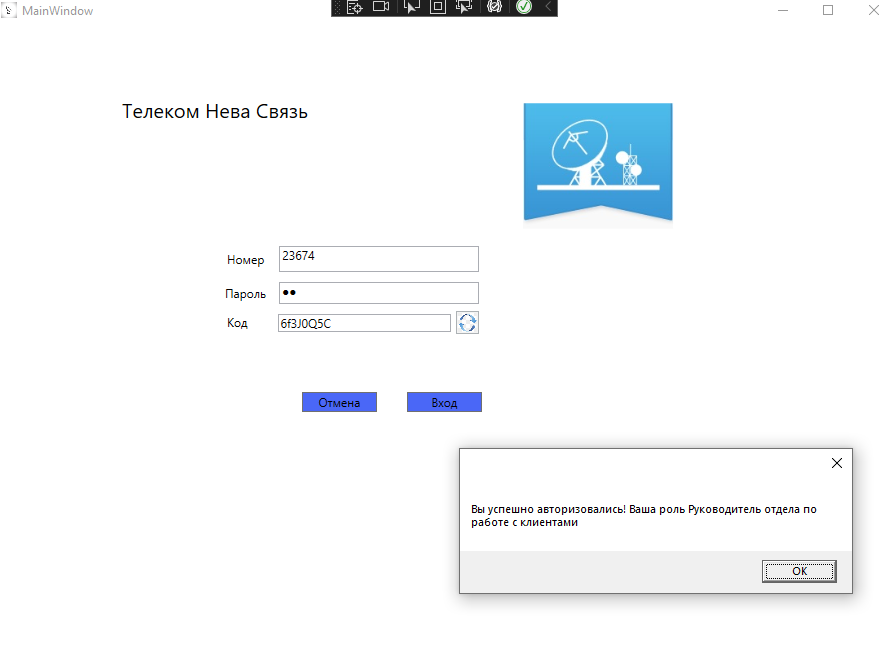


Рисунок 3 – Успешная авторизация, вывод роли

**Модуль работы абонентами**

Создадим новое приложение, которое не будет зависеть от процедуры авторизации в предыдущей сессии, в соответствии с макетом.

Вертикальное меню предполагает работу в двух форматах:

● в развернутом виде меню содержит текстовые и графические пункты управления. При первом входе меню должно быть в развернутом виде с активным пунктом меню Абоненты.

● в свернутом виде - реализуйте схлопывание вертикального меню, при котором все пункты меню будут представлены пиктограммами (предоставленными в ресурсах к заданию). Схлопывание меню можно выполнять по любому элементу интерфейса.

Для выбора авторизованного сотрудника сделаем выпадающий список со всеми пользователями из БД. При изменении выбранного пользователя должна меняться фотография справа и список доступных пунктов меню слева (на основании роли пользователя).

Общие требования к экранным формам:

● вертикальное меню с пунктами: логотип ТНС, Абоненты, Управление оборудованием, Активы, Биллинг, Поддержка пользователей, CRM;

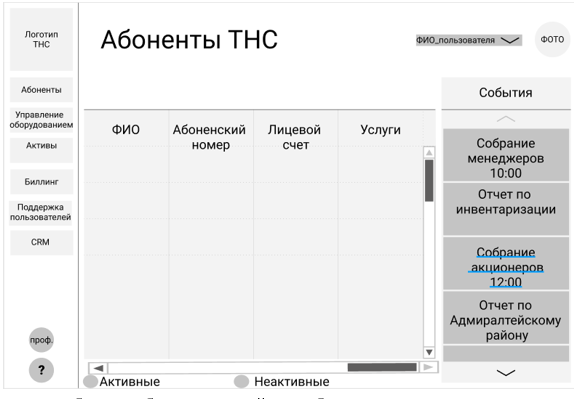
● отдельный блок с названием окна;

● фото пользователя;

● ФИО пользователя.

При первом входе меню должно быть в развернутом виде с активным пунктом меню Абоненты.

При выборе каждого пункта меню в основной части экрана должен быть отображен соответствующий заголовок.



При просмотре отображаются абоненты с активными договорами (выделен пункт “Активные”). Реализуйте возможность отображения абонентов с расторгнутыми договорами при выделении пункта “Неактивные”. Предусмотрите просмотр одновременно и активных, и неактивных договоров.

В правой части окна просмотра абонентов реализуйте блок для отображения активных событий: название события, время (если указано). События отображаются только в соответствии с ролью пользователя и только на актуальную дату.

Реализуйте поиск абонентов по фамилии, району, улице, лицевому счету. При поиске абонентов по улице реализуйте выбор улицы с номером дома с помощью выпадающего списка.

**Модуль «CRM»**

Модуль «CRM» предназначен для работы с заявками. Реализуем интерфейс, позволяющий сформировать заявку.

# **Разработка мобильных приложений**

Задание: на языке Java в Android Studio разработайте приложение в соответствии с заданием

Задачей является разработать приложение для смартфона. Дизайн приложения доступен по адресу: <https://www.figma.com/file/gAonPFbfoILOIIyNHgwY41/MAD-Meditation-App?nodeid=0%3A1&t=AwojHXi9OjzEHCNS-0>.

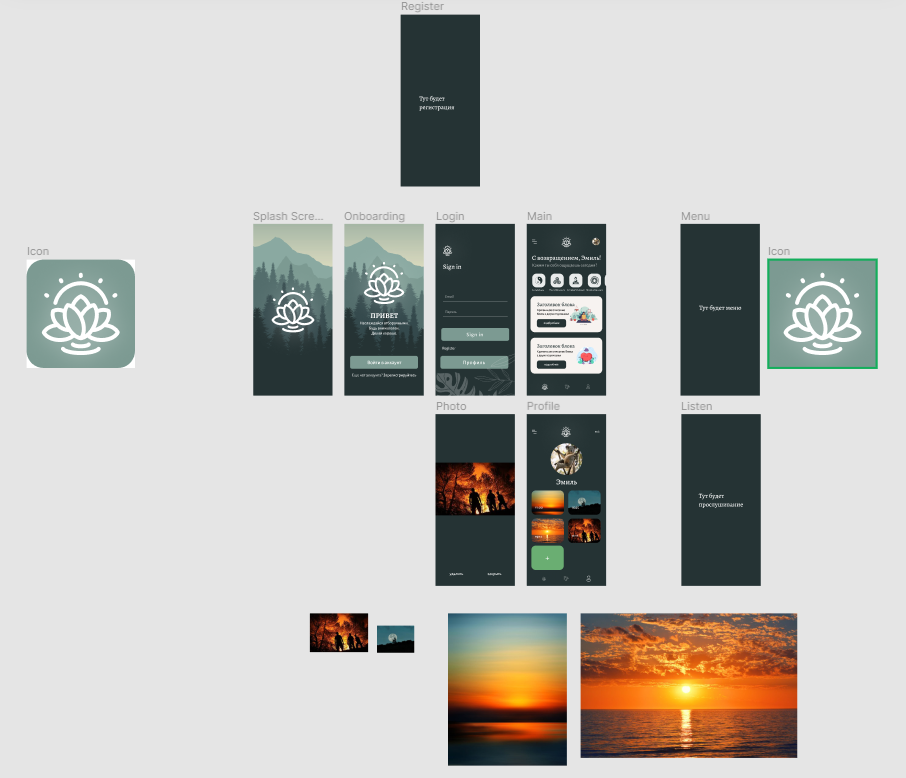


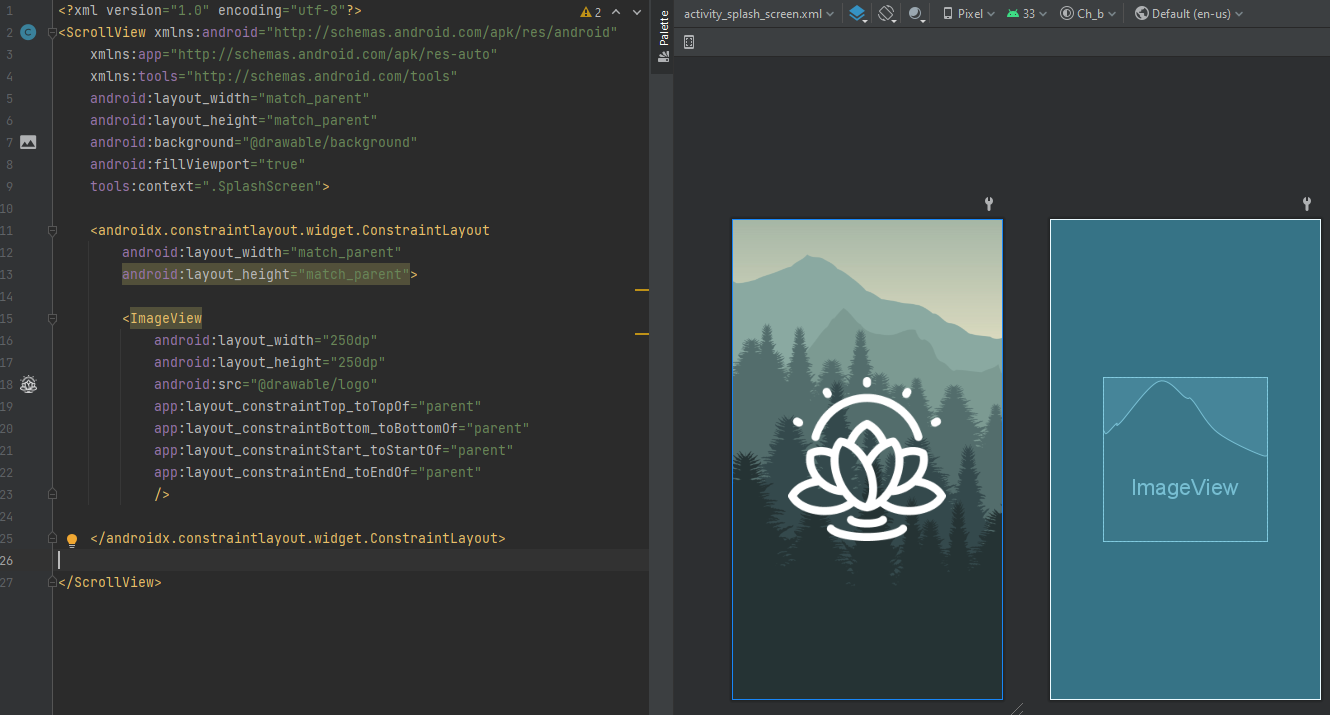
Рисунок 1 – Макет приложения

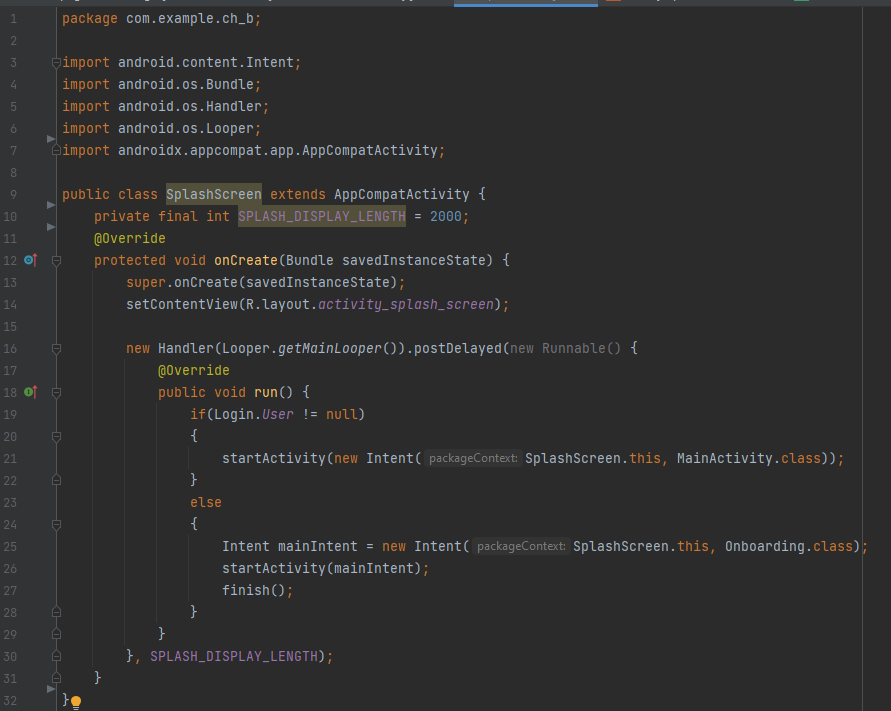
Необходимо следовать дизайну, а также логике переходов, которая реализована в прототипе. То есть все экраны должны соответствовать макету (за исключением экранов-заглушек). Текст должен быть вставлен в виде текстовых блоков, поля для ввода - полями для ввода.

В прототипе присутствуют экраны-заглушки. Переход на них тоже нужно реализовать. Добавьте иконку к приложению и используйте предоставленный шрифт. Начальный URL к API: mskko2021.mad.hakta.pro/api.

**Splash Screen**

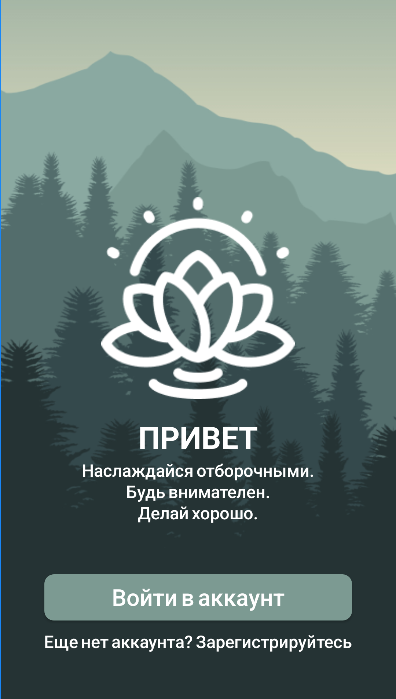
Экран загрузки должен отображаться при каждом запуске приложения.

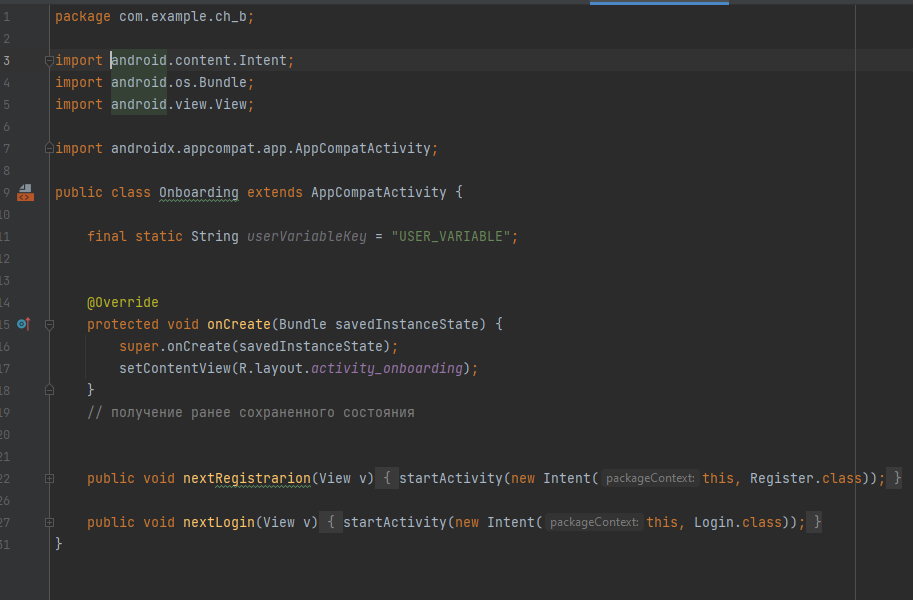




**Onboarding**

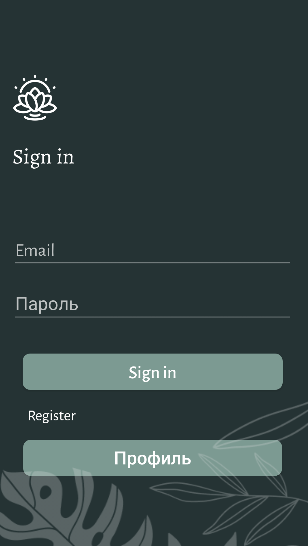
У пользователя должна быть возможность перейти на окно входа и экран заглушку - регистрацию.





**Login**

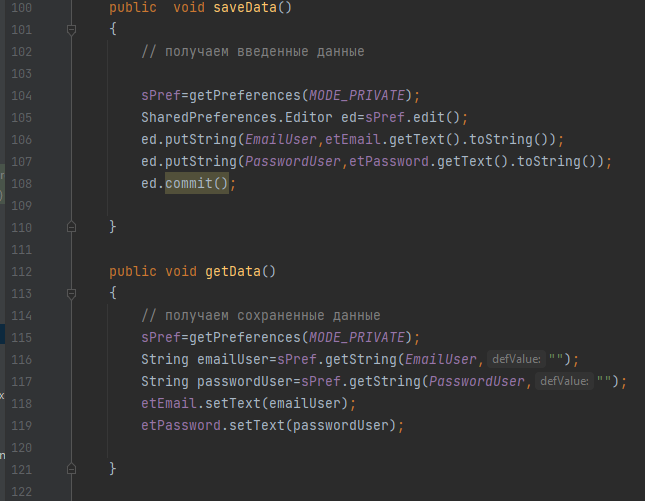
На данном экране пользователь должен иметь возможность ввести email и пароль к своей учетной записи и получить токен авторизации.

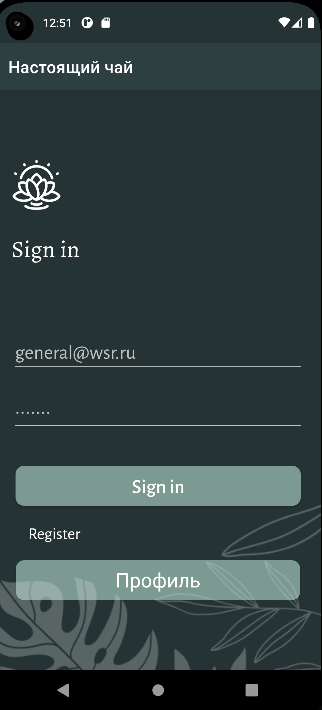


Перед тем, как отправить данные на сервер для получения токена, необходимо проверить заполненность полей, а также наличие символа @ в поле для Email.



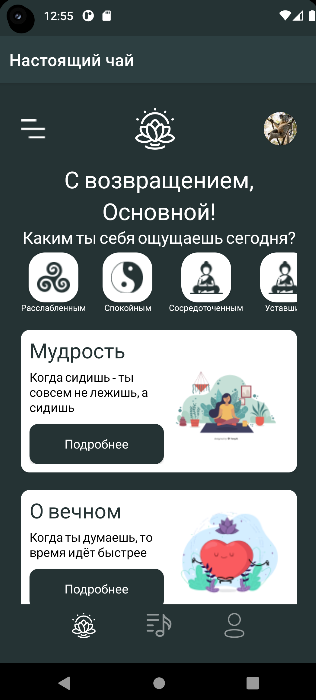
Данные авторизации должны сохраняться на устройстве и после перезагрузки приложения, то есть пользователь при повторном запуске будет пропускать экраны onboarding и login



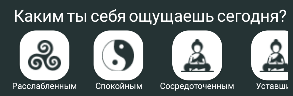


**Main**

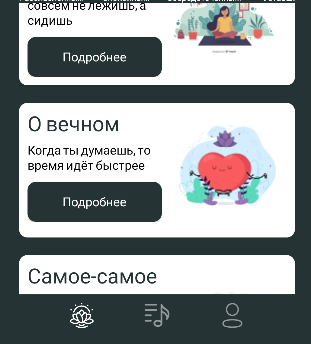
На главном экране необходимо выводить полученную информацию о текущем пользователе (изображение профиля и имя).



На экране должен выводиться список ощущений, получаемый с севера. Список должен листаться по горизонтали. Список ощущений должен быть отсортирован по параметру “position” по возрастанию.

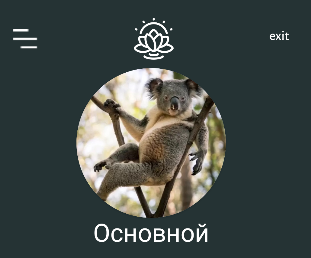


На экране должны выводиться блоки с короткими цитатами, получаемые с сервера. В случае, если список цитат не помещается на экран, то необходимо позволить пользователю листать экран полностью (за исключением нижнего таббара).



**Profile**

На экране профиля добавьте кнопку “выход”, после которой приложение открывает окно авторизации. На экране должны быть отображены: имя и фотография пользователя.



В нижней части экрана должна быть показана галерея фотографий, которые были загружены из стандартной галереи приложения. На каждой фотографии должно отображаться время её загрузки. Самым последним элементом всегда должна быть кнопка добавления фотографии, при нажатии на которую открывается выбор фотографий из стандартной галереи. Фотографии должны сохраняться в памяти приложения, чтобы они остались в приложении в случае удалениях их из стандартной галереи. Фотографии должны сохранятся и после перезагрузки приложения. После нажатия на фотографию, должен открываться экран Photo.

**Photo**

Фотография должна вписываться в экран по большей стороне. При нажатии на кнопку “удалить”, фотография должна быть удалена из приложения. При нажатии на кнопку “закрыть”, пользователь должен вернуться на экран Profile. Реализуйте управление экраном с помощью жестов. Если пользователь проводит по экрану слева направо, то он закрывает окно просмотра, а если справа налево, то удаляет фотографию. Добавьте возможность приближать фотографию с помощью двойного нажатия на неё. После двойного нажатия фотография должна увеличиться в 2 раза. В случае повторного двойного нажатия, фотография должна вернуться в исходное состояние

# **ЗАКЛЮЧЕНИЕ**

В процессе прохождения учебной практики по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем было разработано настольное приложение.

# **ПРИЛОЖЕНИЯ**

**ПРИЛОЖЕНИЕ А**