**Содержание**

1 Анализ предметной области. ERD-диаграмма 1

2 Анализ предметной области. Диаграмма прецедентов 1

3 Анализ предметной области. Диаграмма деятельности 1

4 Анализ предметной области. Физическая модель базы данных 1

5 Анализ предметной области. Словарь данных 1

6 Импорт данных 1

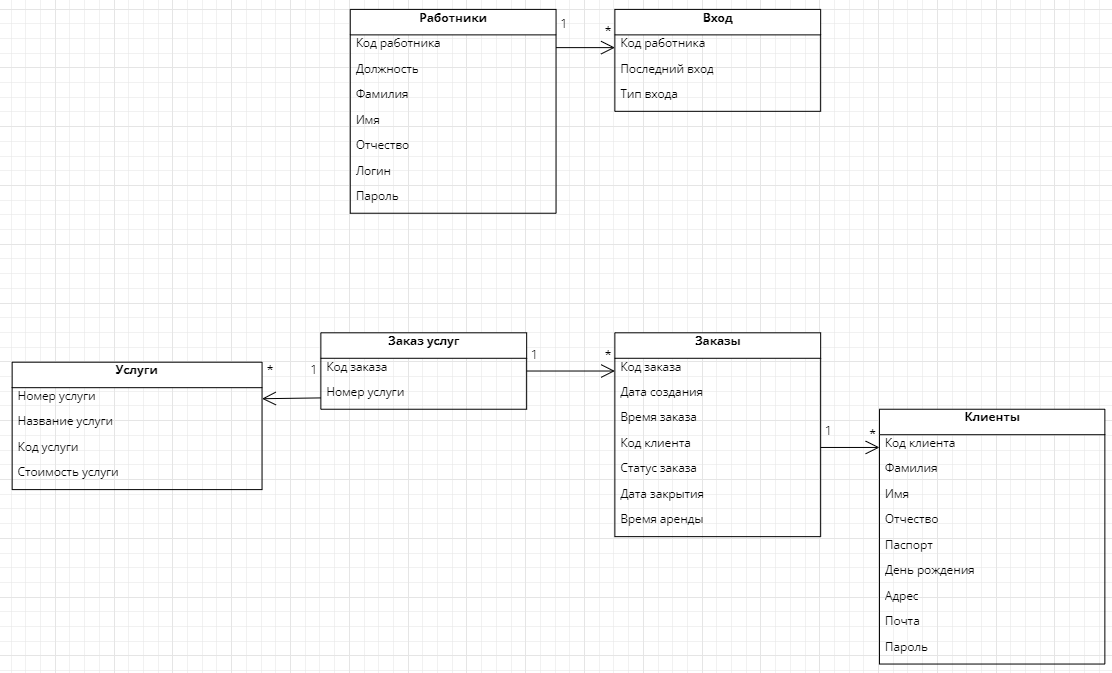
7 Пользовательский интерфейс приложения 1

8 Функционал программного решения 1

Нумерацию необходимо исправить – необходимо сделать тогда, когда отчет будет готов.

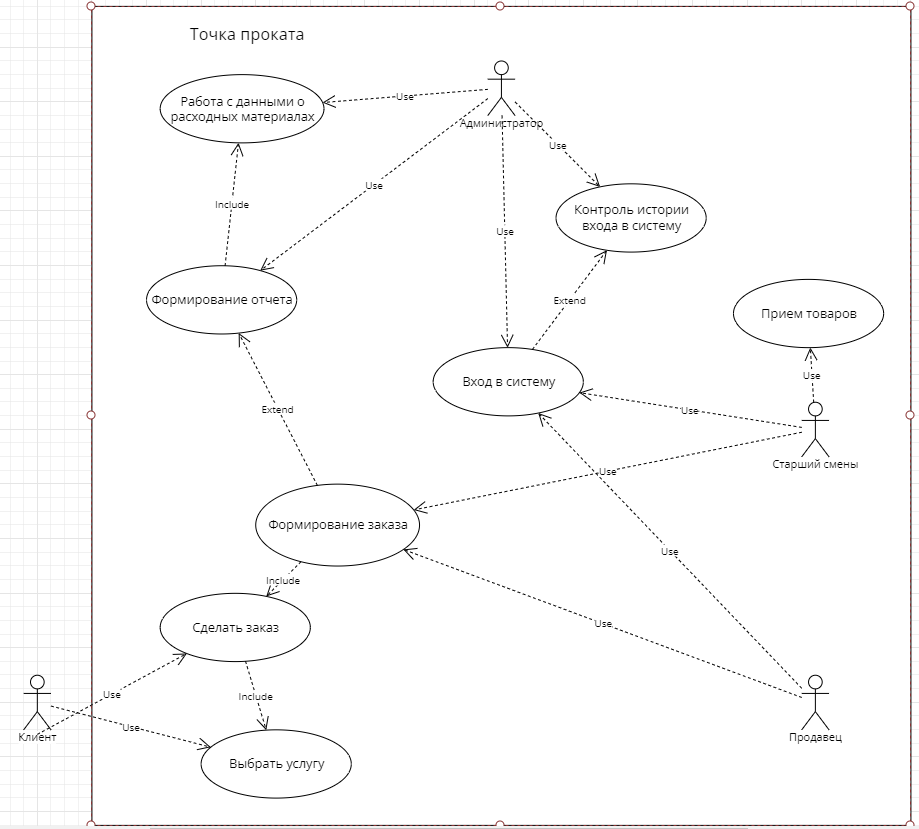
**1 Анализ предметной области. ERD-диаграмма**

ER-диаграмма — модель данных, позволяющая описывать концептуальные схемы предметной области. В ER-диаграмме показанной на рисунке 1 показана логическая модель данных позволяющая понять схему связей между таблицами.

Рисунок 1 — ER-диаграмма

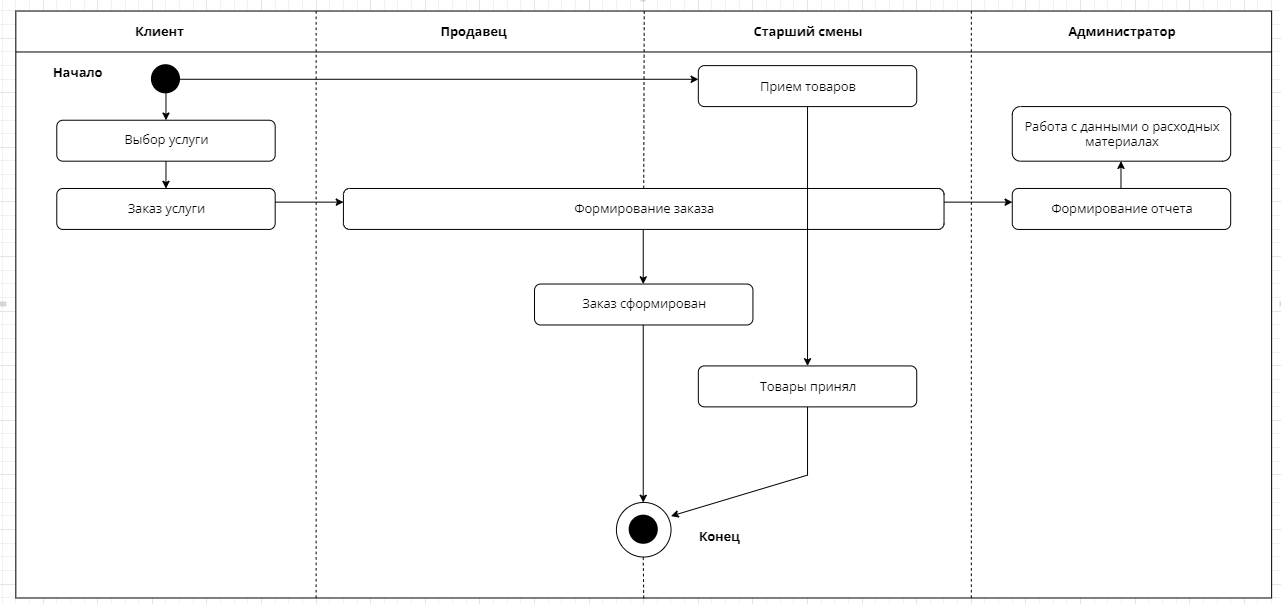
**2 Анализ предметной области. Диаграмма прецедентов**

Диаграмма прецедентов (рисунок 2) или диаграмма вариантов использования — диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Основное назначение диаграммы прецедентов — описание функциональности и поведения, позволяющее заказчику, конечному пользователю или разработчику совместно обсуждать проектируемую или существующую систему.

Рисунок 2 — Диаграмма прецедентов

**3 Анализ предметной области. Диаграмма деятельности**

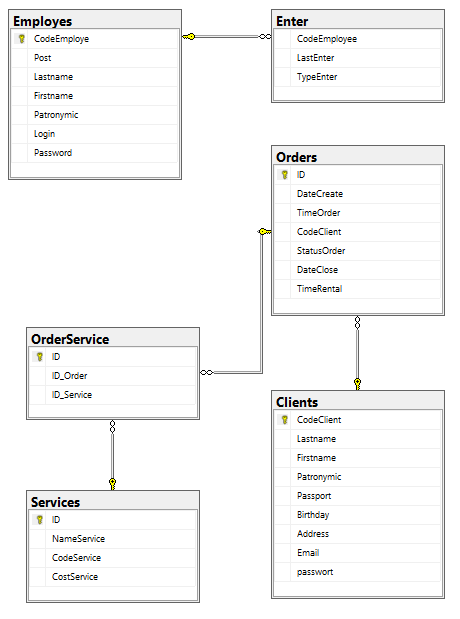
Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграммах состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчиненных элементов — вложенных видов деятельности и отдельных действий, соединенных между собой потоками, которые идут от выходов одного узла ко входам другого.

Рисунок 3 — Диаграмма деятельности

**4 Анализ предметной области. Физическая модель базы данных**

Физи́ческая моде́ль — физическое представление системы, объекта или процесса с целью их исследования, то есть это представление с помощью другого физического, реального объекта, имеющего в том или ином аспекте схожую динамику поведения.

Физическая модель базы данных — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных.

Рисунок 4 — Физическая модель базы данных

В нашей базе данных используются такие сущности как Employes (Сотрудники), Enter (Вход), Clients (Клиенты), Orders (Заказы), OrderService (Заказ услуг) и Services (Услуги).

Связь «Employes – Enter» нужна для взаимодействия с сущностью Enter и нормализации данных.

Связь «Orders – Clients» нужна для учета взаимодействий клиентов с компанией.

Связи «Orders – OrderService» и «Services – OrderService» требуются для нормализации таблиц, разбиении связи «многие-ко-многим» и правильной работоспособности программы.

**5 Анализ предметной области. Словарь данных**

Словарь данных — центральное хранилище информации о данных, такой как значение, взаимосвязи с другими данными, их источник, применение и формат.

Таблица 1 — Словарь данных

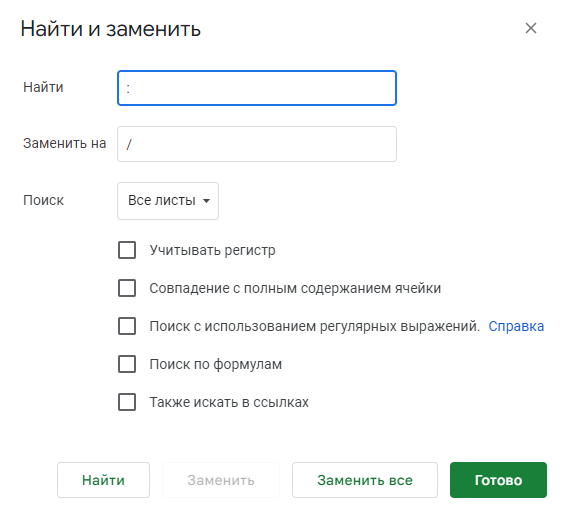
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Client** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Code** | VARCHAR (6) | Y |  |
|  | **LastName** | VARCHAR (100) | Y |  |
|  | **FirstName** | VARCHAR (50) | Y |  |
|  | **Patronymic** | VARCHAR (50) | Y |  |
|  | **Gender** | CHAR(1) | Y | Check Constraint (Допускается только 'F' или 'M') |
|  | **Phone** | VARCHAR (20) | Y |  |
|  | **Email** | VARCHAR (50) | Y |  |
| **Country** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
|  | **Code** | VARCHAR (3) | Y | Трехзначный код из классификатора стран мира |
|  | **Name** | VARCHAR (100) | Y |  |
|  | **Flag** | BINARY |  |  |
| **Service** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | INT | Y | Auto increment |
| FK | **Name** | VARCHAR (200) | Y |  |
| FK | **PartnerID** | INT | Y | Foreign Key to Partner |
| FK | **HotelID** | INT | Y | Foreign Key to Hotel |
|  | **Price** | DECIMAL(10,2) | Y | Стоимость услуги от поставщика. Позволяет расчитывать стоимость тура для клиентов, чтобы не упасть ниже себестоимости. |
| 1 | **2** | 3 | 4 | 5 |
| 1 | **2** | 3 | 4 | 5 |
| **ServiceByTour** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | INT | Y | Auto increment |
| PK | **ServiceID** | INT | Y | Foreign Key to Услуга |
|  | **IsControl** | BOOLEAN |  | Признак. 1 - услуга в туре должна быть обязательной (например, страховка); 0 - услуга не требуется (например, виза) |

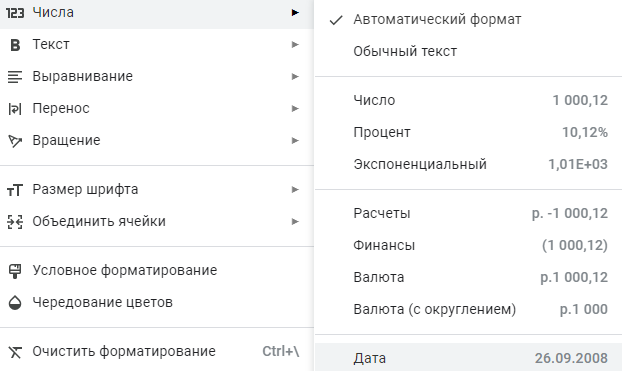
В данном словаре обозначено использование значений для атрибутов таблицы, таблицы используемые программой и описание каждого атрибута.

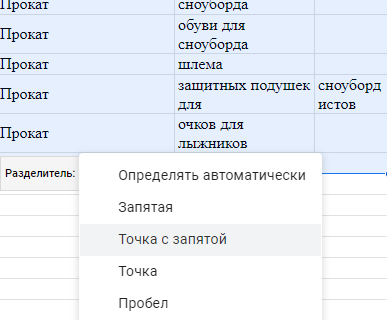
**6 Импорт данных**

Для успешной работы с импортом данных я работал с Online Excel, из предварительных документов было замечено что некоторые данные не нужны для дальнейшей работы, либо совершенно бесполезны.

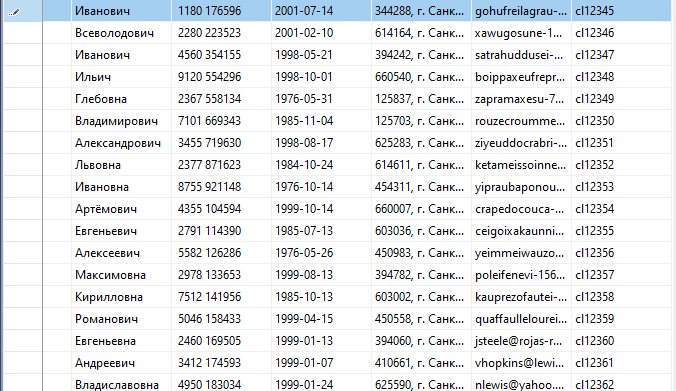
Все действия по нормализации были выполнены при помощи замены (рисунок 5) комбинация клавиш (Ctrl+H), изменения формата на дату (рисунок 6) для успешной вставки, а также разбиение таких столбцов как «ФИО» на отдельные ячейки при помощи разделителя текста на столбцы (рисунок 7).

Рисунок 5 — Замена

Рисунок 6 — Изменение формата

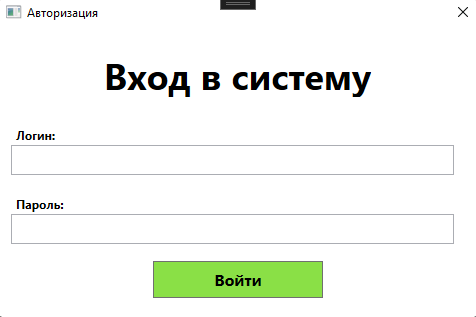
Рисунок 7 - Разделитель

Непосредственно в базу данных строки из excel импортировались в ручную (рисунок 8), база данных была написана на SQL после распределения данных в excel и заполнена готовыми данными.

Рисунок 8 — Заполнение базы данных

**7 Пользовательский интерфейс приложения**

В данном задании мной было реализовано приложение по предметной области, из предметной области было выявлено 2 формы создания.

Рисунок 9 — Форма авторизации

Данная форма сделана в минималистичном стиле, выполнены все требования такие как:

1. Проверка на введенные логин и пароль (рисунки 10-11).

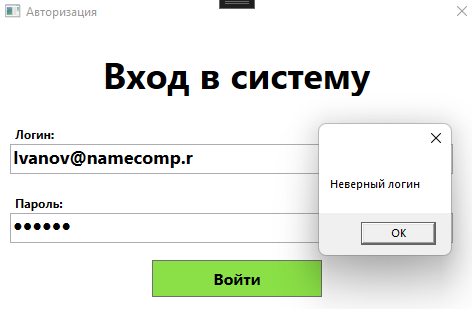
2. Маска на ввод пароля (рисунок 11).

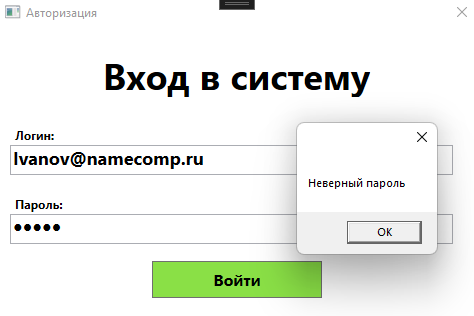
3. Разграничение прав доступа (рисунки 12-15).

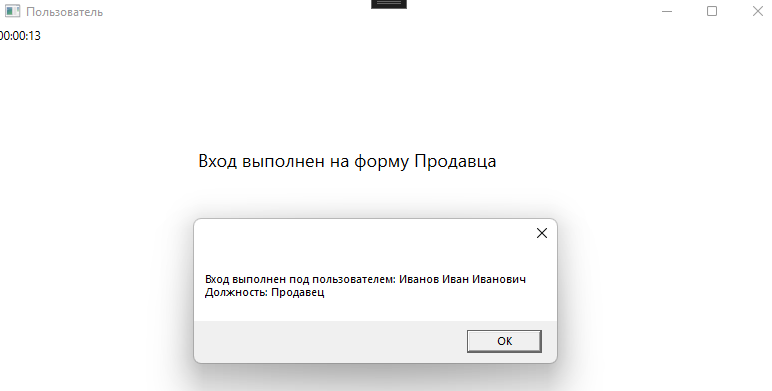
4. Рабочий таймер (рисунок 16).

5. Капча (рисунок 17).

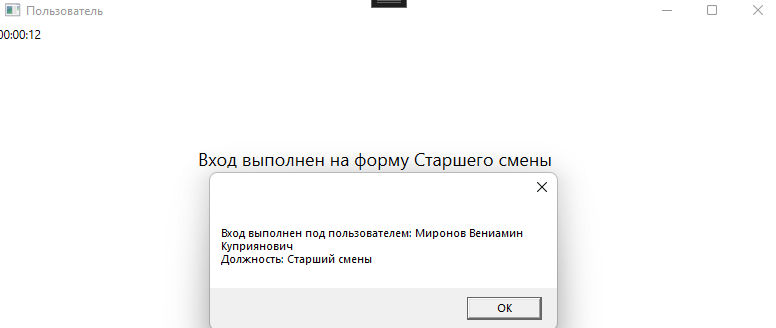
6. Блокировка, в случае неуспешного входа (рисунок 18).

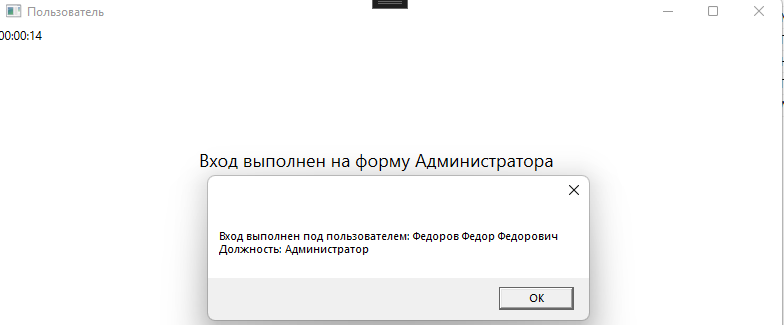
Рисунок 10 — Неверный логин

Рисунок 11 — Неверный пароль и маска

Рисунок 12 — Вход на форму продавца

Формы пользователей не содержат каких либо функций и показывают только разграничение прав пользователей, в дальнейшем есть возможность сделать формы для каждого типа пользователей.

Рисунок 13 — Вход на форму старшего смены

Рисунок 14 — Вход на форму администратора

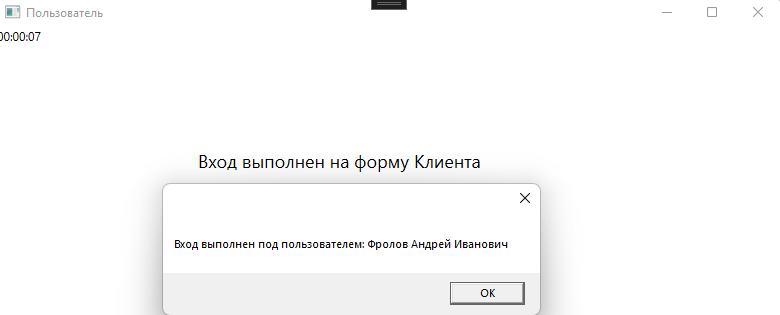
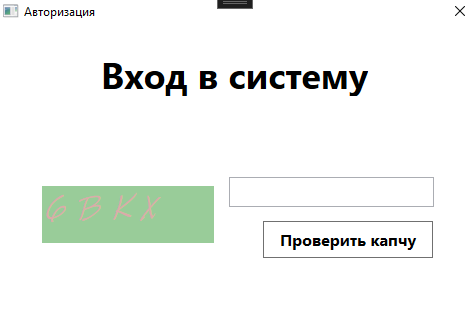
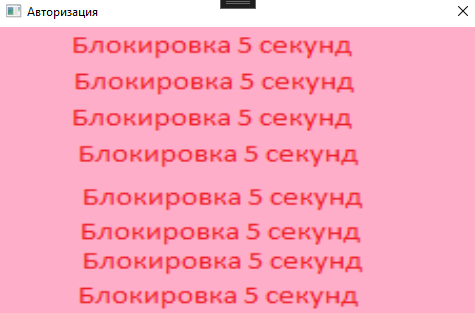
Рисунок 15 — Вход на форму клиента

Рисунок 16 — Таймер

Таймер на форме пользователей запускается при открытии формы и начинает уменьшенный отсчет от 15 секунд, по истечении этого времени пользователь автоматически выходит из системы.

Рисунок 17 - Капча

Капча выдается пользователю в случае если он 2 раза неправильно вводит логин или пароль.

Рисунок 18 — Блокировка экрана

Блокировка экрана срабатывает в случае если пользователь неправильно вводит капчу 2 раза.

**8 Функционал программного решения**

В данной программе были реализованы такие функции как:

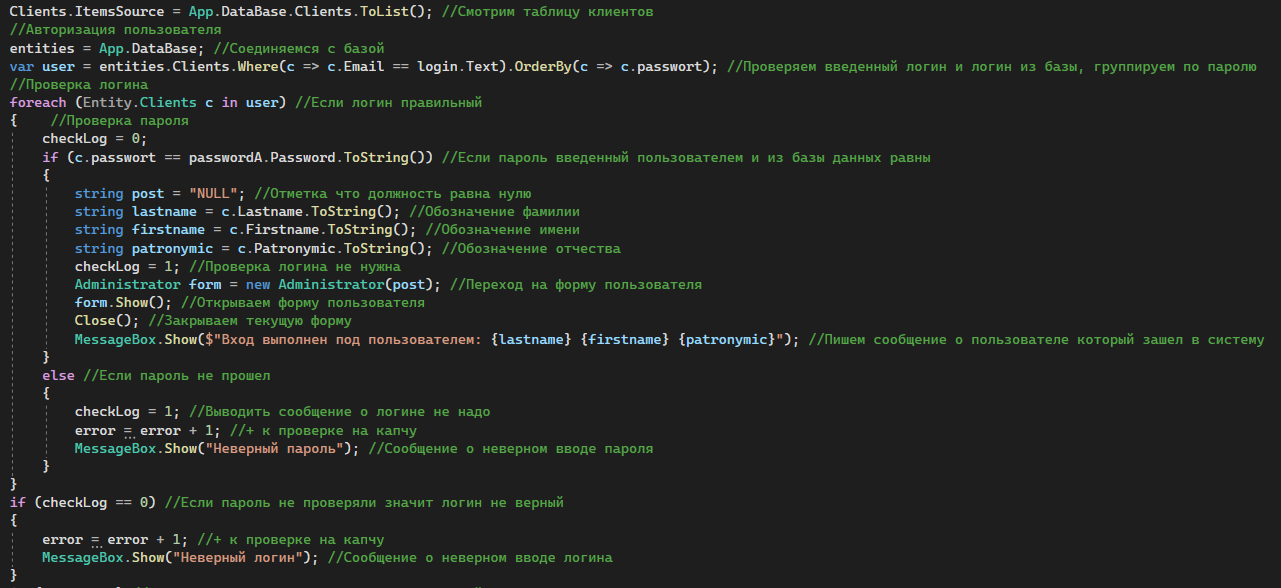
1. Проверка на введенные логин и пароль (рисунки 19-20).

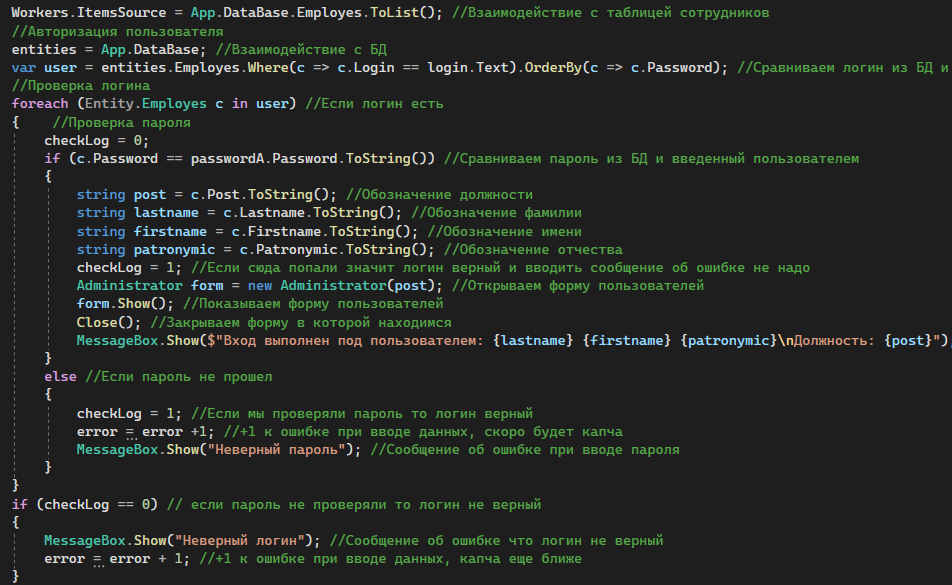
2. Разграничение прав доступа (рисунки 21-22).

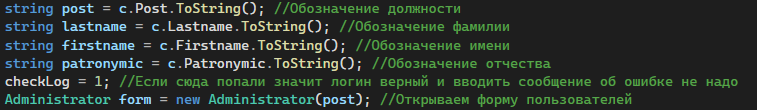
3. Рабочий таймер (рисунок 23).

4. Капча (рисунки 24-26).

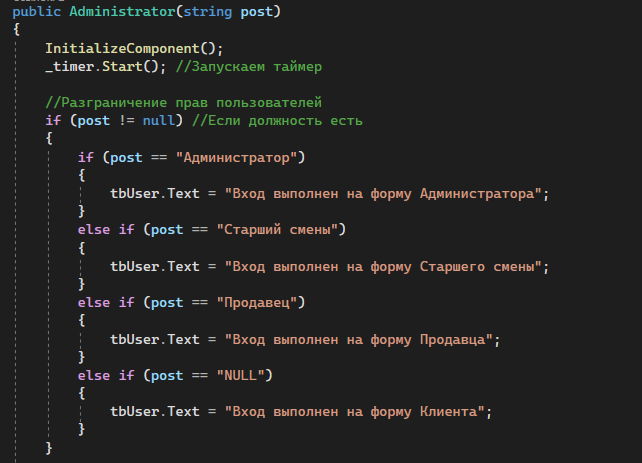
5. Блокировка, в случае неуспешного входа (рисунок 27).

Рисунок 19 — Проверка логина и пароля для клиента

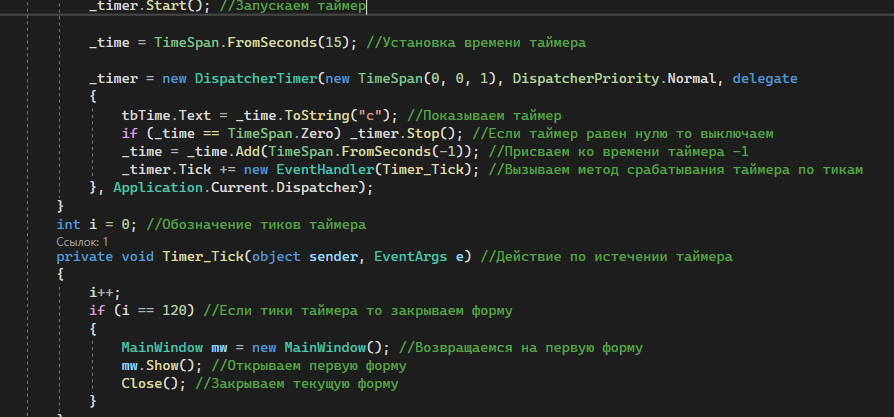
Рисунок 20 — Проверка логина и пароля для сотрудника

Рисунок 21 — Разграничение по должности

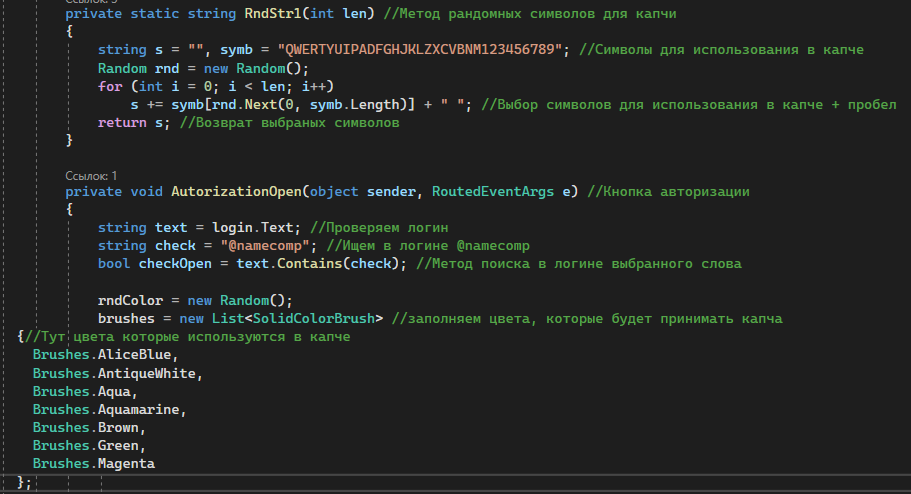
Разграничение происходит путем проверки должности у сотрудника и передаче ее на следующую форму.

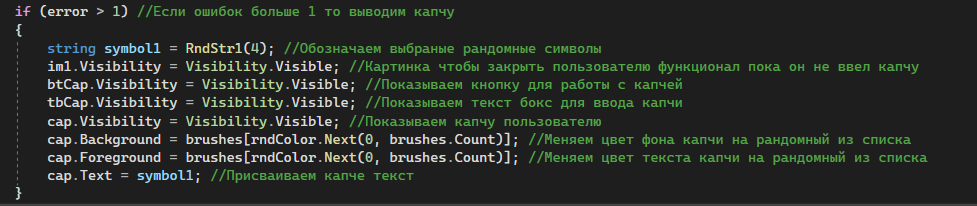
Рисунок 22 — Разграничение на форме пользователей

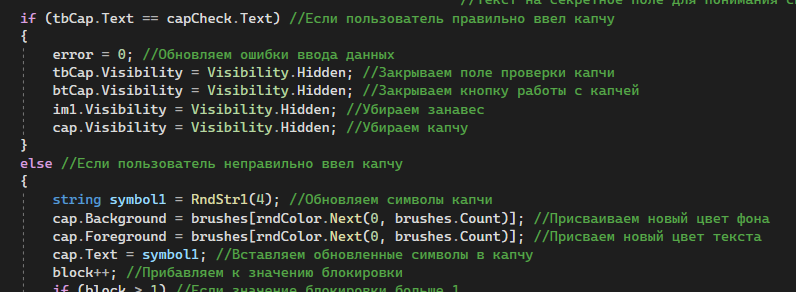
На форме пользователей программа проверяет должность пользователя входящего в систему и выдает ему функционал доступный для него.

Рисунок 23 - Таймер

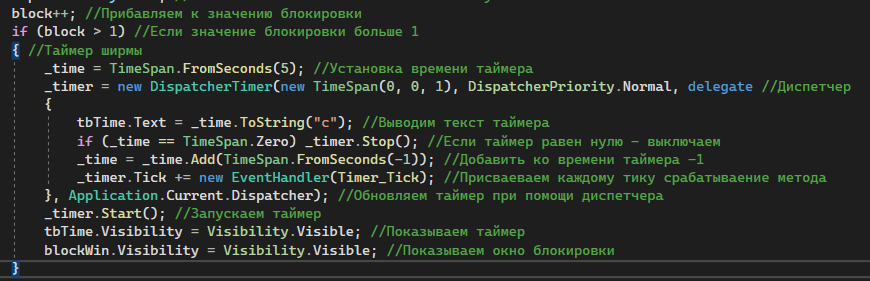
При переходе на вторую форму таймер запускается в инициализации и сразу же начинает работу, время для таймера укорочено для проверки работоспособности, через 15 секунд по окончании таймера пользователь выйдет из учетной записи.

Рисунок 24 — Методы рандома символов и цвета

Рисунок 25 — Капча

Рисунок 26 — Проверка правильности ввода капчи

Капча срабатывает в случае если пользователь допустил более одной ошибки при вводе данных.

Рисунок 27 - Блокировка

В случае если пользователь допускает более одной ошибки при вводе капчи его окно блокируется на 5 секунд.

**Приложение А**