**Содержание**

1 Анализ предметной области. ERD-диаграмма 4

2 Анализ предметной области. Диаграмма прецедентов 5

3 Анализ предметной области. Диаграмма деятельности 6

4 Анализ предметной области. Физическая модель базы данных 7

5 Анализ предметной области. Словарь данных 9

6 Импорт данных…………………………………………………………………..11

7 Проектирование модулей программного решения……………………………14

8 Пользовательский интерфейс приложения…………………………………….18

9 Функционал программного решения…………………………………………..22

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

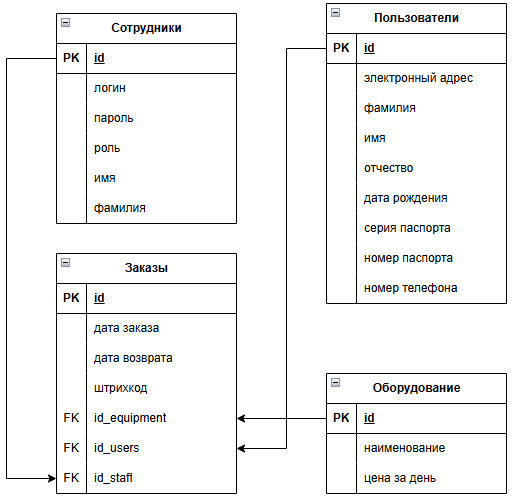
****

Рисунок 1 – Модель базы данных

В разработанной модели БД объектами являются таблицы:

1) «staff» (Сотрудники) – предназначена для хранения информации о сотрудниках.

2) «users» (Клиенты) – предназначена для хранения информации о клиентах.

3) «orders» (Заказы) – предназначена для хранения информации о заказах.

4) «equipment» (Услуги) – предназначена для хранения информации об услугах.

**2 Анализ предметной области. Диаграмма прецедентов**

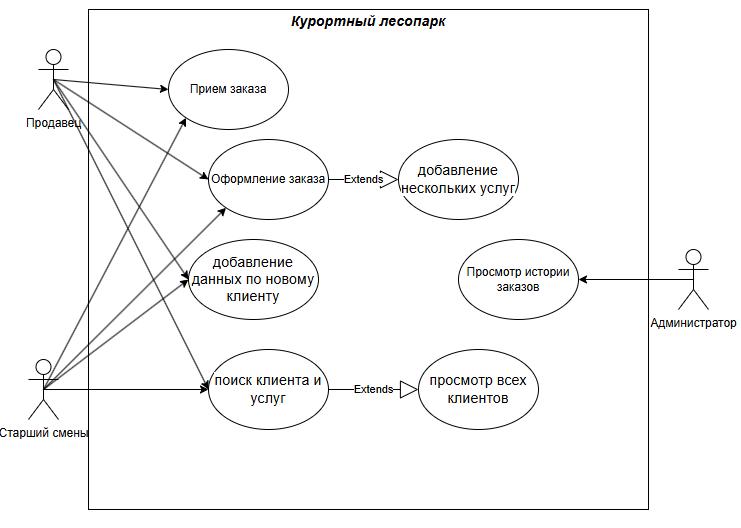
****

Рисунок 2 – Архитектура программного решения

С приложением могут работать следующие пользователи:

1) Администратор – имеет возможность просматривать историю заказов по точкам проката.

2) Старший смены – имеет возможность принимать заказы, оформлять заказы с добавлением нескольких услуг, добавлять, т.е. регистрировать нового клиента в системе и просматривать всех клиентов и услуг.

3) Продавец - имеет возможность принимать заказы, оформлять заказы с добавлением нескольких услуг, добавлять, т.е. регистрировать нового клиента в системе и просматривать всех клиентов и услуг.

**3 Анализ предметной области. Диаграмма деятельности**

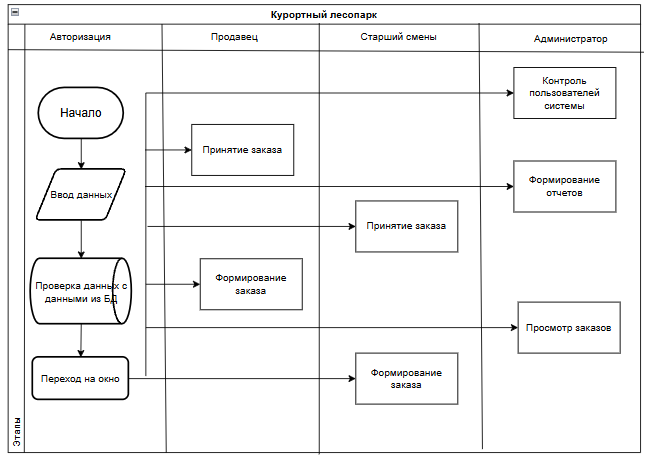
****

Рисунок 3 – Диаграмма деятельности

Диаграмма деятельности описывает процесс входа сотрудников в систему с учетом дифференцированного доступа. После успешного входа, в зависимости от уровня доступа, сотрудник получает доступ к одному из трех этапов.

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

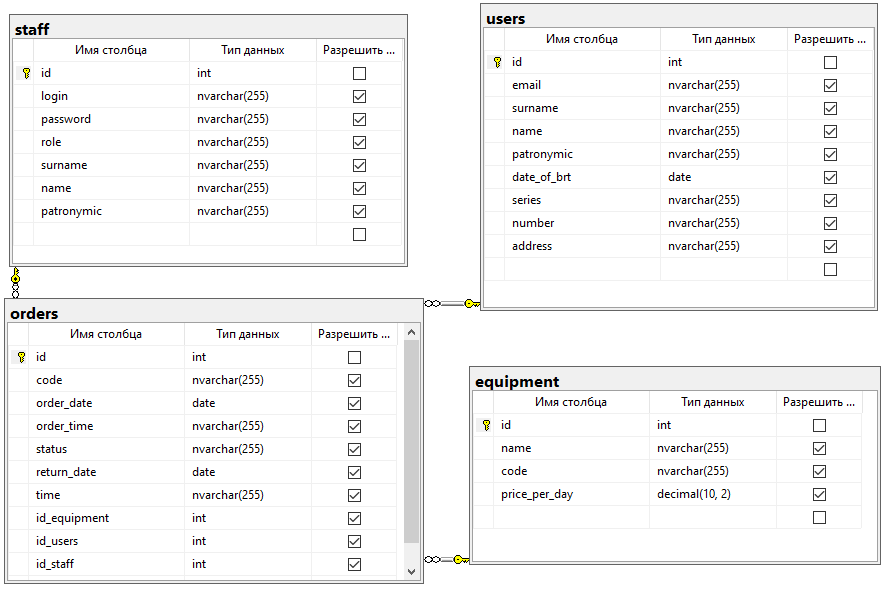


Рисунок 4 – Физическая модель БД

Сущность «orders» связана с сущностью «staff» типом связи «один-ко-многим», так как один сотрудник может регистрировать множество заказов.

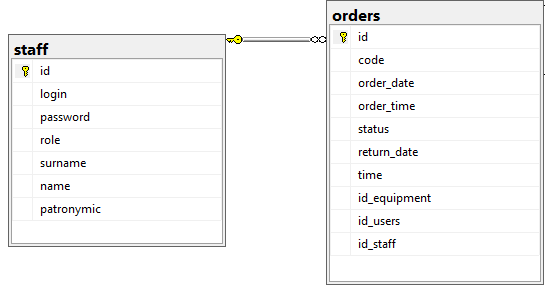


Рисунок 5 – Ассоциация «orders» и «staff»

Сущность «orders» связана с сущностью «users» типом связи «один-ко-многим», так как клиент заказывает множество заказов.

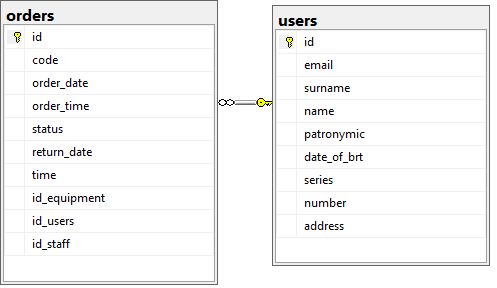


Рисунок 6 - Ассоциация «orders» и «users»

Сущность «orders» связана с сущностью «equipment» типом связи «один-ко-многим», так как в одном заказе может быть множество услуг.

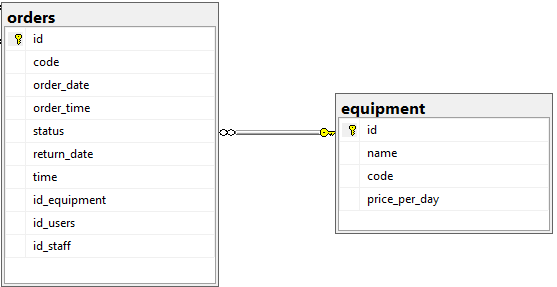


Рисунок 7 - Ассоциация «orders» и «equipment»

**5 Анализ предметной области. Словарь данных**

****

Рисунок 8 – Словарь данных

При проектировании словаря данных были задействованы основные типы данных, такие как «nvarchar» и «int».

По мимо вышеперечисленных типов данных были использованы:

- «date» - использовался для хранения даты в одном столбце;

- «decimal(10,2) – использовался для хранения цифрового типа данных;

- «time» - использовался для хранения данных о времени.

**6 Импорт данных**

Импорт данных в таблицу сотрудники.

Для импорта осуществили разделение информации о ФИО пользователя на отдельные таблицы при помощи функции «Текст по столбцам». Выберем необходимый столбец, затем ориентир, по которому будет осуществляться разделение в нашем случае это пробел.

Перед разделение необходимо подготовить пустые поля для наших новых столбцов.

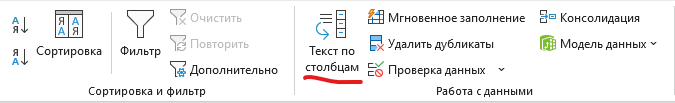


Рисунок 9 – Функция «Текст по столбцам»

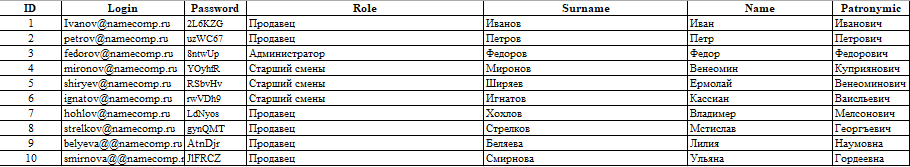


Рисунок 10 – Измененные данные сотрудников

В таблице пользователи было осуществлено разделение данных о ФИО, серии и номера паспорта и адреса проживания при помощи функции «Текст по столбцам».



Рисунок 11 - Измененные данные о клиентах

Таблица услуги остается без изменений.

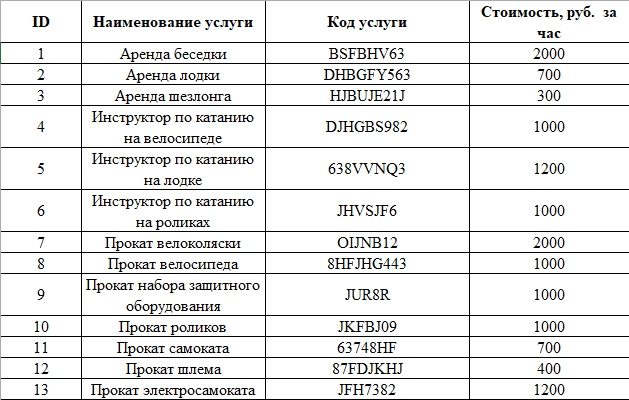


Рисунок 12 – Данные об услугах

В таблице заказы время аренды было переведено в единую систему счисления при помощи функции «ЕСЛИ».

Были удалены названия единиц измерения при помощи функции «Найти и заменить».

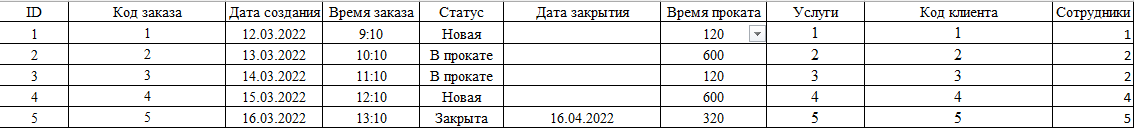
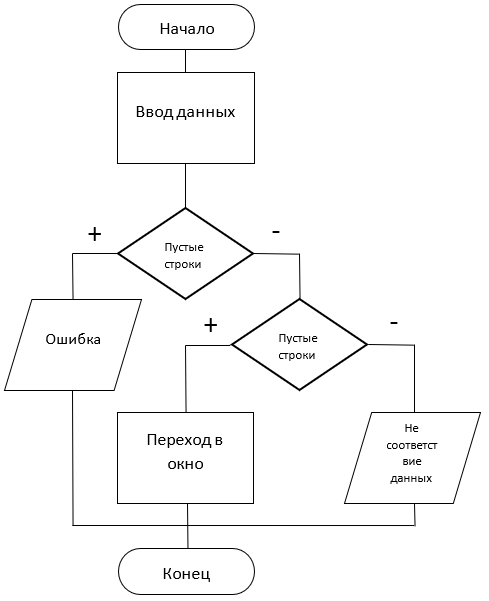


Рисунок 13 – Измененные данные о заказах

Импорт в базу данных осуществлялся копированием данных из таблиц в exel и вставкой в пустые таблицы в БД.

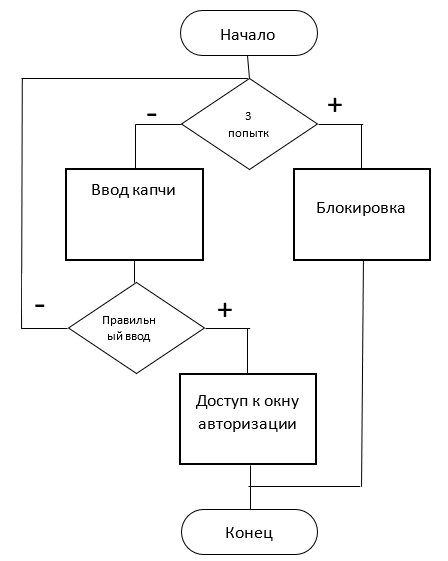
**7 Проектирование модулей программного решения**

****

Блок – схема 1 – Авторизация

Блок-схема 1 иллюстрирует процесс авторизации пользователя: при вводе данных система проверяет на наличие пустых полей. Если поля пусты, выводится сообщение об ошибке. В противном случае, система сверяет введенные данные с базой данных. При совпадении данных пользователь попадает в функциональное окно.

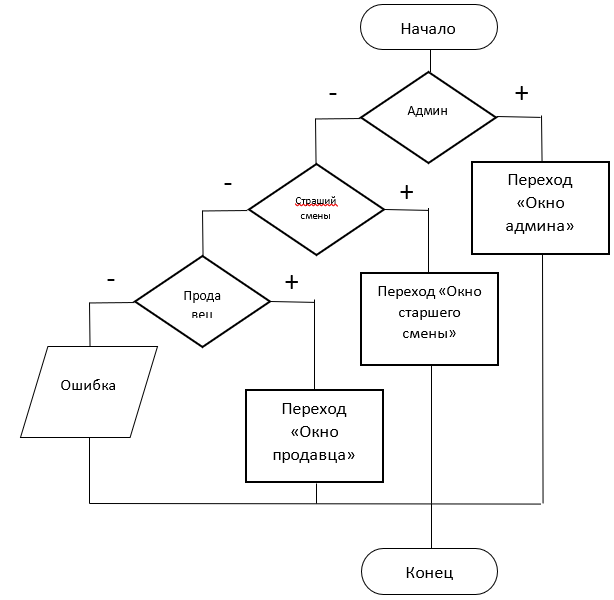
В случае несовпадения выводится сообщение об ошибке, и доступ в функциональное окно не предоставляется.



Блок – схема 2 – Капча

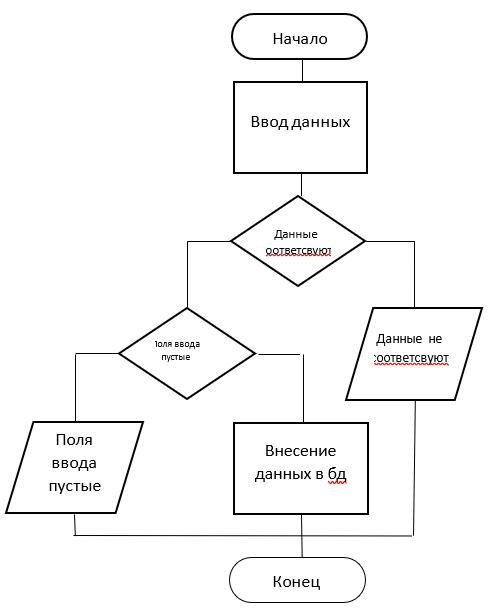
Блок-схема 2 описывает механизм отображения капчи после неудачной попытки авторизации (неверный логин или пароль). Пользователю предоставляется три попытки для ввода капчи. После трех неудачных попыток доступ к системе блокируется на определённый период.

Успешный ввод капчи позволяет пользователю повторить попытку ввода логина и пароля.



Блок – схема 3 – Разграничение прав доступа

Блок-схема 3 отображает механизм контроля доступа после успешной авторизации. Система проверяет роль пользователя, сопоставляя его логин и пароль с ролями, определёнными в системе. Если роль найдена, пользователь получает доступ к соответствующему меню и функциям, исключая доступ к функциям других ролей. В случае ошибки или отсутствия роли, пользователю выводится сообщение об ошибке, и доступ к функциональным возможностям не предоставляется.



Блок – схема 4 – Оформление заказа

Блок-схема 4 иллюстрирует процесс оформления заказа. Система проверяет введённые пользователем данные на полноту и соответствие требованиям. При обнаружении ошибок или пустых полей выводится сообщение об ошибке. В случае корректного ввода данных заказ оформляется успешно.

**8 Пользовательский интерфейс приложения**

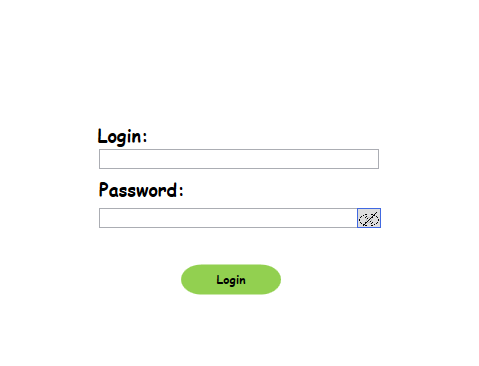
****

Рисунок 14 – Окно авторизации

На окне авторизации имеются текстовые поля для ввода логина и пароля, кнопки. Пароль скрыт маской ввода, если пользователь желает увидеть пароль, ему необходимо поставить галочку и пароль станет видимым. Кнопка «Login» отвечает за переход пользователя в меню. В случае если пароль будет введен неверно, то откроется капча для проверки пользователя/



Рисунок 15 – Окно капчи

На окне капчи расположено место с генерацией картинки и сочетания букв, текстовое поле для ввода капчи, кнопка «Проверить» для проверки капчи на правильность ввода и в случае правильности пользователя пересылает в окно авторизации и ему дается еще одна попытка для ввода логина или пароля. В случае если пользователь ввел капчу неверно, то ему дается еще 2 попытки, в общей сумме 3 попытки на ввод. Кнопка «Обновить» отвечает за создание нового сочетания картинки и слова.



Рисунок 16 – Окно администратора

В окне администратора представлена таблица, в которой располагается информация о заказах и кнопка «Выход», отвечающая за переход пользователя в окно «Авторизации», а также таймер.

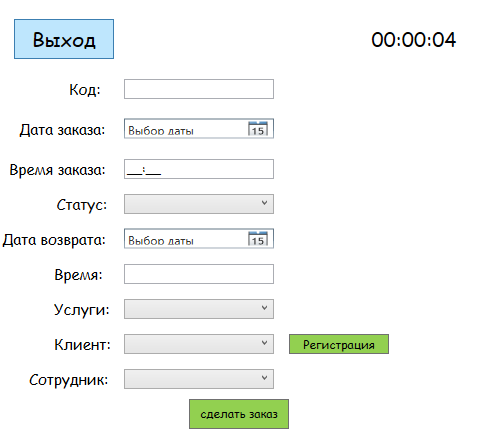


Рисунок 17 – Окно продавца и старшего смены

В окне продавца и старшего смены располагается кнопка «Регистрация», отвечающая за добавление данных о клиентах (Рисунок 18). Также присутствуют поля для добавления нового заказа, кнопка «Выход» и таймер.

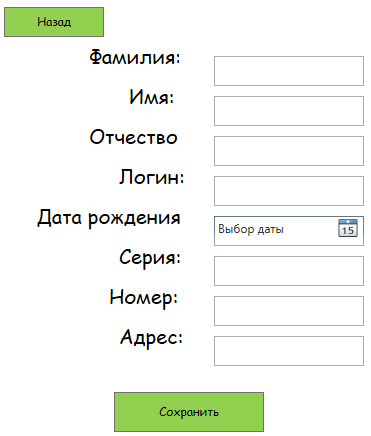


Рисунок 18 – Окно регистрации клиента

В окне регистрации клиента располагаются текстовые поля для ввода информации о клиенте и кнопка, которая осуществляет добавление данных и переход в окно с добавлением заказа.

**9 Функционал программного решения**

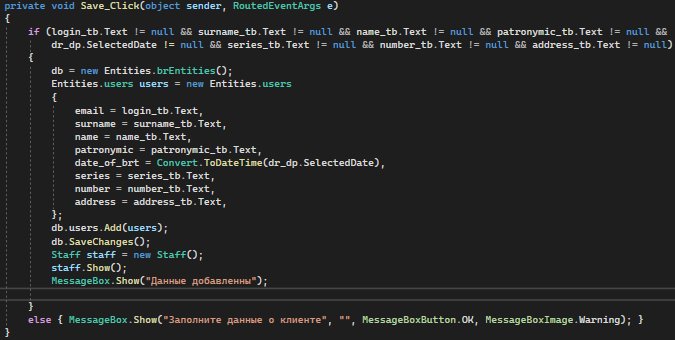
****

Рисунок 19 – Добавление нового клиента

Данный код обрабатывает событие нажатия кнопки "Сохранить". Он проверяет, заполнены ли все поля формы. Если все поля заполнены, данные сохраняются в базу данных. После успешного сохранения открывается форма «Staff» и выводится сообщение об успешном добавлении. Если поля не заполнены, выводится предупреждение.

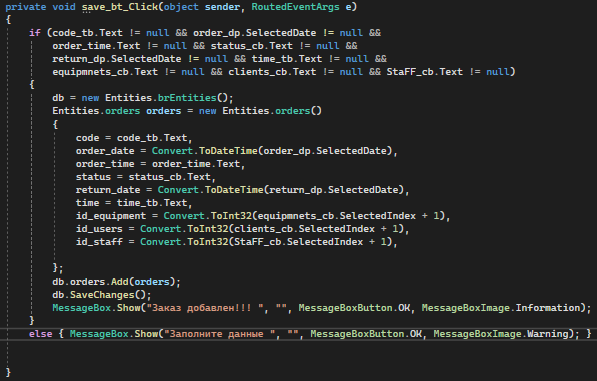


Рисунок 20 – Добавление заказа

Данный код обрабатывает нажатие кнопки «Сохранить». Он проверяет, заполнены ли все поля формы. Если все поля заполнены, данные о заказе сохраняются в базу данных. После сохранения выводится сообщение об успехе. Если поля не заполнены, выводится предупреждающее сообщение.

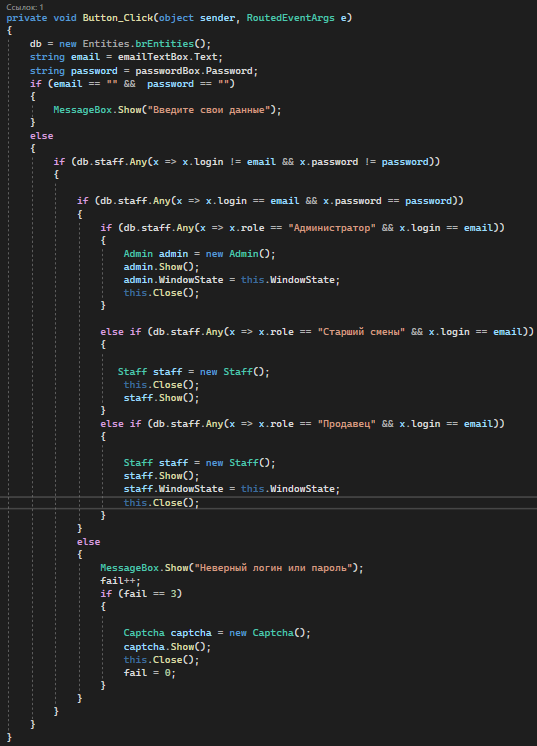


Рисунок 21 – Разграничение прав доступа

Данный код обрабатывает вход пользователя. Он проверяет поля логин и пароль. Если оба пусты, выводится сообщение об ошибке. В противном случае, он проверяет наличие пользователя с введенными данными в базе данных. Если пользователь найден, открывается форма «Admin», «Staff» (в зависимости от роли пользователя), а текущая форма закрывается. Если данные неверны, выводится сообщение об ошибке. После трех неудачных попыток открывается форма «Captcha».

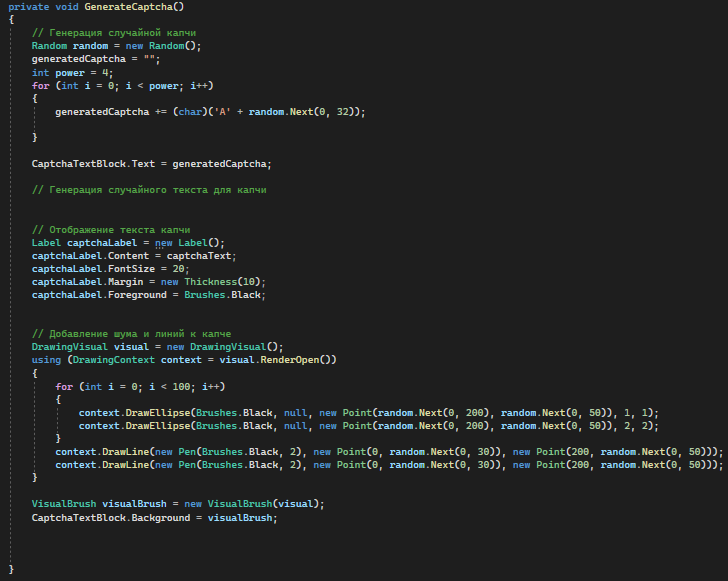


Рисунок 22 – Генерация капчи

Этот код генерирует и отображает случайную капчу. Сначала генерируется 4-символьная строка из кириллических букв. Затем создается визуальный элемент, на котором рисуются случайные точки и линии, создавая эффект шума. Наконец, этот визуальный элемент устанавливается в качестве фона, где отображается сгенерированная текстовая капча.

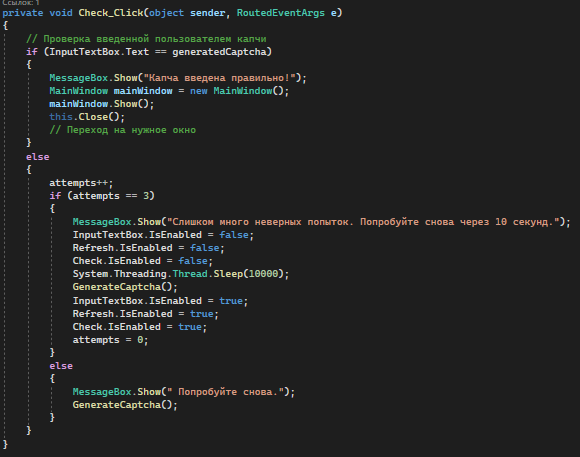


Рисунок 23 – Проверка введенной капчи

Этот код представляет собой обработчик события нажатия кнопки «Check», который проверяет введённый пользователем текст капчи.

Проверка капчи: если текст совпадает с сгенерированной капчей, выводится сообщение о правильности ввода, создаётся и открывается новое окно, а текущее окно закрывается.

Неверный ввод: если введённая капча неверна, увеличивается счётчик попыток. Если количество попыток достигает 3, пользователю показывается сообщение о превышении лимита попыток. Поля ввода и кнопки отключаются на 10 секунд, после чего капча обновляется, и элементы управления снова становятся доступными. Если попыток меньше трёх, выводится сообщение о необходимости повторной попытки, и капча генерируется заново. Таким образом, код реализует механизм проверки капчи с ограничением на количество попыток.

**Приложение А**