**Содержание**

[1 Анализ предметной области. ERD-диаграмма 4](#_Toc8291)

[2 Анализ предметной области. Диаграмма прецедентов 6](#_Toc17037)

[3 Анализ предметной области. Диаграмма деятельности 8](#_Toc20733)

[4 Анализ предметной области. Физическая модель базы данных 10](#_Toc7704)

[5 Анализ предметной области. Словарь данных 13](#_Toc2034)

[6 Импорт данных 16](#_Toc26512)

[7 Проектирование модулей программного решения 29](#_Toc32549)

[8 Пользовательский интерфейс приложения 33](#_Toc11080)

[9 Функционал программного решения 38](#_Toc19374)

# 1 Анализ предметной области. ERD-диаграмма

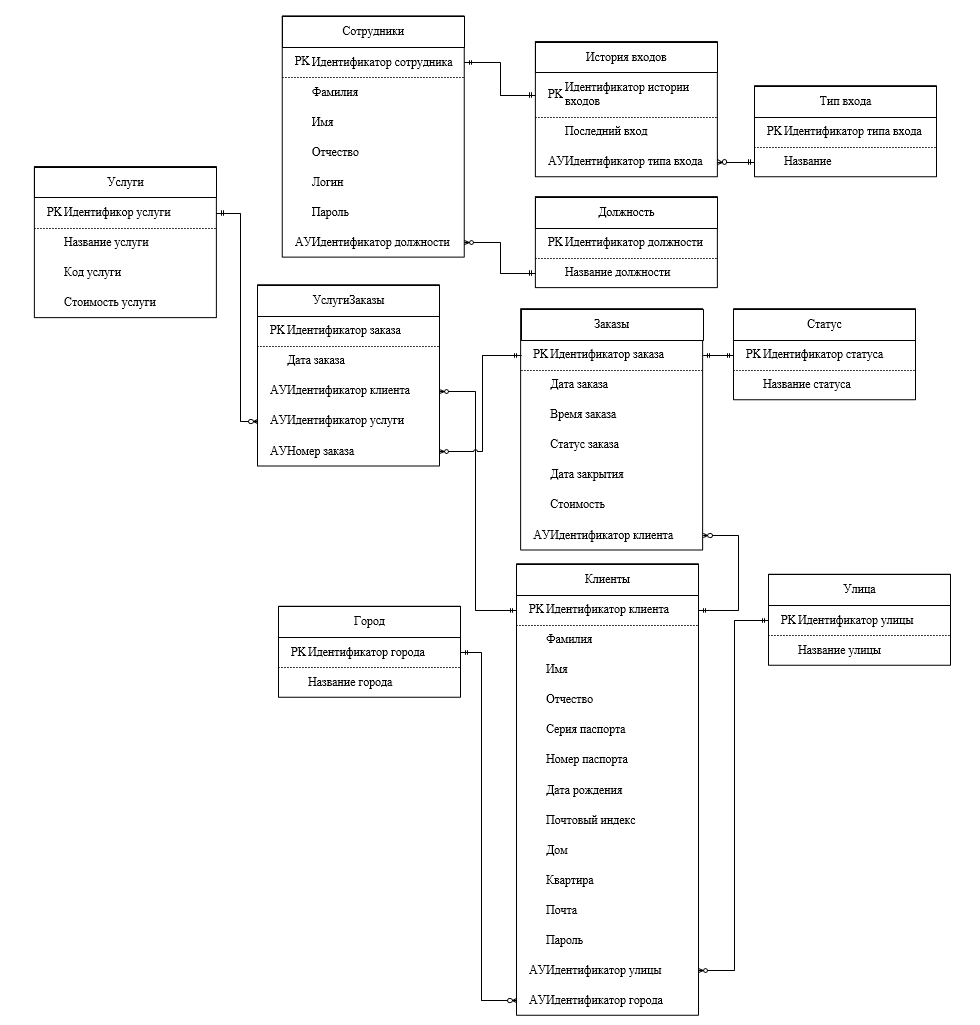


Рисунок 1 – Логическая модель

На рисунке 1 представлена логическая модель предметной области компании «Охта Парк». Предметная область разделяется на две части, одна содержит данные о взаимодействии клиентов с услугами, а вторая о самих сотрудниках.

Часть с данными о сотрудниках содержит следующие сущности:

* «Тип входа» - обеспечивает хранение информации об успешном или неуспешном входе пользователя в систему;
* «История входа» - обеспечивает хранение дату последнего успешного или неуспешного входа пользователя в систему, а также тип входа;
* «Должность» - обеспечивает хранение всех должностей для дальнейшего присвоения конкретным сотрудникам;
* «Сотрудник» - обеспечивает хранение ФИО, должности, а также логина и пароля для осуществления входа в систему.

Часть, содержащая данные о клиентах и услугах, имеет следующие сущности:

* «Улица» - обеспечивает хранение название улицы, на которой осуществляется предоставление услуг для клиента;
* «Город» - обеспечивает хранение название города для обеспечения дальнейшего выбора услуг доступных в данном городе;
* «Статус» - обеспечивает хранение информации о статусе заказа клиента;
* «Услуги» - обеспечивает хранение информации об услуге, которую могут предоставить, а также ее стоимость;
* «Заказ» - обеспечивает хранение информации о дате и времени создания заказа, статусе заказа, стоимости заказа, а также дате закрытия заказа;
* «Услуги и заказы» - обеспечивает хранение информации об дате и номере заказа, а также выбранных услуг клиента;
* «Клиент» - обеспечивает хранение информации о ФИО, серии и номера паспорта, дате рождения, адресе проживания, почте и пароле для системы.

# 2 Анализ предметной области. Диаграмма прецедентов

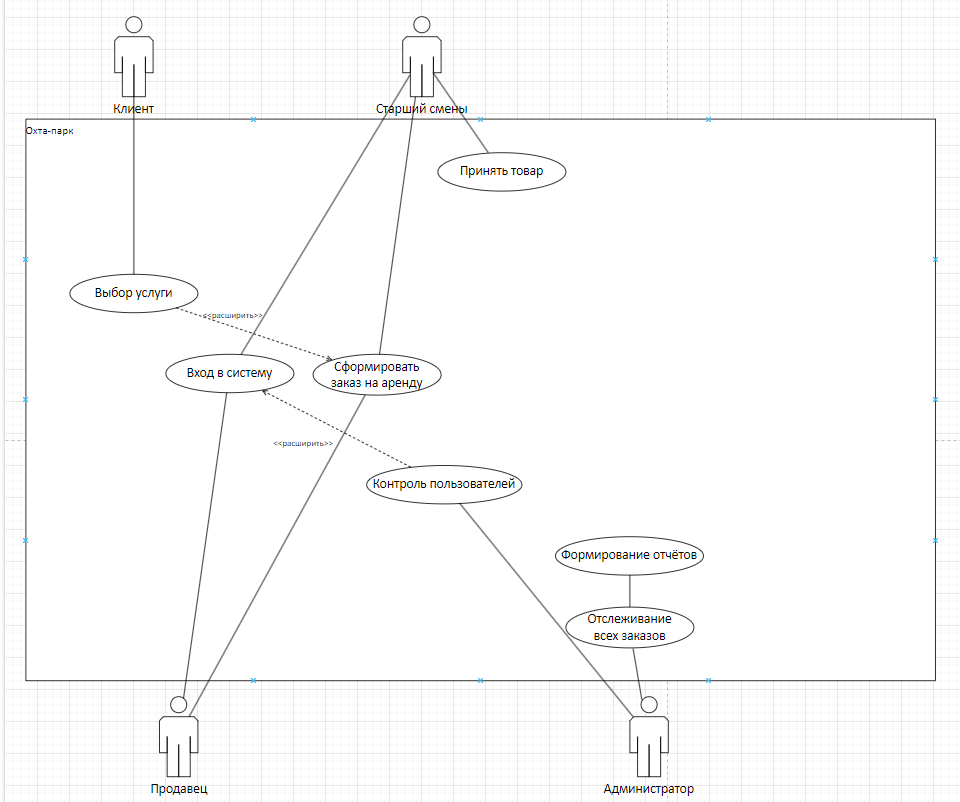


Рисунок 2 – Диаграмма прецедентов

На рисунке 2 представлена диаграмма прецедентов для системы управления заказами в компании «Охта парк». Данная модель содержит следующих актеров:

* «Администратор» - обеспечивает формирование отчетов по проанализированной информации о заказах. Имеет возможность контролирования пользователей, которые осуществили вход в систему;
* «Продавец» и «Старший смены» - обеспечивают формирование заказа в аренду в зависимости от выбранных услуг клиента. Старший смены так же осуществляет прием товара;
* «Клиент» - осуществляет выбор услуг, которые предоставляет компания.

# 3 Анализ предметной области. Диаграмма деятельности

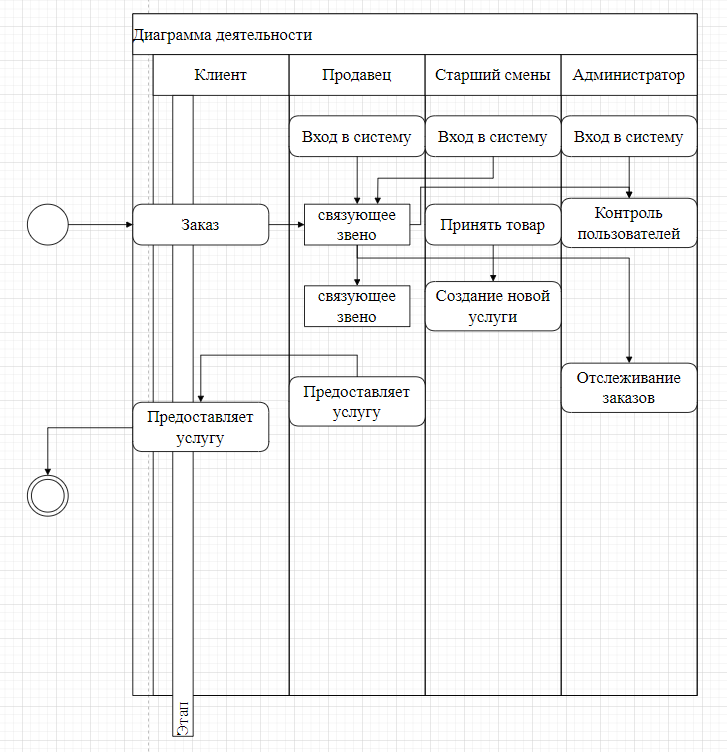


Рисунок 3 – Диаграмма деятельности

На рисунке 3 представлена диаграмма деятельности сотрудников компании «Охта парк».

Данная модель работает по следующему принципу:

* начало: Клиент обращается к продавцу или старшему смены с запросом на аренду нужного ему оборудования;
* действие 1: Продавец или старший смены осуществляют вход в систему;
* переход к действию 2: Выдача клиенту выбранного оборудования для дальнейшего использования;
* завершение действия 1: Оборудование передается клиенту, а заказ на аренду сформирован;
* начало действия 2: Старший смены принимает оборудование обратно на склад;
* завершение действия 2: В базе данных регистрируется возврат оборудования, и заказ на аренду закрывается;
* администратор точки проката формирует отчеты о состоянии заказов и анализирует данные о прокате оборудования для оптимизации работы точки проката, а также обеспечивает контроль работы сотрудников.

# 4 Анализ предметной области. Физическая модель базы данных

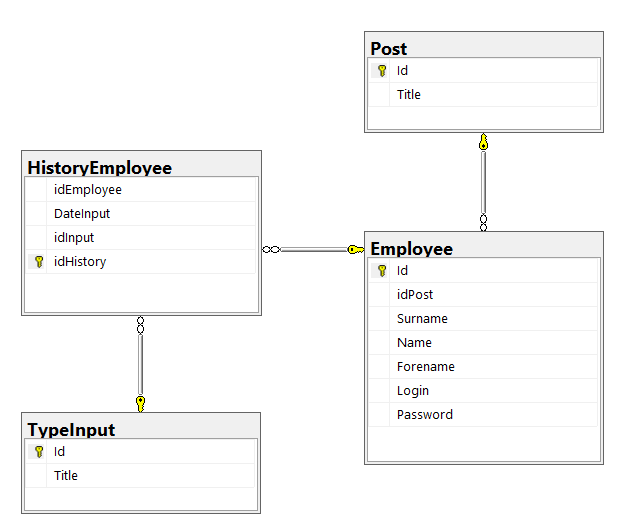


Рисунок 4 – Физическая модель 1

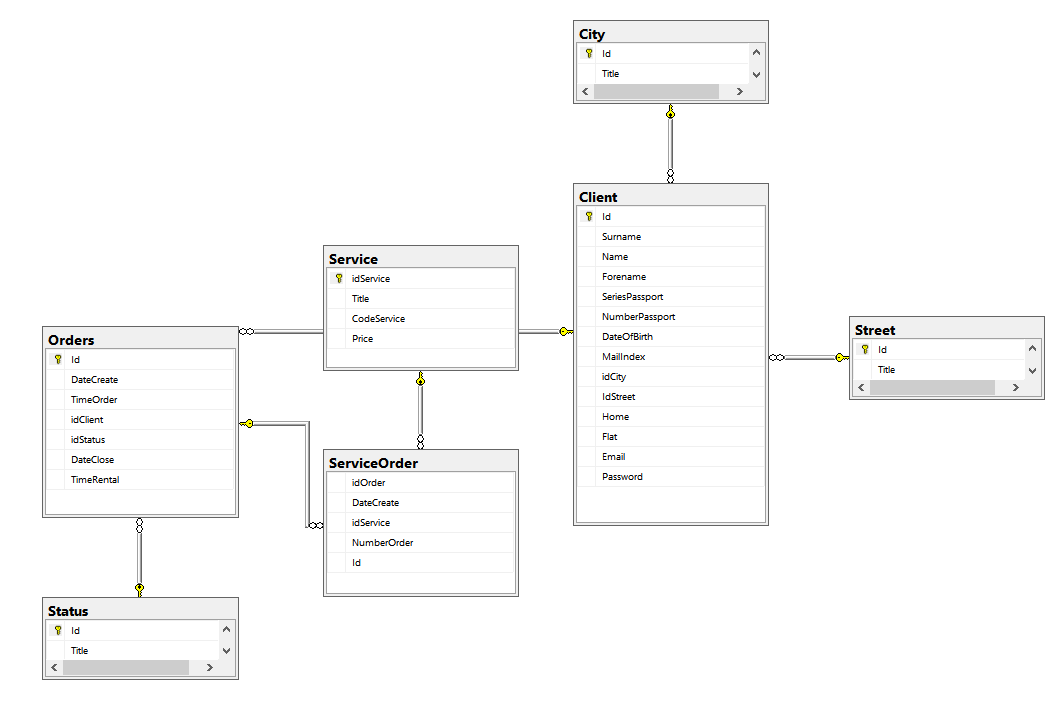


Рисунок 5 – Физическая модель 2

Физическая модель базы данных - это структура данных, состоящая из таблиц, колонок, индексов и других объектов, которые определяют, как данные хранятся и обрабатываются в базе.

На рисунках 4 и 5 представлена физическая модель базы данных предметной области «Охта парк», которая имеет структурированные атрибуты и связи логическая модель.

Описание связей в физической модели 1:

* сущности «Post» и «Employee» связаны типом связи «один-ко-многим», так как на одну должность может приходиться несколько записей о сотрудниках;
* сущности «Employee» и «HistoryEmployee» связаны типом связи «один-ко-многим», так как у одного сотрудника может быть несколько записей о входе;
* сущности «TypeInput» и «HistoryEmployee» связаны типом связи «один-ко-многим», так как у многих записей о истории входа сотрудников может быть только одна запись о типе входа.

Описание связей в физической модели 2:

* сущности «City» и «Client» связаны типом связи «один-ко-многим», так как у одного города может быть множество записей о сотрудниках;
* сущности «Street» и «Client» связаны типом связи «один-ко-многим», так как у одной улицы может быть множество записей о сотрудниках;
* сущности «Client» и «Orders» связаны типом связи «один-ко-многим», так как у одного клиента может быть множество заказов;
* сущности «Client» и «ServiceOrder» связаны типом связи «один-ко-многим», так как в таблице может храниться несколько заказов у одного клиента;
* сущности «Orders» и «Service» связаны типом связи «многие-ко-многим» через таблицу «ServiceOrder», так как у одного заказа может быть несколько услуг, и одна услуга может находиться в нескольких заказах;
* сущности «Status» и «Orders» связаны типом связи «один-ко-многим», так как один статус может быть у нескольких заказов.

# 5 Анализ предметной области. Словарь данных

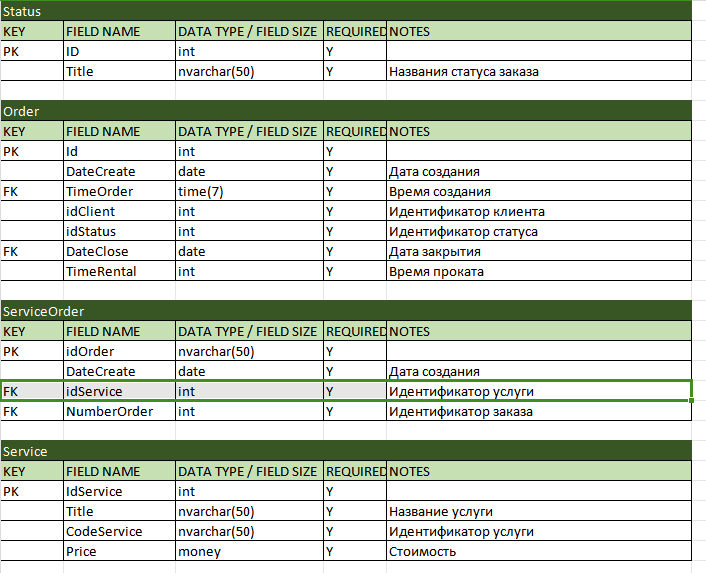


Рисунок 6 – Словарь данных 1



Рисунок 7 – Словарь данных 2

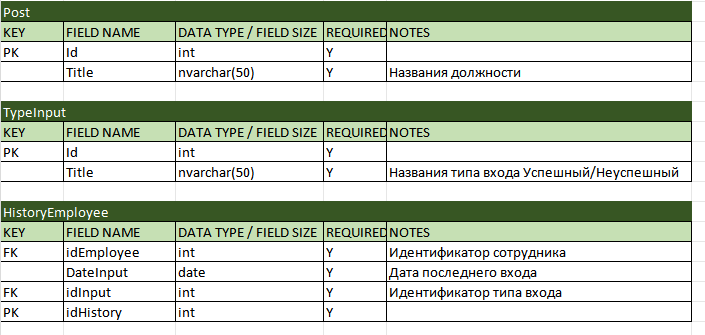


Рисунок 8 – Словарь данных 3

Словарь данных - это набор данных, который содержит информацию о других данных. Он используется для описания структуры данных, их типов, связей между ними и других характеристик.

Словарь данных может быть использован для создания физической модели данных, которая представляет собой визуальное представление структуры данных и их связей.

В словаре данных содержатся следующие сущности: Status, Orders, ServiceOrder, Service, Client, City, Street, Employee, Post, TypeInput, HistoryEmployee.

# 6 Импорт данных

Для того, чтобы привести данные к структурированному виду необходимо определить, что можно вынести в отдельную таблицу. Для этого необходимо прочитать описание предметной области и осмотреть на данные, содержащиеся в Excel-файле, которые повторяются несколько раз, как на рисунке 9.

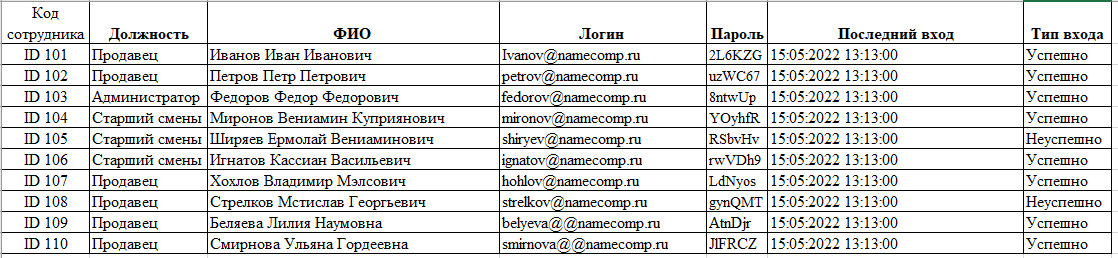


Рисунок 9 – Excel-файл «Сотрудники»

Исходя из описания предметной области компании, в «Охта Парк» можно вынести отдельно данные, которые фиксируют последний вход и его результат. Для этого нам необходимо скопировать данные из столбцов: «Код сотрудника», «Последний вход» и «Тип входа» в отдельные столбцы (рисунок 10).

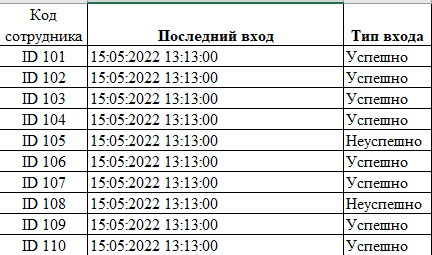


Рисунок 10 – Вынесение данных об последнем входе

После того, как мы вынесли данные, можно заметить, что столбец «Тип входа» содержит одинаковые данные, которые повторяются, и отсюда следует, что их можно вынести в отдельную таблицу. Для этого копируем столбец «Тип входа» и вставляем в два свободных столбца (рисунок 11).

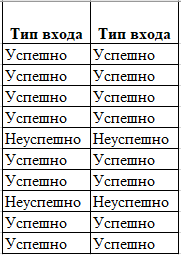


Рисунок 11 – Вынесение столбца «Тип входа»

Теперь нам необходимо провести очистку повторяющихся значений в одном из столбца. Для это выделяем нужный столбец и переходим в раздел «Данные», где в группе «Работа с данными» выбираем «Удалить дубликаты» (рисунок 12).

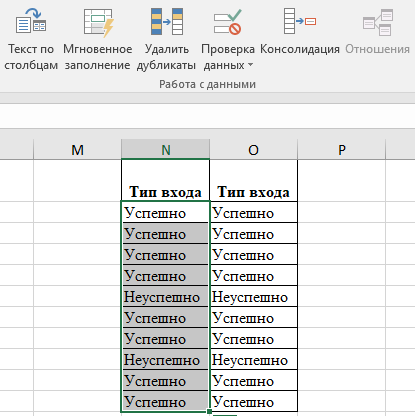


Рисунок 12 – Удаление дубликатов

В появившемся диалоговом окне необходимо выбрать пункт «сортировать в пределах указанного выделения» и выбираем «Удалить дубликаты…» (рисунок 13).

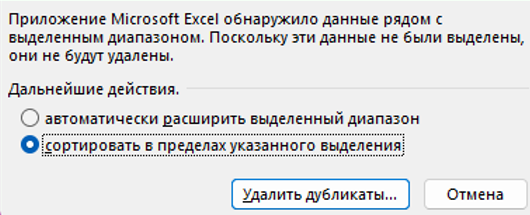


Рисунок 13 – Выбор сортировки после удаления дубликатов

Если было выбрано несколько столбцов, то в следующем окне можно выбрать те, которые подлежат удалению дубликатов, но так как выбран был один столбец, то оставляем все по умолчанию и нажимаем «ОК» (рисунок 14).

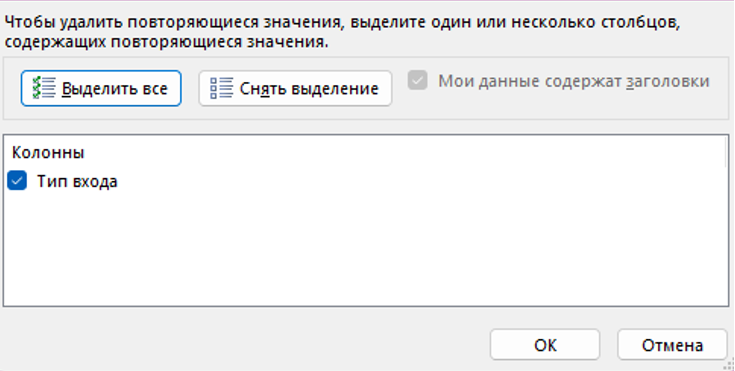


Рисунок 14 – Выбор столбцов для удаления дубликатов

После того, как дубликаты удалены, необходимо провести сортировку значений. Для этого выбираем значения столбца, в котором произвели удаление дубликатов и переходим в раздел «Данные», где в группе «Сортировка и фильтр» необходимо выбрать «Сортировать от А до Я» (рисунок 15).

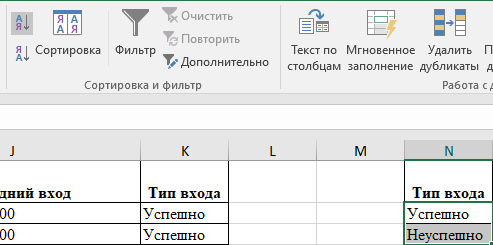


Рисунок 15 – Сортировка значений

В появившемся окне выбираем «сортировать в пределах указанного диапазона» и нажимаем «ОК» (рисунок 16).

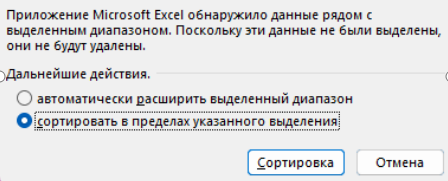


Рисунок 16 – Сортировка от А до Я

Далее нам необходимо установить идентификаторы для каждой строки, так как данных мало, то можно прописать вручную, но если их будет много, то можно потянуть за «язычок» в правом нижнем углу до определенной строки (рисунок 17).

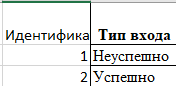


Рисунок 17 – Установка идентификатора

На этом таблицы «Тип входа» уже является сделанной, осталось только заменить значение в таблице, которая содержит эти значения. Для этого нам необходимо написать следующую формулу в ячейку, в которой будем осуществлять замену: «=ПРОСМОТР()», а после чего необходимо потянуть формулу вниз, до конечной строки. Необходимо ставить специальный знак «$», после каждой буквы в диапазонах того, что хотим найти и на что заменить. рисунок (18)

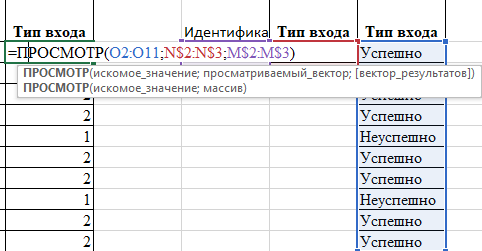


Рисунок 18 – Замена слов на идентификаторы

После чего необходимо заменить формат ячейке, в которой содержится дата и время последнего входа на значение «Время». Для этого выделяем столбец, переходим в раздел «Главная» и в группе «Число» из раскрывающегося списка выбираем «Время» рисунок 19.

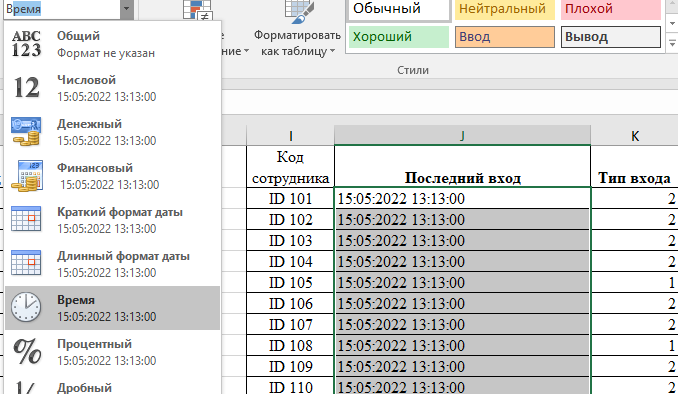


Рисунок 19 – Изменение формата ячеек

После чего таблица, хранящая значения последнего входа – готова. Точно такие же действия делаем с полем «Должность», производим замену текста на идентификатор. В конце она должна будет выглядеть, как на рисунке 20.



Рисунок 20 – Установка идентификатора «Должность»

Столбец «ФИО» имеет значения, который следует разделить по отдельным столбцам. Для этого нам сначала нужно столбцы «Логин» и «Пароль» передвинуть вправо на два столбца, а после чего выделить значения столбца «ФИО» и перейти во вкладку «Данные», где в группе «Работа с данными» выбрать «Текст по столбцам» рисунок (21).

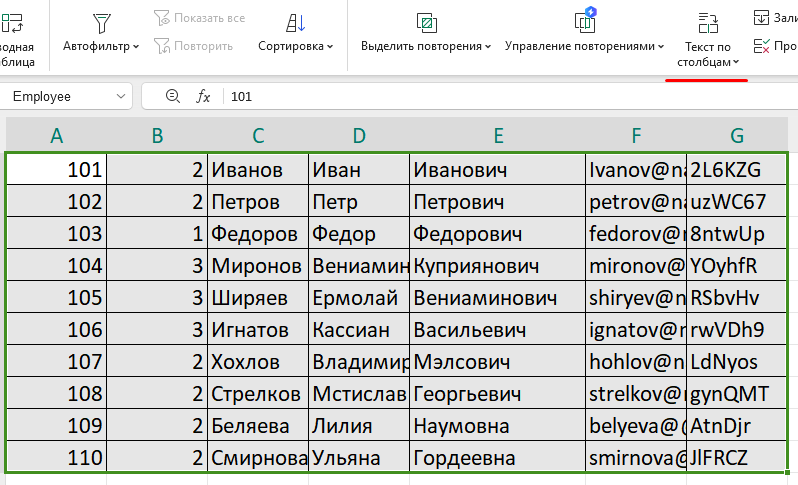


Рисунок 21 – Разбитие значений по столбцам

Откроется окно, в котором необходимо выбрать пункт «с разделителями» и нажать «Далее» (рисунок 22).

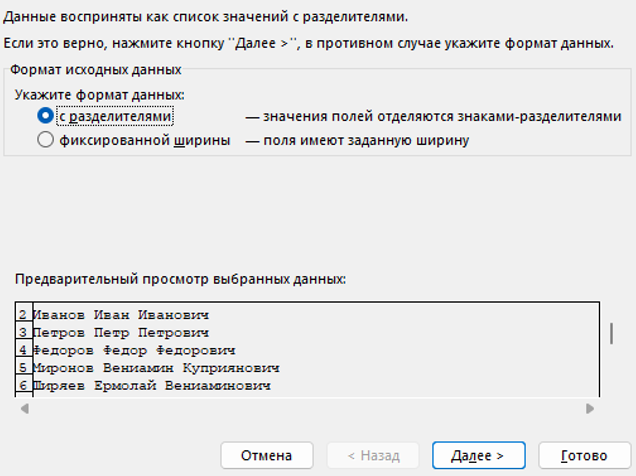


Рисунок 22 – Выбор формата данных

В следующем окне нам нужно выбрать «Символ-разделителем является» - «пробел» и нажать «Далее» (рисунок 23).

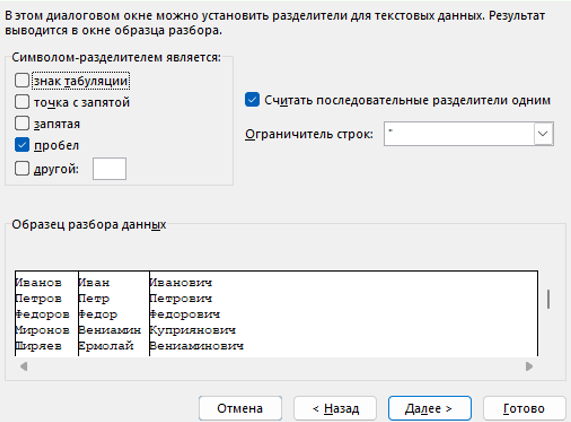


Рисунок 23 – Символ-разделителя

После чего демонстрируется, как будут выглядеть данные, а также можно выбрать формат данных столбца, нажимаем «Готово» (рисунок 24).

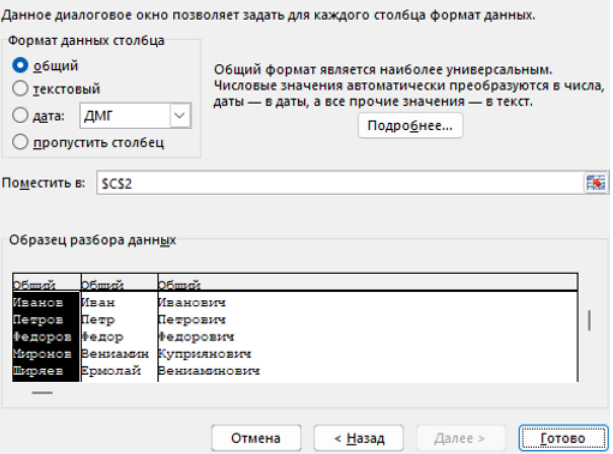


Рисунок 24 – Формат данных столбца

Появится предупреждение, что произойдет замена данных в выбранном столбце – нажимаем «ОК» (рисунок 25).

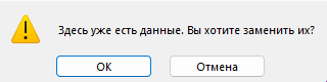


Рисунок 25 – Подтверждение замены

После чего можно дать название для столбцов, которые получили после разделения по столбцам.

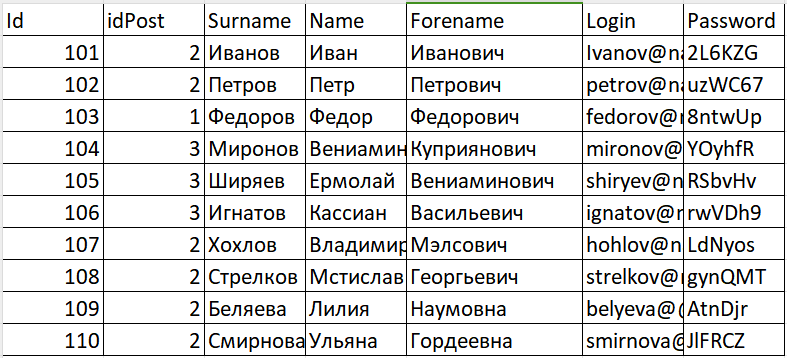


Рисунок 26 – Таблицы «Сотрудники»

После того, как полученные данные были правильно структурированы, необходимо их импортировать. Есть несколько способов для того, чтобы импортировать данные в базу, и для этого я буду пользоваться методом вставки данных. Суть данного метода заключается в том, чтобы скопировать нужные нам значения и вставить их в нужную нам таблицу.

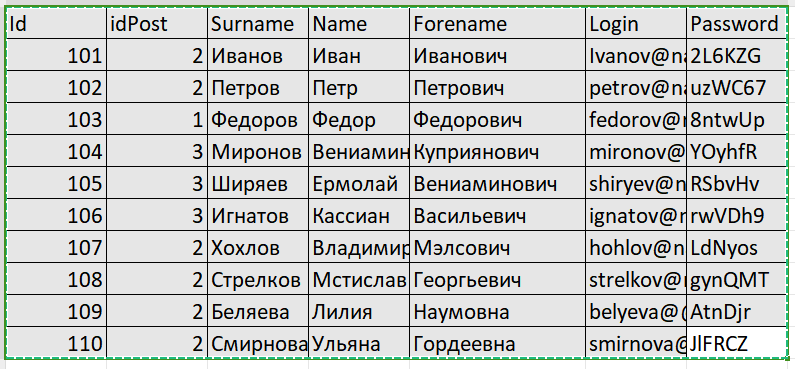


Рисунок 27 – Копирование данных для вставки

После чего мы заходим в используемую нами СУБД, жмём «ПКМ» по нужной таблице и выбираем «Изменить первые 200 строк».

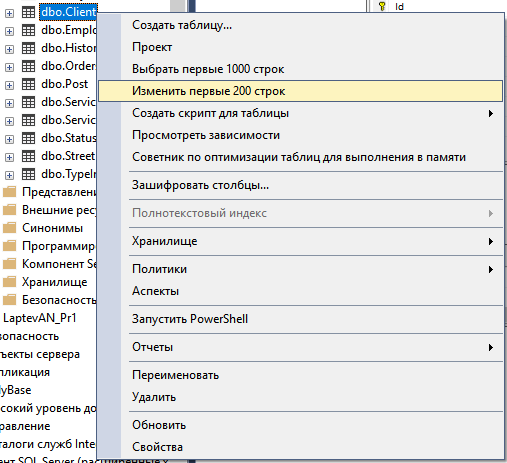


Рисунок 28 – Выбор таблицы для внесения данных

После чего в открывшемся окне нажимаем на «язычок» слева и при помощи «CTRL+V» вставляем данные в эту таблицу.

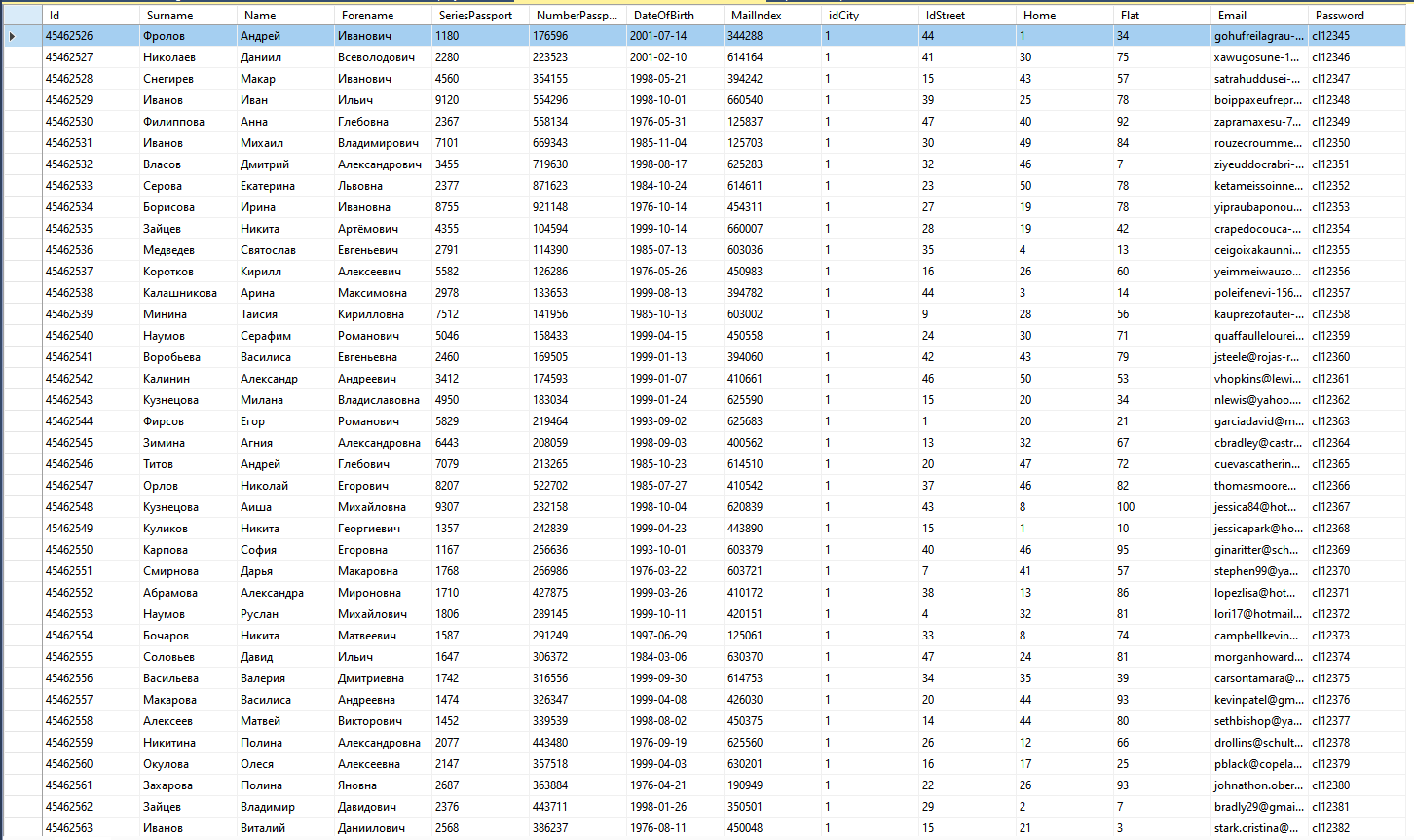
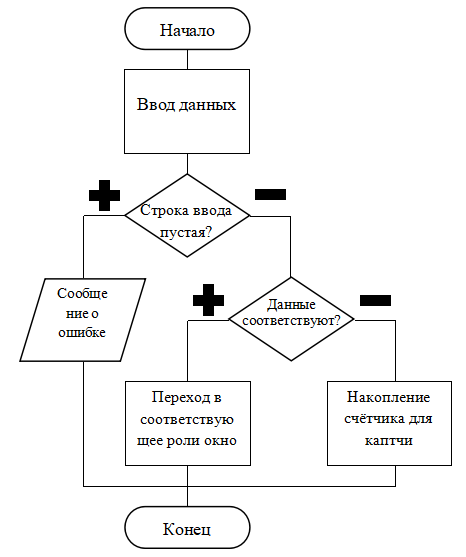


Рисунок 29 – Вставленные данные

Делаем тоже самое и для других Excel-файлов.

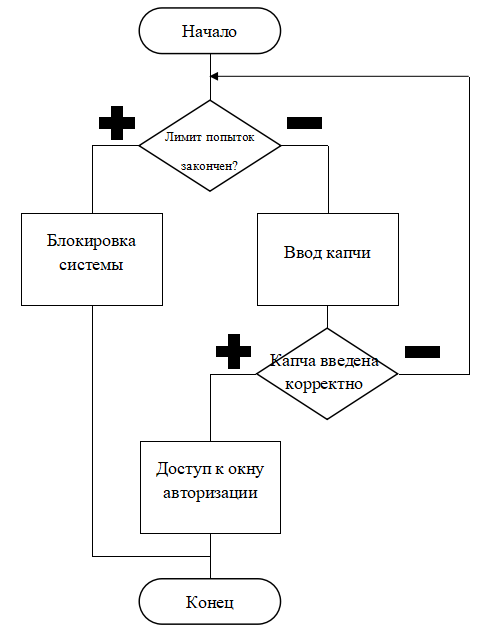
# 7 Проектирование модулей программного решения

Блок-схема 1 - Авторизация



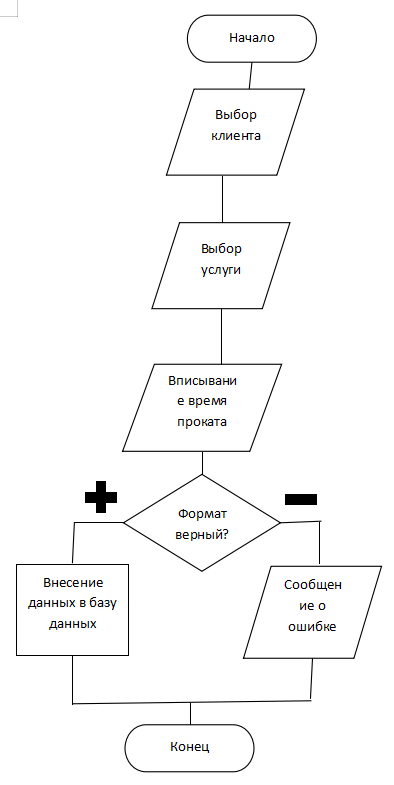
На блок-схеме 1 изображен процесс авторизации в приложении. Пользователь вводит данные в поля, и они проверяются ну пустоту. Если строка ввода пустая, то выводится сообщение об этом. Если же поля заполнены, проверяется корректность введенных данных, если данные некорректны, то происходит накопление счётчика каптчи. Если же все поля заполнены и значения в них корректны, происходит переход на соответствующую форму.

Блок-схема 2 - CAPTCHA



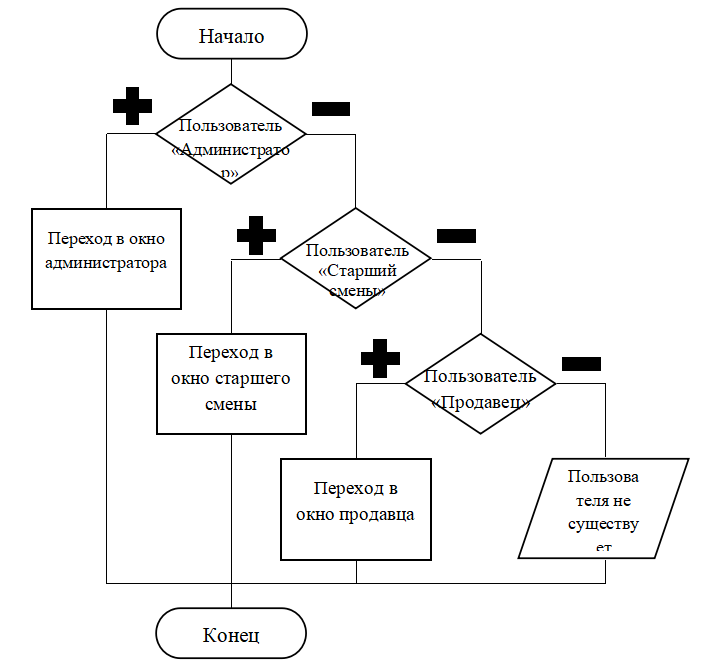
На блок-схеме 2 изображен процесс работы каптчи. При ошибке ввода данных пользователь накапливает попытки входа. При достижении лимита попыток, на экране пользователя появляется каптча. Если пользователь вводит каптчу некорректно, то она обновляется и даётся ещё шанс на попытку её ввода. Если попытки ввода заканчиваются, то система блокируется на некоторое время. Если же каптча вводится верно, то возвращается доступ к окну авторизации.

Блок-схема 3 - Добавление заказа



На блок-схеме 3 изображен процесс оформления заказа. Сначала выбираются клиент и услуга, после вписывает в поле время проката. Если формат времени верный, то происходит добавление данных в БД. Иначе, появляется сообщение о ошибке.

Блок-схема 4 - Разграничение прав доступа



На блок-схема 4 демонстрируется разграничение прав доступа. В зависимости от должности происходит переход на соответствующую форму.

# 8 Пользовательский интерфейс приложения

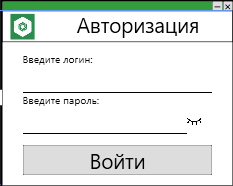


Рисунок 30 – Форма авторизации

Данная форма предоставляет возможность всем сотрудником компании «Охта парк» осуществлять вход в систему. Форма содержит компоненты для ввода логина и пароля сотрудника, кнопки для сворачивания и закрытия окна, глазик, позволяющий показывать вводимый пароль, а также кнопку осуществляющую вход в саму систему путем проверки наличия данного логина и пароля в базе данных.

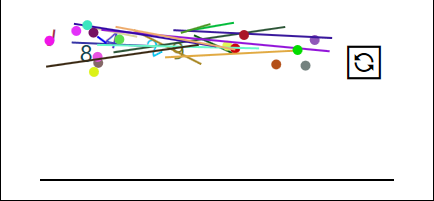


Рисунок 31 – CAPTCHA

При неверном вводе пароля или логина появляется ошибка, которая говорит, о том, что было неверно введено, если пароль или логин был некорректно введен 2 раза, появиться каптча, которую можно обновлять путем нажатия на соответствующую кнопку, а дальнейший вход в систему будет требовать ввода верного логина, пароля и каптчи.

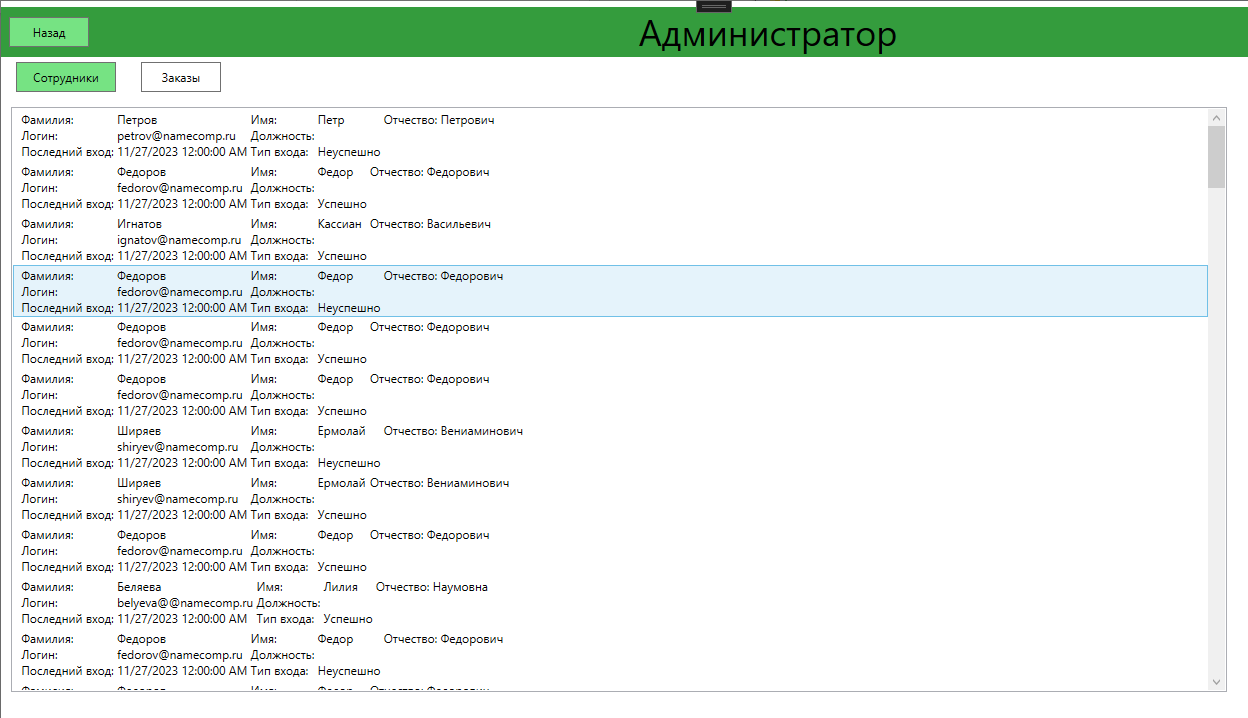


Рисунок 32 - Форма администратора

Форма администратора содержит следующие компоненты: кнопки для переключения между просмотром заказов и просмотром истории входа сотрудников, кнопку «Назад» .

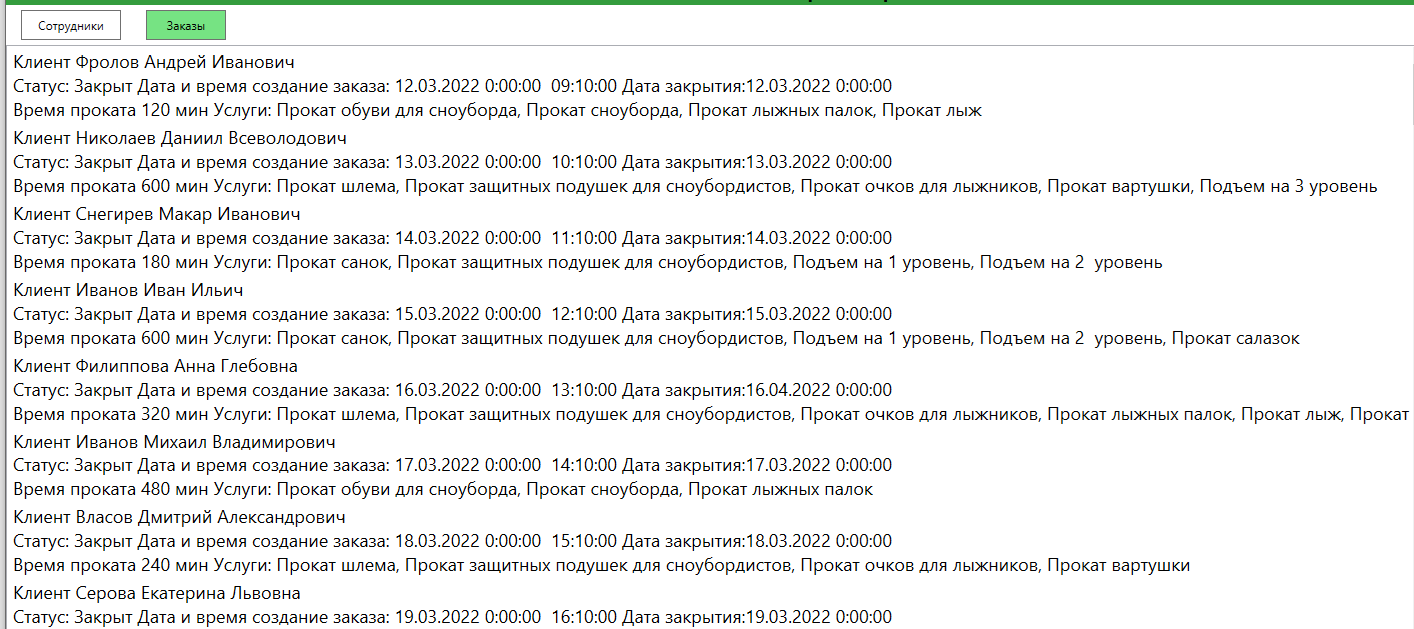


Рисунок 33 - Вкладка «Заказы»

Находясь на вкладке «Заказы» можно увидеть элемент «ListView», позволяющий выводить информацию обо всех заказах в компании «Охта парк» в удобном формате, который содержит самую необходимую информацию об заказе.

Перейдя на вкладку «Сотрудники» можно  
увидеть также элемент «ListView», который отображает всех сотрудников, которые осуществляли вход в систему и итог их попытки входа: успешно или неуспешно.

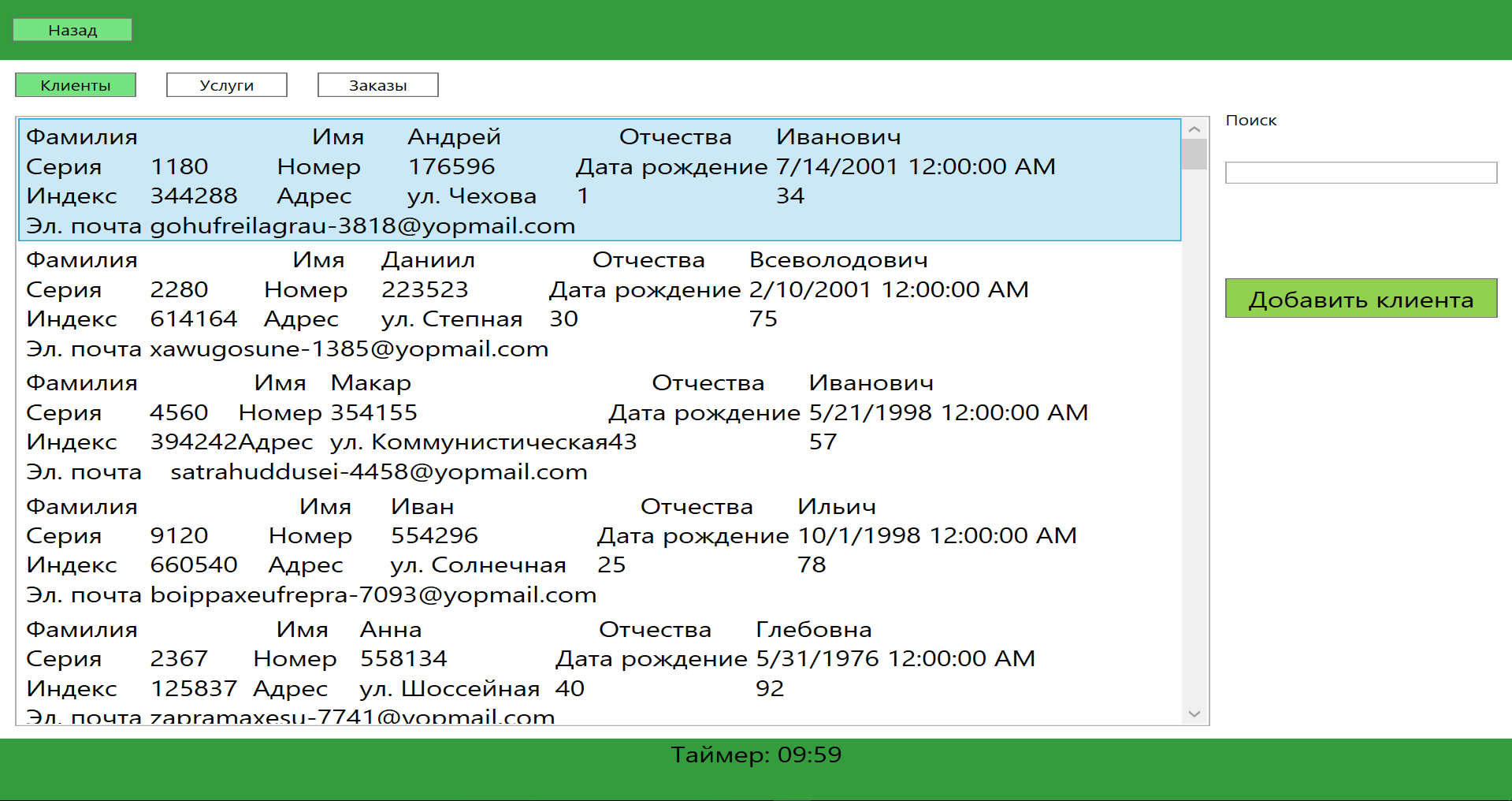


Рисунок 34 - Форма старшего смены и продавца

На форме для продавца и старшего сотрудника представлены следующие элементы: кнопки для вывода клиентов, услуг и заказов, кнопка «Назад» и таймер. В зависимости от выбранной вкладки меняются и некоторые элементы. Для формы с выводом клиентов выводится дополнительно кнопка «Добавить клиента», которая переводит нас на окно добавления клиента.

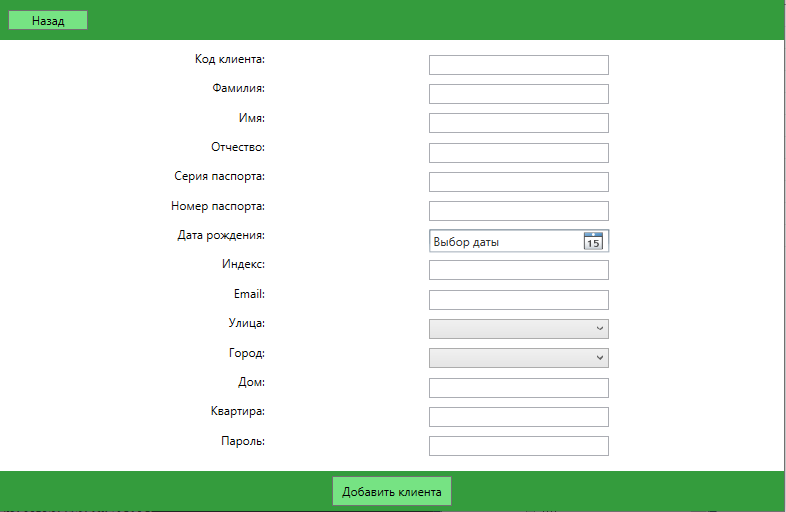


Рисунок 35 – Форма добавления клиента

Данная форма содержит элементы для ввода и выбора основных данных об клиенте, а также кнопку для добавления клиента. Для добавления нового города или улицы необходимо раскрыть список и в самом начале будет находиться кнопка для добавления и текстовое поле для ввода названия города или улицы.

# 9 Функционал программного решения

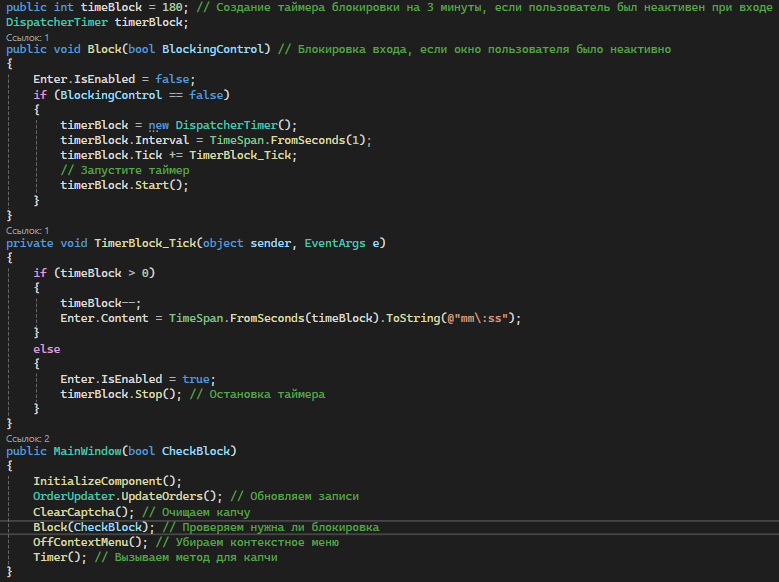


Рисунок 36 – Код блокировки входа при неактивности для окна авторизации

В данном коде необходимо объявить таймер, затем проверить то, что пользователя деавторизировало с аккаунта автоматически, а после назначить время каждого тика длиною в одну секунду, после вызвать событие «Тика», в которой будет использоваться переменная с длиною 180 секунд. После в событие «Тик» происходит вычитание единицы с каждой секундой, пока количество секунд не будет равно нулю. Далее мы инициализируем нужные события.

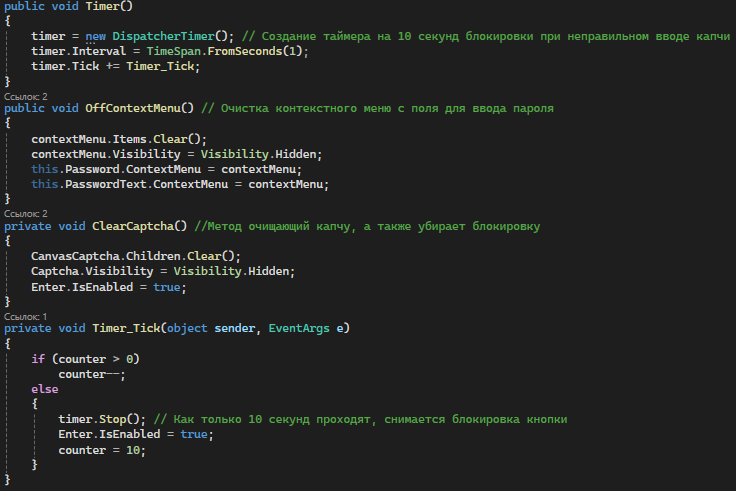


Рисунок 37 – Методы окна авторизации

При некорректном вводе каптчи необходимо осуществлять блокировку входа на 10 секунд, для этого создаётся новый таймер, к которому присваивается событие «Тик» и каждую секунду вычитывает из переменной целого числа единицу, как только число становится равно нулю, блокировка снимается.

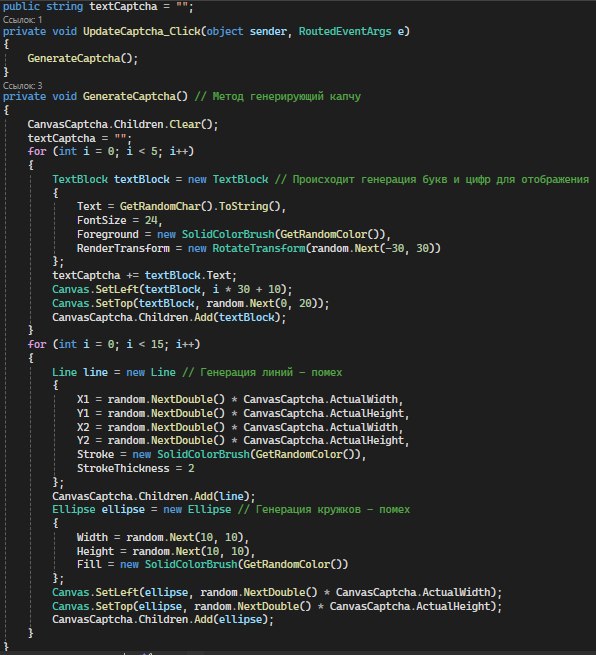


Рисунок 38 – Генерация каптчи

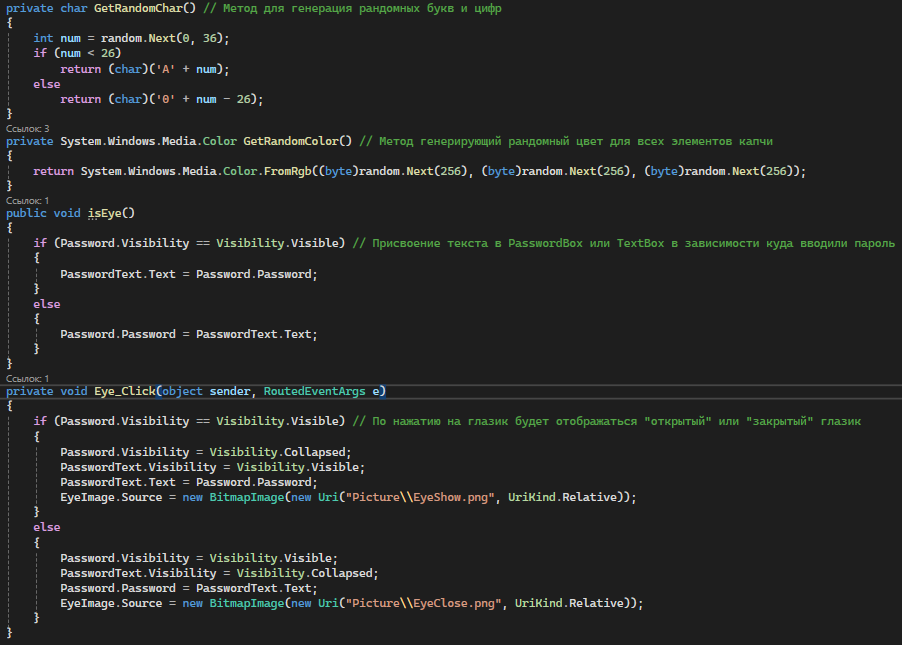


Рисунок 39 – Генерация каптчи и отображение введенного пароля

На рисунках 38 и 39 представлен код, который генерирует каптчу. Для начала нужно очистить то поле куда она будет выводить, после присвоить случайный текст, для этого использовался метод «GetRandomChar», который генерировал случайный набор букв и цифр. Затем нужно отобразить сгенерированные символы, а после создается текстовое поле для каждого символа отдельно, текстовое поле имеет случайный цвет, угол наклона и фиксированный размер, после чего для каждого текстового поля указывается его расположение.

Для реализации графического шума используется рисование кружков и линий, которые также имеют случайный цвет и расположение.

Для реализации возможности просмотра пароля используется специальный символ в виде «глазика», нажатие по которому позволяет передать введенный пароль в скрытое маской поле в обычное текстовое поле, при каждом нажатии происходит скрытие путем присвоения параметру видимости соответствующий значений, а также изменение специального символа.

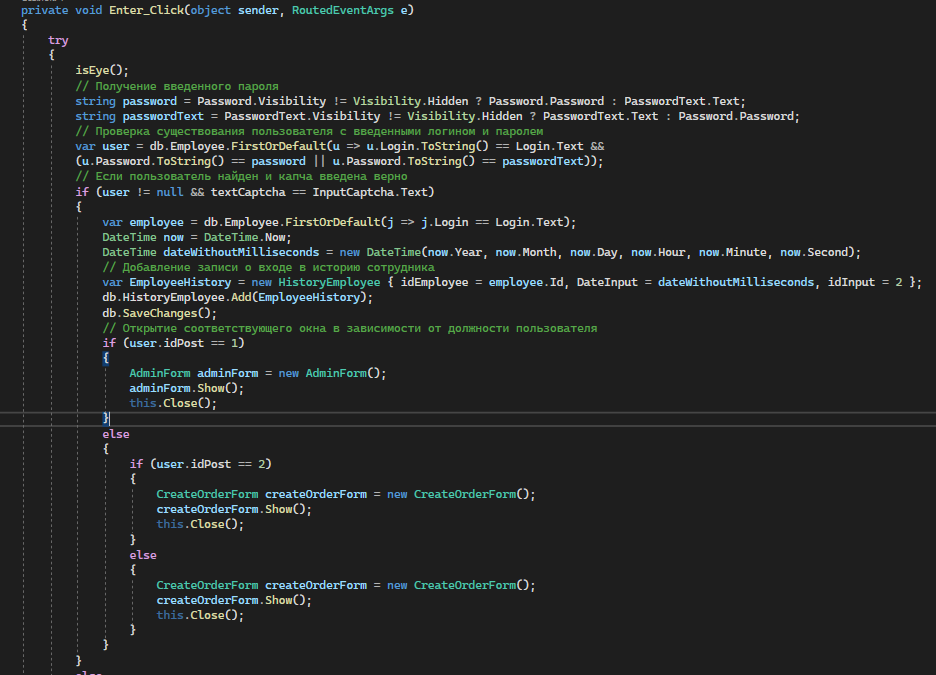


Рисунок 40 – Кнопка входа 1

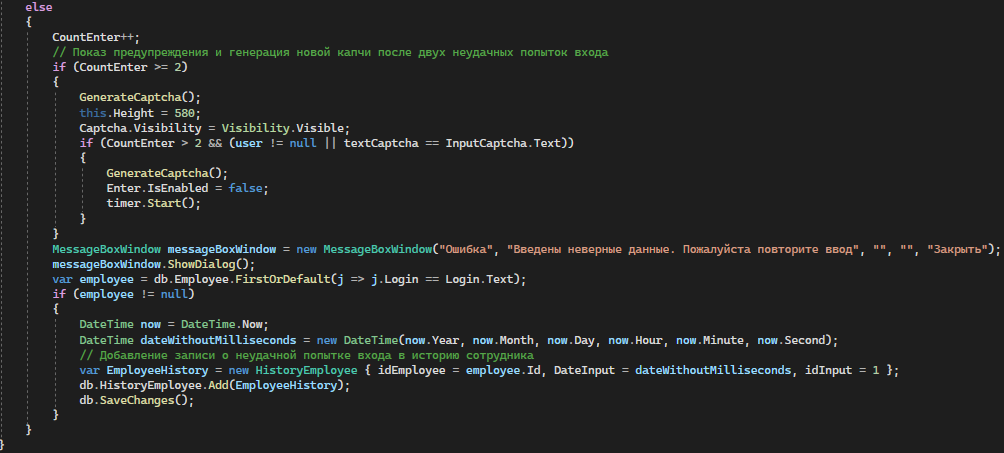


Рисунок 41 – Кнопка входа 2

Осуществление входа происходит путем нажатия на кнопку, в которой происходит присвоения текста в зависимости от того, что открыто для ввода пароля, после переменной присваивается значение верного поля. Затем проверяется наличие данного пользователя в базе данных. Если такой пользователь есть, в базу данных заносятся данные о входе. После чего проверяется должность пользователя и в соответствие с ней, открывается соответствующая форма.

Если логин или пароль были некорректно введены два раза форма расширяется и появляются элементы для каптчи.

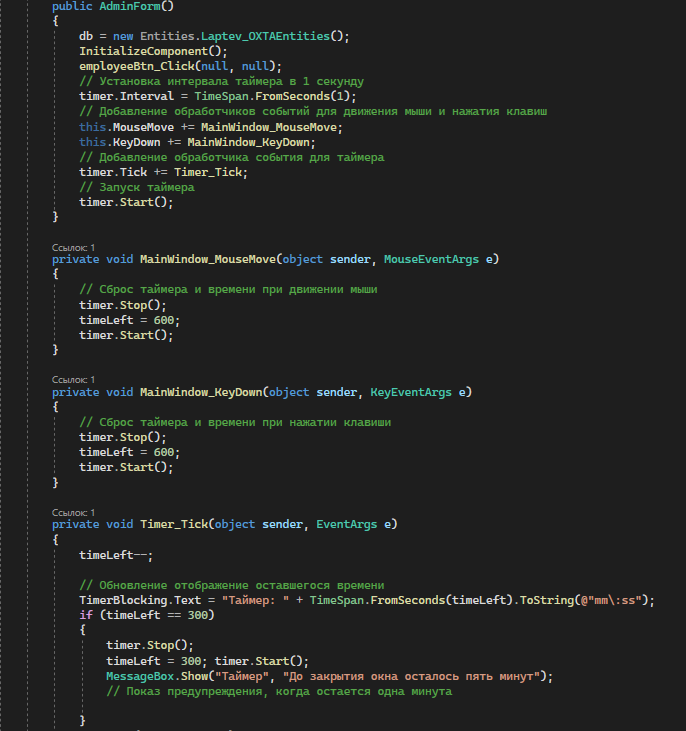


Рисунок 42 – Окно администратора

На показанном рисунке выше (рисунок 42) продемонстрирован код инициализации окна администратора а также методы для работы с таймером.

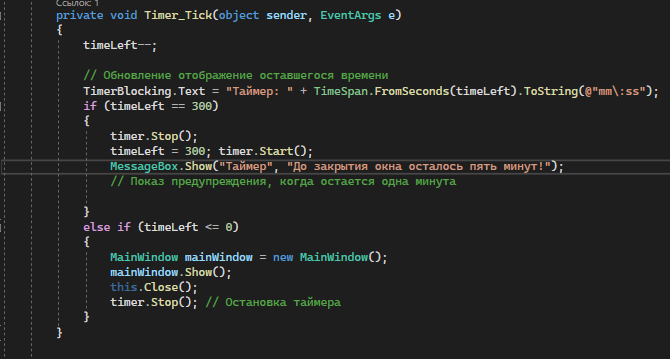


Рисунок 43 – Таймер блокировки при неактивности

На показанном рисунке выше (рисунок 43) продемонстрирован код события «Тика», где по истечению времени происходит переход обратно на форму авторизации.

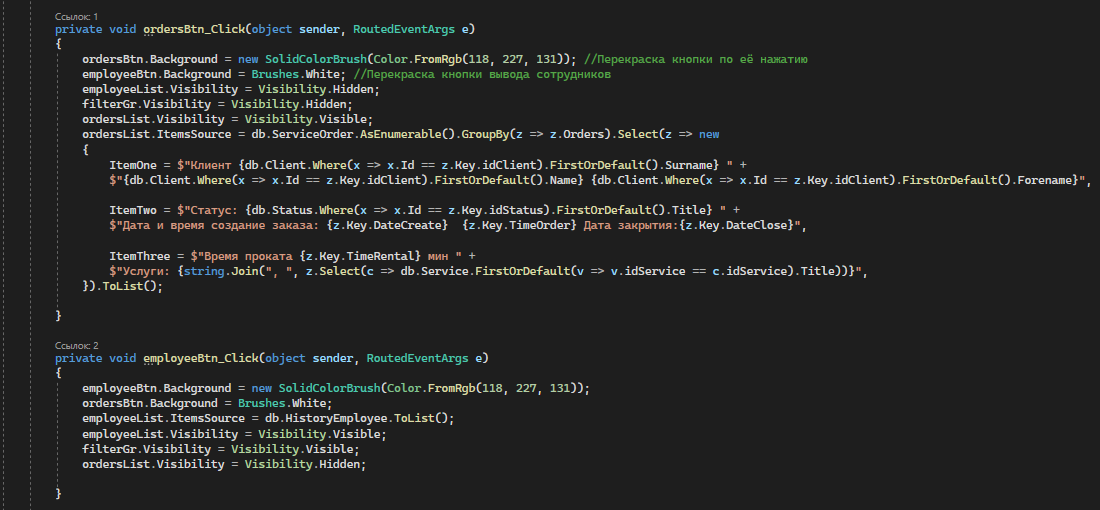


Рисунок 44 - Вывод заказов и сотрудников

В данном фрагменте кода происходит вывод заказов и сотрудников в элемент «ListView». Вывод заказов сделан через переменные «ItemOne» и так далее, для более структурированного отображения.

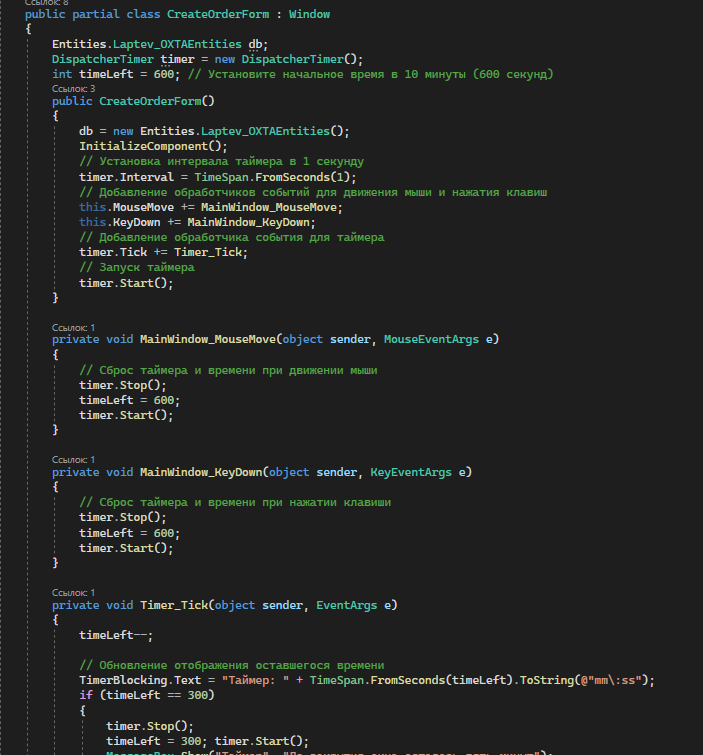


Рисунок 45 – Окно продавца/старшего смены

На окне продавца/старшего смены также происходит объявление таймера для дальнейшей деавторизации при неактивности.

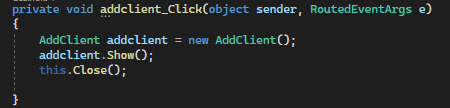


Рисунок 46 – Открытие окна добавления клиента

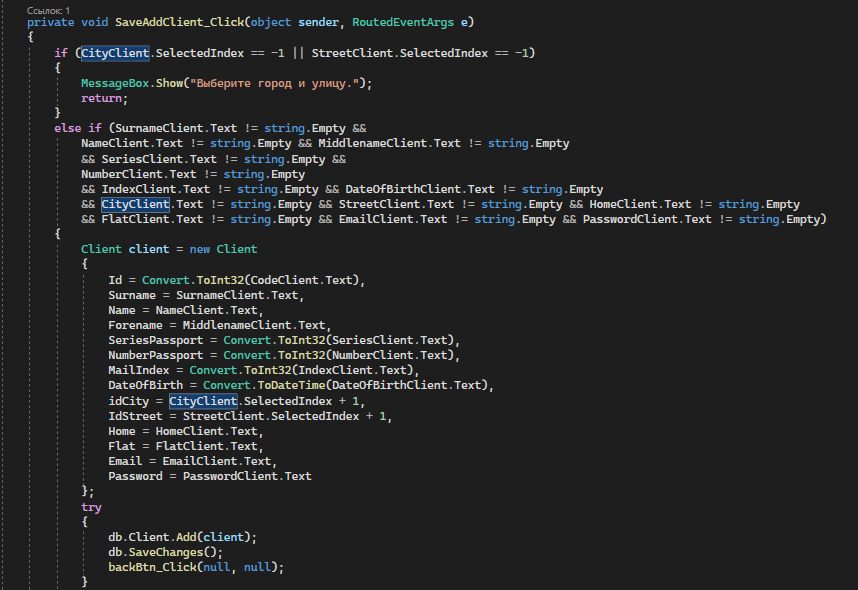


Рисунок 47 - Добавление клиента

В данном фрагменте кода сначала происходит проверка, выбраны ли значения в элементе «Combobox». Если они не выбраны, то появляется сообщение об этом, иначе происходит проверка на заполненность значений. Если значения заполнены и корректно введены, то происходит добавление данных из полей в базу данных и возвращение на окно продавца/старшего смены.

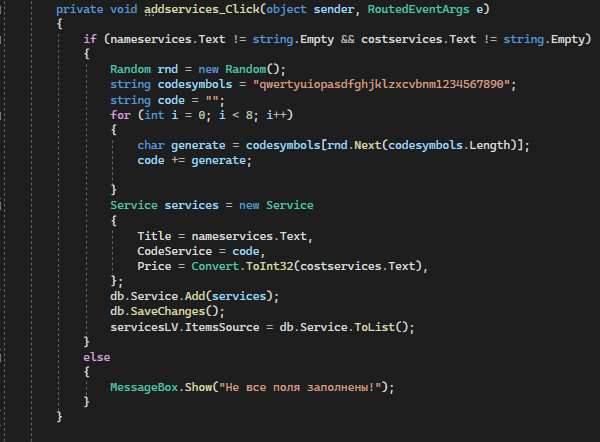


Рисунок 48 – Добавление услуги и генерация кода услуги

На рисунке 48 представлен код для добавления услуги. Так как каждый код имеет свой уникальный номер, то необходимо генерировать его случайным набором символом. После чего происходит проверка того, что поля, в которые вводятся данные самим пользователем не должны быть пустыми, если все хорошо, то происходит добавление данных в соответствующую таблицу в базе данных.

Генерация номера происходит путем присвоения случайных букв и символов, после чего запускается цикл, который будет осуществлять генерацию номера.

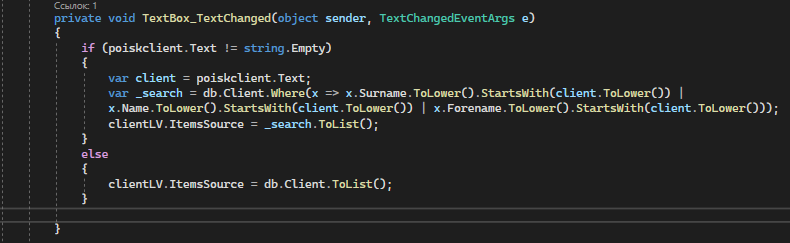


Рисунок 49 – Поиск клиента

Поиск клиента осуществляется по ФИО. Если в поле поиска ввести часть от имени, фамилии или отчества, то будет использоваться метод «StartWith», который ищет значения, начинающиеся с введенного текста.

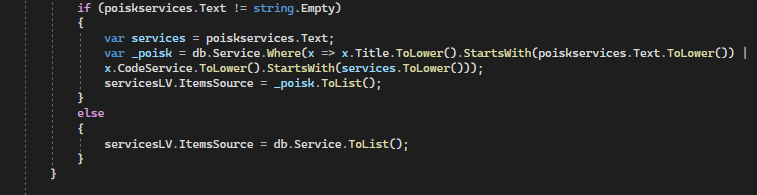


Рисунок 50 – Поиск услуги

Поиск услуги (рисунок 50) аналогичен такому же поиску, как и у поиска клиента.

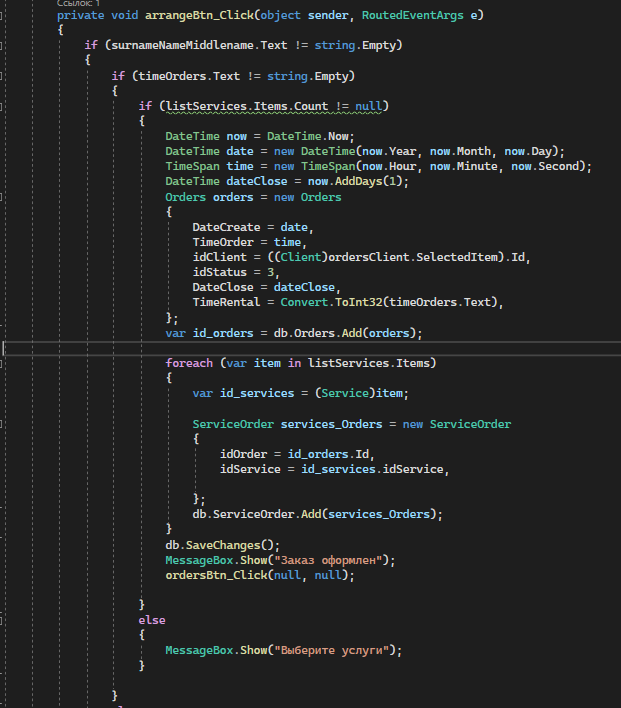


Рисунок 52 – Выбор услуг и клиента

Выбор клиента и услуг осуществляется по нажатию на Item из соответствующего ListView. Если выбран клиент и выбраны нужные услуги, а также заполнено время аренды, осуществляется оформление заказа.

**Приложение А**