Государственное автономное профессиональное образовательное учреждение Саратовской области   
«Новоузенский агротехнологический техникум»

Специальность 09.02.07   
«Информационные системы и  
программирование»  
Курс II  
Группа П-21

**Курсовой проект**

По ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем»

Тема: «Автоматизированное рабочее место диспетчера такси»

Руководитель:  
Бекбулатов Р.Ж.  
Разработал студент:  
Малова Степанида Валерьевна.

г. Новоузенск, 2021

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Теоретические основы создания системы автоматизации деятельности таксопарка |  |
| 1.1 Анализ деятельности таксопарка | 4 |
| 1.2 Выбор жизненного цикла автоматизированной информационной системы | 5 |
| 1.3 Выбор системы управления базами данных | 6 |
| 1.4 Выбор языка программирования | 7 |
| 2 Разработка программы для автоматизации деятельности таксопарка |  |
| 2.1 Анализ требований заказчика | 9 |
| 2.2 Проектирование системы автоматизации деятельности таксопарка | 10 |
| 2.3 Создание программы «Автоматизированное рабочее место диспетчера такси» | 12 |
| 2.4 Создание форм и запросов | 13 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**ВВЕДЕНИЕ**

До появления подобных программ, вся информация о заказах, водителях и другие данные хранилась в бумажном виде и как следствие могла содержать ошибки (неправильно указан номер дома или название улицы, стоимость поездки и т.д.) или просто затеряться. Данное АРМ диспетчера такси имеет возможность внести сведения в базу данных о поездках и водителях и может брать оттуда различные доступные данные.

Программа может найти применение во всех компаниях такси, так как наглядно автоматизирует и облегчает работу диспетчера.

**Цель**: разработка программы автоматизации деятельности диспетчера такси.

**Задачи**:

Исследовать сферу деятельности таксопарка;

Проанализировать существующие аналоги автоматизации деятельности диспетчера такси;

Выбрать СУБД и язык программирования;

Построить структуру СУБД;

Написать программный код;

1. **Теоретические основы создания системы автоматизации деятельности таксопарка**
   1. **Анализ деятельности таксопарка**

Работа таксопарка осуществляется следующим образом: каждый водитель, заступая на смену, связывается с диспетчером, и тот вносит его в карточку работающих в данный момент. Сведения карточки отображаются на экране. Водители держат обратную связь по радио.

Заказы поступают по телефону к диспетчеру, он записывает необходимые данные в базу заказов. Дата и время поступления заказа диспетчер вводится автоматически.

Потом из списка свободных водителей выбирает того, кто будет выполнять заказ (если в машинах встроены маяки выбирает того кто ближе к месту), или водитель сам отвечает по обратной связи, что примет заказ. После подтверждения водителем диспетчер ставит на заказ статус «В пути»

После выполнения заказа водитель по рации отчитывается перед диспетчером, тот ставит заказу статус "Выполнен".

После каждой смены диспетчер формирует отчет, который показывает, сколько заказов поступило, сколько выполнилось и отменено.

В конце месяца администратор, пользуясь отчетами, формирует отчет по итогам месяца и отправляет его высшему руководству таксопарка.

Еще администратор занимается кадрами. Он формирует дела новых сотрудников, и вносит необходимые поправки в дела уже работающих на этом предприятии. На администраторе лежит ответственность за правильное формирование дел и их сохранность.

Проблемы, которые могут возникнуть при осуществлении данной деятельности такие:

Потеря данных диспетчером, или неправильная их трактовка.

Выбор не оптимального маршрута следования таксистов

Не корректная форма представления отчетов и т.д.

Пути решения этих проблем могут быть следующие:

Использование стандартных форм для заполнения данных и ввода/вывода отчетов

Использование систем ГИС и математических формул для определения оптимальных маршрутов из пункта в пункт и их стоимости

Создание при помощи картографических средств перечня районов и улиц, с целью исключения ошибок диспетчера.

* 1. **Выбор жизненного цикла автоматизированной информационной системы**

Жизненный цикл АИС - это непрерывный процесс, начинающийся, с момента принятия решения о создании АИС и заканчивается, в момент его полного изъятия из эксплуатации.

Существует две модели жизненного цикла Автоматизированных систем: каскадная и спиральная модель.

Каскадная модель состоит из следующих этапов: анализ, проектирование, конструирование, тестирование сдача. На каждом этапе оформляется пакет документов и ошибки предыдущего этапа видны только на следующем. Достоинства в том, что надежность таких систем очень высокая, а недостаток то, что очень сложно вносить изменения в функциональность продукта.

Спиральная модель или итерационная состоит из этапов: определение требований и анализ, проектирование, реализация и внедрение. Спиральная модель на каждом своём витке предоставляет новую версию продукта с доработками и возможными функциональными изменениями, то есть добавление новых функций. Достоинства легко вносить изменения, нет избыточной документаций. Схема этапов спиральной модели изображена на рис.1.



**Рис. 1 Спиральная модель**

Для программы "Автоматизированное рабочее место диспетчера такси" была использована спиральная модель жизненного цикла, потому что

в программе есть только базовый набор функций, который можно будет расширить в следующей версии программы

Процессы проектирования и разработки проходят быстрее, чем при использовании каскадной модели

Использования предыдущих итераций для создания новой версии, существенно облегчают созданий

* 1. **Выбор системы управления базами данных**

База данных - это средство накопления и организации больших массивов информации об объектах некоторой предметной области. БД должна отображать текущие данные о предметной области, накапливать, хранить информацию и предоставлять различным категориям пользователей быстрый доступ к данным. Для этого данные в базе должны быть структурированы в соответствии с некоторой моделью, отражающей основные объекты ПО, их свойства и связи между ними.

Все данные в базах находятся в таблицах и разделены по смысловой нагрузке. Каждая таблица для описания содержимого атрибутов использует типы данных (домены), домены Microsoft SQL Server подходят для описания данных любой предметной области.

MS SQL Server это настольная, реляционная система управления базами данных от корпорации Microsoft. Благодаря тому, что она разработана Microsoft, она может легко создавать связь с другими приложениями. Ее средствами можно создавать различные запросы, отчеты и формы для заполнения.

* 1. **Выбор языка программирования**

Для создания программы "Автоматизированное рабочее место диспетчера такси" был использован язык программирования C#, потому что:

C#, объектно-ориентированный язык программирования, то есть состоит из набора объектов, каждый из которых обладает определенными параметрами и может выполнять ряд конкретных функций.

Выбор компонентов C# можно расширять, а также самому описывать методы, типы данных, объекты и действия производимые в них.

Главный плюс высокоуровневых языков программирования это визуализация объектов, но при этом ход действия производимых этими объектами скрыты. Еще одна особенность: один объект может использоваться помимо своего основного назначения в других функциях, а также на один и тот же объект можно поставить несколько событий.

Средствами C# легко создать связь с СУБД Access через технологию SqlClient, что имеет ряд своих преимуществ, таких как:

Возможность вручную создавать запросы средствами C#

Отображать необходимые атрибуты базы данных в интерфейсе программного продукта и создание удобных карточек заполнения атрибутов БД: простота создания связей и их модернизация.

1. **Разработка программы для автоматизации деятельности таксопарка**

* 1. **Анализ требований заказчика**

Программа Автоматизированное рабочее место диспетчера такси разработана по спиральной модели жизненного цикла автоматизированных информационных систем. На каждом этапе создания были проведены следующие действия:

определение требований.

В соответствии с техническим заданием были определены основные требования и методы реализации этих требований.

Основные требования к программе:

Вход по специальному паролю, для каждого диспетчера он свой;

Заполнение карточки водителя с указанием ФИО и подробного описания автомобиля;

Печать сводных отчетов по диспетчерам и заказам за любой срок;

Заполнение карточки заказа;

Изменение статуса заказа;

Печать отчета за смену;

На этапе анализа были проведены:

Анализ предметной области, где были выявлены необходимые функции для работы таксопарка.

Анализ аналогов, где были проанализированы достоинства и недостатки продуктов, применяемых в данной предметной области. Пример интерфейса.

* 1. **Проектирование системы автоматизации деятельности таксопарка**

Диспетчер напрямую работает с клиентами, в его обязанности входит регистрация заказов с последующей передачей их водителям. После каждой смены диспетчер создает и помещает общую базу отчетов.

База данных data\_taxi состоит из следующих сущностей (таблиц): "Drivers", "Admins", "Sessions" и "Orders".

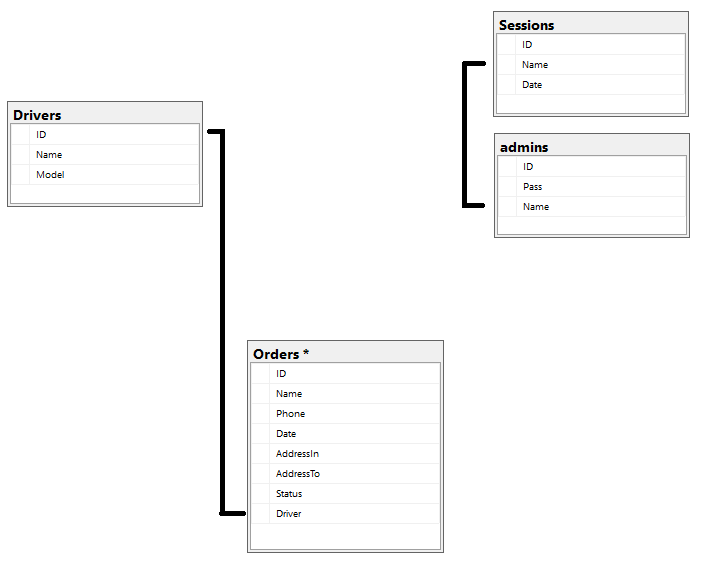
Схема связей базы данных data\_taxi, показывает какие сущности соединены между собой и по каким атрибутам: 

Схема базы данных Taxi

Каждая таблица необходима для выполнения конкретных функций.

В таблицу "Orders" заносятся все сведения о поступивших заказах: какой водитель будет исполнять заказ, ФИО и контактный телефон абонента, места отправления и назначения, статус.

Заказы

|  |  |  |
| --- | --- | --- |
| наименование атрибута | тип данных | описание |
| ID | счетчик | уникальный номер заказ |
| Name | Varchar | ФИО клиента |
| Phone | Varchar | Номер телефона клиента |
| Date | Datetime | Дата и время заказа |
| AddressIn | Varchar | Адрес клиента |
| AddressTo | Varchar | Место назначения |
| Status | Integer | Статус заказа |
| Driver | Integer | ID водителя, принявшего заказ |

Таблицы "Водители" и "Диспетчеры" выполняют схожие функции - это хранение данных о сотрудниках, работающих в данном таксопарке, естественно данные различаются по специфике выполняемых операций. Например, у водителей существуют поля, в которых находятся сведения о транспортном средстве, а у диспетчеров наличие поля "Password", так как оно необходимы для входа в систему. Атрибуты таблиц "Drivers", "Admins"и их домены представлены ниже:

Водители:

|  |  |  |
| --- | --- | --- |
| наименование атрибута | тип данных | описание |
| ID | счётчик | Уникальный номер водителя |
| Name | текстовый | ФИО водителя |
| Model | текстовый | Марка, модель, цвет и гос. номер автомобиля |

Диспетчеры:

|  |  |  |
| --- | --- | --- |
| наименование атрибута | тип данных | описание |
| ID | счётчик | Уникальный номер диспетчера |
| Name | Varchar | ФИО диспетчера |
| Pass | Varchar | Пароль для входа в систему |

* 1. **Создание программы "Автоматизированное рабочее место диспетчера такси"**

Основная связь используемая в программе - это связь с базой данных по средствам технологии SqlClient. Основное преимущество подобной связи в том, что через связь установленную в SQLConnection средствами SQLCommand и SQLAdapter, можно вносить изменения в базу данных, и получать из нее необходимые данные, а так же проводить сравнивания данных и прочие операции.

Связь с базой осуществляется при помощи многоступенчатой связи. Многоступенчатая связь с базой технологией ADO выглядит следующим образом:

Компонент SQLConnection устанавливает связь приложения с конкретной базой, адрес которой прописывается в строке Connection String, это первая ступень. К компоненту SQLConnection присоединяются один или несколько компонентов SQLAdapter и SQLCommand, это зависит от количества атрибутов базы, которые нужно отобразить в приложении с последующим внесением изменений в них. Это компоненты второй ступени. Компонент SQLAdapter устанавливает связь с конкретной таблицей, подсоединенной базы. Компонент SQLCommand необходим для создания запросов, сравнения данных в двух различных атрибутов одной базы, либо вставка, выборка и другие действия реляционной алгебры. Третьей ступенью являются компоненты из вкладки data Access, в частности Data Source, как более часто используемый при создании приложений. Этот компонент своего рода служит переходником от компонентов второй ступени к компонентам четвертой ступени. Компоненты четвертой системы - это компоненты отображающие содержимое базы на форме приложения. Они выглядят и называются как компоненты находящиеся в вкладке Standart, только с приставной DB, что указывает на их взаимосвязь с базой. Это такие компоненты как: DataGridView, ComboBox и TextBox. Помимо отображения содержимого атрибутов базы данных, с их помощью можно вставлять, редактировать, удалять данные из базы.

Для печати отчетов используется технология PrintDocument. Благодаря установлению этой связи можно напрямую распечатать таблицу из приложения, так же в папке с исполняемым приложением создаётся png-файл, который можно будет сохранить на своём локальном диске.

Помимо компонентов связи с базой данных использовались

компоненты:

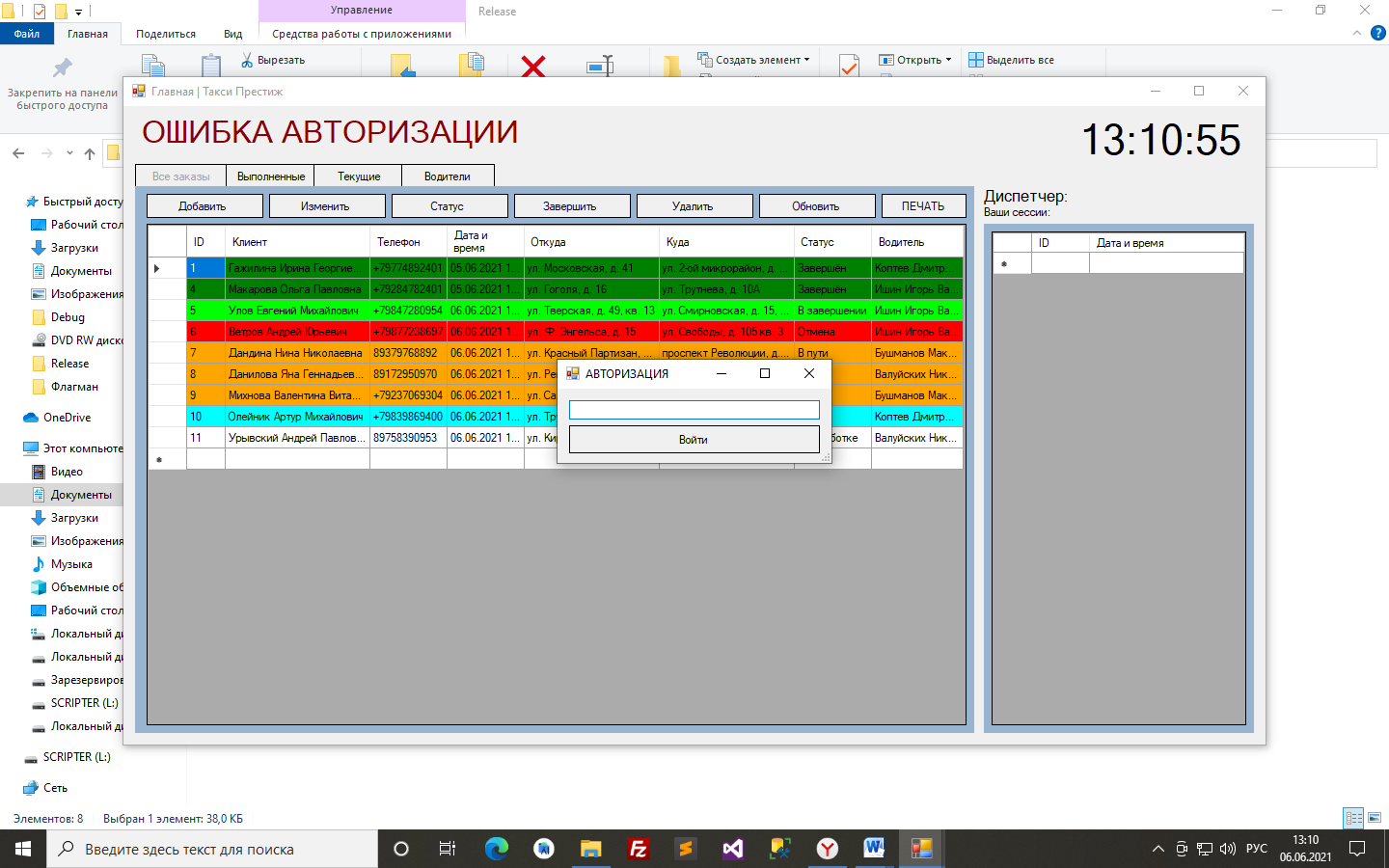
Button: Все кнопки выполняющие действия закрытия, занесения данных в базу, переход между формами.

Label: Все надписи, которые служат подсказками или появляющейся информацией, преимущество - пользователь не может изменить содержание Label.

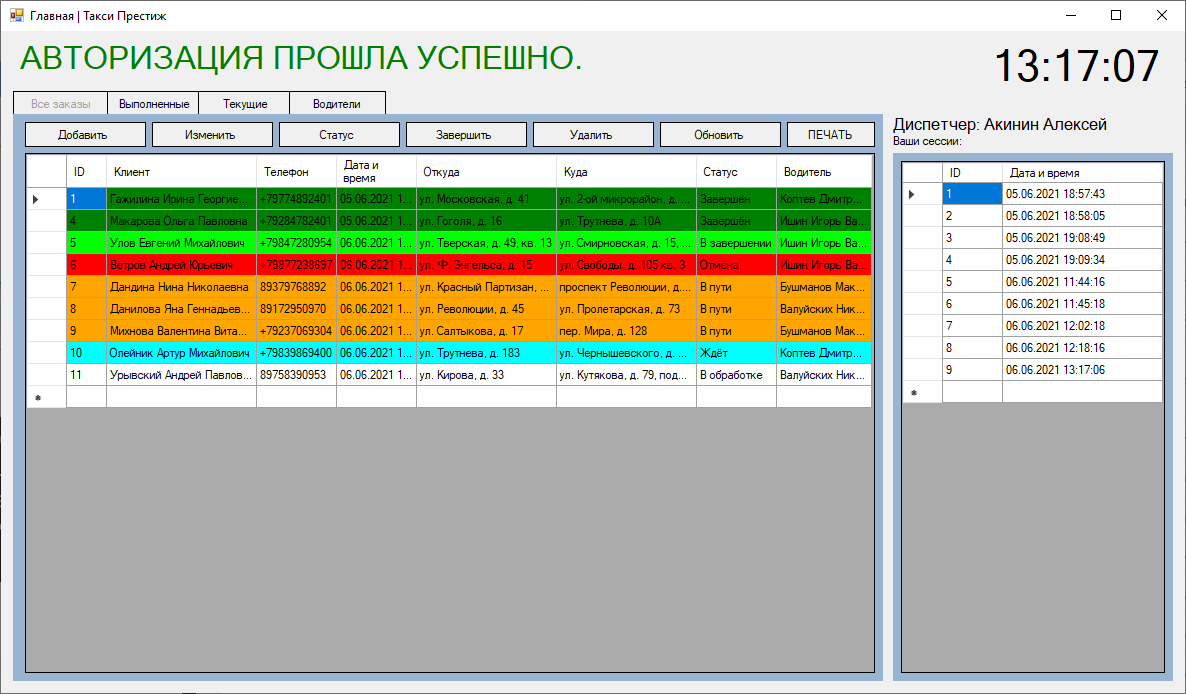
Combobox: выпадающий список.

* 1. **Создание форм и запросов**

При запуске программы нас встречает главная форма, и сразу же появляется окно авторизации:



Вводим личный пароль диспетчера, и начинаем работу:



Любой заказ можно:

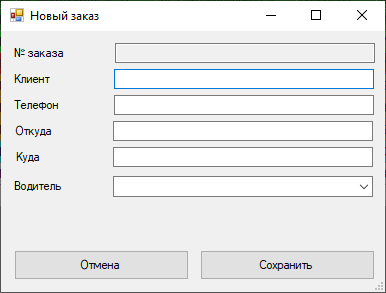
* Добавить**;**
* Редактировать;
* Изменить статус;
* Завершить;
* Удалить;
* Обновить.

Имеется меню для быстрого перехода между таблицами:

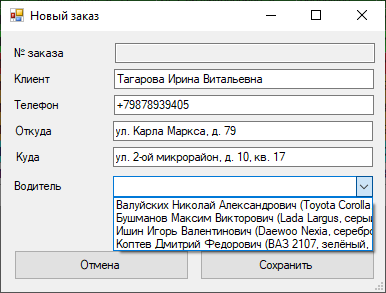


Данные кнопки могут отобразить все заказы, выполненные и текущие заказы, а также всех водителей, сохранённых в базе данных.

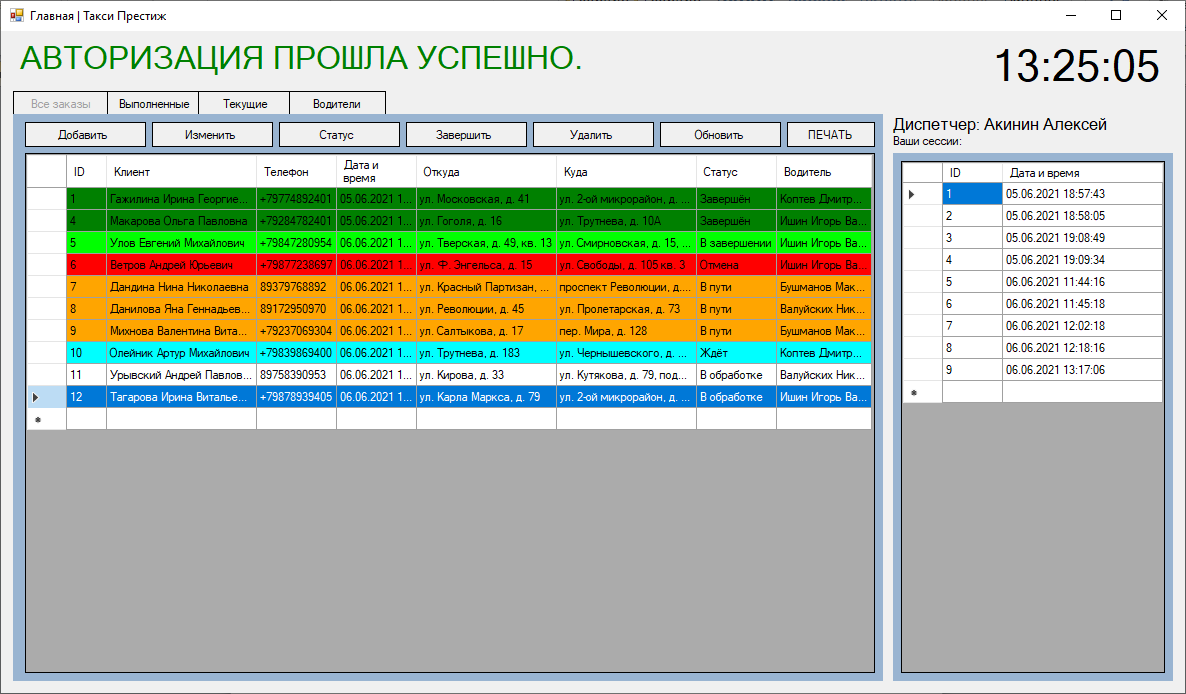
Для регистрации нового заказа жмём «Добавить»:



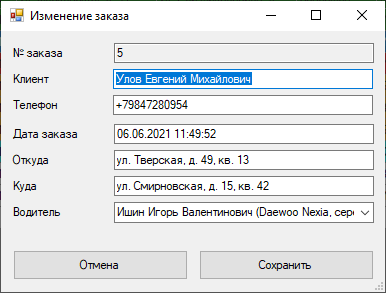
Вводим необходимые данные, и жмём «Сохранить»:



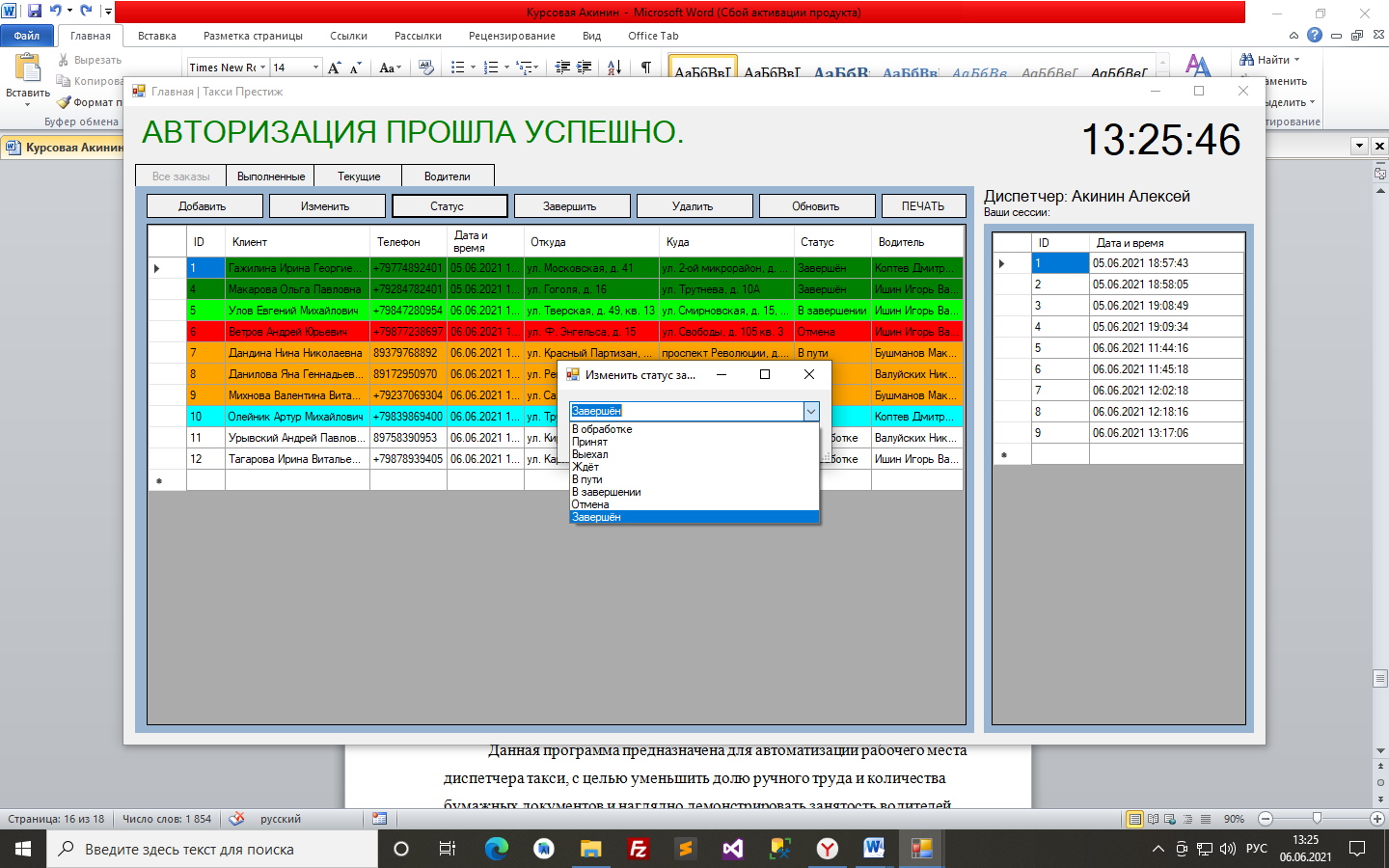
Как мы видим, сразу после закрытия данного диалога таблица обновилась с учётом введённых новых данных:



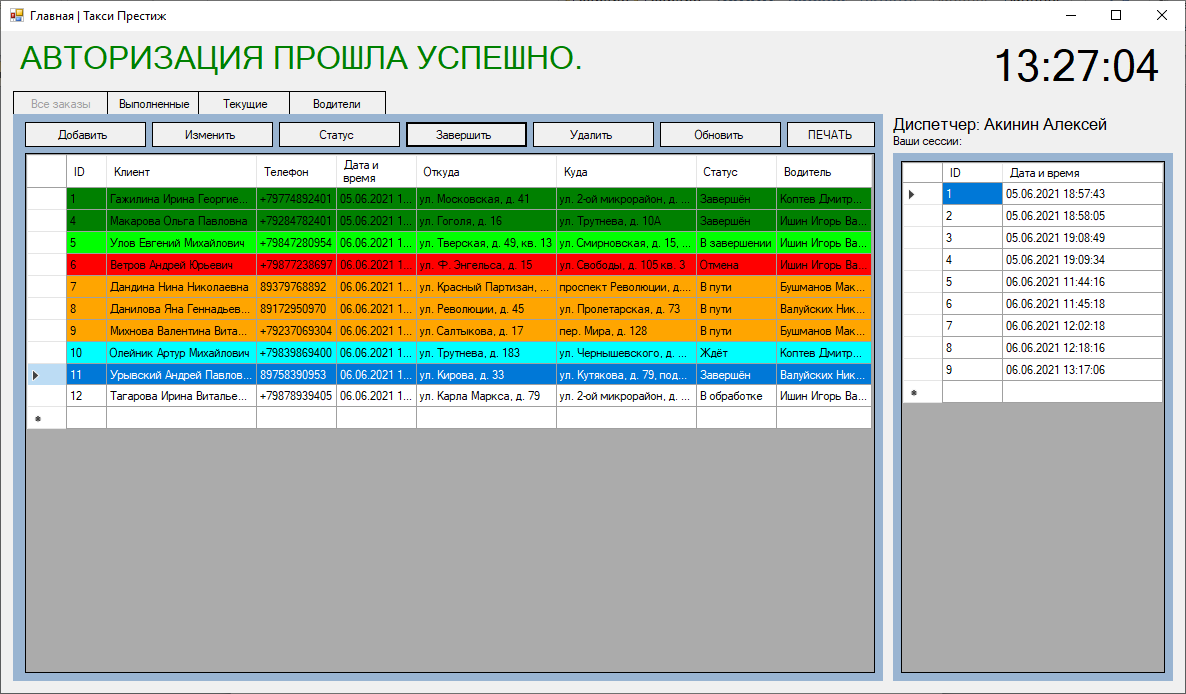
Любой заказ можно отредактировать, предварительно выбрав его в таблице, и нажав на кнопку «Редактировать»:



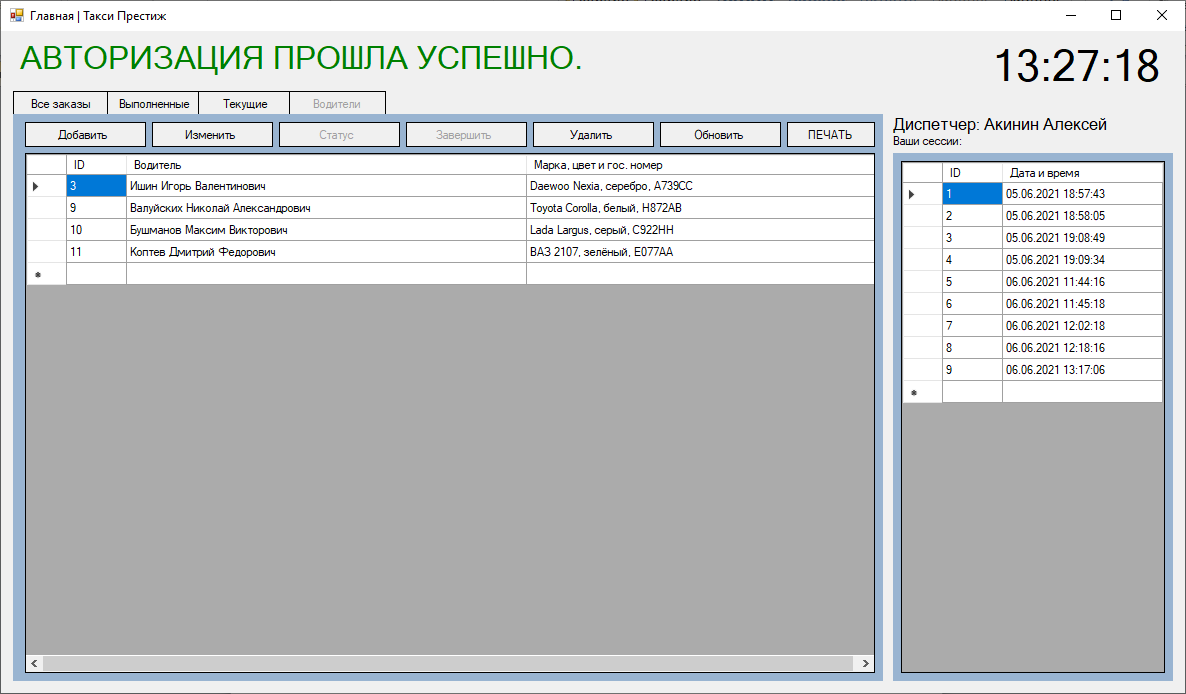
Каждый заказ имеет свой статус, который можно изменить с помощью кнопки «Статус»:



Если заказ завершён, необязательно изменять его статус, для этого можно нажать кнопку «Завершить» в меню таблицы:

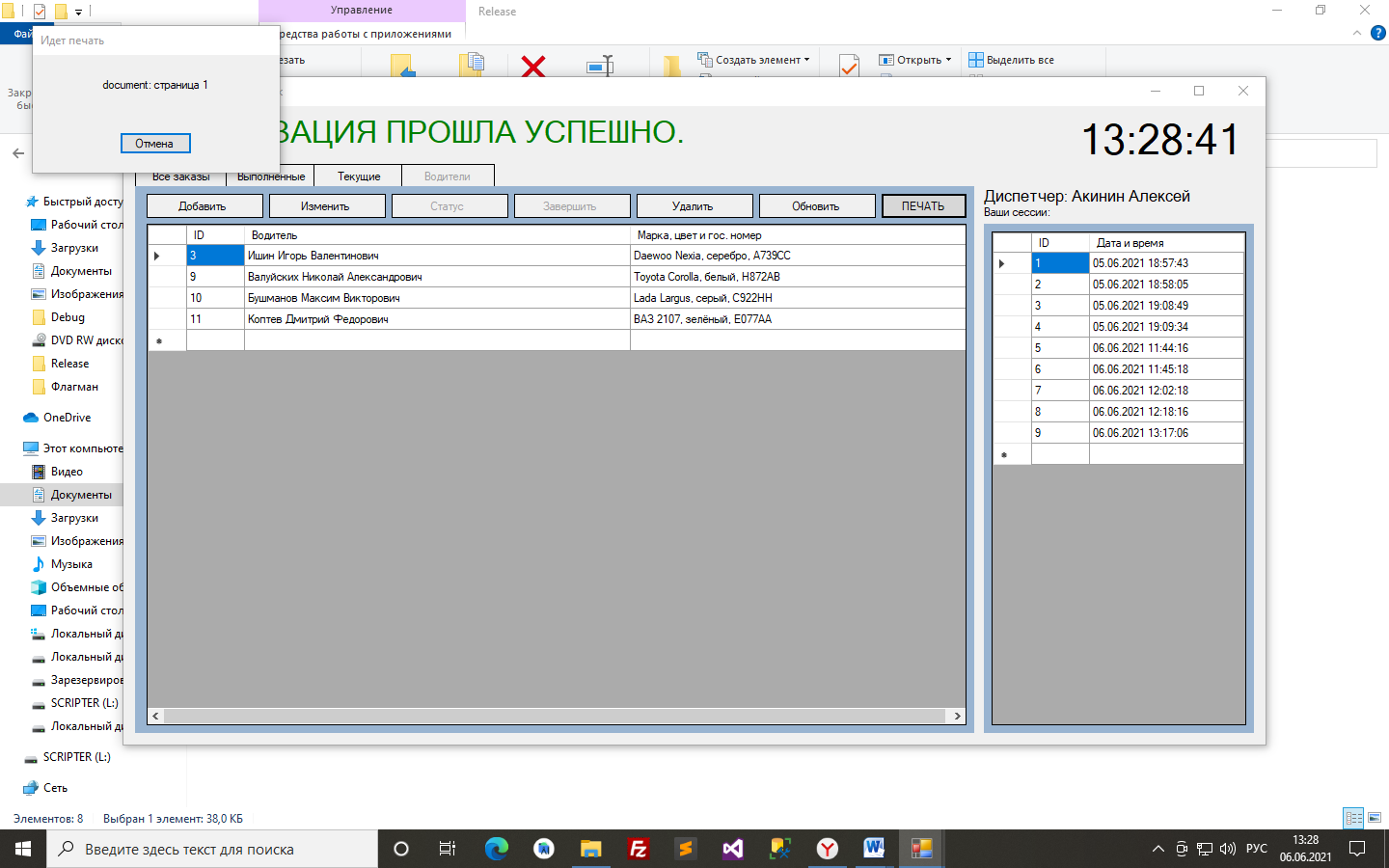


Таблицу водителей так же можно редактировать, удалять и добавлять:

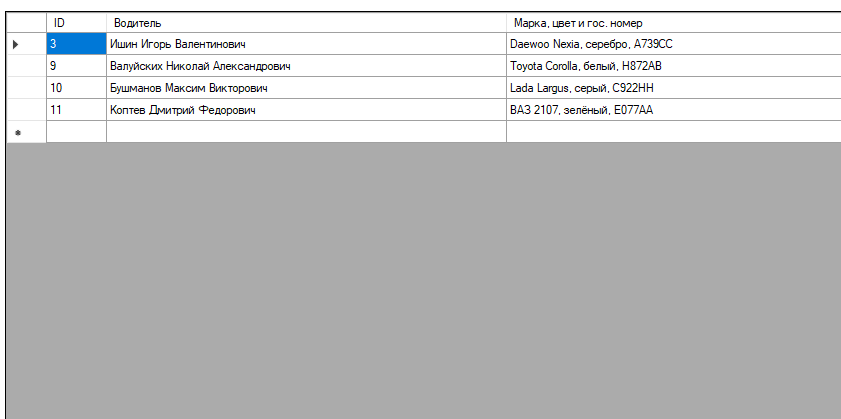


* 1. **Создание отчётов**

Для создания отчёта достаточно нажать кнопку «Печать» в меню, и приложение автоматически начнёт печать выбранной таблицы, а также создаст файл last\_print.png в папке с приложением:



last\_print.png:



**ЗАКЛЮЧЕНИЕ**

Данная программа предназначена для автоматизации рабочего места диспетчера такси, с целью уменьшить долю ручного труда и количества бумажных документов и наглядно демонстрировать занятость водителей.

Программный продукт "Автоматизированное рабочее место диспетчера такси" выполняет поставленные задачи:

Регистрация новых водителей и заказов;

Печать отчетов по заказам клиентов;

Изменение данных таблиц;

При входе на рабочем месте появляется имя и фамилия работающего диспетчера, а так же его последние 20 сессий;

Прием заказов.

Недостатки: Очень большую роль играет "Человеческий фактор". При халатном введении базы данных, могут забить значения не существующих улиц, именами водителей и клиентов в виде непонятного набора букв («шыыпывл») и т.д.

**Список используемой литературы**

* Тельнов Ю. Т Реинжиниринг бизнес процессов / Москва. "Финансы и Статистика" 2004.
* Калашян А. Н., Калясов Г. Н Структурное моделирование бизнеса: DFD-технология/ Москва. "Финансы и Статистика" 2001.
* Фаронов, Delphi 7. Программирование баз данных.
* Дарахвелидзе П., Марков Е. Программирование в Delphi 7/Санкт-Петербург "БХВ-Петербург 2003;
* Филонович С.Р., Использование моделей жизненного цикла в организации и диагностике /Москва 2005;
* Батлер Э., Microsoft Office Access 2007: профессиональное программирование / Вильямс 2009;
* Кириллов В.В., Громов Г.Ю., Введение в реляционные базы данных /Москва BHV 2009;
* Автоматизированное рабочее место в системе управления предприятием, Сборник научных трудов, СПб, 2002 г;
* Гофман В., Хомоненко А., Работа с базами данных в Delphi/ Санкт-Петербург "БХВ-Петербург 2003;
* Коннолли Т., Бегг К., Базы данных. Проектирование, реализация и сопровождение / Москва, Санкт-Петербург, Киев 2003;
* Кантарь И.Л. Автоматизированные рабочие места управленческого аппарата, 1990.
* Рапопорт Э.Я. Структурное моделирование объектов и систем управления. М: финансы и статистика, 2003 г.

**ПРИЛОЖЕНИЕ**

**Инструкция по установке:**

* Скачиваем архив с приложением, открываем проект через Visual Studio (желательно версия 2015 года);
* Переходим в настройки (Обозреватель решения > Properties > Settings) и изменяем строку stringConnect, а именно: ALEX\SQL на название своего SQL сервера, data\_taxi на название своей базы данных (лучше оставить название прежним);
* Компилируем проект, открываем исполняемый файл и начинаем пользоваться.

**Листинг программы.**

Главная форма

