Содержание

1. Введение
2. Основная часть
   1. Характеристика технических средств для решения задачи
   2. Характеристика программных средств для решения задачи
3. Проектная часть
   1. Описание предметной области/Постановка задачи

3.1 Техническое задание

3.2 Разработка структуры приложения

3.3 Разработка концептуальной модели базы данных

3.4 Дизайн приложения

3.5 Разработка основных и графических модулей приложения

3.6 Основные алгоритмы приложения

3.7 Безопасность данных

3.8 Тестирование и анализ результатов

3.9 Руководство пользователя

4. Охрана труда

4.1 Заключение

4.2 Список использованных источников

Введение

В какой период развития человечества и общества в целом берут свое начало информационные технологии? Если говорить без точной даты, то основопалагающим будет цифра «20», означающая двадцатый век нашей эры. Более точной датой начала информационых технологий является

1 Характеристика технических средств для решения поставленной задачи

Технические средства являются неотъемлемой и наиболее существенной составляющей информационной технологии, выполняя ту же роль, что и средства производства в трудовой деятельности.

Компьютер - совокупность электронных компонентов, механических деталей и программного обеспечения. Его называют электронно-вычислительная машина (ЭВМ) или персональный компьютер (ПК). Работает ПК по строгому алгоритму, заложенному в программном обеспечении.

Компьютер - это инструмент, который помогает человеку быстро решать задачи в работе или повседневной жизни**.** Он используется для поиска информации, для работы с текстами и таблицами, для игр и развлечений, для покупок, для общения и хранения информации.

Основными техническими характеристиками компьютера являются: объем жесткого диска, тактовая частота процессора и объем оперативной памяти.

Приложение разрабатывалось на компьютере со следующими характеристиками:

* видеокарта: Nvidia GeForce 1050 ti;
* процессор: Intel Core i3-7100;
* материнская плата: Asus H110M-K;
* блок питания: DeepCool PK750D 750Вт;
* оперативная память: Kingston 16 Гб (1x);
* жесткий диск: WD Blue 1000 ГБ.

Процессором называется устройство, электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера.

Оперативная память – это энергозависимая память ПК, обладающая высокой скоростью чтения/записи по сравнению с ПЗУ.

Основное назначение оперативной памяти временное хранение данных, к которым можно получить быстрый доступ: код программы, кэш, промежуточные вычисления, текущие параметры операционной системы, настройки драйверов и т.д.

Видеокарта - устройство, преобразующее графический образ, хранящийся как содержимое памяти компьютера в форму, пригодную для дальнейшего вывода на экран монитора.

Материнская плата - это сложная многослойная печатная плата, на которой устанавливаются основные компоненты персонального компьютера (центральный процессор, ОЗУ, загрузочное ПЗУ, контроллеры базовых интерфейсов ввода-вывода).

Блок питания - устройство, предназначенное для формирования напряжения, необходимого системе, из напряжения электрической сети.

Жесткий диск - это запоминающее устройство для хранения ваших файлов и данных в течение длительного времени.

# 2 Характеристика программных средств для решения поставленной задачи

Программные средства - это компьютерные (машинные) программы, представленные на языке программирования или в машинном коде описания действий, которые должна выполнить электронно-вычислительная машина в соответствии с алгоритмом решения конкретной задачи или группы задач.

Для решения поставленной задачи использовались следующие программные средства:

* продукты Microsoft Office (офисный пакет приложений для работы с различными типами документов: текстами, электронными таблицами, базами данных и др.), основными из которых является Word, PowerPoint, Excel;
* MySQL - свободная реляционная система управления базами данных (СУБД) корпорации Oracle. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных;
* Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, поддерживаемых Windows, Windows Mobile, Windows CE, NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.
* Microsoft Visual Basic — язык программирования «Visual Basic» относится к категории императивных. Написанные на нем программы представляют собой строгую последовательность команд (приказов — отсюда и название «императивный»), причем каждая последующая использует данные, полученные при исполнении предыдущей. Кроме того, Visual Basic объединяет в себе элементы сразу нескольких парадигм (методологий) программирования.

3 Разработка программного обеспечения

3.1 Постановка задачи

Создать средствами СУБД MySQL базу данных «Железнодорожный вокзал». Для создания базы данных требуется:

* произвести анализ предметной области;
* выбрать структуру таблиц;
* определить внешние и первичные ключи;
* определить схему (связи) базы данных;
* ввести данные в таблицы.

Установить связи между таблицами можно как в СУБД, так и в процессе разработки прикладной программы.

Разработать прикладную программу на Visual Basic в среде разработки приложений Visual Studio, которая будет служить интерфейсом для работы с созданной базой данных и реализовать следующие возможности:

* просмотр данных из таблиц;
* редактирование данных в таблицах;
* добавление новых данных;
* удаление данных;
* осуществление поиска (фильтрация данных).

Редактирование данных и заполнение новыми данными должны происходить в отдельных формах (окнах программы).

3.2 Описание предметное области

Основной целью железнодорожного воказала является составление системы учета расписания и поездов, для упрощения их отслеживания и создание функции учета пассажиров, за счет цифрового способа создания билетов.

Основными объектами и сущностями базы данных «Железнодородный вокзал» являюся:

* Рейсы: служат для отслеживания текущих поездок
* Пассажиры: для создания билетов и учета пассажиров в рейсе, в случае какой-либо чрезвычайной ситуации или нарушения
* Расписание: включает в себя предыдущие два пункта, за счет которых можно осуществлять учет поездов, соблюдение таймингов прибытия / отбытия и уведомления самих пассажиров.

Таблицы, которые являются составляющими базой данных «Железнодорожный вокзал», представлены ниже.

Таблица «Пассажиры» содержит информацию, связанную с клиентами железнодорожной компании, ее структура представлена в таблице 1.

Таблица 1 - Структура таблицы «Пассажиры»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | | Тип поля | | Описание поля |
| ID | | Int(11) | | Первичный ключ. Индентификатор записи |
| Имя | | varchar(255) | | Первичный ключ. Имя пассажира |
| Фамилия | | varchar(255) | | Первичный ключ. Фамилия пассажира |
| Отчество | | varchar(255) | | Первичный ключ. Отчетсво пассажира |
| Дата\_рождения | | date | | Первичный ключ. Дата рождения пассажира |
| Номер\_паспорта | | Int(6) | | Первичный ключ. Номер паспорта пассажира, для его индентификации при посадке и актуализации аккаунта |
| Продолжение таблицы 1 | | | | |
| Номер\_паспорта | | Int(6) | | Первичный ключ. Номер паспорта пассажира, для его индентификации при посадке и актуализации аккаунта | |

Таблица «Поезда» содержит подробную информацию о поезде. Структура таблицы представлена в таблице 2.

Таблица 2 - Cтруктура таблицы «Поезда»

|  |  |  |
| --- | --- | --- |
| ID | Int(11) | Первичный ключ. Индентификатор записи |
| Номер\_поезда | Int(11) | Первичный ключ. Личный номер поезда |
| Общее\_кол-во\_вагонов | Int(2) | Общее кол-во вагонов, которые закреплены за этим поездом |
| Кол-во\_вагонов\_типа\_Купе | Int(2) | Количество вагонов типа «Купе» |
| Кол-во\_вагонов\_типа\_Плацкарт | Int(2) | Количество вагонов типа «Плацкарт» |
| Кол-во\_вагонов\_типа\_СВ | Int(2) | Количество вагонов типа «СВ» |
| Кол-во\_вагонов\_типа\_Люкс | Int(2) | Количество вагонов типа «Люкс» |
| Изготовитель | Varchar(255) | Изготовитель вагонов и локомотива |
| Вместимость | Int(3) | Общая вместимость пассажиров, включая персонал. |

Таблица «Расписание» содержит в себе точное расписание каждого маршрута, по которому ориентируется пассажиры и машинисты. Структура таблицы представлена в таблице 3.

Таблица 3 - Cтруктура таблицы «Расписание»

|  |  |  |
| --- | --- | --- |
| Код | Int(11) | Первичный ключ. Индентификатор записи |
| Номер\_поезда | Int(11) | Внешний ключ. Личный номер поезда |
| До | Varchar(255) | Место прибытия пассажира |
| От | Varchar(255) | Место отбытия пассажира |
| Прибытие | datetimed | Дата и время прибытия поезда |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| Отправление | datetime | Дата и время отбытия поезда |
| Дата\_составления | date | Дата внесения изменения или составления расписания |
| Вместимость | Int(3) | Общая вместимость данного рейса |

Таблица «Рейсы» необходима для отображения всей необходимой информации для пассажиров, работников вокзала и соотношения прибытия/отбытия с расписанием. Структура таблицы представлена в таблице 4.

Таблица 4 - Cтруктура таблицы «Рейсы»

|  |  |  |
| --- | --- | --- |
| ID | Int(11) | Первичный ключ. Индентификатор записи |
| Номер\_поезда | Int(11) | Внешний ключ. Личный номер поезда |
| От | Varchar(255) | Место отбытия пассажира |
| До | Varchar(255) | Место прибытия пассажира |
| Отправление | datetime | Дата и время отбытия поезда |
| Прибытие | datetime | Дата и время прибытия поезда |

Таблица «Билеты» необходима для ведения отчетности по проданным билетам, выборки необходимых данных и составления электронных билетов. Структура таблицы представлена в таблице 5.

Также, данная таблица нужна для общего отслеживания и проверки пассажиров в рейсе, в случае какой-либо непредвиденной ситуации или возникшей проблеме.

Таблица 5 - Cтруктура таблицы «Билеты»

|  |  |  |
| --- | --- | --- |
| Код\_билета | Int(11) | Первичный ключ. Индентификатор записи |
| Номер\_поезда | Int(11) | Внешний ключ. Номер поезда |
| Имя | Varchar(255) | Внешний ключ. Имя пассажира |
| Фамилия | Varchar(255) | Внешний ключ. Фамииля пассажира |
| Отчество | Varchar(255) | Внешний ключ. Отчество пассажира |
| Продолжение таблицы 5 |  |  |
| Отправление | datetime | Дата и время отбытия поезда |
| От | Varchar(255) | Место отбытия пассажира |
| До | Varchar(255) | Место прибытия пассажира |
| Прибытие | datetime | Дата и время прибытия поезда |
| Номер\_вагона | Varchar(255) | Номер вагона, в котором распологается место пассажира |
| Тип\_вагона | Varchar(100) | Тип вагона, который выбрал пассажир |
| Номер\_места | Int(3) | Номер места пассажира, в вагоне |
| Стоимость\_билета | Int(6) | Стоимость билета пассажира, исходя из выбранного места, типа вагона и дальности пути |

Схема базы данных включает в себя описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных. Постоянные данные в среде базы данных включают в себя схему и базу данных.

На рисунке 1 представлена схема данных.

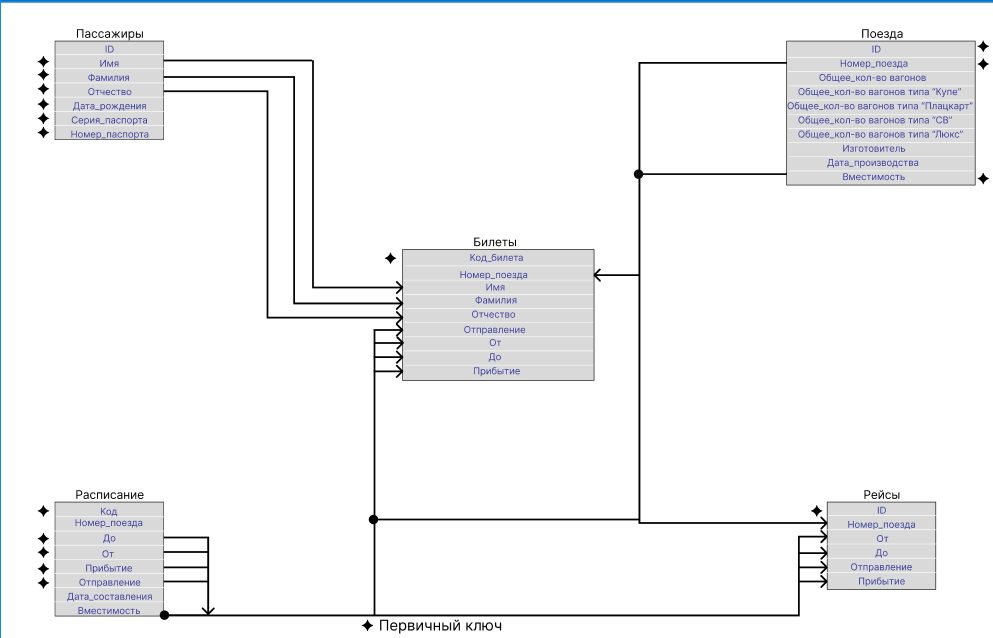


Рисунок 1 - Схема данных

3.3 Взаимосвязь программных модулей

Программный модуль — программа или отдельная ее функциональная часть, рассматриваемая как единое целое в контекстах хранения, замены, трансляции, объединения с другими программными модулями и ее загрузки в оперативную память ЭВМ.

Примером программного модуля может служить DLL - библиотека, которая определенным образом обрабатывает запрос других подсистем и возвращает в ответ полученное значение. Модули формируют структуру программного продукта, позволяют избежать повторяющихся участков кода, делают размер приложения меньше, а его работу - быстрее. Модульное программирование — это методология разработки программного обеспечения, основанная на создании отдельных модулей, которые можно независимо разрабатывать, тестировать и поддерживать. Каждый модуль выполняет конкретную функцию и имеет определённые входные и выходные данные.

Взаимосвязь программных модулей DLL обеспечивает модульность, повторное использование кода и поддержку крупных проектов, разделенных на логические части. Основные аспекты включают:

Экспорт и импорт функций: DLL предоставляет функции и процедуры, которые могут быть использованы другими модулями или основной программой. Экспорт функций означает, что DLL делает эти функции доступными для вызова из других модулей. Импорт функций предполагает использование экспортированных функций в других частях программы. Это осуществляется с помощью деклараций функций и указанием соответствующих библиотек.

Интерфейсы и контракты: Взаимодействие между DLL и основными модулями обычно организовано через интерфейсы и контракты, которые определяют, какие функции и данные будут предоставлены для использования. Это позволяет модулям взаимодействовать друг с другом независимо от их внутренней реализации, обеспечивая гибкость и расширяемость системы.

Загрузка и инициализация DLL: DLL могут загружаться статически или динамически. Статическая загрузка означает, что ссылки на DLL устанавливаются на этапе компиляции, и библиотека загружается при запуске программы. Динамическая загрузка выполняется во время выполнения с использованием функций, таких как LoadLibrary и GetProcAddress, что позволяет загружать библиотеки при необходимости и освобождать ресурсы после завершения их использования.

Обмен данными и память: DLL могут обмениваться данными с основной программой через параметры функций, глобальные переменные или структуры данных. Важно управлять памятью и ресурсами правильно, чтобы избежать утечек памяти и конфликтов. Передача сложных структур данных может потребовать сериализации или использования указателей и соглашений об управлении памятью.

В разработанном приложении были созданы следующие модули:

AddRecRice.vb - форма для добавления записей в форму, отвечающая за рейсы.

AddRecSche.vb - форма для добавления записей в форму, отвечающая за расписание.

AddRecTicke.vb - форма для добавления записей в форму, отвечающая за билеты.

ChangeRecRice.vb - форма для изменения записей в форме, отвечающая за рейсы.

ChangeRecSche.vb - форма для изменения записей в форме, отвечающая за расписание.

ChangeRecTicke.vb - форма для изменения записей в форме, отвечающая за билеты.

Rice.vb - основная форма, отвечающая за таблицу «Рейсы».

Tickets2.vb - основная форма, отвечающая за таблицу «Билеты».

Schedule.vb - основная форма, отвечающая за таблицу «Расписание».

Main.vb - главная форма, которая является основная для взаимодействия с программной и базой данной в целом, также содержащая статус подключения к базе данных.

3.4 Описание программного модуля

Блок структурного программирования представляет собой парадигму разработки программного обеспечения, которая подчеркивает использование четко определенных блоков кода, таких как последовательности, условные операторы и циклы, для достижения логики программы. Каждый блок в этой парадигме следует строгой иерархии, избегая использования операторов перехода (например, GOTO), что способствует улучшению читаемости и понимания кода. Основной целью структурного программирования является упрощение отладки и сопровождения программ путем разбиения их на небольшие, легко управляемые и предсказуемые модули. Таким образом, программы, созданные с использованием структурного подхода, становятся более надежными и менее подверженными ошибкам.

Любой модуль с припиской «AddRec» отвечает за добавление данных, которые относятся к данной области. На форме можно добавлять записи, просматривать текущие и очищать поля, необходимые для ввода данных. Вся информация заносится в соответствующую таблицу и сохраняется в базе данных.

В любом модуле «AddRec» имеется несколько sql запросов, за счет которых обеспечивается добавление данных и конвертация данных типа VB в тип, необходимый языку MySQL. Также, в форме имеется функционал заполнения полей для ввода данных, основанные на существующих записях и работающая при нажатии на существующую запись.

Любой модуль с припиской «ChangeRec» отвечает за изменение существубщих данных, которые относятся к данной области. На форме можно изменять записи, просматривать текущие и очищать поля, необходимые для ввода данных. Вся информация изменяется в соответствующей таблице и обновлятся при закрытии формы в основной таблице.

В осноных формах таблиц, из которых осуществляется переход на формы изменения и удаления, осуществляется функция удаления существующих записей.

Блок-схемы работы алгоритма удаления, добавления и изменения записей на формах представлены на рисунках 2,3,4.



Рисунок 2 - Блок схема алгоритма «Удаление записи»

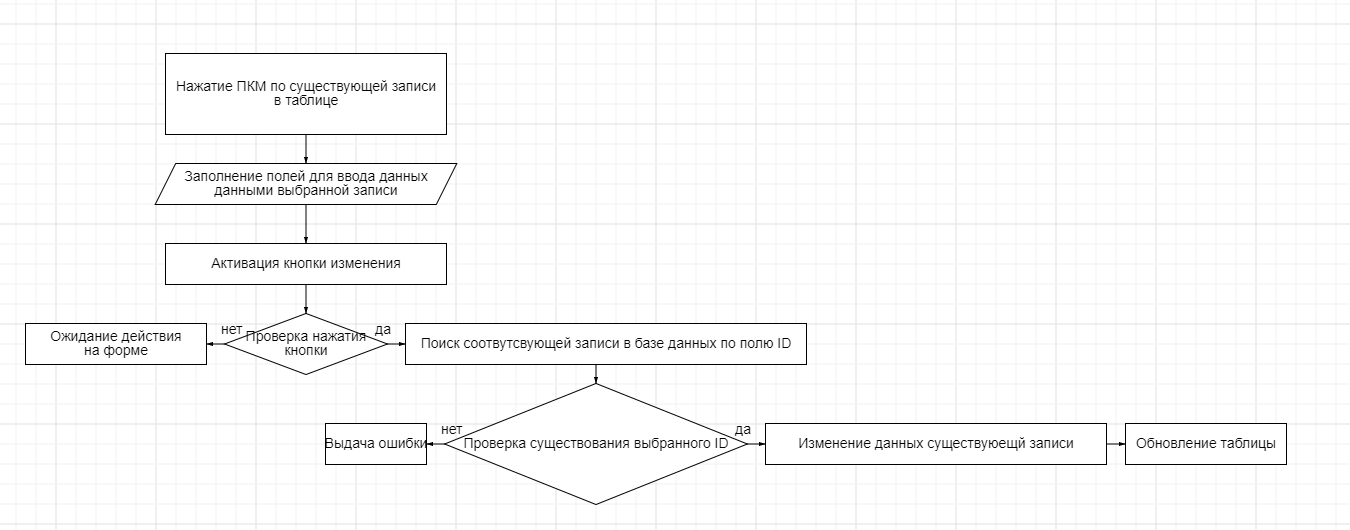


Рисунок 3 - Блок схема алгоритма «Изменение данных»

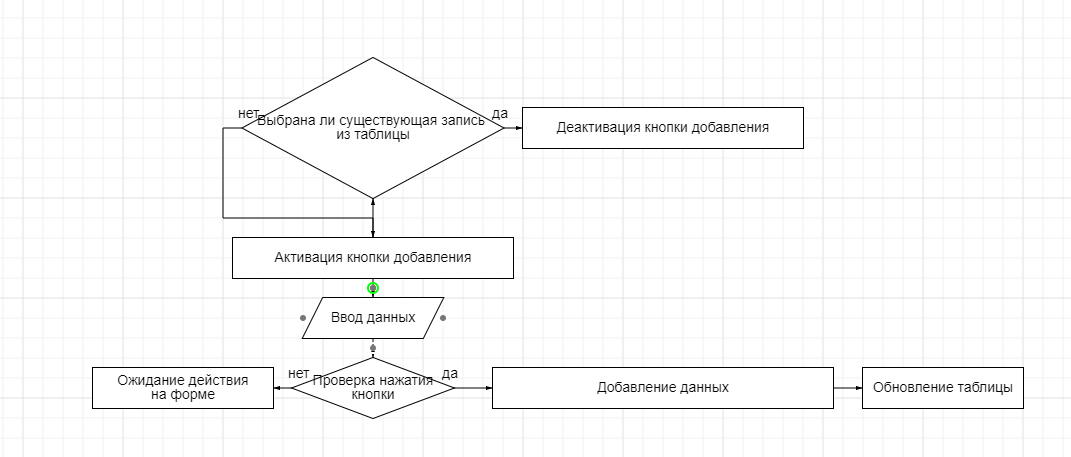


Рисунок 4 - Блок схема алгоритма «Добавление данных»

3.5 Руководство пользователя

Руководство пользователя - это выдержка из полной документации, предназначенная для эксплуатации программы. Она представляет собой независимый документ для пользователя программы, в котором описывается: что делает программа и как им пользоваться.

«Руководство пользователя» должно содержать всю необходимую для пользователя информацию и должна быть ему понятна без дополнительных материалов (без обращения к другим спецификациям). Следовательно, необходимая для этой инструкции информация переписывается полностью из соответствующих спецификаций.

Чтобы запустить программу пользователь должен запустить файл с названием Kursovaya.exe, находящемуся по пути: kursovaya/kursovaya.sln. Вид файла представлен на рисунке 6.



Рисунок 6 - Приложение

При запуске приложения открывается главная форма, на которой представленны основные элементы управления и кнопки, для перехода на другие формы. Также, главная форма позволяет осуществлять поиск и отбор данных, исходя из запросов пассажира, который пришел на железнодорожный вокзал Интерфейс формы представлен на рисунке 7.

На рисунке 8 предствлены две таблицы «Расписание» и «Рейсы», отвечающие за соответсвующие таблицы в базе данных.

На рисунке 9 и 10 показаны элементы управления comboBox, отвечающие за выборку записей в таблице «Расписание» по городам.

При нажатии на данные элементы управления, осуществится открытие списка, в котором будут города, доступные для выборки. После выбора интересующих направлений нужно нажать на кнопку «Найти», чтобы осуществить выборку в таблице «Расписание». Соответсвующая выборка представленна на рисунке 11.

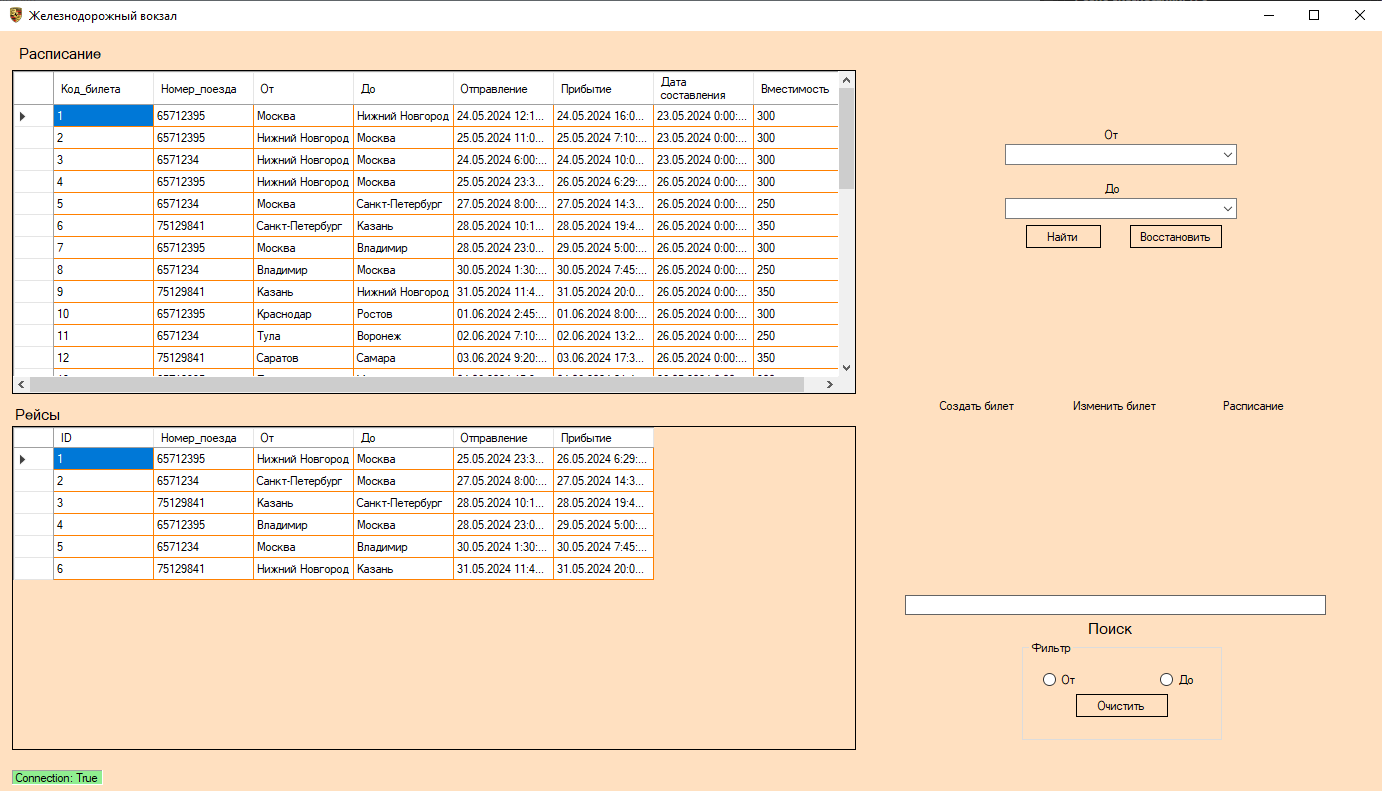


Рисунок 7 - Главная форма

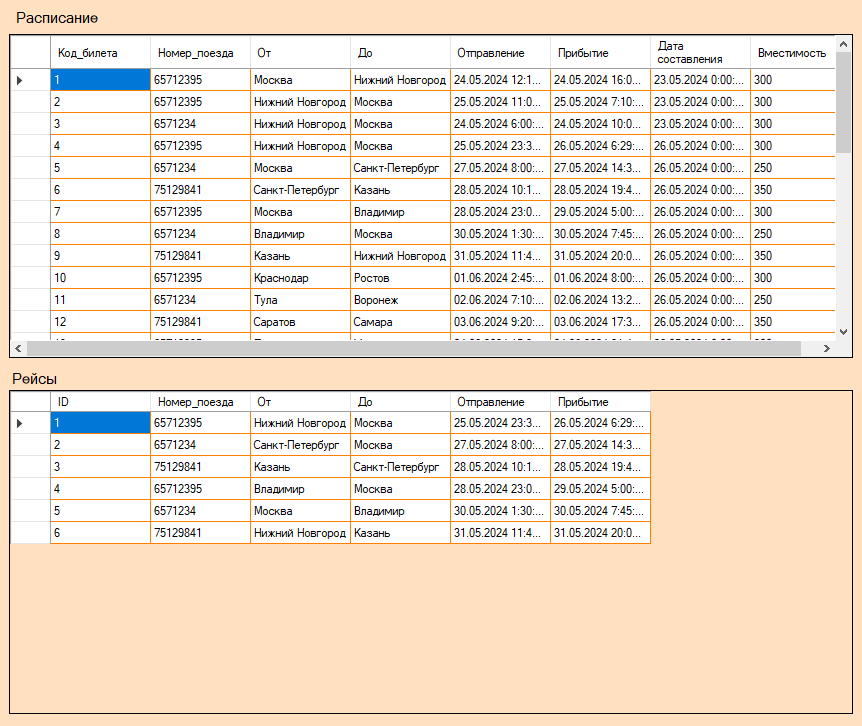


Рисунок 8 - Основные таблицы

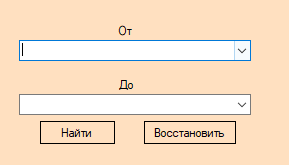


Рисунок 9 - Элементы управления comboBox

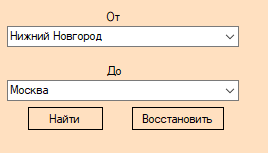


Рисунок 10 - Заполненные элементы управления comboBox

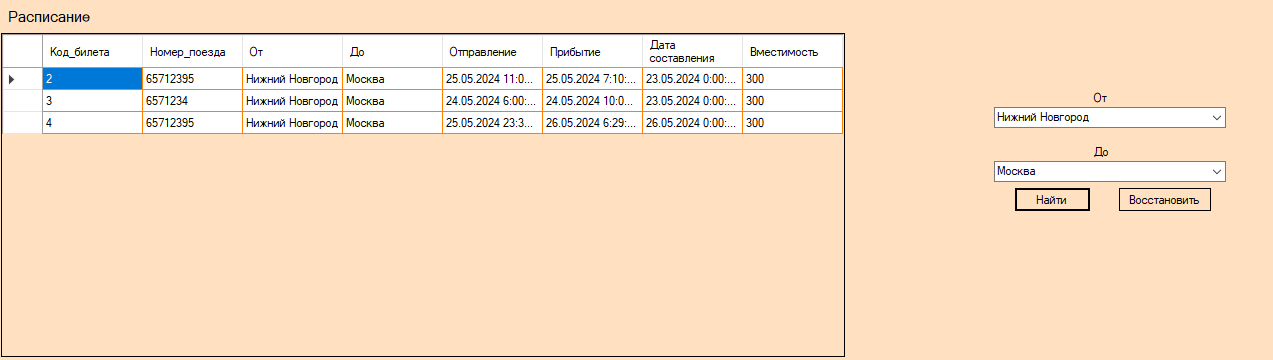


Рисунок 11 - Выборка записей в таблице «Расписание»

На рисунке 12 показаны элементы управления, отвечающие за поиск данных в таблицах «Рейсы» и «Расписание».

Фильтр на данном рисунке предназначен для работы алгоритма поиска, который отбирает записи по заданным критериям. Для полноценной работы алгоритма нужно осуществить нажатие одной из кнопкой, внутри фильтра и начать ввод текста в поле «Поиск». Пример работы показан на рисунке 13.

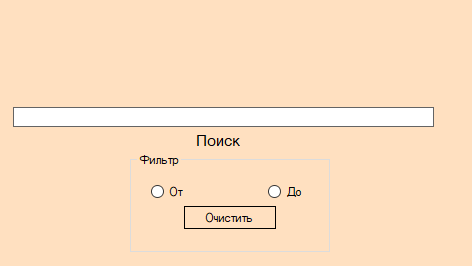


Рисунок 12 - Поиск в главной форме

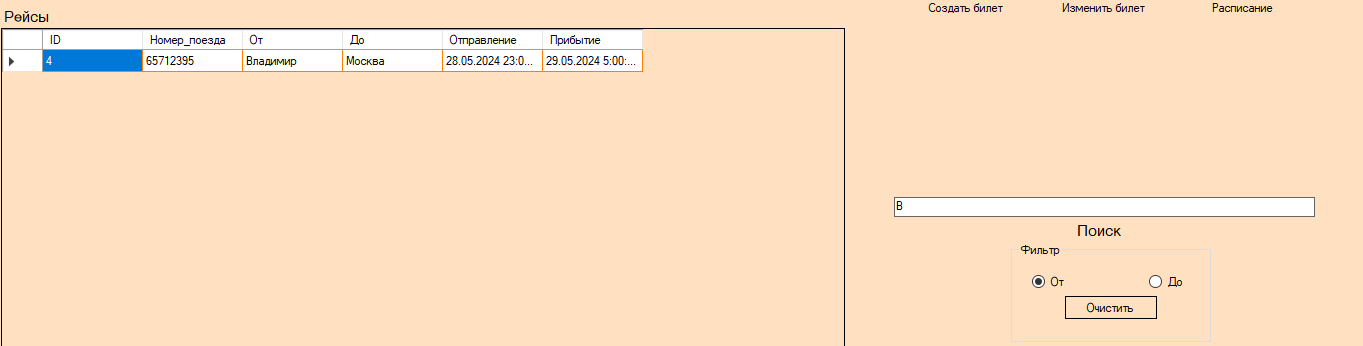


Рисунок 13 - Работа алгоритма «Поиск» в главной форме

На рисунке 14 обозначены 3 кнопки, которые отвечают за переход на другие формы, в соответствии с названием.



Рисунок 14 - Кнопки для перехода на формы

При нажатии на кнопку «Создать билет», откроется форма «Добавление билета». Основные элементы управления и общий интерфейс представлен на рисунке 15.

Для того, чтобы создать билет для пассажира, нужно следовать алгоритму, который описан ниже.

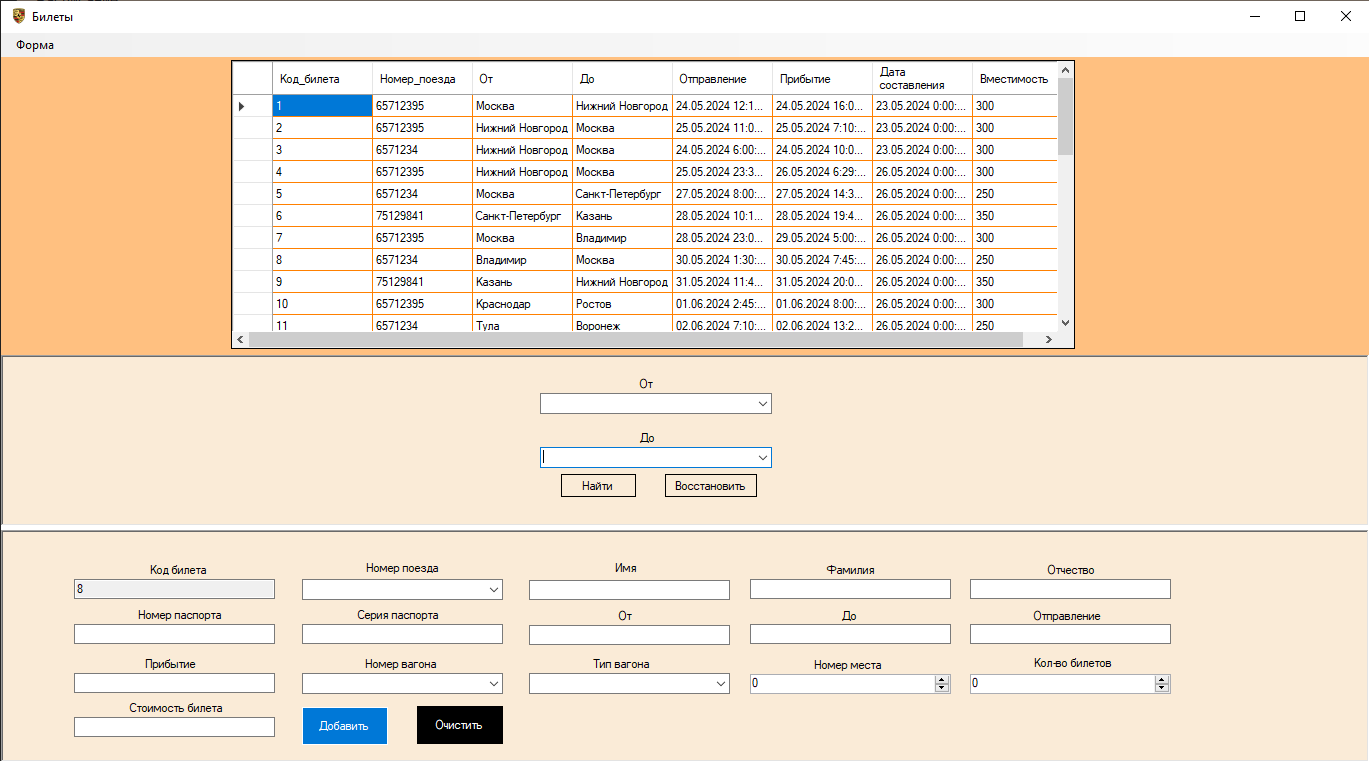


Рисунок 15 - Форма «Билеты»

Алгоритм составления билета:

1. На основе поиска, выбрать соответствующие города в полях «От» и «До». Сам поиск предстален на рисунке 16.

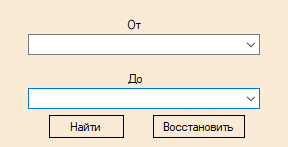


Рисунок 16 - Элементы управления для поиска

1. Отобрав нужный маршрут, в таблице обновиться информация, что позволит найти нужный, соответсвующий запросу пассажира. Результат отбора представлен на рисунке 17.
2. Для того, чтобы заполнить основные поля, отвечающие за состаление общей записи таблицы, нужно осуществить нажатие правой кнопкой мыши по любой записи. Результат заполнения представлен на рисунке 18.

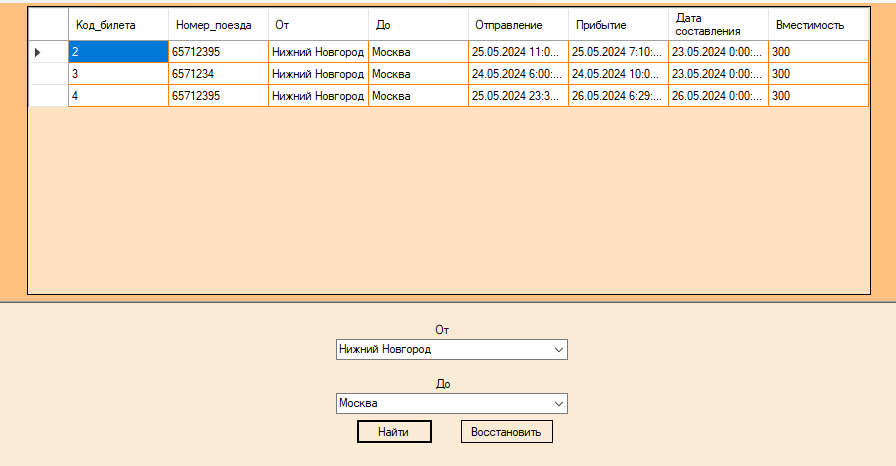


Рисунок 17 - Результат выборки в форме «Билеты»

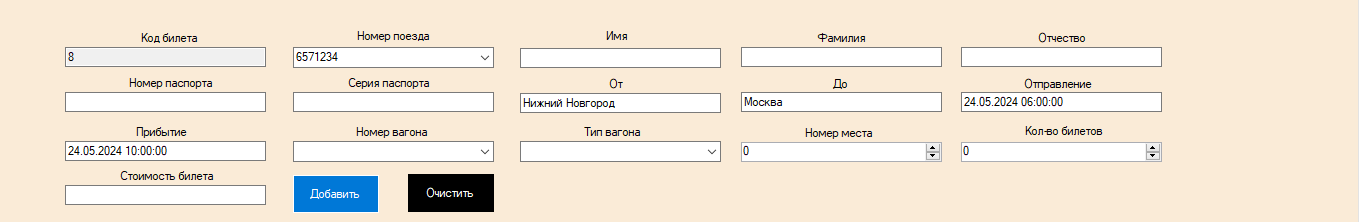


Рисунок 18 - Результат заполнения полей в таблице «Билеты»

1. Остальные поля, которые обязательные к заполнению, нужно заполнить самостоятельно, запрашивая соотвествующую информацию у самого пассажира (номер и серия паспорта, тип вагона, номер места, ФИО). Результат заполенения представлен на рисунке 19.

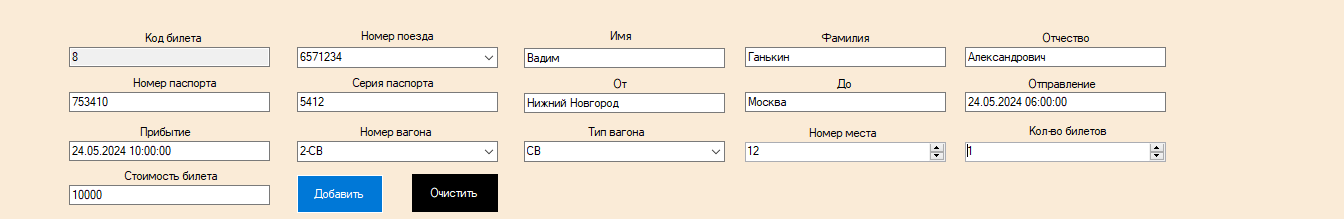


Рисунок 19 - Результат финального заполнения полей, формы «Билеты»

1. После заполнения оставшихся полей, для добавления записи в таблицу и осуществления регистрации места за пассажиром - нужно нажать кнопку «Добавить», которая окрашена синим цветом.

После нажатия будет открыто диологовое окно, которое сообщит о результате работы алгоритма. Само диалговое окно представлено на рисунке 20.

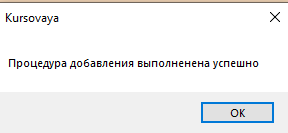


Рисунок 20 - Диалоговое окно добавления записи

В случае какой-либо ошибки сотрудника, в момент составления билетов, у данного сотрудника имеется возможность изменить запись, которая отвечает за данного пассажира. Изменение записи можно осуществить, нажав на кнопку «Измененить билет», в результате действия которой откроется форма соответствующая форма, показанная на рисунке 21.

Общий алгоритм изменения записи аналогичен алгоритму, который описан для таблицы «Билеты» и осуществляющий запись данных. Для изменения записи нужно осуществить поиск необходимого пассажира и нажать правой кнопкой мыши на запись, соответствующая данному пассажиру. После нажатия на соответствующую запись, совешается заполнения представленных полей данными, необходимые для перезаписи.

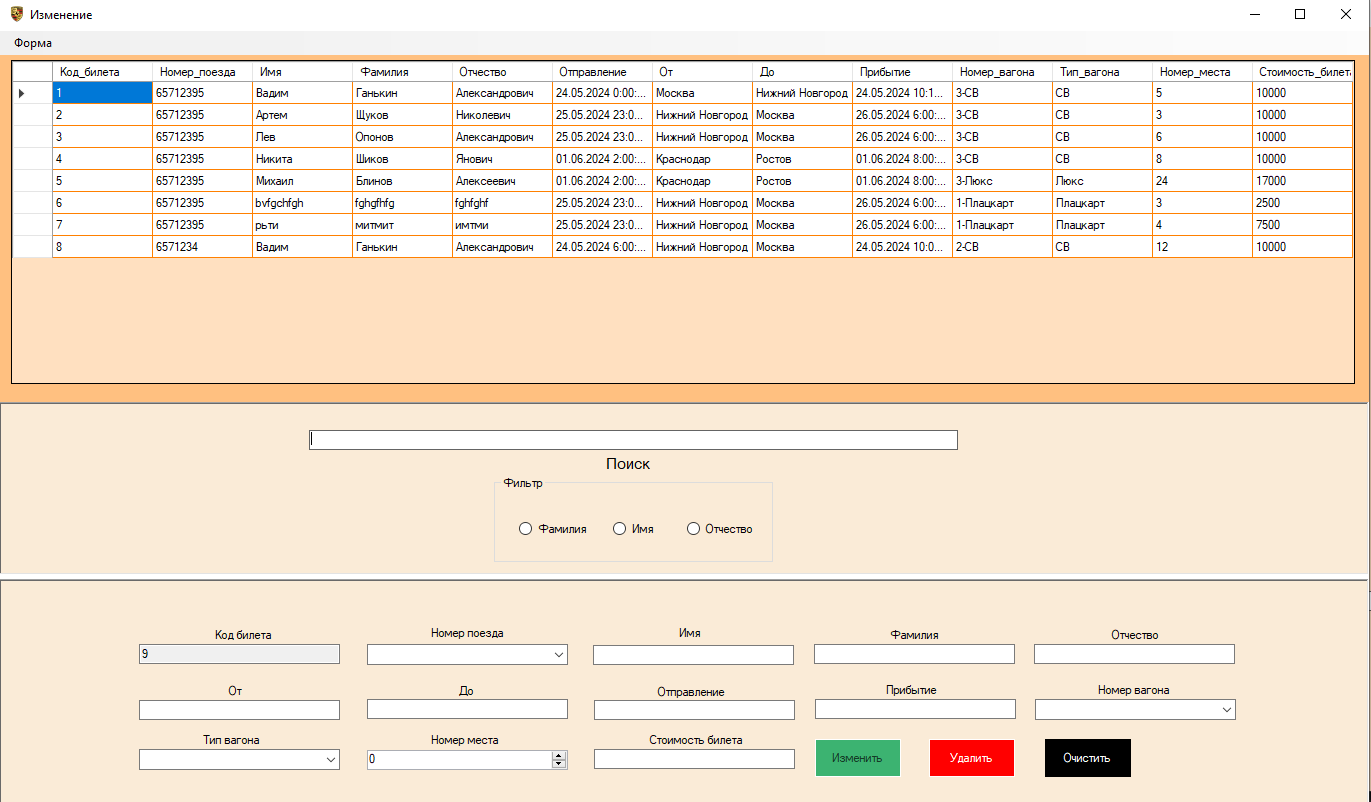


Рисунок 21 - Форма «Изменение билета»

После того, как сотрудник изменит необходимые даннные - он должен нажать на кнопку добавить, которая окрашена зеленым цветом.

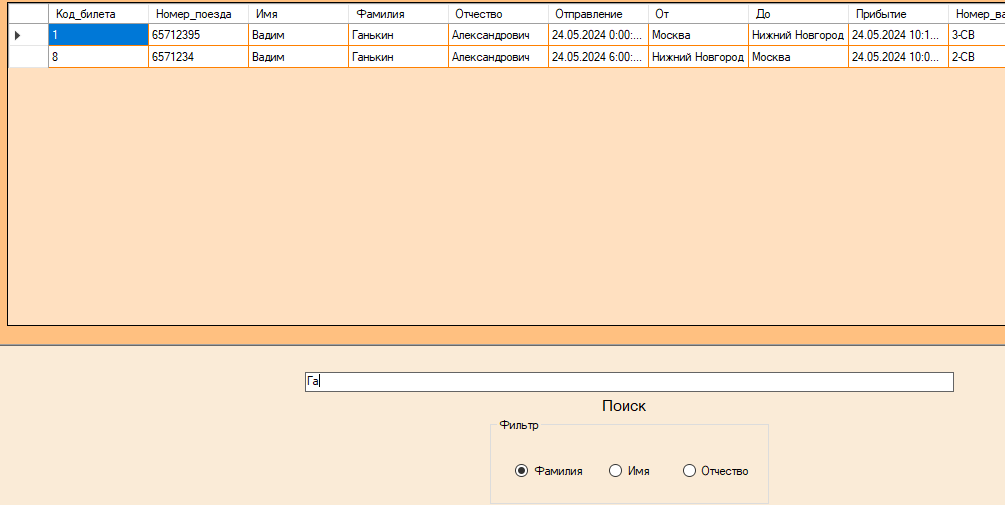


Рисунок 22 - Результат отбора записей, формы «Изменение»

Результат поиска необходимой записи представлен на рисунке 22, результат заполнения полей данными показан на рисунке 23, а результат нажатия кнопки «Изменить» показан на рисунке 24.

На форме «Изменение» также представлена кнопка «Удалить», которая отвечает за удаление соответствующей записи из таблицы «Билеты».

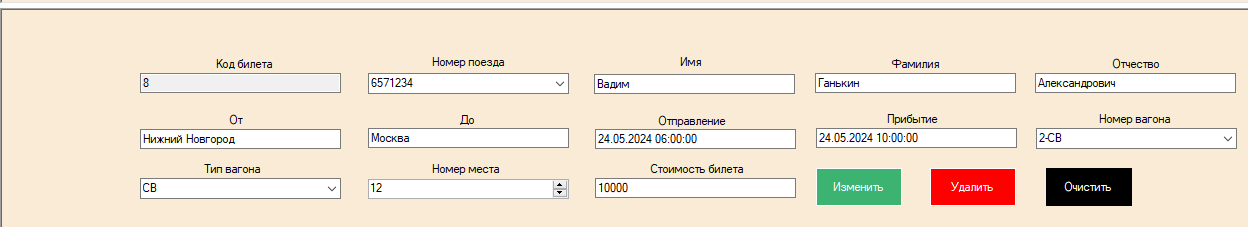


Рисунок 23 - Результат заполнения полей, формы «Изменение»

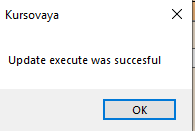


Рисунок 24 - Результат нажатия кнопки «Изменить», формы «Изменение»

Удаление происходит за счет нажатия правой кнопки мыши по записи в таблице и нажатии кнопки «Удаление». Результат удаления представлен на рисунке 25.

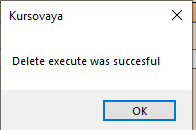


Рисунок 25 - Результат удаления записи

Для составления нового или изменения существующего расписания, на главной форме имеется кнопка «Расписание», которая имеет данные функции и позволяет осуществлять описанные действия. Форма «Расписание» представлена на рисунке 26.

На форме «Расписание» имеется несколько кнопок, отвечающие за: изменение, удаление и добавление записей. За форму изменения записи отвечает кнопка «Изменить», соответстующая форма показана на рисунке 27.

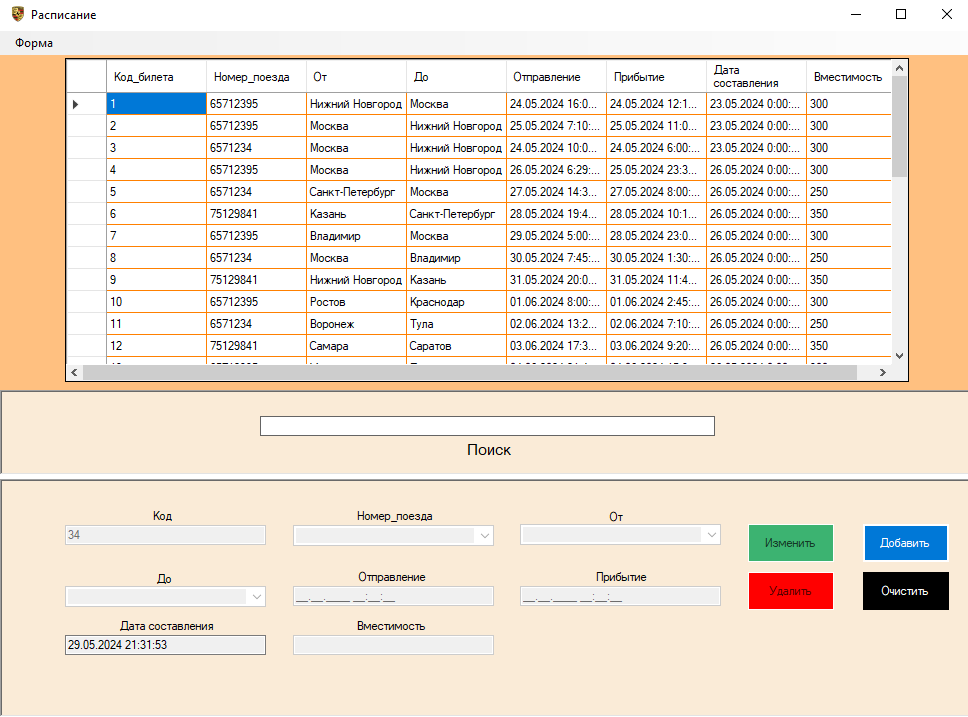


Рисунок 26 - Форма «Расписание»

Для добавления данных существует кнопка «Добавить», которая открывает ответствунную за это форму, представленную на рисунке 28. Если существующую запись расписания необходимо удалить - за это отвечает кнопка «Удалить», алгоритм которой описан выше, для таблицы «Изменение».

Общий алгоритм изменения записи аналогичен алгоритму, который описан для таблицы «Билеты» и осуществляющий запись данных. Для изменения записи нужно осуществить поиск необходимого расписания и нажать правой кнопкой мыши на запись, соответствующая желаниям пассажира. После нажатия на соответствующую запись, совершается заполнения представленных полей данными, необходимые для перезаписи.

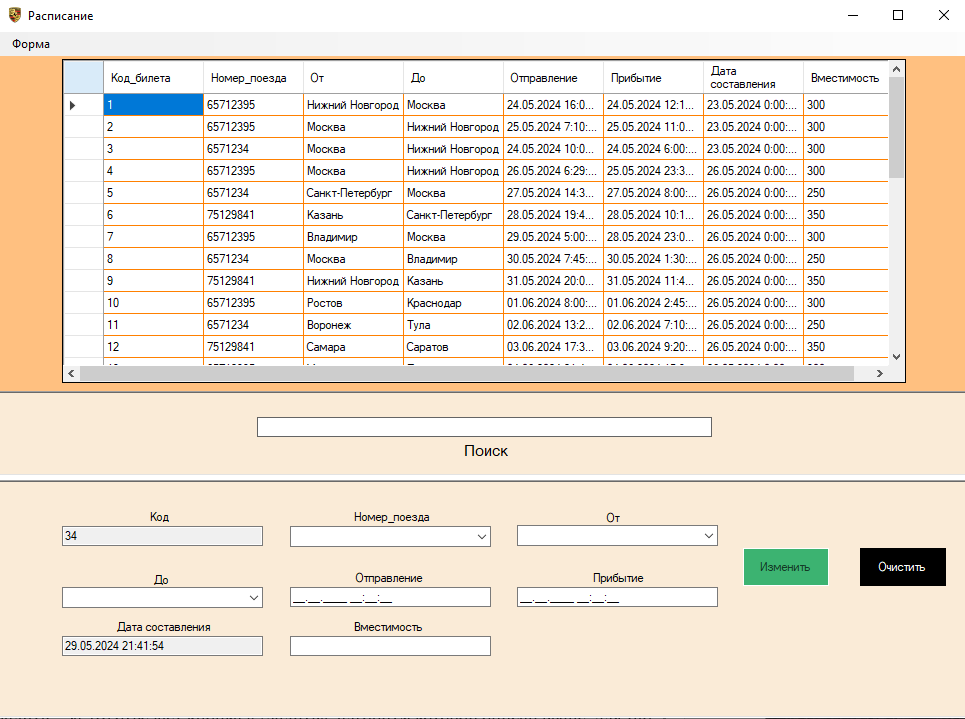


Рисунок 27 - Форма для изменения записей расписания

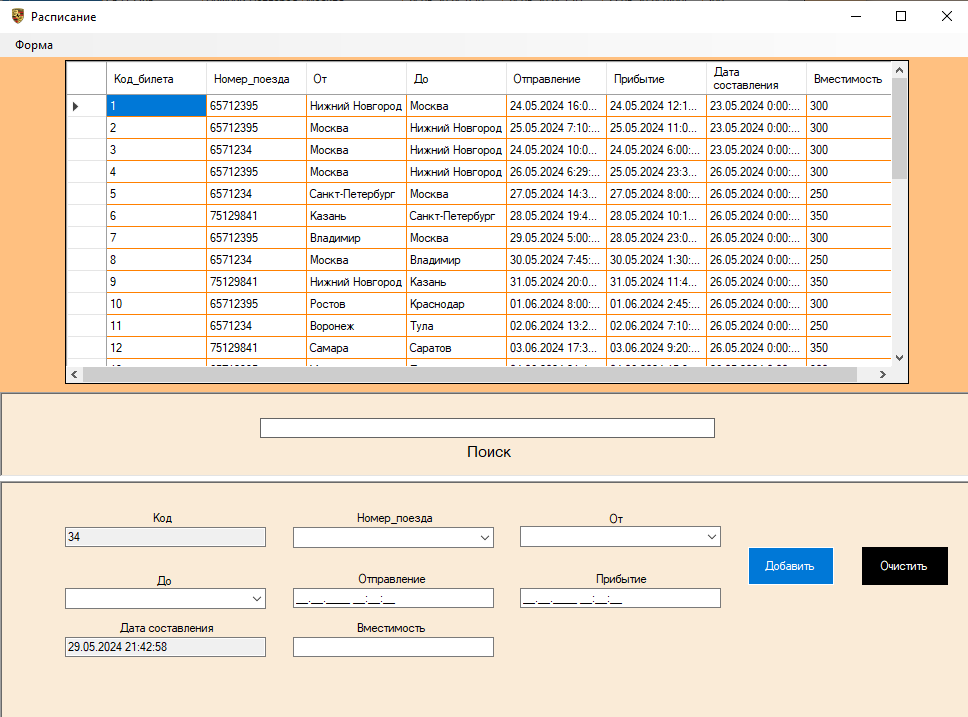


Рисунок 28 - Форма для добавления записей расписания

4 Охрана труда

4.1 Общие требования безопасности

Требования охраны труда распространяются на персонал, эксплуатирующий средства вычислительной техники и периферийное оборудование. Инструкция содержит общие указания по безопасности применения электрооборудования в учреждении. Требования настоящей инструкции являются обязательными, отступления от неё не допускаются. К самостоятельной эксплуатации электроаппаратуры допускаются только специально обученный персонал не моложе 18 лет, пригодный по состоянию здоровья и квалификации к выполнению указанных работ.

4.2 Требования безопасности перед работой на компьютере

Осмотреть и привести в порядок рабочее место. Отрегулировать освещение на рабочем месте, убедится в отсутствии потока встречного света. Проверить правильность подключения электрооборудования в сети. Протереть салфеткой поверхность экрана. Проверить правильность установки стола и клавиатуры.

4.3 Требования безопасности во время работы

Во время работы быть внимательным, не отвлекаться посторонними делами и разговорами. Рабочее место должно быть оборудовано так, чтобы исключить неудобные позы и длительные статические напряжения тела. При работе на ПК должна быть исключена возможность одновременного прикосновения к оборудованию и частям помещения или оборудования, имеющим соединение с землёй (радиатора батарей, металлоконструкций). Во время работы нельзя класть на монитор бумаги, книги и другие предметы, которые могут закрыть его вентиляционные отверстия. Запрещается оставлять без присмотра включенное оборудование; вскрывать устройства ПК.

4.4 Требования безопасности после окончания работы

Отключить ПК от сети, штепсельную вилку при этом держась за корпус. Запрещается отключать ПК за электропривод. При отключении ПК со съёмным шнуром питания сначала необходимо отключить вилку от розетки, а затем отключить питающий шнур от ПК. Привести в порядок рабочее место. Чистку ПК от пыли необходимо производить только после отключения ПК от сети.

4.5 Требования безопасности в аварийных ситуациях

При обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

При обнаружении оборвавшегося провода необходимо немедленно сообщить об этом администрации, принять меры по исключению контакта с ним людей. Прикосновение к проводу опасно для жизни.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему.

Необходимо немедленно начать производить искусственное дыхание, наиболее эффективным из которых является метод рот в рот или рот в нос, а также наружный массаж сердца. Искусственное дыхание поражённому электрическим током производится вплоть до прибытия врача.

На рабочем месте запрещается иметь огнеопасные вещества.

В помещениях запрещается:

* зажигать огонь;
* включать электрооборудование, если в помещении пахнет газом;
* курить;
* сушить что-либо на отопительных приборах;
* закрывать вентиляционные отверстия в электроаппаратуре.

Источниками воспламенения являются:

* искра при разряде статического электричества;
* искры от электрооборудования;
* искры от удара и трения;
* открытое пламя.

При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию. Помещения с электрооборудованием должны быть оснащены огнетушителями типа ОУ-2 или ОУБ-3.

Заключение

В ходе выполнения курсового проекта было разработано приложение, осуществляющие взаимодействие с базой данных «Железнодорожный вокзал» и обеспечивающие автоматизация бизнес процессов, которые происходят на самом вокзале. Также, за счет разработанных алгоритмов подгрузки данных в элементы управления comboBox и автоматизации поиска в большом количестве данных - обеспечивается более интуитивное и удобно управление.

Цели и задачи курсового проекта выполнены успешно.

Список использованных источников

1. Положение по организации выполнения и защиты курсовой работы (проекта) в ГБПОУ «НАМТ», 2024 г.
2. Стандарт предприятия ГБПОУ «НАМТ», 2017 г.
3. Методические указания по выполнению курсового проектирования по МДК.01.01. Разработка программных модулей для студентов специальности 09.02.07 Информационные системы и программирование, 2023 .
4. Бодров О.А., Медведев Р.Е. Предметно-ориентированные экономические информационные системы. - М.: Горячая линия - Телеком, 2013. -250 С.
5. Громов, А. Современные подходы к автоматизации управления железнодорожным транспортом / Громов, А. // Транспорт и логистика. -2020. -№ 86. С. 45-54.
6. Иванов, С. Оптимизация работы железнодорожных вокзалов с использованием информационных технологий / Иванов, С // Информационные технологии. -2018. - № 12. С. 56-68.
7. Петров, В. Проектирование информационных систем на Visual Basic. / Петров, В // Москва: ДМК Пресс. -2015. - № 4. С. 5-10.
8. Рудаков А.В. Технология разработки программных продуктов. Практикум: учеб. пособие для студ. учреждений сред. проф. образования / А.В. Рудаков, Г.Н. Федорова. - 4-е изд. Стер. - М.: Издательский центр «Академия», 2014. -356 С.
9. Сидоров, К. Применение Visual Basic для автоматизации процессов на железнодорожных вокзалах / Сидоров, К // Программные решения. -2019. -№ 67. С. 5-20.
10. Федорова Г.Н Разработка модулей программноо обеспечения для компьютерных систем: учебник для студентов среднего профессионального образования / Г.Н. Федорова. – 3-е ездание. –М.: Издательский центр «Академия», 2019. -384 С.
11. Шилоносов, М. Программирование баз данных на Visual Basic, Москва: МК-Пресс, 2010. -257 С.

Приложение 1

Imports MySql.Data.MySqlClient

Module SQL

Public constring As MySqlConnection

Public command As MySqlCommand

Sub connection(ByRef connBool As Boolean)

Try

constring = New MySqlConnection("server=127.0.0.1; database=железнодорожный\_вокзал; username=root")

constring.Open()

connBool = True

Catch ex As Exception

MsgBox("Connection was failed [ " & ex.Message & " ]")

connBool = False

End Try

End Sub

End Module

Public Class main

Dim connBool As Boolean

Dim i As Integer

Dim dr As MySqlDataReader

Dim picker As String

Private Sub main\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Try

connection(connBool)

If connBool Then

Dim SucColor = RGB(0, 255, 0)

lblCon.Text = "Connection: True"

lblCon.BackColor = Color.LightGreen

End If

Catch ex As Exception

MsgBox(ex.Message)

lblCon.Text = "Connection: False"

lblCon.BackColor = Color.Red

End Try

loadData\_rice()

loadData\_Schedule()

End Sub

Public Sub loadData\_rice()

Try

Dim cmd As New MySqlCommand("Select \* from рейсы", constring)

DataGridView1.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), dr.Item("Отправление"), dr.Item("Прибытие"))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub loadData\_Schedule()

Try

Dim cmd As New MySqlCommand("Select \* from расписание", constring)

DataGridView2.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView2.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), dr.Item("Отправление"), dr.Item("Прибытие"), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub Tickets\_Click(sender As Object, e As EventArgs)

Tickets2.Show()

End Sub

Private Sub bSchedule\_Click(sender As Object, e As EventArgs)

Schedule.Show()

End Sub

Private Sub Train\_Race\_Click(sender As Object, e As EventArgs)

Rice.Show()

End Sub

Private Sub bTickets\_Click(sender As Object, e As EventArgs) Handles bTickets.Click

AddRecTicke.Show()

End Sub

Private Sub bSche\_Click(sender As Object, e As EventArgs) Handles bSche.Click

Schedule.Show()

End Sub

Private Sub bRice\_Click(sender As Object, e As EventArgs)

Rice.Show()

End Sub

Private Sub MonthCalendar1\_DateSelected(sender As Object, e As DateRangeEventArgs)

End Sub

Private Sub bChangeTick\_Click(sender As Object, e As EventArgs) Handles bChangeTick.Click

ChangeRecTicke.Show()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

If cbFrom.Text = cbTo.Text Then

MsgBox("Выбранные направления одинковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора направлений")

Else

Try

Dim cmd As New MySqlCommand("Select \* from расписание where От = '" & cbTo.Text & "' and До = '" & cbFrom.Text & "'", constring)

dr = cmd.ExecuteReader

DataGridView2.Rows.Clear()

While dr.Read()

DataGridView2.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), (dr.Item("Отправление")), (dr.Item("Прибытие")), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Close()

cmd.CommandText = "Select \* from рейсы where От = '" & cbFrom.Text & "' and До = '" & cbTo.Text & "'"

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), (dr.Item("Отправление")), (dr.Item("Прибытие")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End If

End Sub

Private Sub bClear\_Click(sender As Object, e As EventArgs) Handles bRecent.Click

dr.Close()

DataGridView1.Rows.Clear()

DataGridView2.Rows.Clear()

loadData\_rice()

loadData\_Schedule()

cbFrom.Text = ""

cbTo.Text = ""

End Sub

Private Sub txtSearch\_TextChanged(sender As Object, e As EventArgs) Handles txtSearch.TextChanged

If rbFrom.Checked = True Then

Try

Dim cmd As New MySqlCommand("Select \* from расписание where От like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView2.Rows.Clear()

While dr.Read()

DataGridView2.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), (dr.Item("Прибытие")), (dr.Item("Отправление")), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Close()

DataGridView1.Rows.Clear()

cmd.CommandText = "Select \* from рейсы where `От` like '" & txtSearch.Text & "%'"

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), (dr.Item("Отправление")), (dr.Item("Прибытие")))

End While

Catch ex As Exception

MsgBox(ex.Message)

End Try

dr.Dispose()

End If

If rbTo.Checked = True Then

Try

dr.Close()

Dim cmd As New MySqlCommand("Select \* from расписание where До like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView2.Rows.Clear()

While dr.Read()

DataGridView2.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), (dr.Item("Прибытие")), (dr.Item("Отправление")), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Close()

DataGridView1.Rows.Clear()

cmd.CommandText = "Select \* from рейсы where `До` like '" & txtSearch.Text & "%'"

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), (dr.Item("Отправление")), (dr.Item("Прибытие")))

End While

Catch ex As Exception

MsgBox(ex.Message)

End Try

End If

End Sub

Private Sub bClear\_Click\_1(sender As Object, e As EventArgs) Handles bClear.Click

dr.Close()

txtSearch.Text = ""

rbTo.Checked = False

rbFrom.Checked = False

loadData\_rice()

loadData\_Schedule()

End Sub

End Class

Imports System.ComponentModel

Imports System.Data.SqlTypes

Imports System.Windows.Forms.VisualStyles.VisualStyleElement.ProgressBar

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class AddRecTicke

Dim a As Integer

Dim dr As MySqlDataReader

Dim picker As String

Dim count

Private Sub Train\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

loadData()

load\_txtNumT()

txtido.ReadOnly = True

End Sub

Public Sub splitDateToSQL(comboDate As TextBox, ByRef picker As String)

Dim y, monh, day, h, m, s, ttime As String

y = CStr(DatePart("yyyy", comboDate.Text))

monh = CStr(DatePart("m", comboDate.Text))

day = CStr(DatePart("d", comboDate.Text))

ttime = Split(comboDate.Text)(1)

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

If Len(CStr(Hour(ttime))) Or Len(CStr(Minute(ttime))) Or Len(CStr(Second(ttime))) <= 1 Then

If Len(CStr(Hour(ttime))) <= 1 Then

h = "0" & CStr(Hour(ttime))

Else

h = CStr(Hour(ttime))

End If

If Len(CStr(Minute(ttime))) <= 1 Then

m = "0" & CStr(Minute(ttime))

Else

m = CStr(Minute(ttime))

End If

If Len(CStr(Second(ttime))) <= 1 Then

s = "0" & CStr(Second(ttime))

Else

s = CStr(Second(ttime))

End If

End If

picker = y & "-" & monh & "-" & day & " " & h & ":" & m & ":" & s

End Sub

Public Sub splitDateToText(valDate As DateTime, ByRef picker As String)

Dim y, monh, day, h, m, s As String

y = CStr(DatePart("yyyy", valDate))

monh = CStr(Month(valDate))

day = CStr(DatePart("d", valDate))

If Len(CStr(valDate.Hour)) Or Len(CStr(valDate.Minute)) Or Len(CStr(valDate.Second)) <= 1 Then

If Len(CStr(valDate.Hour)) <= 1 Then

h = "0" & CStr(valDate.Hour)

Else

h = CStr(valDate.Hour)

End If

If Len(CStr(valDate.Minute)) <= 1 Then

m = "0" & CStr(valDate.Minute)

Else

m = CStr(valDate.Minute)

End If

If Len(CStr(valDate.Second)) <= 1 Then

s = "0" & CStr(valDate.Second)

Else

s = CStr(valDate.Second)

End If

End If

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

'tdate = Split(txtDate.Text, " ")(0)

'ttime = Split(txtDate.Text, " ")(1)

picker = day & "." & monh & "." & y & " " & h & ":" & s & ":" & s

End Sub

Public Sub updateId()

Dim ido As Integer

ido = CInt(txtido.Text) + 1

CountRec(count)

Try

While ido <= count

Dim cmd As New MySqlCommand("UPDATE `билеты` SET `Код\_билета`=@Код\_билета WHERE `Код\_билета`=@Код\_билета2", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Код\_билета", ido - 1)

cmd.Parameters.AddWithValue("@Код\_билета2", ido)

cmd.ExecuteNonQuery()

ido += 1

End While

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub SaveData()

Try

Dim cmd As New MySqlCommand("Select \* from билеты", constring)

Dim NumC, CarrT, numSit, numT

Dim check As Boolean

dr.Close()

dr = cmd.ExecuteReader

While dr.Read()

NumC = dr.Item("Номер\_вагона")

CarrT = dr.Item("Тип\_вагона")

numSit = dr.Item("Номер\_места")

If NumC = txtNumCarr.Text And CarrT = txtTypeCarr.Text And numSit = NumericUpDown1.Value Then

MessageBox.Show("Данное место уже занято")

check = False

Exit While

Else

check = True

End If

End While

dr.Dispose()

If check = True Then

cmd.CommandText = "INSERT INTO `билеты`(`Код\_билета`,`Номер\_поезда`, `Имя`, `Фамилия`, `Отчество`, `Отправление`, `От`, `До`, `Прибытие`,`Номер\_вагона`,`Тип\_вагона`, `Номер\_места`,`Стоимость\_билета`) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?)"

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("p1", txtido.Text)

cmd.Parameters.AddWithValue("p2", txtNumT.Text)

cmd.Parameters.AddWithValue("p3", txtFName.Text)

cmd.Parameters.AddWithValue("p4", txtSName.Text)

cmd.Parameters.AddWithValue("p5", txtThName.Text)

splitDateToSQL(DateTimePicker2, picker)

cmd.Parameters.AddWithValue("p6", picker)

cmd.Parameters.AddWithValue("p7", txtFrom.Text)

cmd.Parameters.AddWithValue("p8", txtTo.Text)

splitDateToSQL(DateTimePicker1, picker)

cmd.Parameters.AddWithValue("p9", picker)

cmd.Parameters.AddWithValue("p10", txtNumCarr.Text)

cmd.Parameters.AddWithValue("p11", txtTypeCarr.SelectedItem)

cmd.Parameters.AddWithValue("p12", NumericUpDown1.Value)

cmd.Parameters.AddWithValue("p13", txtPrice.Text)

cmd.ExecuteNonQuery()

MsgBox("Процедура добавления выполненена успешно")

loadData()

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

CountRec()

End Sub

Public Sub loadData()

Try

Dim cmd As New MySqlCommand("Select \* from расписание", constring)

DataGridView1.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), dr.Item("Отправление"), dr.Item("Прибытие"), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CountRec()

End Sub

Public Sub CountRec(Optional ByRef count As Integer = 0)

Try

Dim cmd As New MySqlCommand("Select COUNT(\*) as coun from билеты", constring)

cmd.Parameters.Clear()

dr = cmd.ExecuteReader

While dr.Read()

txtido.Text = dr.Item("coun") + 1

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

txtNumT.Text = DataGridView1.CurrentRow.Cells(1).Value

splitDateToText(DataGridView1.CurrentRow.Cells(4).Value, picker)

DateTimePicker2.Text = picker

txtFrom.Text = DataGridView1.CurrentRow.Cells(2).Value

txtTo.Text = DataGridView1.CurrentRow.Cells(3).Value

splitDateToText(DataGridView1.CurrentRow.Cells(5).Value, picker)

DateTimePicker1.Text = picker

End Sub

Private Sub ВыйтиToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ВыйтиToolStripMenuItem.Click

If MsgBox("Вы уверенны, что хотите закрыть приложение?", MsgBoxStyle.Question + vbYesNo, Title:="Выход из приложения") = vbYes Then

Application.Exit()

Else

Return

End If

End Sub

Private Sub bClear\_Click\_1(sender As Object, e As EventArgs) Handles bClear.Click

txtFrom.Text = ""

txtSName.Text = ""

txtFName.Text = ""

txtTo.Text = ""

txtNumT.Text = ""

txtThName.Text = ""

txtido.Text = ""

txtPrice.Text = ""

txtTypeCarr.Text = ""

txtNumCarr.Text = ""

DateTimePicker1.Text = ""

DateTimePicker2.Text = ""

txtNumPasp.Text = ""

txtSerPasp.Text = ""

NumericUpDown1.Value = 0

CountRec()

End Sub

Private Sub bSave\_Click(sender As Object, e As EventArgs) Handles bSave.Click

Dim picker2 As String

splitDateToSQL(DateTimePicker2, picker)

splitDateToSQL(DateTimePicker1, picker2)

Dim cmd As New MySqlCommand("Select COUNT(\*) as count from билеты where Номер\_поезда = " & txtNumT.SelectedItem & " and От like '" & txtFrom.Text & "' and До like '" & txtTo.Text & "' and Отправление = '" & picker & "' and Прибытие = '" & picker2 & "'", constring)

Dim countRow, txtNumPas

dr.Close()

dr = cmd.ExecuteReader

txtNumCarr.Items.Clear()

While dr.Read()

countRow = dr.Item("count")

End While

dr.Close()

cmd.CommandText = "Select \* from поезда where Номер\_поезда = " & txtNumT.SelectedItem

dr = cmd.ExecuteReader

While dr.Read()

txtNumPas = dr.Item("Вместимость")

End While

dr.Dispose()

If txtFrom.Text = txtTo.Text And DateTimePicker1.Text = DateTimePicker2.Text Then

MsgBox("Выбранные поля даты, места отправления и прибытия одинаковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка составления расписания")

ElseIf DateTimePicker1.Text = DateTimePicker2.Text Then

MsgBox("Выбранное время неверное так, как выбранные поля отправления и прибытия одинаковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора даты")

ElseIf txtFrom.Text = txtTo.Text Then

MsgBox("Выбранный маршрут неверный так, как выбранный города одинаковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора городов")

ElseIf txtFName.Text = "" Or txtSName.Text = "" Or txtThName.Text = "" Then

MsgBox("Введите ФИО", MsgBoxStyle.Information + vbOK, Title:="Ошибка заполнения")

ElseIf countRow > txtNumPas Then

MsgBox("Все места на данный рейс заняты", MsgBoxStyle.Information + vbOK, Title:="Переполнение мест на рейсе")

Else

SaveData()

End If

End Sub

Sub load\_txtNumT()

Dim cmd As New MySqlCommand("Select \* from поезда", constring)

dr = cmd.ExecuteReader

txtNumT.Items.Clear()

While dr.Read()

txtNumT.Items.Add(dr.Item("Номер\_поезда"))

End While

dr.Dispose()

End Sub

Private Sub AddRecTicke\_Closing(sender As Object, e As CancelEventArgs) Handles Me.Closing

Tickets2.loadData()

End Sub

Private Sub txtNumCarr\_Click(sender As Object, e As EventArgs) Handles txtNumCarr.Click

Dim nameCarr As String

Dim countCarr, numPlac, numCup, numSv, numLux As Integer

a = 1

Try

Dim cmd As New MySqlCommand("Select \* from поезда where Номер\_поезда like '" & txtNumT.SelectedItem & "'", constring)

nameCarr = "Плацкарт"

dr = cmd.ExecuteReader

txtNumCarr.Items.Clear()

While dr.Read()

countCarr = dr.Item("Общее\_кол\_во\_вагонов")

numPlac = dr.Item("Кол\_во\_вагонов\_типа\_Плацкарт")

numCup = dr.Item("Кол\_во\_вагонов\_типа\_Купе")

numSv = dr.Item("Кол\_во\_вагонов\_типа\_СВ")

numLux = dr.Item("Кол\_во\_вагонов\_типа\_Люкс")

End While

dr.Dispose()

For i = 1 To countCarr

If nameCarr = "Плацкарт" Then

For a = a To numPlac

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numPlac Then

a = 0

nameCarr = "Купе"

End If

End If

If nameCarr = "Купе" Then

For a = a To numCup

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numCup Then

a = 0

nameCarr = "СВ"

End If

End If

If nameCarr = "СВ" Then

For a = a To numSv

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numSv Then

a = 0

nameCarr = "Люкс"

End If

End If

If nameCarr = "Люкс" Then

For a = a To numLux

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numLux Then

a = 0

nameCarr = ""

Exit For

End If

End If

Next i

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub txtTypeCarr\_Click(sender As Object, e As EventArgs) Handles txtTypeCarr.Click

Try

If txtNumCarr.Text.Split("-")(1) = "Купе" Then

txtTypeCarr.Text = "Купе"

End If

If txtNumCarr.Text.Split("-")(1) = "Плацкарт" Then

txtTypeCarr.Text = "Плацкарт"

End If

If txtNumCarr.Text.Split("-")(1) = "СВ" Then

txtTypeCarr.Text = "СВ"

End If

If txtNumCarr.Text.Split("-")(1) = "Люкс" Then

txtTypeCarr.Text = "Люкс"

End If

Catch ex As Exception

End Try

End Sub

Private Sub GroupBox1\_Enter(sender As Object, e As EventArgs)

End Sub

Private Sub Label5\_Click(sender As Object, e As EventArgs)

End Sub

Private Sub countTick\_ValueChanged(sender As Object, e As EventArgs) Handles countTick.ValueChanged

If txtTypeCarr.Text = "Купе" Then

txtPrice.Text = CStr(5000 \* countTick.Value)

End If

If txtTypeCarr.Text = "Плацкарт" Then

txtPrice.Text = CStr(2500 \* countTick.Value)

End If

If txtTypeCarr.Text = "СВ" Then

txtPrice.Text = CStr(10000 \* countTick.Value)

End If

If txtTypeCarr.Text = "Люкс" Then

txtPrice.Text = CStr(17000 \* countTick.Value)

End If

End Sub

Private Sub bRecent\_Click(sender As Object, e As EventArgs) Handles bRecent.Click

loadData()

End Sub

Private Sub bSearch\_Click(sender As Object, e As EventArgs) Handles bSearch.Click

If cbFrom.Text = cbTo.Text Then

MsgBox("Выбранные направления одинковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора направлений")

Else

Try

Dim cmd As New MySqlCommand("Select \* from расписание where От = '" & cbTo.Text & "' and До = '" & cbFrom.Text & "'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("От"), dr.Item("До"), (dr.Item("Отправление")), (dr.Item("Прибытие")), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End If

End Sub

End Class

Imports System.ComponentModel

Imports System.Data.SqlTypes

Imports System.Diagnostics.Tracing

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class ChangeRecRice

Dim i As Integer

Dim dr As MySqlDataReader

Dim picker As String

Dim count

Private Sub Schedule\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

loadData()

txtido.ReadOnly = True

End Sub

Public Overloads Sub splitDateToSQL(txtDate As TextBoxBase, ByRef picker As String)

Dim y, m, d, tdate, ttime As String

y = CStr(DatePart("yyyy", txtDate.Text))

m = CStr(Month(txtDate.Text))

d = CStr(DatePart("d", txtDate.Text))

If Len(m) <= 1 Then

m = "0" & m

End If

If Len(d) <= 1 Then

d = "0" & d

End If

If Split(txtDate.Text, " ").GetLength(0) > 1 Then

tdate = Split(txtDate.Text, " ")(0)

ttime = Split(txtDate.Text, " ")(1)

picker = y & "-" & m & "-" & d & " " & ttime

Else

tdate = Split(txtDate.Text, " ")(0)

picker = y & "-" & m & "-" & d

End If

End Sub

Public Overloads Sub splitDateToSQL(comboDate As ComboBox, ByRef picker As String)

Dim y, m, d, tdate, ttime As String

y = CStr(DatePart("yyyy", comboDate.Text))

m = CStr(Month(comboDate.Text))

d = CStr(DatePart("d", comboDate.Text))

If Len(m) <= 1 Then

m = "0" & m

End If

If Len(d) <= 1 Then

d = "0" & d

End If

If Split(comboDate.Text, " ").GetLength(0) > 1 Then

tdate = Split(comboDate.Text, " ")(0)

ttime = Split(comboDate.Text, " ")(1)

picker = y & "-" & m & "-" & d & " " & ttime

Else

tdate = Split(comboDate.Text, " ")(0)

picker = y & "-" & m & "-" & d

End If

End Sub

Public Sub splitDateToText(valDate As DateTime, ByRef picker As String)

Dim y, monh, day, h, m, s As String

y = CStr(DatePart("yyyy", valDate))

monh = CStr(Month(valDate))

day = CStr(DatePart("d", valDate))

If Len(CStr(valDate.Hour)) Or Len(CStr(valDate.Minute)) Or Len(CStr(valDate.Second)) <= 1 Then

If Len(CStr(valDate.Hour)) <= 1 Then

h = "0" & CStr(valDate.Hour)

Else

h = CStr(valDate.Hour)

End If

If Len(CStr(valDate.Minute)) <= 1 Then

m = "0" & CStr(valDate.Minute)

Else

m = CStr(valDate.Minute)

End If

If Len(CStr(valDate.Second)) <= 1 Then

s = "0" & CStr(valDate.Second)

Else

s = CStr(valDate.Second)

End If

End If

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

'tdate = Split(txtDate.Text, " ")(0)

'ttime = Split(txtDate.Text, " ")(1)

picker = day & "." & monh & "." & y & " " & h & ":" & s & ":" & s

End Sub

Public Sub updateId(ID As Integer)

Dim ido As Integer

ido = CInt(txtido.Text) + 1

CountRec(count)

Try

While ido <= count

Dim cmd As New MySqlCommand("UPDATE `расписание` SET `ID`=@ID WHERE `ID`=@ID2", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@ID", ido - 1)

cmd.Parameters.AddWithValue("@ID2", ido)

cmd.ExecuteNonQuery()

ido += 1

End While

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub loadData()

Try

Dim cmd As New MySqlCommand("Select \* from рейсы", constring)

DataGridView1.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), (dr.Item("Отправление")), (dr.Item("Прибытие")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CountRec()

End Sub

Public Sub updateData()

Try

Dim cmd As New MySqlCommand("UPDATE `рейсы` SET `ID`=@ID,`Номер\_поезда`=@Номер\_поезда,`От`=@От,`До`=@До,`Отправление`=@Отправление,`Прибытие`=@Прибытие WHERE `ID`=@ID2", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@ID", txtido.Text)

cmd.Parameters.AddWithValue("@Номер\_поезда", txtNumT.Text)

cmd.Parameters.AddWithValue("@До", txtTo.SelectedItem)

cmd.Parameters.AddWithValue("@От", txtFrom.SelectedItem)

splitDateToSQL(DateTimePicker2, picker)

cmd.Parameters.AddWithValue("@Прибытие", picker)

splitDateToSQL(DateTimePicker1, picker)

cmd.Parameters.AddWithValue("@Отправление", picker)

cmd.Parameters.AddWithValue("@ID2", txtido.Text)

cmd.ExecuteNonQuery()

MsgBox("Процедура изменения выполненена успешно")

loadData()

Catch ex As Exception

MsgBox(ex.Message)

End Try

picker = ""

End Sub

Public Sub CountRec(Optional ByRef count As Integer = 0)

Dim cmd As New MySqlCommand("Select COUNT(\*) as count from рейсы", constring)

cmd.Parameters.Clear()

dr.Close()

dr = cmd.ExecuteReader

While dr.Read()

txtido.Text = dr.Item("count") + 1

End While

dr.Dispose()

cmd.Parameters.Clear()

cmd.CommandText = "Select MAX(`ID`) as id from рейсы"

dr.Close()

dr = cmd.ExecuteReader

While dr.Read()

count = CInt(dr.Item("id"))

End While

dr.Dispose()

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

txtido.Text = DataGridView1.CurrentRow.Cells(0).Value

txtNumT.Text = DataGridView1.CurrentRow.Cells(1).Value

txtFrom.SelectedItem = DataGridView1.CurrentRow.Cells(2).Value

txtTo.SelectedItem = DataGridView1.CurrentRow.Cells(3).Value

splitDateToText(DataGridView1.CurrentRow.Cells(4).Value, picker)

DateTimePicker1.Text = picker

splitDateToText(DataGridView1.CurrentRow.Cells(5).Value, picker)

DateTimePicker2.Text = picker

bChange.Enabled = True

End Sub

Private Sub TextBox1\_TextChanged(sender As Object, e As EventArgs) Handles txtSearch.TextChanged

Try

Dim cmd As New MySqlCommand("Select \* from рейсы where ID like '" & txtSearch.Text & "' or Номер\_поезда like '" & txtSearch.Text & "%' or До like '" & txtSearch.Text & "%' or От like '" & txtSearch.Text & "%' or Прибытие like '" & txtSearch.Text & "%' or Отправление like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), (dr.Item("Отправление")), (dr.Item("Прибытие")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub ВыйтиToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ВыйтиToolStripMenuItem.Click

If MsgBox("Вы уверенны, что хотите закрыть приложение?", MsgBoxStyle.Question + vbYesNo, Title:="Выход из приложения") = vbYes Then

Application.Exit()

Else

Return

End If

End Sub

Private Sub bChange\_Click\_1(sender As Object, e As EventArgs) Handles bChange.Click

updateData()

End Sub

Private Sub bClear\_Click\_1(sender As Object, e As EventArgs) Handles bClear.Click

bclears()

CountRec()

End Sub

Public Sub bclears()

txtFrom.SelectedItem = Nothing

txtTo.SelectedItem = Nothing

DateTimePicker1.Text = Nothing

DateTimePicker2.Text = Nothing

txtNumT.Text = Nothing

txtido.Text = ""

bChange.Enabled = False

End Sub

Private Sub bSave\_Click(sender As Object, e As EventArgs)

AddRecRice.Show()

loadData()

End Sub

Private Sub txtNumT\_Click(sender As Object, e As EventArgs) Handles txtNumT.Click

Dim cmd As New MySqlCommand("Select \* from поезда", constring)

dr = cmd.ExecuteReader

txtNumT.Items.Clear()

While dr.Read()

txtNumT.Items.Add(dr.Item("Номер\_поезда"))

End While

dr.Dispose()

End Sub

Private Sub DateTimePicker1\_Click(sender As Object, e As EventArgs) Handles DateTimePicker1.Click

Try

Dim cmd As New MySqlCommand("Select \* from расписание where От like '" & txtFrom.SelectedItem & "' and До like '" & txtTo.SelectedItem & "'", constring)

dr = cmd.ExecuteReader

DateTimePicker1.Items.Clear()

While dr.Read()

DateTimePicker1.Items.Add(dr.Item("Отправление"))

End While

dr.Dispose()

Catch ex As Exception

End Try

End Sub

Private Sub DateTimePicker2\_Click(sender As Object, e As EventArgs) Handles DateTimePicker2.Click

Try

Dim cmd As New MySqlCommand("Select \* from расписание where От like '" & txtFrom.SelectedItem & "' and До like '" & txtTo.SelectedItem & "'", constring)

dr = cmd.ExecuteReader

DateTimePicker2.Items.Clear()

While dr.Read()

DateTimePicker2.Items.Add(dr.Item("Прибытие"))

End While

dr.Dispose()

Catch ex As Exception

End Try

End Sub

Private Sub ChangeRecRice\_Closing(sender As Object, e As CancelEventArgs) Handles Me.Closing

Rice.loadData()

End Sub

Private Sub MenuStrip1\_ItemClicked(sender As Object, e As ToolStripItemClickedEventArgs) Handles MenuStrip1.ItemClicked

End Sub

End Class

Imports System.ComponentModel

Imports System.Data.SqlTypes

Imports System.Diagnostics.Tracing

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class ChangeRecSche

Dim dr As MySqlDataReader

Dim picker As String

Private Sub Schedule\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

loadData()

txtido.ReadOnly = True

bChange.Enabled = False

txtDateCreate.ReadOnly = True

txtDateCreate.Text = Now()

End Sub

Public Sub splitDateToSQL(txtDate As TextBoxBase, ByRef picker As String)

Dim y, m, d, tdate, ttime As String

y = CStr(DatePart("yyyy", txtDate.Text))

m = CStr(Month(txtDate.Text))

d = CStr(DatePart("d", txtDate.Text))

If Len(m) <= 1 Then

m = "0" & m

End If

If Len(d) <= 1 Then

d = "0" & d

End If

If Split(txtDate.Text, " ").GetLength(0) > 1 Then

tdate = Split(txtDate.Text, " ")(0)

ttime = Split(txtDate.Text, " ")(1)

picker = y & "-" & m & "-" & d & " " & ttime

Else

tdate = Split(txtDate.Text, " ")(0)

picker = y & "-" & m & "-" & d

End If

End Sub

Public Sub splitDateToText(valDate As DateTime, ByRef picker As String)

Dim y, monh, day, h, m, s As String

y = CStr(DatePart("yyyy", valDate))

monh = CStr(Month(valDate))

day = CStr(DatePart("d", valDate))

If Len(CStr(valDate.Hour)) Or Len(CStr(valDate.Minute)) Or Len(CStr(valDate.Second)) <= 1 Then

If Len(CStr(valDate.Hour)) <= 1 Then

h = "0" & CStr(valDate.Hour)

Else

h = CStr(valDate.Hour)

End If

If Len(CStr(valDate.Minute)) <= 1 Then

m = "0" & CStr(valDate.Minute)

Else

m = CStr(valDate.Minute)

End If

If Len(CStr(valDate.Second)) <= 1 Then

s = "0" & CStr(valDate.Second)

Else

s = CStr(valDate.Second)

End If

End If

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

'tdate = Split(txtDate.Text, " ")(0)

'ttime = Split(txtDate.Text, " ")(1)

picker = day & "." & monh & "." & y & " " & h & ":" & s & ":" & s

End Sub

Public Sub loadData()

Try

Dim cmd As New MySqlCommand("Select \* from расписание", constring)

DataGridView1.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), (dr.Item("Прибытие")), (dr.Item("Отправление")), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CountRec()

End Sub

Public Sub updateData()

Try

Dim cmd As New MySqlCommand("UPDATE `расписание` SET `Номер\_поезда`=@Номер\_поезда, `До`=@До, `От`=@От, `Прибытие`=@Прибытие, `Отправление`=@Отправление, `Дата\_составления`=@Дата\_составления, `Вместимость`=@Вместимость WHERE `Код`=@Код", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Номер\_поезда", txtNumT.Text)

cmd.Parameters.AddWithValue("@До", txtTo.SelectedItem)

cmd.Parameters.AddWithValue("@От", txtFrom.SelectedItem)

splitDateToSQL(DateTimePicker1, picker)

cmd.Parameters.AddWithValue("@Прибытие", picker)

splitDateToSQL(DateTimePicker2, picker)

cmd.Parameters.AddWithValue("@Отправление", picker)

splitDateToSQL(txtDateCreate, picker)

cmd.Parameters.AddWithValue("@Дата\_составления", picker)

cmd.Parameters.AddWithValue("@Вместимость", CInt(txtNumPass.Text))

cmd.Parameters.AddWithValue("@Код", txtido.Text)

cmd.ExecuteNonQuery()

MsgBox("Процедура изменения выполненена успешно")

loadData()

Catch ex As Exception

MsgBox(ex.Message)

End Try

picker = ""

End Sub

Public Sub CountRec(Optional ByRef count As Integer = 0)

Dim cmd As New MySqlCommand("Select COUNT(\*) as count from расписание", constring)

cmd.Parameters.Clear()

dr.Close()

dr = cmd.ExecuteReader

While dr.Read()

txtido.Text = dr.Item("count") + 1

End While

dr.Dispose()

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

txtido.Text = DataGridView1.CurrentRow.Cells(0).Value

txtNumT.Text = DataGridView1.CurrentRow.Cells(1).Value

txtFrom.SelectedItem = DataGridView1.CurrentRow.Cells(2).Value

txtTo.SelectedItem = DataGridView1.CurrentRow.Cells(3).Value

splitDateToText(DataGridView1.CurrentRow.Cells(4).Value, picker)

DateTimePicker1.Text = picker

splitDateToText(DataGridView1.CurrentRow.Cells(5).Value, picker)

DateTimePicker2.Text = picker

txtNumPass.Text = DataGridView1.CurrentRow.Cells(7).Value

bChange.Enabled = True

End Sub

Private Sub TextBox1\_TextChanged(sender As Object, e As EventArgs) Handles txtSearch.TextChanged

Try

Dim cmd As New MySqlCommand("Select \* from расписание where Код like '" & txtSearch.Text & "' or Номер\_поезда like '" & txtSearch.Text & "%' or До like '" & txtSearch.Text & "%' or Дата\_составления like '" & txtSearch.Text & "%' or От like '" & txtSearch.Text & "%' or Прибытие like '" & txtSearch.Text & "%' or Отправление like '" & txtSearch.Text & "%' or Вместимость like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), dr.Item("Прибытие"), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub ВыйтиToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ВыйтиToolStripMenuItem.Click

If MsgBox("Вы уверенны, что хотите закрыть приложение?", MsgBoxStyle.Question + vbYesNo, Title:="Выход из приложения") = vbYes Then

Application.Exit()

Else

Return

End If

End Sub

Private Sub bChange\_Click\_1(sender As Object, e As EventArgs) Handles bChange.Click

If txtFrom.SelectedItem = txtTo.SelectedItem And DateTimePicker1.Text = DateTimePicker2.Text Then

MsgBox("Выбранные поля даты, места отправления и прибытия одинаковы. Измените данные поля", MsgBoxStyle.Information + vbOK, Title:="Ошибка составления расписания")

ElseIf DateTimePicker1.Text = DateTimePicker2.Text Then

MsgBox("Выбранное время неверное так, как выбранные поля отправления и прибытия одинаковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора даты")

ElseIf txtFrom.SelectedItem = txtTo.SelectedItem Then

MsgBox("Выбранный маршрут неверный так, как выбранный города одинаковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора городов")

Else

updateData()

End If

End Sub

Private Sub bClear\_Click\_1(sender As Object, e As EventArgs) Handles bClear.Click

bclears()

CountRec()

bChange.Enabled = False

End Sub

Public Sub bclears()

txtFrom.SelectedItem = Nothing

txtTo.SelectedItem = Nothing

DateTimePicker1.Text = ""

DateTimePicker2.Text = ""

txtNumT.Text = ""

txtNumPass.Text = ""

txtido.Text = ""

End Sub

Private Sub bSave\_Click(sender As Object, e As EventArgs)

AddRecSche.Show()

loadData()

End Sub

Private Sub txtNumT\_Click(sender As Object, e As EventArgs) Handles txtNumT.Click

Dim cmd As New MySqlCommand("Select \* from поезда", constring)

dr = cmd.ExecuteReader

txtNumT.Items.Clear()

While dr.Read()

txtNumT.Items.Add(dr.Item("Номер\_поезда"))

End While

dr.Dispose()

End Sub

Private Sub ChangeRecSche\_Closing(sender As Object, e As CancelEventArgs) Handles Me.Closing

Schedule.loadData()

End Sub

Private Sub txtNumPass\_Click(sender As Object, e As EventArgs) Handles txtNumPass.Click

Dim cmd As New MySqlCommand("Select \* from поезда where Номер\_поезда = '" & txtNumT.Text & "'", constring)

Dim countPass

dr = cmd.ExecuteReader

txtNumT.Items.Clear()

While dr.Read()

txtNumPass.Text = dr.Item("Вместимость")

End While

dr.Dispose()

End Sub

End Class

Imports System.ComponentModel

Imports System.Data.SqlTypes

Imports System.Windows.Forms.VisualStyles.VisualStyleElement.ProgressBar

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class ChangeRecTicke

Dim i, a As Integer

Dim dr As MySqlDataReader

Dim picker As String

Dim count

Public Sub splitDateToSQL(comboDate As TextBox, ByRef picker As String)

Dim y, monh, day, h, m, s, ttime As String

y = CStr(DatePart("yyyy", comboDate.Text))

monh = CStr(DatePart("m", comboDate.Text))

day = CStr(DatePart("d", comboDate.Text))

ttime = Split(comboDate.Text)(1)

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

If Len(CStr(Hour(ttime))) Or Len(CStr(Minute(ttime))) Or Len(CStr(Second(ttime))) <= 1 Then

If Len(CStr(Hour(ttime))) <= 1 Then

h = "0" & CStr(Hour(ttime))

Else

h = CStr(Hour(ttime))

End If

If Len(CStr(Minute(ttime))) <= 1 Then

m = "0" & CStr(Minute(ttime))

Else

m = CStr(Minute(ttime))

End If

If Len(CStr(Second(ttime))) <= 1 Then

s = "0" & CStr(Second(ttime))

Else

s = CStr(Second(ttime))

End If

End If

picker = y & "-" & monh & "-" & day & " " & h & ":" & m & ":" & s

End Sub

Public Sub splitDateToText(valDate As DateTime, ByRef picker As String)

Dim y, monh, day, h, m, s As String

y = CStr(DatePart("yyyy", valDate))

monh = CStr(Month(valDate))

day = CStr(DatePart("d", valDate))

If Len(CStr(valDate.Hour)) Or Len(CStr(valDate.Minute)) Or Len(CStr(valDate.Second)) <= 1 Then

If Len(CStr(valDate.Hour)) <= 1 Then

h = "0" & CStr(valDate.Hour)

Else

h = CStr(valDate.Hour)

End If

If Len(CStr(valDate.Minute)) <= 1 Then

m = "0" & CStr(valDate.Minute)

Else

m = CStr(valDate.Minute)

End If

If Len(CStr(valDate.Second)) <= 1 Then

s = "0" & CStr(valDate.Second)

Else

s = CStr(valDate.Second)

End If

End If

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

'tdate = Split(txtDate.Text, " ")(0)

'ttime = Split(txtDate.Text, " ")(1)

picker = day & "." & monh & "." & y & " " & h & ":" & m & ":" & s

End Sub

Public Sub loadData()

Try

Dim cmd As New MySqlCommand("Select \* from билеты", constring)

DataGridView1.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код\_билета")), dr.Item("Номер\_поезда"), dr.Item("Имя"), dr.Item("Фамилия"), dr.Item("Отчество"), dr.Item("Отправление"), dr.Item("От"), dr.Item("До"), dr.Item("Прибытие"), dr.Item("Номер\_вагона"), dr.Item("Тип\_вагона"), dr.Item("Номер\_места"), dr.Item("Стоимость\_билета"))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CountRec()

End Sub

Public Sub updateData()

Try

Dim cmd As New MySqlCommand("UPDATE `билеты` SET `Номер\_поезда`=@Номер\_поезда,`Имя`=@Имя,`Фамилия`=@Фамилия,`Отчество`=@Отчество,`Отправление`= @Отправление,`От`=@От,`До`=@До,`Прибытие`=@Прибытие, `Номер\_вагона`=@Номер\_вагона, `Тип\_вагона`=@Тип\_вагона, `Номер\_места`=@Номер\_места, `Стоимость\_билета`=@Стоимость\_билета WHERE `Код\_билета`= @Код\_билета", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Номер\_поезда", txtNumT.Text)

cmd.Parameters.AddWithValue("@Имя", txtFName.Text)

cmd.Parameters.AddWithValue("@Фамилия", txtSName.Text)

cmd.Parameters.AddWithValue("@Отчество", txtThName.Text)

splitDateToSQL(DateTimePicker2, picker)

cmd.Parameters.AddWithValue("@Отправление", picker)

cmd.Parameters.AddWithValue("@От", txtFrom.Text)

cmd.Parameters.AddWithValue("@До", txtTo.Text)

splitDateToSQL(DateTimePicker1, picker)

cmd.Parameters.AddWithValue("@Прибытие", picker)

cmd.Parameters.AddWithValue("@Код\_билета", txtido.Text)

cmd.Parameters.AddWithValue("@Номер\_вагона", txtNumCarr.Text)

cmd.Parameters.AddWithValue("@Тип\_вагона", txtTypeCarr.Text)

cmd.Parameters.AddWithValue("@Номер\_места", NumCarr.Text)

cmd.Parameters.AddWithValue("@Стоимость\_билета", txtPrice.Text)

cmd.ExecuteNonQuery()

MsgBox("Update execute was succesful")

loadData()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub CountRec(Optional ByRef count As Integer = 0)

Try

Dim cmd As New MySqlCommand("Select COUNT(\*) as count from билеты", constring)

cmd.Parameters.Clear()

dr = cmd.ExecuteReader

While dr.Read()

txtido.Text = dr.Item("count") + 1

End While

dr.Dispose()

cmd.Parameters.Clear()

cmd.CommandText = "Select MAX(`Код\_билета`) as id from билеты"

dr = cmd.ExecuteReader

While dr.Read()

count = CInt(dr.Item("id"))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub TextBox1\_TextChanged(sender As Object, e As EventArgs) Handles txtSearch.TextChanged

If rbSName.Checked = True Then

Try

Dim cmd As New MySqlCommand("Select \* from билеты where Фамилия like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код\_билета")), dr.Item("Номер\_поезда"), dr.Item("Имя"), dr.Item("Фамилия"), dr.Item("Отчество"), dr.Item("Отправление"), dr.Item("От"), dr.Item("До"), dr.Item("Прибытие"), dr.Item("Номер\_вагона"), dr.Item("Тип\_вагона"), dr.Item("Номер\_места"), dr.Item("Стоимость\_билета"))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End If

If rbFName.Checked = True Then

Try

Dim cmd As New MySqlCommand("Select \* from билеты where Имя like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код\_билета")), dr.Item("Номер\_поезда"), dr.Item("Имя"), dr.Item("Фамилия"), dr.Item("Отчество"), dr.Item("Отправление"), dr.Item("От"), dr.Item("До"), dr.Item("Прибытие"), dr.Item("Номер\_вагона"), dr.Item("Тип\_вагона"), dr.Item("Номер\_места"), dr.Item("Стоимость\_билета"))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End If

If rbThName.Checked = True Then

Try

Dim cmd As New MySqlCommand("Select \* from билеты where Отчество like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код\_билета")), dr.Item("Номер\_поезда"), dr.Item("Имя"), dr.Item("Фамилия"), dr.Item("Отчество"), dr.Item("Отправление"), dr.Item("От"), dr.Item("До"), dr.Item("Прибытие"), dr.Item("Номер\_вагона"), dr.Item("Тип\_вагона"), dr.Item("Номер\_места"), dr.Item("Стоимость\_билета"))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End If

End Sub

Private Sub ВыйтиToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ВыйтиToolStripMenuItem.Click

If MsgBox("Вы уверенны, что хотите закрыть приложение?", MsgBoxStyle.Question + vbYesNo, Title:="Выход из приложения") = vbYes Then

Application.Exit()

Else

Return

End If

End Sub

Private Sub bChange\_Click\_1(sender As Object, e As EventArgs) Handles bChange.Click

If txtFrom.Text = txtTo.Text And DateTimePicker1.Text = DateTimePicker2.Text Then

MsgBox("Выбранные поля даты, места отправления и прибытия одинаковы. Измените данные поля", MsgBoxStyle.Information + vbOK, Title:="Ошибка составления расписания")

ElseIf DateTimePicker1.Text = DateTimePicker2.Text Then

MsgBox("Выбранное время неверное так, как выбранные поля отправления и прибытия одинаковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора даты")

ElseIf txtFrom.Text = txtTo.Text Then

MsgBox("Выбранный маршрут неверный так, как выбранный города одинаковы", MsgBoxStyle.Information + vbOK, Title:="Ошибка выбора городов")

Else

updateData()

End If

End Sub

Private Sub bClear\_Click\_1(sender As Object, e As EventArgs) Handles bClear.Click

txtFrom.Text = ""

txtSName.Text = ""

txtFName.Text = ""

txtTo.Text = ""

txtNumT.Text = ""

txtThName.Text = ""

txtido.Text = ""

txtPrice.Text = ""

txtTypeCarr.Text = ""

txtNumCarr.Text = ""

DateTimePicker1.Text = ""

DateTimePicker2.Text = ""

NumCarr.Value = 0

bChange.Enabled = False

CountRec()

End Sub

Sub load\_txtNumT()

Dim cmd As New MySqlCommand("Select \* from поезда", constring)

dr.Close()

dr = cmd.ExecuteReader

txtNumT.Items.Clear()

While dr.Read()

txtNumT.Items.Add(dr.Item("Номер\_поезда"))

End While

dr.Dispose()

End Sub

Private Sub ChangeRecTicke\_Closing(sender As Object, e As CancelEventArgs) Handles Me.Closing

Tickets2.loadData()

End Sub

Private Sub txtNumCarr\_Click(sender As Object, e As EventArgs) Handles txtNumCarr.Click

Dim nameCarr As String

Dim countCarr, numPlac, numCup, numSv, numLux As Integer

a = 1

Try

Dim cmd As New MySqlCommand("Select \* from поезда where Номер\_поезда like '" & txtNumT.Text & "'", constring)

nameCarr = "Плацкарт"

dr = cmd.ExecuteReader

txtNumCarr.Items.Clear()

While dr.Read()

countCarr = dr.Item("Общее\_кол\_во\_вагонов")

numPlac = dr.Item("Кол\_во\_вагонов\_типа\_Плацкарт")

numCup = dr.Item("Кол\_во\_вагонов\_типа\_Купе")

numSv = dr.Item("Кол\_во\_вагонов\_типа\_СВ")

numLux = dr.Item("Кол\_во\_вагонов\_типа\_Люкс")

End While

dr.Dispose()

For i = 1 To countCarr

If nameCarr = "Плацкарт" Then

For a = a To numPlac

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numPlac Then

a = 0

nameCarr = "Купе"

End If

End If

If nameCarr = "Купе" Then

For a = a To numCup

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numCup Then

a = 0

nameCarr = "СВ"

End If

End If

If nameCarr = "СВ" Then

For a = a To numSv

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numSv Then

a = 0

nameCarr = "Люкс"

End If

End If

If nameCarr = "Люкс" Then

For a = a To numLux

txtNumCarr.Items.Add(CStr(i) & "-" & nameCarr)

a += 1

Exit For

Next a

If a = numLux Then

a = 0

nameCarr = ""

Exit For

End If

End If

Next i

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub txtTypeCarr\_Click(sender As Object, e As EventArgs) Handles txtTypeCarr.Click

Try

If txtNumCarr.Text.Split("-")(1) = "Купе" Then

txtTypeCarr.Text = "Купе"

End If

If txtNumCarr.Text.Split("-")(1) = "Плацкарт" Then

txtTypeCarr.Text = "Плацкарт"

End If

If txtNumCarr.Text.Split("-")(1) = "СВ" Then

txtTypeCarr.Text = "СВ"

End If

If txtNumCarr.Text.Split("-")(1) = "Люкс" Then

txtTypeCarr.Text = "Люкс"

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub txtTypeCarr\_SelectedValueChanged(sender As Object, e As EventArgs) Handles txtTypeCarr.SelectedValueChanged

If txtTypeCarr.Text = "Купе" Then

txtPrice.Text = CStr(5000)

End If

If txtTypeCarr.Text = "Плацкарт" Then

txtPrice.Text = CStr(2500)

End If

If txtTypeCarr.Text = "СВ" Then

txtPrice.Text = CStr(10000)

End If

If txtTypeCarr.Text = "Люкс" Then

txtPrice.Text = CStr(17000)

End If

End Sub

Private Sub ChangeRecTicke\_Load\_1(sender As Object, e As EventArgs) Handles MyBase.Load

loadData()

load\_txtNumT()

txtido.ReadOnly = True

bChange.Enabled = False

End Sub

Public Sub updateId()

Dim ido As Integer

ido = CInt(txtido.Text) + 1

CountRec(count)

Try

While ido <= count

Dim cmd As New MySqlCommand("UPDATE `билеты` SET `Код\_билета`=@Код\_билета WHERE `Код\_билета`=@Код\_билета2", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Код\_билета", ido - 1)

cmd.Parameters.AddWithValue("@Код\_билета2", ido)

cmd.ExecuteNonQuery()

ido += 1

End While

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Sub delete()

If MsgBox("Вы уверенны, что хотите удалить данную запись", MsgBoxStyle.Question + vbYesNo, Title:="Удаление записи") = vbYes Then

Try

Dim cmd As New MySqlCommand("DELETE FROM `билеты` WHERE `Код\_билета`= @Код\_билета", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Код\_билета", CInt(txtido.Text))

cmd.ExecuteNonQuery()

MsgBox("Delete execute was succesful")

Catch ex As Exception

MsgBox(ex.Message)

End Try

Else

Return

End If

updateId()

loadData()

End Sub

Private Sub bDelete\_Click(sender As Object, e As EventArgs) Handles bDelete.Click

delete()

End Sub

Private Sub DataGridView1\_CellClick\_1(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

Try

txtido.Text = DataGridView1.CurrentRow.Cells(0).Value

txtNumT.Text = DataGridView1.CurrentRow.Cells(1).Value

txtFName.Text = DataGridView1.CurrentRow.Cells(2).Value

txtSName.Text = DataGridView1.CurrentRow.Cells(3).Value

txtThName.Text = DataGridView1.CurrentRow.Cells(4).Value

splitDateToText(DataGridView1.CurrentRow.Cells(5).Value, picker)

DateTimePicker2.Text = picker

txtFrom.Text = DataGridView1.CurrentRow.Cells(6).Value

txtTo.Text = DataGridView1.CurrentRow.Cells(7).Value

splitDateToText(DataGridView1.CurrentRow.Cells(8).Value, picker)

DateTimePicker1.Text = picker

txtNumCarr.Text = DataGridView1.CurrentRow.Cells(9).Value

txtTypeCarr.Text = DataGridView1.CurrentRow.Cells(10).Value

NumCarr.Value = CInt(DataGridView1.CurrentRow.Cells(11).Value)

txtPrice.Text = DataGridView1.CurrentRow.Cells(12).Value

bChange.Enabled = True

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

End Class

Imports System.Data.SqlTypes

Imports System.Diagnostics.Tracing

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class Rice

Dim i As Integer

Dim dr As MySqlDataReader

Dim picker As String

Dim count

Private Sub Schedule\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

loadData()

txtido.ReadOnly = True

bDelete.Enabled = False

bEntryRice.Enabled = False

txtFrom.Enabled = False

txtTo.Enabled = False

DateTimePicker1.Enabled = False

DateTimePicker2.Enabled = False

txtNumT.Enabled = False

txtido.Enabled = False

End Sub

Public Overloads Sub splitDateToSQL(txtDate As TextBoxBase, ByRef picker As String)

Dim y, m, d, tdate, ttime As String

y = CStr(DatePart("yyyy", txtDate.Text))

m = CStr(Month(txtDate.Text))

d = CStr(DatePart("d", txtDate.Text))

If Len(m) <= 1 Then

m = "0" & m

End If

If Len(d) <= 1 Then

d = "0" & d

End If

If Split(txtDate.Text, " ").GetLength(0) > 1 Then

tdate = Split(txtDate.Text, " ")(0)

ttime = Split(txtDate.Text, " ")(1)

picker = y & "-" & m & "-" & d & " " & ttime

Else

tdate = Split(txtDate.Text, " ")(0)

picker = y & "-" & m & "-" & d

End If

End Sub

Public Overloads Sub splitDateToSQL(comboDate As ComboBox, ByRef picker As String)

Dim y, m, d, tdate, ttime As String

y = CStr(DatePart("yyyy", comboDate.Text))

m = CStr(Month(comboDate.Text))

d = CStr(DatePart("d", comboDate.Text))

If Len(m) <= 1 Then

m = "0" & m

End If

If Len(d) <= 1 Then

d = "0" & d

End If

If Split(comboDate.Text, " ").GetLength(0) > 1 Then

tdate = Split(comboDate.Text, " ")(0)

ttime = Split(comboDate.Text, " ")(1)

picker = y & "-" & m & "-" & d & " " & ttime

Else

tdate = Split(comboDate.Text, " ")(0)

picker = y & "-" & m & "-" & d

End If

End Sub

Public Sub splitDateToText(valDate As DateTime, ByRef picker As String)

Dim y, monh, day, h, m, s As String

y = CStr(DatePart("yyyy", valDate))

monh = CStr(Month(valDate))

day = CStr(DatePart("d", valDate))

If Len(CStr(valDate.Hour)) Or Len(CStr(valDate.Minute)) Or Len(CStr(valDate.Second)) <= 1 Then

If Len(CStr(valDate.Hour)) <= 1 Then

h = "0" & CStr(valDate.Hour)

Else

h = CStr(valDate.Hour)

End If

If Len(CStr(valDate.Minute)) <= 1 Then

m = "0" & CStr(valDate.Minute)

Else

m = CStr(valDate.Minute)

End If

If Len(CStr(valDate.Second)) <= 1 Then

s = "0" & CStr(valDate.Second)

Else

s = CStr(valDate.Second)

End If

End If

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

'tdate = Split(txtDate.Text, " ")(0)

'ttime = Split(txtDate.Text, " ")(1)

picker = day & "." & monh & "." & y & " " & h & ":" & s & ":" & s

End Sub

Public Sub updateId(ID As Integer)

Dim ido As Integer

ido = CInt(txtido.Text) + 1

CountRec(count)

Try

While ido <= count

Dim cmd As New MySqlCommand("UPDATE `расписание` SET `ID`=@ID WHERE `ID`=@ID2", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@ID", ido - 1)

cmd.Parameters.AddWithValue("@ID2", ido)

cmd.ExecuteNonQuery()

ido += 1

End While

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub loadData()

Try

Dim cmd As New MySqlCommand("Select \* from рейсы", constring)

DataGridView1.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), (dr.Item("Отправление")), (dr.Item("Прибытие")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CountRec()

End Sub

Public Sub deleteData()

If MsgBox("Вы уверенны, что хотите удалить данную запись", MsgBoxStyle.Question + vbYesNo, Title:="Удаление записи") = vbYes Then

Try

Dim cmd As New MySqlCommand("DELETE FROM `рейсы` WHERE `ID`= @ID", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@ID", CInt(txtido.Text))

cmd.ExecuteNonQuery()

MsgBox("Процедура удаления выполненена успешно")

Catch ex As Exception

MsgBox(ex.Message)

End Try

Else

Return

End If

updateId(CInt(txtido.Text))

loadData()

bclears()

CountRec()

End Sub

Public Sub CountRec(Optional ByRef count As Integer = 0)

Dim cmd As New MySqlCommand("Select COUNT(\*) as count from рейсы", constring)

cmd.Parameters.Clear()

dr.Close()

dr = cmd.ExecuteReader

While dr.Read()

txtido.Text = dr.Item("count") + 1

End While

dr.Dispose()

cmd.Parameters.Clear()

cmd.CommandText = "Select MAX(`ID`) as id from рейсы"

dr.Close()

dr = cmd.ExecuteReader

While dr.Read()

count = CInt(dr.Item("id"))

End While

dr.Dispose()

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

txtido.Text = DataGridView1.CurrentRow.Cells(0).Value

txtNumT.Text = DataGridView1.CurrentRow.Cells(1).Value

txtFrom.SelectedItem = DataGridView1.CurrentRow.Cells(2).Value

txtTo.SelectedItem = DataGridView1.CurrentRow.Cells(3).Value

splitDateToText(DataGridView1.CurrentRow.Cells(4).Value, picker)

DateTimePicker1.Text = picker

splitDateToText(DataGridView1.CurrentRow.Cells(5).Value, picker)

DateTimePicker2.Text = picker

bSave.Enabled = False

bDelete.Enabled = True

bEntryRice.Enabled = True

End Sub

Private Sub TextBox1\_TextChanged(sender As Object, e As EventArgs) Handles txtSearch.TextChanged

Try

Dim cmd As New MySqlCommand("Select \* from рейсы where ID like '" & txtSearch.Text & "' or Номер\_поезда like '" & txtSearch.Text & "%' or До like '" & txtSearch.Text & "%' or От like '" & txtSearch.Text & "%' or Прибытие like '" & txtSearch.Text & "%' or Отправление like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("ID")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), (dr.Item("Отправление")), (dr.Item("Прибытие")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub ВыйтиToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ВыйтиToolStripMenuItem.Click

If MsgBox("Вы уверенны, что хотите закрыть приложение?", MsgBoxStyle.Question + vbYesNo, Title:="Выход из приложения") = vbYes Then

Application.Exit()

Else

Return

End If

End Sub

Private Sub bChange\_Click\_1(sender As Object, e As EventArgs) Handles bChange.Click

ChangeRecRice.Show()

End Sub

Private Sub bDelete\_Click\_1(sender As Object, e As EventArgs) Handles bDelete.Click

deleteData()

End Sub

Private Sub bClear\_Click\_1(sender As Object, e As EventArgs) Handles bClear.Click

bclears()

CountRec()

bSave.Enabled = True

bEntryRice.Enabled = False

End Sub

Public Sub bclears()

txtFrom.SelectedItem = Nothing

txtTo.SelectedItem = Nothing

DateTimePicker1.Text = Nothing

DateTimePicker2.Text = Nothing

txtNumT.Text = Nothing

txtido.Text = ""

End Sub

Private Sub bSave\_Click(sender As Object, e As EventArgs) Handles bSave.Click

AddRecRice.Show()

loadData()

End Sub

Private Sub txtNumT\_Click(sender As Object, e As EventArgs) Handles txtNumT.Click

Dim cmd As New MySqlCommand("Select \* from поезда", constring)

dr = cmd.ExecuteReader

txtNumT.Items.Clear()

While dr.Read()

txtNumT.Items.Add(dr.Item("Номер\_поезда"))

End While

dr.Dispose()

End Sub

Private Sub DateTimePicker1\_Click(sender As Object, e As EventArgs) Handles DateTimePicker1.Click

Try

Dim cmd As New MySqlCommand("Select \* from расписание where От like '" & txtFrom.SelectedItem & "' and До like '" & txtTo.SelectedItem & "'", constring)

dr = cmd.ExecuteReader

DateTimePicker1.Items.Clear()

While dr.Read()

DateTimePicker1.Items.Add(dr.Item("Отправление"))

End While

dr.Dispose()

Catch ex As Exception

End Try

End Sub

Private Sub DateTimePicker2\_Click(sender As Object, e As EventArgs) Handles DateTimePicker2.Click

Try

Dim cmd As New MySqlCommand("Select \* from расписание where От like '" & txtFrom.SelectedItem & "' and До like '" & txtTo.SelectedItem & "'", constring)

dr = cmd.ExecuteReader

DateTimePicker2.Items.Clear()

While dr.Read()

DateTimePicker2.Items.Add(dr.Item("Прибытие"))

End While

dr.Dispose()

Catch ex As Exception

End Try

End Sub

Private Sub bEntryRice\_Click(sender As Object, e As EventArgs) Handles bEntryRice.Click

If MsgBox("Вы уверенны, что рейс действительно пришел?", MsgBoxStyle.Question + vbYesNo, Title:="Подтверждение рейса") = vbYes Then

Try

Dim cmd As New MySqlCommand("Delete from `билеты` where `От`=@От and `До`=@До and `Номер\_поезда`=@Номер\_поезда", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@От", txtFrom.SelectedItem)

cmd.Parameters.AddWithValue("@До", txtTo.SelectedItem)

cmd.Parameters.AddWithValue("@Номер\_поезда", txtNumT.Text)

cmd.ExecuteNonQuery()

cmd.CommandText = "Delete from `рейсы` where `ID`=@ID"

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@ID", txtido.Text)

MsgBox("Процедура подтверждения рейса выполненена успешно")

Catch ex As Exception

MsgBox(ex.Message)

End Try

Else

Return

End If

updateId(CInt(txtido.Text))

loadData()

bclears()

CountRec()

End Sub

End Class

Imports System.Data.SqlTypes

Imports System.Diagnostics.Tracing

Imports MySql.Data.MySqlClient

Imports MySql.Data.Types

Public Class Schedule

Dim i As Integer

Dim dr As MySqlDataReader

Dim picker As String

Dim count

Private Sub Schedule\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

loadData()

txtido.ReadOnly = True

bDelete.Enabled = False

bChange.Enabled = False

txtDateCreate.ReadOnly = True

txtDateCreate.Text = Now()

txtFrom.Enabled = False

txtTo.Enabled = False

DateTimePicker1.Enabled = False

DateTimePicker2.Enabled = False

txtNumT.Enabled = False

txtNumPass.Enabled = False

txtido.Enabled = False

End Sub

Public Sub splitDateToSQL(txtDate As TextBoxBase, ByRef picker As String)

Dim y, m, d, tdate, ttime As String

y = CStr(DatePart("yyyy", txtDate.Text))

m = CStr(Month(txtDate.Text))

d = CStr(DatePart("d", txtDate.Text))

If Len(m) <= 1 Then

m = "0" & m

End If

If Len(d) <= 1 Then

d = "0" & d

End If

If Split(txtDate.Text, " ").GetLength(0) > 1 Then

tdate = Split(txtDate.Text, " ")(0)

ttime = Split(txtDate.Text, " ")(1)

picker = y & "-" & m & "-" & d & " " & ttime

Else

tdate = Split(txtDate.Text, " ")(0)

picker = y & "-" & m & "-" & d

End If

End Sub

Public Sub splitDateToText(valDate As DateTime, ByRef picker As String)

Dim y, monh, day, h, m, s As String

y = CStr(DatePart("yyyy", valDate))

monh = CStr(Month(valDate))

day = CStr(DatePart("d", valDate))

If Len(CStr(valDate.Hour)) Or Len(CStr(valDate.Minute)) Or Len(CStr(valDate.Second)) <= 1 Then

If Len(CStr(valDate.Hour)) <= 1 Then

h = "0" & CStr(valDate.Hour)

Else

h = CStr(valDate.Hour)

End If

If Len(CStr(valDate.Minute)) <= 1 Then

m = "0" & CStr(valDate.Minute)

Else

m = CStr(valDate.Minute)

End If

If Len(CStr(valDate.Second)) <= 1 Then

s = "0" & CStr(valDate.Second)

Else

s = CStr(valDate.Second)

End If

End If

If Len(monh) <= 1 Then

monh = "0" & monh

End If

If Len(day) <= 1 Then

day = "0" & day

End If

'tdate = Split(txtDate.Text, " ")(0)

'ttime = Split(txtDate.Text, " ")(1)

picker = day & "." & monh & "." & y & " " & h & ":" & s & ":" & s

End Sub

Public Sub updateId(ID As Integer)

Dim ido As Integer

ido = CInt(txtido.Text) + 1

CountRec(count)

Try

While ido <= count

Dim cmd As New MySqlCommand("UPDATE `расписание` SET `Код`=@Код WHERE `Код`=@Код2", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Код", ido - 1)

cmd.Parameters.AddWithValue("@Код2", ido)

cmd.ExecuteNonQuery()

ido += 1

End While

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub loadData()

Try

Dim cmd As New MySqlCommand("Select \* from расписание", constring)

DataGridView1.Rows.Clear()

dr = cmd.ExecuteReader

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), (dr.Item("Прибытие")), (dr.Item("Отправление")), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CountRec()

End Sub

Public Sub updateData()

Try

Dim cmd As New MySqlCommand("UPDATE `расписание` SET `Номер\_поезда`=@Номер\_поезда, `До`=@До, `От`=@От, `Прибытие`=@Прибытие, `Отправление`=@Отправление, `Дата\_составления`=@Дата\_составления, `Вместимость`=@Вместимость WHERE `Код`=@Код", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Номер\_поезда", txtNumT.Text)

cmd.Parameters.AddWithValue("@До", txtTo.SelectedItem)

cmd.Parameters.AddWithValue("@От", txtFrom.SelectedItem)

splitDateToSQL(DateTimePicker1, picker)

cmd.Parameters.AddWithValue("@Прибытие", picker)

splitDateToSQL(DateTimePicker2, picker)

cmd.Parameters.AddWithValue("@Отправление", picker)

splitDateToSQL(txtDateCreate, picker)

cmd.Parameters.AddWithValue("@Дата\_составления", picker)

cmd.Parameters.AddWithValue("@Вместимость", CInt(txtNumPass.Text))

cmd.Parameters.AddWithValue("@Код", txtido.Text)

cmd.ExecuteNonQuery()

MsgBox("Процедура изменения выполненена успешно")

loadData()

Catch ex As Exception

MsgBox(ex.Message)

End Try

picker = ""

End Sub

Public Sub deleteData()

If MsgBox("Вы уверенны, что хотите удалить данную запись", MsgBoxStyle.Question + vbYesNo, Title:="Удаление записи") = vbYes Then

Try

Dim cmd As New MySqlCommand("DELETE FROM `расписание` WHERE `Код`= @Код", constring)

cmd.Parameters.Clear()

cmd.Parameters.AddWithValue("@Код", CInt(txtido.Text))

cmd.ExecuteNonQuery()

MsgBox("Процедура удаления выполненена успешно")

Catch ex As Exception

MsgBox(ex.Message)

End Try

Else

Return

End If

updateId(CInt(txtido.Text))

loadData()

bclears()

CountRec()

End Sub

Public Sub CountRec(Optional ByRef count As Integer = 0)

Dim cmd As New MySqlCommand("Select COUNT(\*) as count from расписание", constring)

cmd.Parameters.Clear()

dr = cmd.ExecuteReader

While dr.Read()

txtido.Text = dr.Item("count") + 1

End While

dr.Dispose()

cmd.Parameters.Clear()

cmd.CommandText = "Select MAX(`Код`) as id from расписание"

dr.Close()

dr = cmd.ExecuteReader

While dr.Read()

count = CInt(dr.Item("id"))

End While

dr.Dispose()

End Sub

Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

txtido.Text = DataGridView1.CurrentRow.Cells(0).Value

txtNumT.Text = DataGridView1.CurrentRow.Cells(1).Value

txtFrom.SelectedItem = DataGridView1.CurrentRow.Cells(2).Value

txtTo.SelectedItem = DataGridView1.CurrentRow.Cells(3).Value

splitDateToText(DataGridView1.CurrentRow.Cells(4).Value, picker)

DateTimePicker1.Text = picker

splitDateToText(DataGridView1.CurrentRow.Cells(5).Value, picker)

DateTimePicker2.Text = picker

txtNumPass.Text = DataGridView1.CurrentRow.Cells(7).Value

bSave.Enabled = False

bDelete.Enabled = True

bChange.Enabled = True

End Sub

Private Sub TextBox1\_TextChanged(sender As Object, e As EventArgs) Handles txtSearch.TextChanged

Try

Dim cmd As New MySqlCommand("Select \* from расписание where Код like '" & txtSearch.Text & "' or Номер\_поезда like '" & txtSearch.Text & "%' or До like '" & txtSearch.Text & "%' or Дата\_составления like '" & txtSearch.Text & "%' or От like '" & txtSearch.Text & "%' or Прибытие like '" & txtSearch.Text & "%' or Отправление like '" & txtSearch.Text & "%' or Вместимость like '" & txtSearch.Text & "%'", constring)

dr = cmd.ExecuteReader

DataGridView1.Rows.Clear()

While dr.Read()

DataGridView1.Rows.Add(CInt(dr.Item("Код")), CInt(dr.Item("Номер\_поезда")), dr.Item("До"), dr.Item("От"), dr.Item("Прибытие"), dr.Item("Дата\_составления"), CInt(dr.Item("Вместимость")))

End While

dr.Dispose()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub ВыйтиToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ВыйтиToolStripMenuItem.Click

If MsgBox("Вы уверенны, что хотите закрыть приложение?", MsgBoxStyle.Question + vbYesNo, Title:="Выход из приложения") = vbYes Then

Application.Exit()

Else

Return

End If

End Sub

Private Sub bChange\_Click\_1(sender As Object, e As EventArgs) Handles bChange.Click

ChangeRecSche.Show()

End Sub

Private Sub bDelete\_Click\_1(sender As Object, e As EventArgs) Handles bDelete.Click

deleteData()

End Sub

Private Sub bClear\_Click\_1(sender As Object, e As EventArgs) Handles bClear.Click

bclears()

CountRec()

bSave.Enabled = True

bDelete.Enabled = False

End Sub

Public Sub bclears()

txtFrom.SelectedItem = Nothing

txtTo.SelectedItem = Nothing

DateTimePicker1.Text = ""

DateTimePicker2.Text = ""

txtNumT.Text = ""

txtNumPass.Text = ""

txtido.Text = ""

End Sub

Private Sub bSave\_Click(sender As Object, e As EventArgs) Handles bSave.Click

AddRecSche.Show()

loadData()

End Sub

Private Sub txtNumT\_Click(sender As Object, e As EventArgs) Handles txtNumT.Click

Dim cmd As New MySqlCommand("Select \* from поезда", constring)

dr = cmd.ExecuteReader

txtNumT.Items.Clear()

While dr.Read()

txtNumT.Items.Add(dr.Item("Номер\_поезда"))

End While

dr.Dispose()

End Sub

End Class