**Министерство образования и науки Российской федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет технологий и управления имени К.Г. Разумовского (Первый казачий университет)». Университетский колледж информационных технологий**

Специальность 09.02.03 «Программирование в компьютерных системах»

КУРСОВОЙ ПРОЕКТ

ПМ.03 «Разработка и администрирование баз данных»

МДК.03.01 «Технология разработки программного обеспечения»

Тема: Разработка мобильного мессенджера на сайте «Вконтакте»

**Пояснительная записка**

**УКИТ.09.02.03.2018.403.019ПЗ**

|  |  |  |
| --- | --- | --- |
| Группа | П-403 |  |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (личная подпись) | Самсонов Е. С. |
| Руководитель проекта | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (личная подпись) | Кузьменко С.Ю. |
| Нормоконтроль | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (личная подпись) | Гусева Е. Л. |
|  |  |  |

Москва

2019

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc515454948)

[ОБЩАЯ ЧАСТЬ 5](#_Toc515454949)

[1.1 Назначение и цели создания системы 5](#_Toc515454950)

[1.2 Анализ предметной области 5](#_Toc515454951)

[1.3 Жизненный цикл базы данных 7](#_Toc515454952)

[1.4 Выбор и характеристика СУБД 7](#_Toc515454953)

[1.5 Выбор и характеристика среды разработки приложения 8](#_Toc515454954)

[СПЕЦИАЛЬНАЯ ЧАСТЬ 10](#_Toc515454955)

[2.1 Постановка задачи 10](#_Toc515454956)

[2.2 Архитектура информационной системы 10](#_Toc515454957)

[2.3 Логическая модель базы данных 12](#_Toc515454958)

[2.4 Нормализация таблиц 12](#_Toc515454959)

[2.5 Описание таблиц 13](#_Toc515454960)

[2.7 Разработка приложения 20](#_Toc515454961)

[2.8 Инструкция пользователю 24](#_Toc515454962)

[ЗАКЛЮЧЕНИЕ 29](#_Toc515454963)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 31](#_Toc515454964)

[ПРИЛОЖЕНИЕ А Cкрипты на создание и заполнение базы данных 32](#_Toc515454965)

[ПРИЛОЖЕНИЕ Б Исходный код 50](#_Toc515454966)

ВВЕДЕНИЕ

С появлением мессенджеров появилась возможность обмениваться сообщениями и цеплять телочек даже если вы урод, это позволило социофобам не выходить из дома и решать вопросы лежа в своей уютной кроватке.

Так как раньше вся информация была на бумаге, то для того, чтобы найти необходимую информацию, необходимо было изрядно постараться, а с появлением баз данных этот процесс занимает несколько секунд.

Цель курсового проекта – автоматизация процесса резервирования билета на поезд.

Основные задачи проекта:

1. анализ предметной области;
2. определение основных объектов, необходимых для решения данной задачи;
3. разработка базы данных средствами MariaDB 10.2.13;
4. создание приложения средствами C#;
5. тестирование и отладка приложения;
6. защита приложения.

Возможности приложения:

* резервирование места на поезд;
* получение билета в формате pdf для распечатывания;
* администрирование аккаунтов пользователей;
* управление данными о поездах, вагонах, маршрутах, остановках, прибытиях и отправлениях поездов.

Используемые методы при разработке приложения:

1. Анализ;
2. Моделирование;
3. Сравнение;
4. Объектно-ориентированный подход к реализации программы.

ОБЩАЯ ЧАСТЬ

* 1. Назначение и цели создания системы

Назначение курсового проекта – автоматизация процесса резервирования билета на поезд.

Цель – реализация учебного плана и формирование профессиональных компетенций по модулю ПМ02, таких как:

* + разрабатывать объекты базы данных;
    - реализовывать базу данных в конкретной системе управления базами данных (СУБД);
    - решать вопросы администрирования базы данных;
* реализовывать методы и технологии защиты информации в базах данных.
  1. Анализ предметной области

Самые первые информационные системы были предназначены для резервирования билетов, например, «rzd.ru».

Непосредственно эта база данных создаётся для быстрого резервирования билетов на поезд. База данных должна содержать данные о пассажирах, поездах, маршрутах, остановках и предоставлять возможность получать разнообразные отчёты.

В соответствии с предметной областью система строится с учётом следующих особенностей:

* один пассажир может резервировать сразу же несколько мест;
* каждый пассажир уникален и идентифицируется с помощью серии и номера паспорта;
* нужно предусмотреть возможность поиска маршрутов по пункту отправления и пункту назначения.
* по окончанию резервирования, пассажиру формируется билет для распечатывания;
* нужно предусмотреть, что вагоны имеют разные типы, например, есть плацкартные, СВ, купейные и вагоны-рестораны;
* необходимо предусмотреть, что в вагоне-ресторане резервирование места невозможно;

Выделим базовые сущности этой предметной области:

* **Пассажиры**. Атрибуты пассажиров – идентификатор пассажира, ФИО, пол, паспортные данные.
* **Поезда**. Атрибуты поездов – номер поезда, количество вагонов, номер типа поезда, идентификатор маршрута.
* **Типы поездов**. Атрибуты типов поездов – номер типа поезда, тип поезда.
* **Вагоны**. Атрибуты вагонов – номер поезда, номер вагона, номер типа вагона.
* **Типы вагонов.** Атрибуты типов вагонов – номер типа вагона, тип вагона, кол-во мест, цена за билет.
* **Маршруты**. Атрибуты маршрутов – идентификатор маршрута, количество остановок.
* **Места**. Атрибуты мест – номер места, номер поезда, номер вагона.
* **Прибытия.** Атрибуты прибытия – идентификатор прибытия, идентификатор остановки, идентификатор маршрута, время прибытия, дата прибытия.
* **Отправления.** Атрибуты отправления – идентификатор отправления, идентификатор остановки, идентификатор маршрута, время отправления, дата отправления.
* **Остановки.** Атрибуты остановок – идентификатор маршрута, идентификатор остановки, название остановки.
* **Резервирование.** Атрибуты резервирования – идентификатор резервирования, идентификатор пассажира, идентификатор места, идентификатор отправления, идентификатор прибытия.
  1. Жизненный цикл базы данных

Жизненный цикл – это процесс проектирования, реализации и управления базой данных. Он состоит из:

1. Анализа предметной области:
   1. Планирование разработки (01.03.2018);
   2. Определение требований (02.03.2018);
   3. Сбор и анализ требований (03.03.2018 – 10.03.2018);
2. Проектирование базы данных:
   1. Концептуальная модель (11.03.2018 – 11.03.2018);
   2. Логическая модель (12.03.2018 – 28.03.2018);
   3. Физическая модель (12.03.2018 – 28.03.2018);
3. Реализации. В процессе реализации происходит написание всех SQL запросов и разработка приложения для работы с базой данных (28.03.2018 – 19.05.2018);
4. Тестирования (01.05.2018 – настоящее время);
5. Эксплуатации и сопровождения (в ближайшем будущем).
   1. Выбор и характеристика СУБД

Поскольку в данной базе данных хранятся персональные данные пользователей, такие как: номер и серия паспорта, то необходимо выбрать максимально защищенную БД, поэтому в качестве БД был выбран MySQL, который славится своей защищенностью, но так как стандартный MySQL не поддерживает роли пользователей, то было решено выбрать модификацию – MariaDB, в которой роли присутствуют.

Особенности:

1. Быстрая работа;
2. Многопоточность;
3. Высокая надежность;
4. Высокая защита базы данных;
5. Не требователен к ресурсам, поэтому для содержания базы данных не нужен высокопроизводительный сервер;
6. Поддержка транзакций;
7. Открытый исходный код;

Недостатки:

1. Версия для коммерческого использования является платной;
   1. Выбор и характеристика среды разработки приложения

Поскольку разработка достаточно объемного приложения ведется в короткие сроки, то необходим язык с максимально высокой скоростью разработки, таким является язык C#, который позволяет максимально быстро и просто делать приложения. В качестве системы для построения клиентских приложений был выбран достаточно старый WPF, так как необходимо максимально охватить количество компьютеров, которые будут работать с разрабатываемым приложением. Поддерживаются все новые версии Windows, начиная с Windows 7, в отличии от нового UWP.

В качестве разметки интерфейса используется система, под названием XAML. Она обладает очень гибкой настройкой, поэтому приложение будет с актуальным для 2018 года дизайном, в отличии от Windows Forms.

Так же, в C# WPF присутствует официальный драйвер для работы с базой данных MySQL, поэтому c# идеально подходит для MySQL.

Так же можно еще выделить некоторые особенности:

1. Поддержка объектно-ориентированного программирования;
2. Ориентация на безопасность кода;
3. Активно развивается.

Недостатки:

1. Приложения, разработанные на с# работают только на компьютерах с операционной системой Windows;
2. Приложение, разработанное с помощью с# обладает небольшим быстродействием, с++ намного быстрее. Однако, в данном приложении весь функционал реализован с помощью запросов. Приложение только получает готовые результаты и выводит их, поэтому потеря производительности будет минимальной.

СПЕЦИАЛЬНАЯ ЧАСТЬ

* 1. Постановка задачи

Спроектировать средствами MariaDB базу данных и создать средствами C# приложение, позволяющее зарезервировать железнодорожные билеты. В базе данных должна храниться следующая информация: анкетные и паспортные данные пассажира, номер маршрута, дата и время пункта отправления и пункта назначения, тип места (плацкарт, купе, СВ), пункт отправления, пункт назначения, тип поезда, количество вагонов, количество посадочных мест, номер места, цена билета. Предусмотреть возможность поиска маршрутов по пункту отправления и пункту назначения (как вместе, так и по отдельности), добавления, удаления, редактирования записей. Предусмотреть возможность формирования проданных билетов. Обеспечить защиту базы данных и приложения.

* 1. Архитектура информационной системы



Рисунок 1 – Архитектура ИС

Так как архитектура информационной системы является клиент – сервер, то у неё есть преимущества, такие как:

* все запросы выполняются на стороне сервера, а пользователь получает только готовый результат;
* приложением можно пользоваться даже на самых малопроизводительных персональных компьютерах;
* все данные хранятся на стороне сервера;
* разграниченный доступ клиентам со стороны сервера.

Недостатки:

* неработоспособность сервера может сделать неработоспособным всю вычислительную сеть;
* поддержка работы требует отдельного специалиста – системного администратора.
  1. Логическая модель базы данных

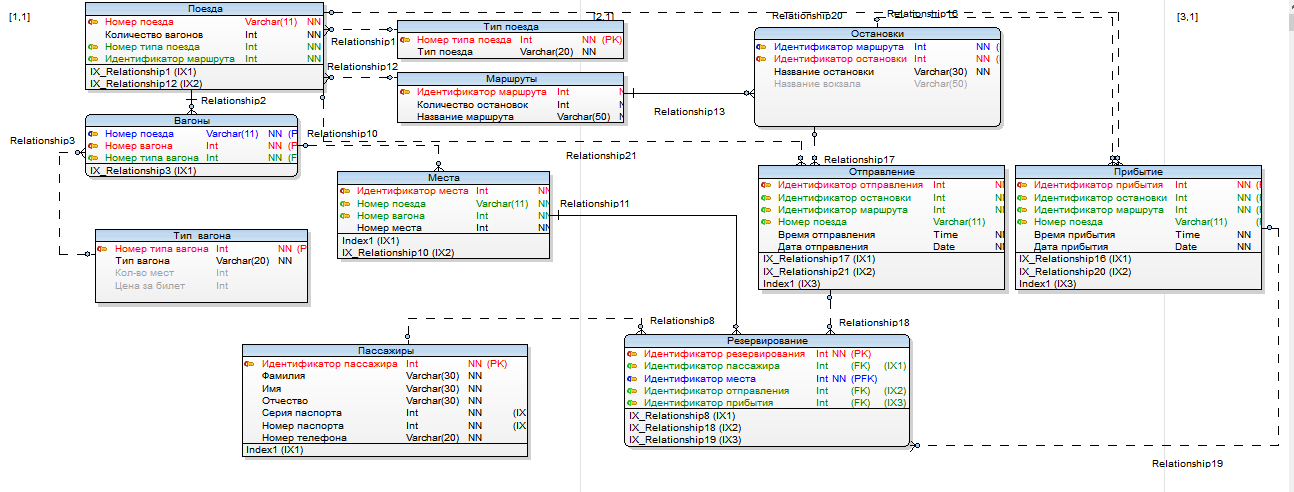


Рисунок 2 – Логическая модель базы данных

* 1. Нормализация таблиц

Все таблицы приведены к 1, 2 и 3 нормальным формам так как в базе данных только атомарные атрибуты, которые невозможно разделить на более простые атрибуты, а в таблицах нет не ключевых атрибутов, которые не зависят от ключа, а не ключевые атрибуты зависят от первичного ключа.

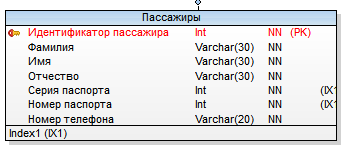


Рисунок 3 ― Таблица пассажиров, которую невозможно разделить на более простые атрибуты и они атомарны.

* 1. Описание таблиц

Ниже приведены таблицы с описанием их атрибутов. Введены следующие обозначения:

* + AI – AUTO INCREMENT
  + PK – PRIMARY KEY
  + FK – FOREIGN KEY
  + NN – NOT NULL
  + UNIQUE – UNIQUE

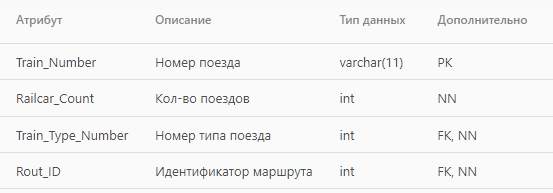


Рисунок 4 ― Таблица «Поезда»

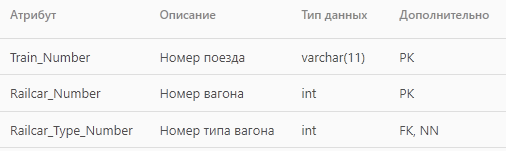


Рисунок 5 ― Таблица «Вагоны»

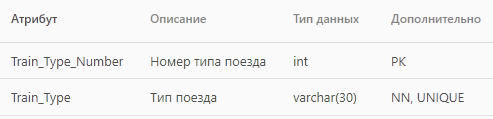


Рисунок 6 ― Таблица «Типы поездов»

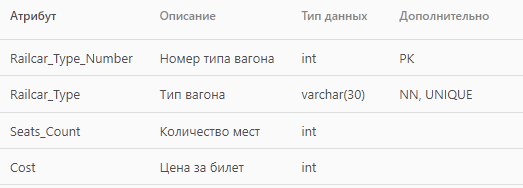


Рисунок 7 ― Таблица «Типы вагонов»

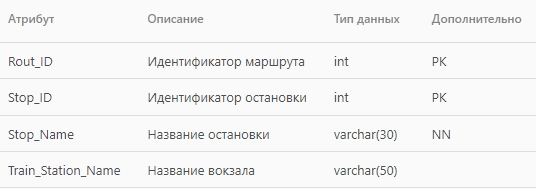


Рисунок 8 ― Таблица «Остановки»

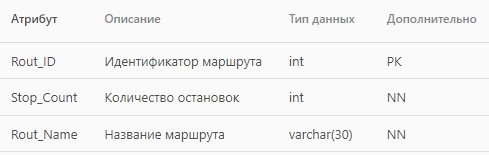


Рисунок 9 ― Таблица «Маршруты»

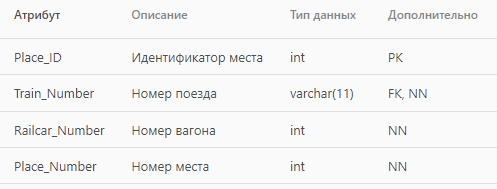


Рисунок 10 ― Таблица «Места»

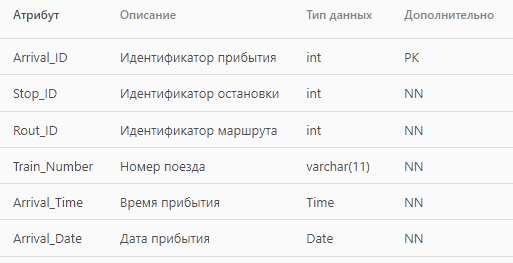


Рисунок 11 ― Таблица «Прибытия»

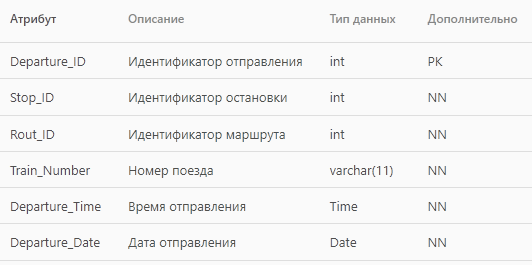


Рисунок 12 ― Таблица «Отправления»

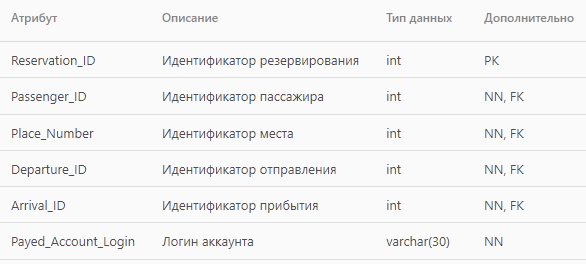


Рисунок 13 ― Таблица «Резервирования»

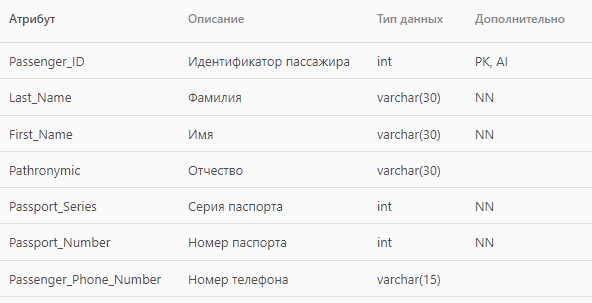


Рисунок 14 ― Таблица «Пассажиры»

* 1. **Другие объекты базы данных (представления**, **хранимые процедуры, триггеры, имена входа, пользователи)**

Список процедур:

* Available\_For\_Planting\_Seats – показывает список свободных мест;
* Available\_Railcar\_Types – показывает список доступных вагонов;
* CancelTicket – отмена резервирования билета;
* EmployPlaces – резервирование на поезд;
* EmptyProcForBlockedUsers – пустая процедура;
* FindPassengerWithPersonalData - поиск пассажира по серии и номеру паспорта;
* FindRout – поиск маршрута;
* newFindTrainList – возвращает список поездов;
* register\_createuser – регистарация пользователя;
* ThrowAllStations – возвращает все остановки;
* throwAvailableTicketsWithInfo – возвращает список купленных билетов;
* throwCountAvailableTickets – возвращает число купленных билетов;
* throwPassengerInfo – возвращает информацию о пассажире;
* throwRailcarInfo – возвращает информацию о вагоне;
* ThrowTrainNumberList – возвращает список доступных билетов;
* TrainInfoTwo – более расширенный показ информации о поезде
* DISPATCHER\_AddArrival – добавление прибытия;
* DISPATCHER\_AddDeparture – добавление отправления;
* DISPATCHER\_AddRailcar – добавление вагона;
* DISPATCHER\_AddTrain – добавление поезда;
* DISPATCHER\_AddStop – добавление остановки;
* DISPATCHER\_DropArrival – удаление прибытия;
* DISPATCHER\_DropDeparture – удаление отправления;
* DISPATCHER\_DropRailcar – удаление вагона;
* DISPATCHER\_DropTrain – удаление поезда;
* DISPATCHER\_DropRout – удаление маршрута;
* DISPATCHER\_DropStop – удаление остановки;
* DISPATCHER\_ShowArrivals – показ прибытий;
* DISPATCHER\_ShowDepartures – показ отправлений;
* DISPATCHER\_ShowRailcars – показ вагонов;
* DISPATCHER\_ShowTrains – показ поездов;
* DISPATCHER\_ShowRoutes – показ маршрутов;
* DISPATCHER\_ShowStops – показ остановок;
* DISPATCHER\_UpdateArrival – редактирование прибытия;
* DISPATCHER\_UpdateDeparture – редактирование отправления;
* DISPATCHER\_UpdateRailcar – редактирование вагона;
* DISPATCHER\_UpdateTrain – редактирование поезда;
* DISPATCHER\_UpdateRout – редактирование маршрута;
* DISPATCHER\_UpdateStop – редактирование остановки;
* \_DISPATCHER\_ThrowAvailableRailcarList – показ доступных вагонов;
* \_DISPATCHER\_ThrowAvailableStopsFindByRout – показ доступных остановок;
* \_DISPATCHER\_ThrowTrainsFindByRout – показ доступных поездов;
* \_DISPATCHER\_ThrowMaxStopCount – показ максимального количества остановок;
* \_DISPATCHER\_ThrowRailcarListCount – показ списка вагонов;
* \_DISPATCHER\_ThrowRailcarTypesList – показ списка типов вагонов;
* \_DISPATCHER\_ThrowRoutsNames – показ маршрутов
* \_DISPATCHER\_ThrowStopNumber – показ номера остановки
* \_DISPATCHER\_ThrowTrainList – показ поездов
* \_DISPATCHER\_ThrowTrainTypes – показ типов поездов
* ADMIN\_ChangeRole – изменение роли;
* ADMIN\_CreateUser – создание пользователя;
* ADMIN\_RefreshPrivileges – выдача прав;
* ADMIN\_ThrowRoles – показ всех ролей;
* ADMIN\_ThrowUsers – показ всех пользователей;

Список фунцкий:

* DISPATCHER\_AddRout – добавление маршрута;
* FindPassenger – поиск пассажира;
* GetArrivalID – получение arrivalid;
* GetDepartureID – получение departureid;
* PassengerAddToDB – добавление пассажира в бд;
* ThrowArrivalStopID – получение arrivalstopid;
* ThrowDepartureStopID – получение departurestopid;
* ThrowEmployedIDs - Вспомогательная функция для резервирования места
* ThrowRoutID – получение routid;

Список триггеров:

* RemovePhantomSeats – удаление места из Places, если удаляется резервирование;

Таблица 1. Список пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| **Логин** | **Пароль** | **Роль** | **Описание** |
| BestAdmin | 1111 | Администратор | Управляет пользователями |
| BestDispatcher | 1111 | Диспетчер | Управляет информацией о поездах |
| BestUser | 1111 | Пользователь | Резервирует места на поезд |
| BestBlocked | 1111 | Заблокированный | Ничего не может делать |

* 1. Разработка приложения
     1. Структурная схема приложения

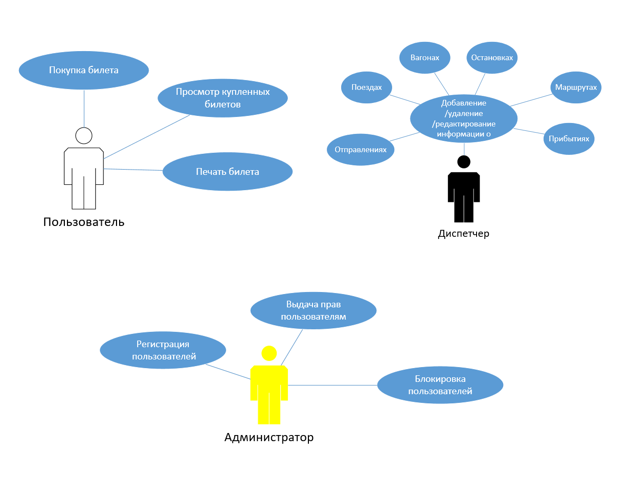


Рисунок 15 ― Диаграмма вариантов использования

Классы в приложении:

* Authorization.cs – интерфейс авторизации;
* ApplicationLogic.cs – логика приложения;
* BestErrors.cs – обработчик ошибок;
* Dispatcher\_Interface\_In\_To\_Face.cs – интерфейс диспетчера;
* Menu.cs – интерфейс пользователя;
* ReservationControl.cs – интерфейс резервирования;
* TableFillKostil.cs – костыль;
  + 1. Функции приложения

У пользователя:

* резервирование места на поезд;
* показ купленных билетов;
* возможность сохранить купленный билет в pdf файле, чтобы его можно было распечатать;
* создание персонального аккаунта в базе данных;
* возможность отменить поездку.

У диспетчера редактирование информации:

* о поездах;
* о вагонах;
* об остановках;
* о маршрутах;
* о приездах поездов;
* о отправлениях поездов;
  + 1. Решение главной задачи проекта

Необходимо:

1. Получить список поездов, которые будут ехать по нужному маршруту

Эквивалент запроса на SQL: call newFindTrainList([Номер\_Маршрута], [Название\_остановки\_отправления], [Дата\_поездки], [Название\_остановки\_прибытия]);

1. определить количество оставшихся билетов на каждый поезд:

Эквивалент запроса на SQL: SELECT CheckFreeSeatsCount([№ поезда]);

1. осуществить бронирование билетов c учётом свободных мест.

Эквивалент запроса на SQL: Call EmployPlaces([Номер поезда], [Номер вагона], [Идентификатор места], [Идентификатор пассажира], [Идентификатор прибытия], [Идентификатор отправления]);

* + 1. Описание процесса отладки и тестирования

В алгоритме для резервирования места, нужно сделать так, чтобы нельзя было зарезервировать место, которое уже было зарезервировано до тех пор, пока первый пассажир, зарезервировавший место не доедет до нужной ему остановки. Для этого было решено сделать приложение, которое будет эмулировать алгоритм резервирования. Когда был написан алгоритм, который прошел все тесты для резервирования в приложении с#, только тогда было решено внедрить его в процедуру для резервирования.

В процессе разработки приложения было допущено большое количество логических ошибок как в процедурах, так и в самом коде приложения, например, в показе списка доступных поездов, была из-за спешки допущена ошибка, из-за которой зарезервировать место можно было только на 1 поезд из всего списка, в настоящий момент такой проблемы нет, благодаря пошаговой отладке её удалось найти и исправить.

Непосредственно тестирование приложения происходило в 3 этапа:

1. Этап тестирования функций пользователя. На этом этапе тестировались такие функции, как поиск поезда, выбор типа вагона, вагона, места и непосредственно резервирование. В первых версиях приложения можно было резервировать места на поезда, на которые уже посадка была завершена и в базе данных не было шифрования персональных данных. В последней версии все ошибки, обнаруженные в результате тестирования уже исправлены.
2. Этап тестирования функций диспетчера. На этом этапе тестировались такие функции, как добавление/удаление/редактирование информации о поездах, маршрутах, остановках, вагонах, прибытиях и отправлениях. Во время тестирования обнаружилось, что информацию о вагоне невозможно редактировать, это связано с ошибкой в проектировании базы данных. Эта проблема актуальна до сих пор.
3. Этап тестирования функций администратора. На этом этапе тестировались такие функции, как регистрация пользователя и выдача прав пользователю. Во время тестирования ошибок обнаружено не было.
   * 1. Защита информационной системы

В приложении есть поля для ввода данных пользователей, поэтому в приложении должна иметься полная защита от sql-инъекций.

Во-первых, в приложении в потенциально уязвимых местах должны использоваться запросы с параметрами.

Во-вторых, у каждого пользователя должен быть доступ только к ограниченному списку процедур и функций, которые используются для резервирования места. Доступа к командам, таких как: “SELECT, UPDATE, DELETE, CREATE” и так далее, у пользователя не может быть в принципе.

В-третьих, у разных типов пользователей должны быть разные возможности. Например, диспетчер может пользоваться списком процедур, которые предназначены для заполнения информации о поездах, вагонах, маршрутах. Так же у диспетчера в приложении должен быть уникальный интерфейс, позволяющий удобно заполнять информацию о поездах, вагонах и маршрутах, а у обычного пользователя должен быть интерфейс, предназначенный для резервирования места и просмотра зарезервированных мест на поезд.

Пароли пользователей хешируются алгоритмом sha1(sha1(“пароль”)) и в базе данных хранятся в захешированном виде.

Так же, персональные данные пользователей зашифрованы с помощью ключа, который находится в ресурсах приложения и в базе данных хранятся в бинарном виде. Для шифрования используется алгоритм AES (Advanced Encryption Standart) Rijndael (Рэндал), который с 2003 года является стандартом шифрования в правительстве США.



Рисунок 16 – Пример серии паспорта, которая хранится в базе данных

* 1. Инструкция пользователю
     1. Общие сведения об информационной системе

Данная информационная система предназначена для быстрого и удобного резервирования места на поезд.

В приложении имеется интерфейс пользователя, диспетчера и администратора

* + 1. Требования к техническим средствам

Системные требования к компьютеру пользователя:

* Операционная система Windows (начиная с Windows 7);
* Процессор Intel Pentium n3710 или любой другой, более производительный;
* ОЗУ объёмом не менее 2 ГБ;
* Дисковой накопитель HDD (не менее 250 ГБ) или SSD (не мене 64 ГБ);
* Видеоадаптер;
* Монитор с разрешением не менее 1280х1024;
* Клавиатура и мышь или тачпад.
  + 1. Требования к программным средствам

Для корректной работы необходимо установить следующее ПО:

* базу данных MariaDB;
  + 1. Настройка информационной системы

Для установки приложения необходимо запустить исполняемый файл setup.exe из установочного диска с программой. После установки приложение можно будет найти в списке приложений под названием «Билеты».

Для настройки базы данных необходимо развернуть содержимое из файла Execute.SQL в базу данных. В процессе развертывания у Вас появятся все процедуры и функции, необходимые для стабильной работы информационной системы, а также все пользователи и их роли.

После этих двух простых действий ваша информационная система готова к работе.

Для регистрации пользователя необходимо нажать на кнопку «Регистрация», после этого откроется форма регистрации. В поля логина и пароля запрещено вводить русские символы. Хоть это и разрешено базой данных, но все равно может привести к непредсказуемым последствиям.

Пользователь после авторизации увидит (Рисунок 17) форму главного меню. В нем необходимо ввести точку отправления и назначения. Эти поля сделаны в виде списка, города в которых берутся непосредственно из самой базы данных. Если необходимого города в этом списке нет, то это значит, что необходимый город отсутствует в базе данных.

После покупки билета пользователь сможет зайти в личный кабинет (Рисунок 18) и увидеть все свои купленные билеты. В этом же личном кабинете он может распечатать свои билеты.

Диспетчер после авторизации увидит (Рисунок 19) интерфейс диспетчера. В этом интерфейсе диспетчер может удобно добавлять/редактировать/удалять информацию о поездах, вагонах, маршрутах, остановках, прибытиях и отправлениях. Все вводимые данные сверяются с данными из базы данных, поэтому диспетчер не сможет ввести тип поезда, которого не существует, или большое количество вагонов у поезда (максимальное количество вагонов – 20).

Администратор после авторизации увидит интерфейс администратора. В этом интерфейсе администратор может добавлять пользователей, менять роль пользователя. Однако, администратору не доступно удаление пользователей. Вместо этого реализовано блокирование пользователей.

* + 1. Формы ввода

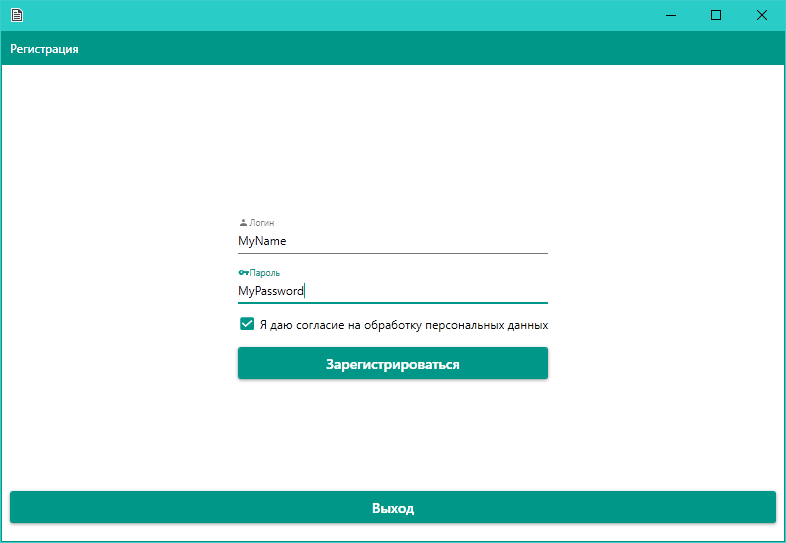


Рисунок 17 ― Форма регистрации

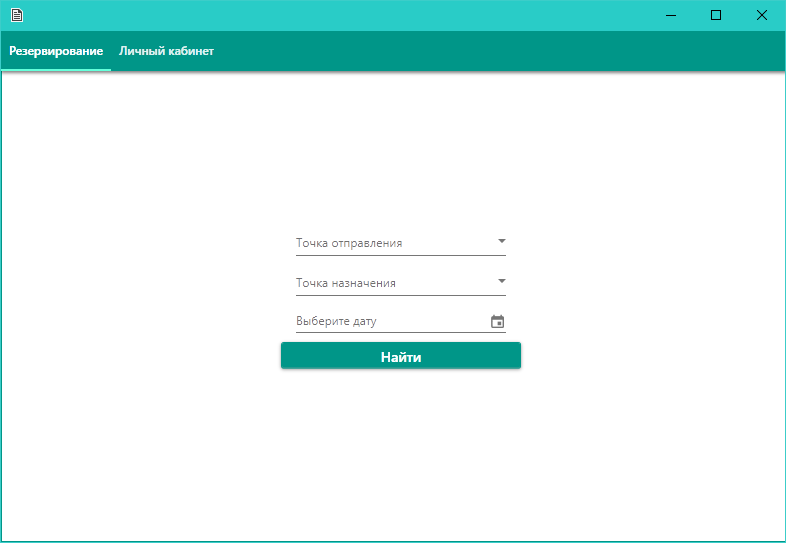


Рисунок 18 ― Форма главного меню

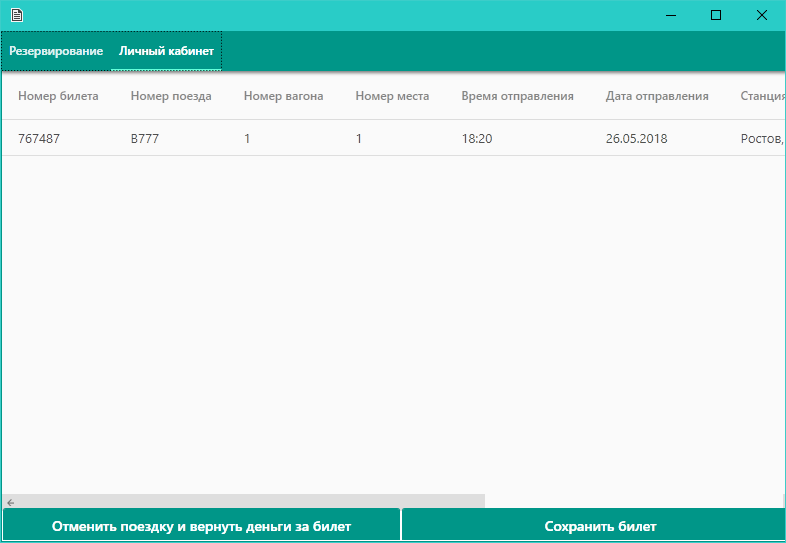


Рисунок 19 ― Личный кабинет

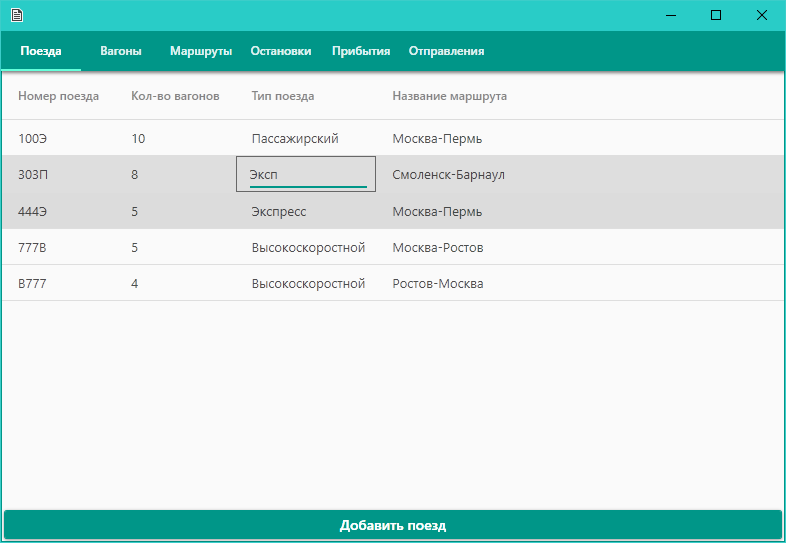


Рисунок 20 ― Интерфейс диспетчера

* + 1. Отчёты

Приложение автоматически генерирует билет на поезд.

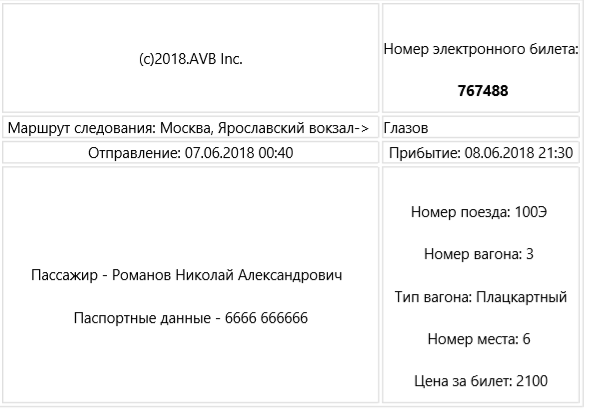


Рисунок 21 ― Пример билета на поезд

ЗАКЛЮЧЕНИЕ

Курсовой проект посвящен резервированию билетов на поезд.

В процессе выполнения курсового проекта:

1. Изучена предметная область;
2. Разработана база данных MariaDB;
3. Разработано приложение для резервирования билетов на поезд на языке C#;
4. Осуществлена защита приложения;

Возможности приложения:

* резервирование билета на поезд;
* отмена резервирования;
* печать билета на поезд;
* полное управление базой данных у диспетчера;
* полное управление пользователями у администратора;

Дальнейшее развитие темы курсового проекта предполагает подключение полноценной оплаты, более тщательное скрытие ключа для шифрования данных.

На разработку информационной системы ушло 3 месяца. Все этапы разработки приложения зафиксированы в git, а скрипты базы данных сохранялись на google drive, чтобы можно было в любой момент их редактировать

Результатом работы стала высокозащищенная информационная система, с материальным дизайном, написанная на c#. Однако, в приложении нужны доработки:

* ключ шифрования хранится в ресурсах, поэтому его можно получить с помощью программы для взлома .net приложений, например, .net reflector. Однако, даже если злоумышленник получит ключ шифрования, то ему нужно будет ещё взломать базу данных, а MariaDB это модификация MySQL, поэтому чтобы взломать используемую базу данных, необходимо сильно постараться;

Во время разработки были закреплены навыки:

* объекто-ориентированный анализ и проектирование;
* работа с WPF;
* работа с адаптивным дизайном;
* разработка базы данных с помощью MariaDB;
* использования системы контроля версий;
* защита приложения;
* защита персональных данных в базе данных;
* защита от SQL-инъекций;
* разработка приложения с материальным дизайном.

Итоги:

* В SQL (хранимых процедурах) всего написано около 1500 строк;
* В XAML (разметке приложения) всего написано около 600 строк;
* В cs (исходном коде приложения) всего написано около 2000 строк;

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1 MariaDB Документация [Электронный ресурс]. – Режим доступа: https://mariadb.com/kb/en/library/documentation/ (28.03.2018);

2 WPF Документация [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/framework/wpf/ (23.04.2018);

3 Интернет – форум Habrahabr [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/> (05.05.2018).

ПРИЛОЖЕНИЕ А  
Cкрипты на создание и заполнение базы данных

DROP DATABASE IF EXISTS rzd;

CREATE DATABASE rzd;

DROP ROLE IF EXISTS Admin;

DROP ROLE IF EXISTS Blocked;

DROP ROLE IF EXISTS RZD\_Dispatcher;

DROP ROLE IF EXISTS user;

CREATE ROLE Admin;

CREATE ROLE Blocked;

CREATE ROLE RZD\_Dispatcher;

CREATE ROLE user;

DROP USER IF EXISTS BestAdmin@localhost;

CREATE USER BestAdmin@localhost IDENTIFIED BY '1111';

DROP USER IF EXISTS BestUser@localhost;

CREATE USER BestUser@localhost IDENTIFIED BY '1111';

DROP USER IF EXISTS BestDispatcher@localhost;

CREATE USER BestDispatcher@localhost IDENTIFIED BY '1111';

DROP USER IF EXISTS BestBlocked@localhost;

CREATE USER BestBlocked@localhost IDENTIFIED BY '1111';

grant Admin to BestAdmin@localhost;

set default role Admin for BestAdmin@localhost;

grant user to BestUser@localhost;

set default role user for BestUser@localhost;

grant RZD\_Dispatcher to BestDispatcher@localhost;

set default role RZD\_Dispatcher for BestDispatcher@localhost;

grant Blocked to BestBlocked@localhost;

set default role Blocked for BestBlocked@localhost;

USE rzd;

CREATE TABLE train\_types (

Train\_Type\_Number INT(11) NOT NULL AUTO\_INCREMENT,

Train\_Type VARCHAR(20) NOT NULL,

PRIMARY KEY (Train\_Type\_Number)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 11,

AVG\_ROW\_LENGTH = 3276,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

DELIMITER $$

CREATE DEFINER = 'root'@'localhost'

PROCEDURE \_DISPATCHER\_ThrowTrainTypes()

BEGIN

SELECT DISTINCT

Train\_Type

FROM train\_types;

END$$

DELIMITER ;

CREATE TABLE routes (

Rout\_ID INT(11) NOT NULL AUTO\_INCREMENT,

Stop\_Count INT(11) NOT NULL,

Rout\_Name VARCHAR(50) NOT NULL,

PRIMARY KEY (Rout\_ID)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 8,

AVG\_ROW\_LENGTH = 8192,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE routes

ADD UNIQUE INDEX UK\_routes\_Rout\_Name(Rout\_Name);

DELIMITER $$

DELIMITER ;

CREATE TABLE trains (

Train\_Number VARCHAR(30) NOT NULL,

Railcar\_Count INT(11) NOT NULL,

Train\_Type\_Number INT(11) NOT NULL,

Rout\_ID INT(11) NOT NULL,

PRIMARY KEY (Train\_Number)

)

ENGINE = INNODB,

AVG\_ROW\_LENGTH = 16384,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE trains

ADD INDEX IX\_Relationship1(Train\_Type\_Number);

ALTER TABLE trains

ADD INDEX IX\_Relationship12(Rout\_ID);

ALTER TABLE trains

ADD CONSTRAINT Relationship1 FOREIGN KEY (Train\_Type\_Number)

REFERENCES train\_types(Train\_Type\_Number) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE trains

ADD CONSTRAINT Relationship12 FOREIGN KEY (Rout\_ID)

REFERENCES routes(Rout\_ID) ON DELETE CASCADE ON UPDATE CASCADE;

DELIMITER $$

CREATE TABLE departures (

Departure\_ID INT(11) NOT NULL AUTO\_INCREMENT,

Stop\_ID INT(11) NOT NULL,

Rout\_ID INT(11) NOT NULL,

Train\_Number VARCHAR(11) NOT NULL,

Departure\_Time TIME NOT NULL,

Departure\_Date DATE NOT NULL,

PRIMARY KEY (Departure\_ID)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 32,

AVG\_ROW\_LENGTH = 1092,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE departures

ADD UNIQUE INDEX Index1(Stop\_ID, Rout\_ID, Train\_Number);

ALTER TABLE departures

ADD CONSTRAINT Relationship17 FOREIGN KEY (Stop\_ID, Rout\_ID)

REFERENCES stops(Stop\_ID, Rout\_ID) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE departures

ADD CONSTRAINT Relationship21 FOREIGN KEY (Train\_Number)

REFERENCES trains(Train\_Number);

DELIMITER ;

CREATE TABLE arrivals (

Arrival\_ID INT(11) NOT NULL AUTO\_INCREMENT,

Stop\_ID INT(11) NOT NULL,

Rout\_ID INT(11) NOT NULL,

Train\_Number VARCHAR(11) NOT NULL,

Arrival\_Time TIME NOT NULL,

Arrival\_Date DATE NOT NULL,

PRIMARY KEY (Arrival\_ID)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 32,

AVG\_ROW\_LENGTH = 1024,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE arrivals

ADD UNIQUE INDEX Index1(Stop\_ID, Rout\_ID, Train\_Number);

ALTER TABLE arrivals

ADD CONSTRAINT Relationship16 FOREIGN KEY (Stop\_ID, Rout\_ID)

REFERENCES stops(Stop\_ID, Rout\_ID) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE arrivals

ADD CONSTRAINT Relationship20 FOREIGN KEY (Train\_Number)

REFERENCES trains(Train\_Number);

DELIMITER $$

DELIMITER ;

CREATE TABLE railcar\_types (

Railcar\_Type\_Number INT(11) NOT NULL AUTO\_INCREMENT,

Railcar\_Type VARCHAR(20) NOT NULL,

Seats\_Count INT(11) DEFAULT NULL,

Cost INT(11) DEFAULT NULL,

PRIMARY KEY (Railcar\_Type\_Number)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 6,

AVG\_ROW\_LENGTH = 3276,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

CREATE TABLE railcars (

Train\_Number VARCHAR(11) NOT NULL,

Railcar\_Number INT(11) NOT NULL,

Railcar\_Type\_Number INT(11) NOT NULL,

PRIMARY KEY (Train\_Number, Railcar\_Number)

)

ENGINE = INNODB,

AVG\_ROW\_LENGTH = 564,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE railcars

ADD CONSTRAINT Relationship2 FOREIGN KEY (Train\_Number)

REFERENCES trains(Train\_Number) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE railcars

ADD CONSTRAINT Relationship3 FOREIGN KEY (Railcar\_Type\_Number)

REFERENCES railcar\_types(Railcar\_Type\_Number);

DELIMITER $$

DELIMITER ;

CREATE TABLE places (

Place\_ID INT(11) NOT NULL AUTO\_INCREMENT,

Train\_Number VARCHAR(11) NOT NULL,

Railcar\_Number INT(11) NOT NULL,

Place\_Number INT(11) NOT NULL,

PRIMARY KEY (Place\_ID)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 96,

AVG\_ROW\_LENGTH = 3276,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE places

ADD CONSTRAINT Relationship10 FOREIGN KEY (Train\_Number, Railcar\_Number)

REFERENCES railcars(Train\_Number, Railcar\_Number) ON DELETE CASCADE ON UPDATE CASCADE;

CREATE TABLE passengers (

Passenger\_ID INT(11) NOT NULL AUTO\_INCREMENT,

Last\_Name VARCHAR(30) NOT NULL,

First\_Name VARCHAR(30) NOT NULL,

Pathronymic VARCHAR(30) NOT NULL,

Passport\_Series VARBINARY(255) NOT NULL,

Passport\_Number VARBINARY(255) NOT NULL,

Passenger\_Phone\_Number VARBINARY(255) NOT NULL,

PRIMARY KEY (Passenger\_ID)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 70,

AVG\_ROW\_LENGTH = 1489,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE passengers

ADD UNIQUE INDEX Index1(Passport\_Series, Passport\_Number);

DELIMITER $$

CREATE TABLE reservation (

Reservation\_ID INT(11) NOT NULL AUTO\_INCREMENT,

Passenger\_ID INT(11) NOT NULL,

Place\_Number INT(11) NOT NULL,

Departure\_ID INT(11) NOT NULL,

Arrival\_ID INT(11) NOT NULL,

Payed\_Account\_Login VARCHAR(30) NOT NULL,

PRIMARY KEY (Reservation\_ID)

)

ENGINE = INNODB,

AUTO\_INCREMENT = 767492,

AVG\_ROW\_LENGTH = 8192,

CHARACTER SET utf8,

COLLATE utf8\_general\_ci;

ALTER TABLE reservation

ADD INDEX IX\_Relationship18(Departure\_ID);

ALTER TABLE reservation

ADD INDEX IX\_Relationship19(Arrival\_ID);

ALTER TABLE reservation

ADD INDEX IX\_Relationship8(Passenger\_ID);

ALTER TABLE reservation

ADD UNIQUE INDEX UK\_reservation(Place\_Number, Departure\_ID, Arrival\_ID);

ALTER TABLE reservation

ADD CONSTRAINT Relationship11 FOREIGN KEY (Place\_Number)

REFERENCES places(Place\_ID) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE reservation

ADD CONSTRAINT Relationship18 FOREIGN KEY (Departure\_ID)

REFERENCES departures(Departure\_ID) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE reservation

ADD CONSTRAINT Relationship19 FOREIGN KEY (Arrival\_ID)

REFERENCES arrivals(Arrival\_ID) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE reservation

ADD CONSTRAINT Relationship8 FOREIGN KEY (Passenger\_ID)

REFERENCES passengers(Passenger\_ID) ON DELETE CASCADE ON UPDATE CASCADE;

DELIMITER $$

INSERT INTO train\_types VALUES(1, 'Пассажирский'),(2, 'Скорый'),(3, 'Экспресс'),(4, 'Скоростной'),(5, 'Высокоскоростной'),

INSERT INTO routes VALUES(1, 5, 'Москва-Пермь'),(2, 5, 'Смоленск-Барнаул'),(5, 3, 'Москва-Ростов'),(6, 3, 'Ростов-Москва');

INSERT INTO trains VALUES('100Э', 10, 1, 1),('303П', 8, 1, 2),('444Э', 5, 3, 1),('777В', 5, 5, 5),('В777', 4, 5, 6);

INSERT INTO railcar\_types VALUES (1, 'Плацкартный', 54, 2100),(2, 'Купе', 36, 3000),(3, 'СВ', 18, 7000),(4, 'Почтовый', NULL, NULL),(5, 'Ресторан', NULL, NULL);

INSERT INTO railcars VALUES('100Э', 3, 1),('100Э', 10, 1),('303П', 3, 1),('303П', 4, 1),('303П', 5, 1),('444Э', 1, 1),('444Э', 2, 1),('444Э', 3, 1),('100Э', 5, 2),('100Э', 8, 2),('303П', 6, 2),('303П', 8, 2),('444Э', 4, 2),('444Э', 5, 2),('777В', 3, 2),('777В', 4, 2),('В777', 1, 2),('В777', 2, 2),('В777', 3, 2),('100Э', 9, 3),('777В', 1, 3),('777В', 2, 3),('100Э', 1, 4),('100Э', 2, 4),('303П', 1, 4),('303П', 2, 4),('303П', 7, 4),('100Э', 6, 5),('В777', 4, 5);

INSERT INTO stops VALUES(1, 1, 'Москва', 'Ярославский вокзал'),(2, 1, 'Смоленск', 'Центральный вокзал'),(5, 1, 'Москва', 'Курский вокзал'),(6, 1, 'Ростов', 'Главный вокзал'),(1, 2, 'Ярославль', NULL),(2, 2, 'Иваново', NULL),(5, 2, 'Воронеж', 'Курский вокзал'),(6, 2, 'Воронеж', ''),(1, 3, 'Галич', NULL),(2, 3, 'Галич', NULL),(5, 3, 'Ростов', 'Главный вокзал'),(6, 3, 'Москва', 'Курский вокзал'),(1, 4, 'Глазов', NULL),(2, 4, 'Пермь', NULL),(1, 5, 'Пермь', NULL),(2, 5, 'Барнаул', NULL);

INSERT INTO places VALUES(87, '100Э', 5, 3),(91, 'В777', 1, 1),(92, '100Э', 3, 6),(93, '100Э', 5, 2),(94, '100Э', 3, 1);

INSERT INTO passengers VALUES(66, 'Самсонов', 'Евгений', 'Сергеевич', x'7C8BA885A07A6C89916F35604C2511D0', x'9B89D60769E0F59A26E529BEBFBF127C', x'43D7D9A380507B64EEEFF3620DF85868'),(67, 'Романов', 'Николай', 'Александрович', x'1AB8B22E7E96FBB88AC94D569196FB5A', x'4BA578B3F54F25A98AB558D67A198395', x'2CC4F2D8354BFBF15CF36B0DFEE3B4A2'),(68, 'Самсонов', 'Евгений', 'Сергеевич', x'C52A892CC5D097E23EFA24BC0DBD28A0', x'408F80DCFFC3709866F4E209663D7794', x'82819900EECE145605F9AA32C76308CE');

INSERT INTO departures VALUES(12, 2, 1, '100Э', '04:40:00', '2018-06-07'),(13, 3, 1, '100Э', '09:40:00', '2018-06-07'),(14, 4, 1, '100Э', '21:40:00', '2018-06-08'),(16, 1, 2, '303П', '00:10:00', '2018-06-07'),(17, 2, 2, '303П', '03:10:00', '2018-06-07'),(18, 3, 2, '303П', '15:10:00', '2018-06-07'),(19, 4, 2, '303П', '23:10:00', '2018-06-07'),(20, 5, 2, '303П', '05:10:00', '2018-06-08'),(21, 5, 1, '100Э', '17:30:00', '2018-06-09'),(22, 1, 5, '777В', '10:10:00', '2018-05-26'),(24, 2, 5, '777В', '15:20:00', '2018-05-26'),(25, 3, 5, '777В', '20:20:00', '2018-05-26'),(26, 1, 6, 'В777', '18:20:00', '2018-05-26'),(28, 2, 6, 'В777', '22:20:00', '2018-05-26'),(29, 3, 6, 'В777', '03:20:00', '2018-05-27'),(31, 1, 1, '100Э', '00:40:00', '2018-06-07');

INSERT INTO arrivals VALUES (12, 2, 1, '100Э', '04:30:00', '2018-06-07'),(13, 3, 1, '100Э', '09:30:00', '2018-06-07'),(14, 4, 1, '100Э', '21:30:00', '2018-06-08'),(16, 1, 2, '303П', '00:00:00', '2018-06-07'),(17, 2, 2, '303П', '03:00:00', '2018-06-07'),(18, 3, 2, '303П', '15:00:00', '2018-06-07'),(19, 4, 2, '303П', '23:00:00', '2018-06-07'),(20, 5, 2, '303П', '05:00:00', '2018-06-08'),(21, 5, 1, '100Э', '17:26:00', '2018-06-09'),(22, 1, 5, '777В', '10:00:00', '2018-05-26'),(23, 2, 5, '777В', '15:00:00', '2018-05-26'),(24, 3, 5, '777В', '20:00:00', '2018-05-26'),(25, 1, 6, 'В777', '18:00:00', '2018-05-26'),(26, 2, 6, 'В777', '22:00:00', '2018-05-26'),(27, 3, 6, 'В777', '03:00:00', '2018-05-27'),(31, 1, 1, '100Э', '00:22:00', '2018-06-07');

INSERT INTO reservation VALUES(767487, 66, 91, 29, 25, 'test@localhost'),(767490, 69, 94, 14, 13, 'ex@localhost');

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.FindPassengerWithPersonalData(IN Passport\_Series\_IN INT, IN Passport\_Number\_IN INT, KeySi TEXT)

BEGIN

SELECT Last\_Name, First\_Name, Pathronymic, AES\_DECRYPT(Passenger\_Phone\_Number, KeySi)

FROM passengers

WHERE Passport\_Series = AES\_ENCRYPT(Passport\_Series\_IN, KeySi) AND Passport\_Number = AES\_ENCRYPT(Passport\_Number\_IN, KeySi);

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.ThrowAllStations()

BEGIN

SELECT DISTINCT Stop\_Name FROM stops;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.Available\_Railcar\_Types(

IN `Train\_Number\_IN` VARCHAR(11)

)

BEGIN

SELECT DISTINCT Railcar\_Type, Cost FROM railcars

INNER JOIN railcar\_types ON railcars.Railcar\_Type\_Number = railcar\_types.Railcar\_Type\_Number

WHERE Train\_Number = Train\_Number\_IN and railcar\_types.Seats\_Count <> 0 AND railcar\_types.Cost <> 0;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.TrainInfoTwo(IN TrainNumber VARCHAR(10), IN Arrival\_ID\_IN INT, IN DepartureID\_IN INT)

BEGIN

DECLARE ArrivalTime time;

DECLARE ArrivalDate Date;

DECLARE StopDeparture int;

DECLARE StopArrival int;

DECLARE DepartureTime time;

DECLARE DepartureDate Date;

DECLARE FirstStation text;

DECLARE LastStation\_ID int;

DECLARE LastStation text;

DECLARE RoutID int;

DECLARE RailwayStation varchar(50);

SELECT arrivals.Rout\_ID INTO RoutID FROM arrivals

WHERE (Arrival\_ID = Arrival\_ID\_IN);

SELECT Stop\_ID INTO StopDeparture FROM arrivals

WHERE Arrival\_ID = Arrival\_ID\_IN;

SELECT Departure\_Time INTO DepartureTime FROM departures

WHERE Train\_Number = TrainNumber and Rout\_ID = RoutID and Stop\_ID = StopDeparture;

SELECT Stop\_ID INTO StopArrival FROM departures

WHERE Departure\_ID = DepartureID\_IN;

SELECT Arrival\_Date INTO ArrivalDate FROM arrivals

WHERE Train\_Number = TrainNumber and Rout\_ID = RoutID and Stop\_ID = StopArrival;

SELECT Arrival\_Time INTO ArrivalTime FROM arrivals

WHERE Train\_Number = TrainNumber and Rout\_ID = RoutID and Stop\_ID = StopArrival;

SELECT Arrivals.Stop\_ID INTO LastStation\_ID FROM Arrivals

WHERE (Train\_Number = TrainNumber)

ORDER BY Stop\_ID DESC LIMIT 1;

SELECT Stop\_Name INTO FirstStation FROM stops

INNER JOIN arrivals ON arrivals.Stop\_ID = stops.Stop\_ID

WHERE (stops.Rout\_ID = RoutID) AND (arrivals.Arrival\_ID = Arrival\_ID\_IN);

SELECT Stop\_Name INTO LastStation FROM stops

INNER JOIN departures ON departures.Stop\_ID = stops.Stop\_ID

WHERE (stops.Rout\_ID = RoutID) AND (departures.Departure\_ID = DepartureID\_IN);

SELECT Train\_Station\_Name INTO RailwayStation FROM stops

WHERE Stop\_ID = StopDeparture and Rout\_ID = RoutID;

IF (RailwayStation is NOT NULL) THEN

SET FirstStation = CONCAT(FirstStation, ', ', RailwayStation);

END IF;

SELECT Train\_Station\_Name INTO RailwayStation FROM stops

WHERE Stop\_ID = StopArrival and Rout\_ID = RoutID;

IF (RailwayStation is NOT NULL) THEN

SET LastStation = CONCAT(LastStation, ', ', RailwayStation);

END IF;

SELECT TrainNumber, TIME\_FORMAT(DepartureTime, '%H:%i'), TIME\_FORMAT(ArrivalTime, '%H:%i'), DATE\_FORMAT(ArrivalDate, '%d.%m.%Y'),

FirstStation, LastStation;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.FindRout(in ArrivalStation\_IN varchar(30),in DepartureStation\_IN varchar(30),in Arrival\_Date\_IN varchar(30))

BEGIN

DECLARE LastID int;

DECLARE ArrivalID int;

DECLARE DepartureID int;

DECLARE i int;

DECLARE TrainID int;

DECLARE TMPRout int;

DECLARE TMPKostil int;

DECLARE TrueDate date;

DECLARE Arrival\_Stop\_ID int;

DROP TABLE IF EXISTS TMPResultTable;

CREATE TEMPORARY TABLE TMPResultTable(RoutID\_ int);

SET trueDate = STR\_TO\_DATE(Arrival\_Date\_IN, '%d.%m.%Y');

SET i = 1;

SELECT Rout\_ID INTO LastID FROM routes

ORDER BY Rout\_ID DESC LIMIT 1;

SET LastID = LastID + 1;

WHILE (i <> LastID) DO

SELECT stops.Rout\_ID INTO TMPRout FROM Stops

INNER JOIN arrivals ON arrivals.Rout\_ID = stops.Rout\_ID

WHERE (stops.Rout\_ID = i) AND (stops.Stop\_Name = ArrivalStation\_IN) AND (arrivals.Arrival\_Date = TrueDate) LIMIT 1;

IF (TMPRout is NOT NULL) THEN

SELECT Rout\_ID INTO TMPKostil FROM Stops

WHERE (Rout\_ID = TMPRout) AND (Stop\_Name = DepartureStation\_IN);

IF (TMPKostil is NOT NULL) THEN

INSERT INTO TMPResultTable (RoutID\_) VALUES (TMPKostil);

END IF;

END IF;

SET i = i + 1;

END WHILE;

SELECT DISTINCT \* FROM TMPResultTable;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.ThrowTrainNumbersList(IN Train\_Number\_IN VARCHAR(11), IN Railcar\_Type\_IN VARCHAR(30))

BEGIN

SELECT DISTINCT Railcar\_Number FROM railcars

INNER JOIN railcar\_types ON railcars.Railcar\_Type\_Number = railcar\_types.Railcar\_Type\_Number

WHERE Train\_Number = Train\_Number\_IN AND Railcar\_Type = Railcar\_Type\_IN;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.Available\_For\_Planting\_Seats(IN TrainNumber\_IN VARCHAR(11), IN Railcar\_Number\_IN INT, IN CurrentArrivalID INT, IN CurrentDepartureID INT)

BEGIN

DECLARE MaxCount int;

DECLARE i int;

DECLARE i2 int;

DECLARE TMPPlaceID int;

DECLARE CurrentStopArrivalID int;

DECLARE CurrentStopDepartureID int;

DECLARE ExistsStopArrivalID int;

DECLARE ExistsStopDepartureID int;

DECLARE MaxCompareCount int;

DECLARE Error int;

DECLARE TMPResult int;

DROP TABLE IF EXISTS TMPFreePlaces;

CREATE TEMPORARY table TMPFreePlaces(

`Place\_ID` int AUTO\_INCREMENT,

`Place\_Number` int,

PRIMARY KEY (`Place\_ID`)

);

SELECT ThrowArrivalStopID(CurrentArrivalID) INTO CurrentStopArrivalID;

SELECT ThrowDepartureStopID(CurrentDepartureID) INTO CurrentStopDepartureID;

SET i = 1;

SET i2 = 1;

SELECT railcar\_types.Seats\_Count INTO MaxCount FROM railcar\_types

INNER JOIN railcars ON railcars.Railcar\_Type\_Number = railcar\_types.Railcar\_Type\_Number

WHERE (railcars.Train\_Number = TrainNumber\_IN) AND (railcars.Railcar\_Number = Railcar\_Number\_IN);

WHILE (i <> MaxCount + 1) DO

SET ERROR = 1;

SET TMPPlaceID = NULL;

SET i2 = 1;

SELECT Place\_ID INTO TMPPlaceID FROM places

WHERE (Train\_Number = TrainNumber\_IN) AND (Railcar\_Number = Railcar\_Number\_IN) AND (Place\_Number = i);

IF (TMPPlaceID is NULL) then

INSERT INTO TMPFreePlaces (Place\_Number) VALUES (i);

ELSE

DROP table IF exists TMPComparetable;

CREATE TEMPORARY table TMPComparetable(

`CompareNumber` int AUTO\_INCREMENT,

`Arrival\_Stop\_ID` int,

`Departure\_Stop\_ID` int,

PRIMARY KEY (`CompareNumber`)

);

INSERT INTO TMPComparetable (Arrival\_Stop\_ID, Departure\_Stop\_ID)

SELECT arrivals.Stop\_ID, departures.Stop\_ID FROM reservation

INNER JOIN departures ON reservation.Departure\_ID = departures.Departure\_ID

INNER JOIN arrivals ON reservation.Arrival\_ID = arrivals.Arrival\_ID

WHERE reservation.Place\_Number = TMPPlaceID;

SELECT COUNT(\*) INTO MaxCompareCount FROM TMPComparetable;

WHILE (i2 <> MaxCompareCount + 1) do

SELECT Arrival\_Stop\_ID INTO ExistsStopArrivalID FROM TMPComparetable

WHERE CompareNumber = i2;

SELECT Departure\_Stop\_ID INTO ExistsStopDepartureID FROM TMPComparetable

WHERE CompareNumber = i2;

SELECT ThrowEmployedIDs(CurrentStopArrivalID, CurrentStopDepartureID, ExistsStopArrivalID, ExistsStopDepartureID) INTO TMPResult;

IF (TMPResult = -1) THEN

set Error = 0;

END IF;

set i2 = i2 + 1;

IF (Error = 0) THEN

set i2 = MaxCompareCount + 1;

end IF;

END WHILE;

IF (Error = 1) THEN

INSERT INTO TMPFreePlaces (Place\_Number) VALUES (i);

END IF;

END IF;

SET i = i + 1;

END WHILE;

SELECT Place\_Number FROM TMPFreePlaces;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.newFindTrainList(

`Rout\_ID\_IN` int,

`Arrival\_Stop\_Name` varchar(30),

`Arrival\_Date\_IN` varchar(30),

`Departure\_Stop\_Name` varchar(30)

)

BEGIN

DECLARE TrainNumber int;

DECLARE CurrentStopID int;

DECLARE CurrentDepartureStopID int;

DECLARE CheckDate Date;

DECLARE trueDate Date;

SET CheckDate = now();

SET trueDate = STR\_TO\_DATE(Arrival\_Date\_IN, '%d.%m.%Y');

IF (trueDate > CheckDate) THEN

SELECT Stop\_ID INTO CurrentStopID FROM stops

WHERE (Rout\_ID = Rout\_ID\_IN) AND (Stop\_Name = Arrival\_Stop\_Name);

SELECT Stop\_ID INTO CurrentDepartureStopID FROM stops

WHERE (Rout\_ID = Rout\_ID\_IN) AND (Stop\_Name = Departure\_Stop\_Name);

if (CurrentStopID < CurrentDepartureStopID) AND (CurrentStopID <> CurrentDepartureStopID) then

SELECT Train\_Number FROM arrivals

WHERE (Arrival\_Date = trueDate) AND (Stop\_ID = CurrentStopID) AND (Rout\_ID = Rout\_ID\_IN);

ELSE

SELECT -1;

END IF;

ELSE

SELECT -1;

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.throwPassengerInfo(IN Reservation\_ID\_IN INT, IN KeySi TEXT)

BEGIN

DECLARE PassNum int;

SELECT Passenger\_ID INTO PassNum FROM reservation

WHERE Reservation\_ID = Reservation\_ID\_IN;

SELECT Last\_Name, First\_Name, Pathronymic, AES\_DECRYPT(Passport\_Series, KeySi), AES\_DECRYPT(Passport\_Number, KeySi) FROM passengers

WHERE Passenger\_ID = PassNum;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION rzd.GetDepartureID(DepartureStopName\_IN VARCHAR(30), DepartureRoutID\_IN INT, TrainNumber\_IN VARCHAR(30))

RETURNS int(11)

BEGIN

DECLARE TMPID int;

SELECT departures.Departure\_ID INTO TMPID

FROM departures

INNER JOIN stops ON departures.Stop\_ID = stops.Stop\_ID AND departures.Rout\_ID = stops.Rout\_ID

WHERE stops.Stop\_Name = DepartureStopName\_IN AND departures.Rout\_ID = DepartureRoutID\_IN AND departures.Train\_Number = TrainNumber\_IN;

RETURN TMPID;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION rzd.GetArrivalID(ArrivalStopName\_IN VARCHAR(30), ArrivalRoutID\_IN INT, TrainNumber\_IN VARCHAR(30))

RETURNS int(11)

BEGIN

DECLARE TMPID int;

SELECT Arrivals.Arrival\_ID INTO TMPID

FROM arrivals

INNER JOIN stops ON arrivals.Stop\_ID = stops.Stop\_ID AND arrivals.Rout\_ID = stops.Rout\_ID

WHERE stops.Stop\_Name = ArrivalStopName\_IN AND arrivals.Rout\_ID = ArrivalRoutID\_IN AND arrivals.Train\_Number = TrainNumber\_IN;

RETURN TMPID;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION rzd.FindPassenger(

`Passport\_Series\_IN` int,

`Passport\_Number\_IN` int,

`KeySi` TEXT

)

RETURNS int(11)

BEGIN

DECLARE tmpid int;

SELECT Passenger\_ID INTO tmpid

FROM passengers

WHERE Passport\_Series = AES\_ENCRYPT(Passport\_Series\_IN, KeySi) AND Passport\_Number = AES\_ENCRYPT(Passport\_Number\_IN, KeySi);

IF (tmpid is NOT NULL) THEN

RETURN tmpid;

ELSE

RETURN -1;

END IF;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION rzd.PassengerAddToDB(Last\_Name\_IN VARCHAR(30), First\_Name\_IN VARCHAR(30), Pathronymic\_IN VARCHAR(30), Passport\_Series\_IN INT(11), Passport\_Number\_IN INT(11), Passenger\_Phone\_Number\_IN VARCHAR(20), KeySi text)

RETURNS int(11)

BEGIN

DECLARE LastInsertID int;

INSERT INTO passengers (Last\_Name, First\_Name, Pathronymic, Passport\_Series, Passport\_Number, Passenger\_Phone\_Number)

VALUES (Last\_Name\_IN, First\_Name\_IN, Pathronymic\_IN, AES\_ENCRYPT(Passport\_Series\_IN, KeySi), AES\_ENCRYPT(Passport\_Number\_IN, KeySi), AES\_ENCRYPT(Passenger\_Phone\_Number\_IN, KeySi));

SET LastInsertID = last\_insert\_id();

RETURN LastInsertID;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION rzd.ThrowRoutID(Train\_Number\_IN VARCHAR(30))

RETURNS int(11)

BEGIN

DECLARE CurrentRoutID int;

SELECT Rout\_ID INTO CurrentRoutID FROM trains

WHERE Train\_Number = Train\_Number\_IN;

IF (CurrentRoutID is NOT NULL) THEN

RETURN CurrentRoutID;

ELSE

RETURN -1;

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.throwCountAvailableTickets(IN UserLogin\_IN VARCHAR(30))

BEGIN

SELECT COUNT(\*) FROM reservation

WHERE Payed\_Account\_Login = UserLogin\_IN;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.CancelTicket(IN TicketNumber INT)

BEGIN

DELETE FROM reservation

WHERE Reservation\_ID = TicketNumber;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.throwAvailableTicketsWithInfo(IN User\_Login\_IN VARCHAR(30))

BEGIN

-- TMP Vars

DECLARE \_TMPReservationID int;

DECLARE \_TMPPlaceNumber int;

DECLARE \_TMPDepartureID int;

DECLARE \_TMPArrivalID int;

-- Other Vars

DECLARE \_Train\_Number VARCHAR(11);

DECLARE \_Railcar\_Number int;

DECLARE StopDeparture int;

DECLARE StopArrival int;

DECLARE \_Place\_Number int;

DECLARE \_Arrival\_Date Date;

DECLARE \_Arrival\_Time Time;

DECLARE \_Arrival\_Stop\_Name varchar(30);

DECLARE \_Arrival\_Railway\_Station varchar(30);

DECLARE \_Arrival\_Rout\_ID int;

DECLARE \_Departure\_Date Date;

DECLARE \_Departure\_Time Time;

DECLARE \_Departure\_Stop\_Name varchar(30);

DECLARE \_Departure\_Railway\_Station varchar(30);

-- Cursor Vars

DECLARE done INT DEFAULT 0;

DECLARE cur1 CURSOR FOR

SELECT Reservation\_ID, Place\_Number, Departure\_ID, Arrival\_ID FROM reservation

WHERE Payed\_Account\_Login = User\_Login\_IN;

DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;

-- Code

OPEN cur1;

DROP TABLE IF EXISTS ResultTable;

CREATE TEMPORARY table ResultTable(

`Reservation\_ID` int,

`Train\_Number` VARCHAR(11),

`Railcar\_Number` int,

`Place\_Number` int,

`Arrival\_Time` time,

`Arrival\_Date` date,

`Arrival\_Stop\_Name` varchar(30),

`Departure\_Time` time,

`Departure\_Date` date,

`Departure\_Stop\_Name` varchar(30)

);

WHILE (done <> 1) DO

FETCH cur1 INTO \_TMPReservationID,\_TMPPlaceNumber, \_TMPDepartureID, \_TMPArrivalID;

SELECT Train\_Number INTO \_Train\_Number FROM places

WHERE Place\_ID = \_TMPPlaceNumber;

SELECT Railcar\_Number INTO \_Railcar\_Number FROM places

WHERE Place\_ID = \_TMPPlaceNumber;

SELECT Place\_Number INTO \_Place\_Number FROM places

WHERE Place\_ID = \_TMPPlaceNumber;

SELECT Rout\_ID INTO \_Arrival\_Rout\_ID FROM arrivals

WHERE Arrival\_ID = \_TMPArrivalID;

SELECT Stop\_ID INTO StopDeparture FROM arrivals

WHERE Arrival\_ID = \_TMPArrivalID;

SELECT Stop\_ID INTO StopArrival FROM departures

WHERE Departure\_ID = \_TMPDepartureID;

SELECT Arrival\_Date INTO \_Arrival\_Date FROM arrivals

WHERE Train\_Number = \_Train\_Number and Rout\_ID = \_Arrival\_Rout\_ID and Stop\_ID = StopArrival;

SELECT Arrival\_Time INTO \_Arrival\_Time FROM arrivals

WHERE Train\_Number = \_Train\_Number and Rout\_ID = \_Arrival\_Rout\_ID and Stop\_ID = StopArrival;

SELECT stops.Stop\_Name INTO \_Arrival\_Stop\_Name FROM arrivals

INNER JOIN stops ON arrivals.Stop\_ID = stops.Stop\_ID

WHERE arrivals.Arrival\_ID = \_TMPArrivalID AND stops.Rout\_ID = \_Arrival\_Rout\_ID;

SELECT Train\_Station\_Name INTO \_Arrival\_Railway\_Station FROM stops

WHERE Stop\_ID = StopDeparture and Rout\_ID = \_Arrival\_Rout\_ID;

IF (\_Arrival\_Railway\_Station IS NOT NULL) THEN

SET \_Arrival\_Stop\_Name = CONCAT(\_Arrival\_Stop\_Name, ', ', \_Arrival\_Railway\_Station);

END IF;

SELECT Departure\_Time INTO \_Departure\_Time FROM departures

WHERE Train\_Number = \_Train\_Number and Rout\_ID = \_Arrival\_Rout\_ID and Stop\_ID = StopDeparture;

SELECT Departure\_Date INTO \_Departure\_Date FROM departures

WHERE Train\_Number = \_Train\_Number and Rout\_ID = \_Arrival\_Rout\_ID and Stop\_ID = StopDeparture;

SELECT stops.Stop\_Name INTO \_Departure\_Stop\_Name FROM departures

INNER JOIN stops ON departures.Stop\_ID = stops.Stop\_ID

WHERE departures.Departure\_ID = \_TMPDepartureID AND stops.Rout\_ID = \_Arrival\_Rout\_ID;

SELECT Train\_Station\_Name INTO \_Departure\_Railway\_Station FROM stops

WHERE Stop\_ID = StopArrival and Rout\_ID = \_Arrival\_Rout\_ID;

IF (\_Departure\_Railway\_Station IS NOT NULL) THEN

SET \_Departure\_Stop\_Name = CONCAT(\_Departure\_Stop\_Name, ', ', \_Departure\_Railway\_Station);

END IF;

INSERT INTO ResultTable (Reservation\_ID, Train\_Number, Railcar\_Number, Place\_Number, Arrival\_Time, Arrival\_Date, Arrival\_Stop\_Name, Departure\_Time, Departure\_Date, Departure\_Stop\_Name)

VALUES (\_TMPReservationID, \_Train\_Number, \_Railcar\_Number, \_Place\_Number, \_Arrival\_Time, \_Arrival\_Date, \_Arrival\_Stop\_Name, \_Departure\_Time, \_Departure\_Date, \_Departure\_Stop\_Name);

END WHILE;

SELECT DISTINCT Reservation\_ID, Train\_Number, Railcar\_Number, Place\_Number,

TIME\_FORMAT(Departure\_Time, '%H:%i'), DATE\_FORMAT(Departure\_Date, '%d.%m.%Y'),

Arrival\_Stop\_Name, TIME\_FORMAT(Arrival\_Time, '%H:%i'), DATE\_FORMAT(Arrival\_Date, '%d.%m.%Y'),

Departure\_Stop\_Name FROM ResultTable;

DROP TABLE IF EXISTS ResultTable;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.EmployPlaces(

IN `Train\_Number\_IN` VARCHAR(11),

IN `Railcar\_Number\_IN` int,

IN `Place\_Number\_IN` int,

IN `Passenger\_ID\_IN` int,

IN `Arrival\_ID\_IN` int,

IN `Departure\_ID\_IN` int

)

begin

DECLARE CurrentDepartureID int;

DECLARE CurrentStopArrivalID int;

DECLARE CurrentStopDepartureID int;

DECLARE ExistsStopDepartureID int;

declare TmpDepartureID int;

declare CurrentPlaceID int;

DECLARE CompareTableCount int;

DECLARE i int;

DECLARE Reservation int;

DECLARE MaxCompareCount int;

DECLARE EA int;

DECLARE ED int;

DECLARE CurrentUserLogin varchar(30);

SET i = 1;

SET Reservation = 1;

SELECT Place\_ID INTO CurrentPlaceID FROM places

WHERE ((Train\_Number = Train\_Number\_IN) AND (Railcar\_Number = Railcar\_Number\_IN) AND (Place\_Number = Place\_Number\_IN));

SELECT Stop\_ID INTO CurrentStopArrivalID FROM arrivals

WHERE (Arrival\_ID = Arrival\_ID\_IN);

SELECT Stop\_ID INTO CurrentStopDepartureID FROM departures

WHERE (Departure\_ID = Departure\_ID\_IN);

if (CurrentStopArrivalID < CurrentStopDepartureID) AND (CurrentStopArrivalID <> CurrentStopDepartureID) then

if (CurrentPlaceID is NULL) then

INSERT INTO Places (Train\_Number, Railcar\_Number, Place\_Number) VALUES (Train\_Number\_IN, Railcar\_Number\_IN, Place\_Number\_IN);

SET CurrentPlaceID = LAST\_INSERT\_ID();

SELECT USER() INTO CurrentUserLogin;

INSERT INTO Reservation (Passenger\_ID, Place\_Number, Departure\_ID, Arrival\_ID, Payed\_Account\_Login) VALUES (Passenger\_ID\_IN, CurrentPlaceID, Departure\_ID\_IN, Arrival\_ID\_IN, CurrentUserLogin);

ELSE

DROP TABLE IF EXISTS TMPCompareTable;

CREATE TEMPORARY TABLE TMPCompareTable(

`CompareNumber` int AUTO\_INCREMENT,

`Arrival\_Stop\_ID` int,

`Departure\_Stop\_ID` int,

PRIMARY KEY (`CompareNumber`)

);

INSERT INTO TMPCompareTable (Arrival\_Stop\_ID, Departure\_Stop\_ID)

SELECT arrivals.Stop\_ID, departures.Stop\_ID

FROM reservation

INNER JOIN departures ON reservation.Departure\_ID = departures.Departure\_ID

INNER JOIN arrivals ON reservation.Arrival\_ID = arrivals.Arrival\_ID

WHERE reservation.Place\_Number = CurrentPlaceID;

SELECT COUNT(\*) INTO MaxCompareCount

FROM TMPCompareTable;

WHILE (i <> MaxCompareCount + 1) do

SELECT Arrival\_Stop\_ID INTO EA FROM TMPCompareTable

WHERE CompareNumber = i;

SELECT Departure\_Stop\_ID INTO ED FROM TMPCompareTable

WHERE CompareNumber = i;

IF (EA <> NULL) and (ED <> NULL) THEN

IF (ThrowEmployedIDs(CurrentStopArrivalID, CurrentStopDepartureID, EA, ED) = -1) THEN

SET Reservation = 0;

END IF;

SET i = i + 1;

IF (Reservation = 0) THEN

SET i = MaxCompareCount + 1;

END IF;

ELSE

SET i = i + 1;

END IF;

END WHILE;

if (Reservation = 0) then

signal sqlstate '45000' SET message\_text = 'К сожалению, выбранное вами место уже занято';

ELSE

SELECT USER() INTO CurrentUserLogin;

INSERT INTO Reservation (Passenger\_ID, Place\_Number, Departure\_ID, Arrival\_ID, Payed\_Account\_Login) VALUES (Passenger\_ID\_IN, CurrentPlaceID, Departure\_ID\_IN, Arrival\_ID\_IN, CurrentUserLogin);

END IF;

END IF;

ELSE

signal sqlstate '45000' SET message\_text = 'Произошла техническая ошибка, обратитесь к администратору';

end if;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.register\_createuser(IN UserLogin varchar(30), IN UserPassword varchar(30))

BEGIN

SET @query1 = CONCAT('CREATE USER "',UserLogin,'"@"localhost" IDENTIFIED BY "',UserPassword,'" ');

PREPARE stmt FROM @query1;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

SET @query2 = CONCAT('grant user to "', UserLogin,'"@"localhost"');

PREPARE stmt FROM @query2;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

SET @query3 = CONCAT('set default role user for "', UserLogin,'"@"localhost"');

PREPARE stmt FROM @query3;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

FLUSH PRIVILEGES;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.throwRailcarInfo(IN Train\_Number\_IN VARCHAR(11), IN Railcar\_Number\_IN INT)

BEGIN

SELECT railcar\_types.Railcar\_Type, railcar\_types.Cost FROM railcars

INNER JOIN railcar\_types ON railcar\_types.Railcar\_Type\_Number = railcars.Railcar\_Type\_Number

WHERE Train\_Number = Train\_Number\_IN AND Railcar\_Number = Railcar\_Number\_IN;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.EmptyProcForBlockedUsers()

BEGIN

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION ThrowArrivalStopID(Arrival\_ID\_IN int)

RETURNS int(11)

BEGIN

DECLARE CurrentStopID int;

SELECT Stop\_ID INTO CurrentStopID FROM arrivals

WHERE (Arrival\_ID = Arrival\_ID\_IN);

IF (CurrentStopID is NOT NULL) THEN

RETURN CurrentStopID;

ELSE

signal sqlstate '45000' SET message\_text = 'Во время запроса произошла ошибка. Виноват в этом только программист.';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION ThrowDepartureStopID(Departure\_ID\_IN int)

RETURNS int(11)

BEGIN

DECLARE CurrentStopID int;

SELECT Stop\_ID INTO CurrentStopID FROM departures

WHERE (Departure\_ID = Departure\_ID\_IN);

IF (CurrentStopID is NOT NULL) THEN

RETURN CurrentStopID;

else

signal sqlstate '45000' SET message\_text = 'Во время запроса произошла ошибка. Виноват в этом только программист.';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION ThrowEmployedIDs(CA INT, CD INT, EA INT, ED INT)

RETURNS int(11)

BEGIN

IF (((CA <= ED) AND (CD <= EA)) OR ((CA >= ED) AND (CD >= ED))) THEN

RETURN 1;

ELSE

RETURN -1;

END IF;

END

CREATE DEFINER = 'root'@'localhost'

TRIGGER RemovePhantomSeats

AFTER DELETE ON reservation

FOR EACH ROW

BEGIN

DECLARE CurrentCount int;

SELECT COUNT(\*) INTO CurrentCount FROM reservation

WHERE Place\_Number = old.Place\_Number;

if (CurrentCount = 0) then

DELETE FROM places

WHERE Place\_ID = old.Place\_Number;

END IF;

END

DELIMITER $$

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_AddTrain(IN Train\_Number\_IN varchar(30), IN Railcar\_Count\_IN int, IN Train\_Type\_IN varchar(30), IN Rout\_Name\_IN varchar(50))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID INT;

DECLARE \_TMPNum INT;

IF (Railcar\_Count\_IN < 20) THEN

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN LIMIT 1;

SELECT Train\_Type\_Number INTO \_TMPNum FROM train\_types

WHERE Train\_Type = Train\_Type\_IN;

IF (\_TMPNum is NULL) THEN

INSERT INTO train\_types (Train\_Type) VALUES (Train\_Type\_IN);

SET \_TMPNum = LAST\_INSERT\_ID();

END IF;

IF (\_TMPRoutID is NOT NULL) then

INSERT INTO trains (Train\_Number, Railcar\_Count, Train\_Type\_Number, Rout\_ID)

VALUES (Train\_Number\_IN, Railcar\_Count\_IN, \_TMPNum, Rout\_ID\_IN);

ELSE

signal sqlstate '45000' SET message\_text = 'Такого маршрута не существует. Перепроверьте введенные данные';

END IF;

ELSE

signal sqlstate '45000' SET message\_text = 'Вы ввели слишком большое количество вагонов. Перепроверьте введенные данные';

END IF;

END

DROP PROCEDURE IF EXISTS rzd.DISPATCHER\_ShowTrains;

DELIMITER $$

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_ShowTrains()

DETERMINISTIC

BEGIN

SELECT Train\_Number, Railcar\_Count, train\_types.Train\_Type, routes.Rout\_Name FROM trains

INNER JOIN train\_types ON trains.Train\_Type\_Number = train\_types.Train\_Type\_Number

INNER JOIN routes ON trains.Rout\_ID = routes.Rout\_ID;

END

DROP PROCEDURE IF EXISTS rzd.DISPATCHER\_DropTrain;

DELIMITER $$

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_DropTrain(IN Train\_Number\_IN varchar(30))

DETERMINISTIC

BEGIN

DECLARE EXIT HANDLER FOR SQLSTATE '23000'

BEGIN

signal sqlstate '45000' SET message\_text = 'При удалении поезда произошла ошибка. Убедитесь, что у удаляемого поезда нет записей в таблицах "Отправления" и "Прибытия"';

END;

DELETE FROM trains WHERE Train\_Number = Train\_Number\_IN;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_UpdateTrain(IN Train\_Number\_IN varchar(30), IN Railcar\_Count\_IN int, IN Train\_Type\_IN varchar(30), IN Rout\_Name\_IN varchar(50))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID INT;

DECLARE \_TMPNum INT;

IF (Railcar\_Count\_IN < 20) THEN

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN LIMIT 1;

SELECT Train\_Type\_Number INTO \_TMPNum FROM train\_types

WHERE Train\_Type = Train\_Type\_IN;

IF (\_TMPNum is NULL) THEN

INSERT INTO train\_types (Train\_Type) VALUES (Train\_Type\_IN);

SET \_TMPNum = LAST\_INSERT\_ID();

END IF;

IF (\_TMPRoutID is NOT NULL) THEN

UPDATE trains SET Railcar\_Count = Railcar\_Count\_IN, Train\_Type\_Number = \_TMPNum, Rout\_ID = \_TMPRoutID

WHERE Train\_Number = Train\_Number\_IN;

ELSE

signal sqlstate '45000' SET message\_text = 'Такого маршрута не существует. Перепроверьте введенные данные';

END IF;

ELSE

signal sqlstate '45000' SET message\_text = 'Вы ввели слишком большое количество вагонов. Перепроверьте введенные данные';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_AddRailcar(IN Train\_Number\_IN varchar(30), IN Railcar\_Number\_IN int, IN Railcar\_Type\_IN varchar(30))

DETERMINISTIC

BEGIN

DECLARE \_TMPNum int;

DECLARE \_MaxRailcars\_Count int;

SELECT Railcar\_Type\_Number INTO \_TMPNum FROM railcar\_types

WHERE Railcar\_Type = Railcar\_Type\_IN;

INSERT INTO railcars (Train\_Number, Railcar\_Number, Railcar\_Type\_Number) VALUES (Train\_Number\_IN, Railcar\_Number\_IN, \_TMPNum);

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_ShowRailcars()

DETERMINISTIC

BEGIN

SELECT Train\_Number, Railcar\_Number, railcar\_types.Railcar\_Type FROM railcars

INNER JOIN railcar\_types ON railcars.Railcar\_Type\_Number = railcar\_types.Railcar\_Type\_Number;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_DropRailcar(IN Train\_Number\_IN VARCHAR(30), IN Railcar\_Number\_IN int)

DETERMINISTIC

BEGIN

DELETE FROM railcars WHERE Train\_Number = Train\_Number\_IN AND Railcar\_Number = Railcar\_Number\_IN;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_UpdateRailcar(IN Train\_Number\_IN VARCHAR(30), IN Railcar\_Number\_IN int, IN Railcar\_Type\_IN varchar(30))

DETERMINISTIC

BEGIN

DECLARE \_TMPNum int;

DECLARE \_MaxRailcars\_Count int;

DECLARE \_Error text;

SELECT Railcar\_Type\_Number INTO \_TMPNum FROM railcar\_types

WHERE Railcar\_Type = Railcar\_Type\_IN;

IF (\_TMPNum is NOT NULL) THEN

SELECT Railcar\_Count INTO \_MaxRailcars\_Count FROM trains

WHERE Train\_Number = Train\_Number\_IN;

SET \_Error = CONCAT('У данного поезда всего ', \_MaxRailcars\_Count, ' вагонов.');

IF (\_MaxRailcars\_Count > Railcar\_Number\_IN) then

SIGNAL SQLSTATE '45000' SET message\_text = \_Error;

ELSE

UPDATE railcars SET Railcar\_Number = Railcar\_Number\_IN, Railcar\_Type\_Number = \_TMPNum;

END IF;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Введенного вами типа вагона не существует';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

FUNCTION rzd.DISPATCHER\_AddRout(Stop\_Count\_IN int, Rout\_Name\_IN varchar(50))

RETURNS int(11)

BEGIN

DECLARE TMPNum int;

INSERT INTO routes (Stop\_Count, Rout\_Name) VALUES (Stop\_Count\_IN, Rout\_Name\_IN);

SET TMPNum = LAST\_INSERT\_ID();

RETURN TMPNum;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_DropRout(IN Rout\_ID\_IN INT)

DETERMINISTIC

BEGIN

DELETE FROM routes WHERE Rout\_ID = Rout\_ID\_IN;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_UpdateRout(IN Rout\_ID\_IN INT, IN Stop\_Count\_IN int, IN Rout\_Name\_IN varchar(50))

DETERMINISTIC

BEGIN

UPDATE routes SET Stop\_Count = Stop\_Count\_IN, Rout\_Name = Rout\_Name\_IN

WHERE Rout\_ID = Rout\_ID\_IN;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_ShowRoutes()

DETERMINISTIC

BEGIN

SELECT Rout\_ID, Stop\_Count, Rout\_Name FROM routes;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_AddStop(IN Rout\_Name\_IN varchar(50), IN Stop\_ID\_IN int, IN Stop\_Name\_IN varchar(30), IN Train\_Station\_Name\_IN varchar(50))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPMaxStopCount int;

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SELECT Stop\_Count INTO \_TMPMaxStopCount FROM routes

WHERE Rout\_ID = \_TMPRoutID;

IF (\_TMPRoutID is NOT NULL) THEN

IF (\_TMPMaxStopCount > Stop\_ID\_IN) THEN

INSERT INTO stops (Rout\_ID, Stop\_ID, Stop\_Name, Train\_Station\_Name) VALUES (\_TMPRoutID, Stop\_ID\_IN, Stop\_Name\_IN, Train\_Station\_Name\_IN);

END IF;

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowStopNumber(IN Rout\_Name\_IN varchar(50))

DETERMINISTIC

BEGIN

DECLARE MaxCount int;

DECLARE i int;

DROP TABLE IF EXISTS TMPStopListCount;

CREATE TEMPORARY table TMPStopListCount(

`Stop\_ID` int AUTO\_INCREMENT,

`Stop\_Number` int,

PRIMARY KEY (`Stop\_ID`)

);

SELECT Stop\_Count INTO MaxCount FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SET i = 1;

WHILE (i <> Stop\_Count + 1) DO

INSERT INTO TMPStopListCount (Stop\_Number) VALUES (i);

SET i = i + 1;

END WHILE;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_UpdateStop(IN Rout\_Name\_IN varchar(50), IN Stop\_ID\_IN int, IN Stop\_Name\_IN varchar(30), IN Train\_Station\_Name\_IN varchar(50))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPMaxStopCount int;

DECLARE \_error TEXT;

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

IF (\_TMPRoutID is not null) THEN

SELECT Stop\_Count INTO \_TMPMaxStopCount FROM routes

WHERE Rout\_ID = \_TMPRoutID;

IF (\_TMPMaxStopCount > Stop\_ID\_IN) THEN

IF (Train\_Station\_Name\_IN = '') THEN

UPDATE stops SET Stop\_Name = Stop\_Name\_IN, Train\_Station\_Name = NULL

WHERE Rout\_ID = \_TMPRoutID and Stop\_ID = Stop\_ID\_IN;

ELSE

UPDATE stops SET Stop\_Name = Stop\_Name\_IN, Train\_Station\_Name = Train\_Station\_Name\_IN

WHERE Rout\_ID = \_TMPRoutID and Stop\_ID = Stop\_ID\_IN;

END IF;

ELSE

SET \_error = CONCAT('При обновлении записи произошла ошибка. В выбранном маршруте всего ', \_TMPMaxStopCount, ' остановок');

SIGNAL SQLSTATE '45000' SET message\_text = \_error;

END IF;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Введенного вами маршрута не существует';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_DropStop(IN Rout\_Name\_IN varchar(50), IN Stop\_ID\_IN int)

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPMaxStopCount int;

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

IF (\_TMPRoutID is NOT NULL) THEN

DELETE FROM stops WHERE Rout\_ID = \_TMPRoutID AND Stop\_ID = Stop\_ID\_IN;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При удалении записи произошла невозможная ошибка';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_ShowStops()

DETERMINISTIC

BEGIN

SELECT routes.Rout\_Name, Stop\_ID, Stop\_Name, Train\_Station\_Name FROM stops

INNER JOIN routes ON stops.Rout\_ID = routes.Rout\_ID;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_AddArrival(IN Rout\_Name\_IN varchar(30), IN Stop\_Name\_IN varchar(30), IN Train\_Number\_IN varchar(11), IN Arrival\_Time\_IN time, IN Arrival\_Date\_IN varchar(11))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPStopID int;

SET trueDate = STR\_TO\_DATE(Arrival\_Date\_IN, '%d.%m.%Y');

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SELECT Stop\_ID INTO \_TMPStopID FROM Stops

WHERE Rout\_ID = \_TMPRoutID AND Stop\_Name = Stop\_Name\_IN;

IF (\_TMPStopID is NOT NULL) THEN

INSERT INTO arrivals (Stop\_ID, Rout\_ID, Train\_Number, Arrival\_Time, Arrival\_Date) VALUES (\_TMPStopID, \_TMPRoutID, Train\_Number\_IN, Arrival\_Time\_IN, TrueDate);

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowAvailableStopsFindByRout(IN Rout\_Name\_IN varchar(30))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SELECT Stop\_Name FROM stops

WHERE Rout\_ID = \_TMPRoutID;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowAvailableTrainsFindByRout(IN Rout\_Name\_IN varchar(30))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SELECT Train\_Number FROM trains

WHERE Rout\_ID = \_TMPRoutID;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_UpdateArrival(IN Rout\_Name\_IN varchar(30), IN Stop\_Name\_IN varchar(30), IN Train\_Number\_IN varchar(11), IN Arrival\_Time\_IN time, IN Arrival\_Date\_IN varchar(11))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPStopID int;

DECLARE TrueDate date;

SET trueDate = STR\_TO\_DATE(Arrival\_Date\_IN, '%d.%m.%Y');

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

IF EXISTS(SELECT \* FROM trains WHERE Train\_Number = Train\_Number\_IN) THEN

IF (\_TMPRoutID is NOT NULL) THEN

SELECT Stop\_ID INTO \_TMPStopID FROM Stops

WHERE Rout\_ID = \_TMPRoutID AND Stop\_Name = Stop\_Name\_IN;

IF (\_TMPStopID is NOT NULL) THEN

UPDATE arrivals SET Train\_Number = Train\_Number\_IN, Arrival\_Time = Arrival\_Time\_IN, Arrival\_Date = trueDate

WHERE Stop\_ID = \_TMPStopID AND Rout\_ID = \_TMPRoutID;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Введенной вами остановки не существует';

END IF;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Введенного вами маршрута не существует';

END IF;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Поезда с введеным вами номером не существует';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_DropArrival(IN Rout\_Name\_IN varchar(30), IN Stop\_Name\_IN varchar(30), IN Train\_Number\_IN varchar(11))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPStopID int;

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SELECT Stop\_ID INTO \_TMPStopID FROM Stops

WHERE Rout\_ID = \_TMPRoutID AND Stop\_Name = Stop\_Name\_IN;

IF (\_TMPStopID is NOT NULL) THEN

DELETE FROM arrivals WHERE Stop\_ID = \_TMPStopID AND Rout\_ID = \_TMPRoutID;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При удалении записи произошла неизвестная ошибка';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_ShowArrivals()

DETERMINISTIC

BEGIN

SELECT stops.Stop\_Name, routes.Rout\_Name, Train\_Number, TIME\_FORMAT(Arrival\_Time, '%H:%i'), DATE\_FORMAT(Arrival\_Date, '%d.%m.%Y') FROM arrivals

INNER JOIN stops ON arrivals.Stop\_ID = stops.Stop\_ID AND arrivals.Rout\_ID = stops.Rout\_ID

INNER JOIN routes ON arrivals.Rout\_ID = routes.Rout\_ID;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_AddDeparture(IN Rout\_Name\_IN varchar(30), IN Stop\_Name\_IN varchar(30), IN Train\_Number\_IN varchar(11), IN Departure\_Time\_IN time, IN Departure\_Date\_IN varchar(11))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPStopID int;

DECLARE trueDate date;

SET trueDate = STR\_TO\_DATE(Arrival\_Date\_IN, '%d.%m.%Y');

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SELECT Stop\_ID INTO \_TMPStopID FROM Stops

WHERE Rout\_ID = \_TMPRoutID AND Stop\_Name = Stop\_Name\_IN;

IF (\_TMPStopID is NOT NULL) THEN

INSERT INTO departures (Stop\_ID, Rout\_ID, Train\_Number, Departure\_Time, Departure\_Date) VALUES (\_TMPStopID, \_TMPRoutID, Train\_Number\_IN, Departure\_Time\_IN, trueDate);

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_UpdateDeparture(IN Stop\_Name\_IN varchar(30), IN Rout\_Name\_IN varchar(30), IN Train\_Number\_IN varchar(11), IN Departure\_Time\_IN time, IN Departure\_Date\_IN varchar(11))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPStopID int;

DECLARE TrueDate date;

SET trueDate = STR\_TO\_DATE(Departure\_Date\_IN, '%d.%m.%Y');

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

IF EXISTS(SELECT \* FROM trains WHERE Train\_Number = Train\_Number\_IN) THEN

IF (\_TMPRoutID is NOT NULL) THEN

SELECT Stop\_ID INTO \_TMPStopID FROM Stops

WHERE Rout\_ID = \_TMPRoutID AND Stop\_Name = Stop\_Name\_IN;

IF (\_TMPStopID is NOT NULL) THEN

UPDATE departures SET Train\_Number = Train\_Number\_IN, Departure\_Time = Departure\_Time\_IN, Departure\_Date = trueDate

WHERE Stop\_ID = \_TMPStopID AND Rout\_ID = \_TMPRoutID;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Введенной вами остановки не существует';

END IF;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Введенного вами маршрута не существует';

END IF;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При обновлении записи произошла ошибка. Поезда с введеным вами номером не существует';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_DropDeparture(IN Rout\_Name\_IN varchar(30), IN Stop\_Name\_IN varchar(30), IN Train\_Number\_IN varchar(11))

DETERMINISTIC

BEGIN

DECLARE \_TMPRoutID int;

DECLARE \_TMPStopID int;

SELECT Rout\_ID INTO \_TMPRoutID FROM routes

WHERE Rout\_Name = Rout\_Name\_IN;

SELECT Stop\_ID INTO \_TMPStopID FROM Stops

WHERE Rout\_ID = \_TMPRoutID AND Stop\_Name = Stop\_Name\_IN;

IF (\_TMPStopID is NOT NULL) THEN

DELETE FROM departures WHERE Stop\_ID = \_TMPStopID AND Rout\_ID = \_TMPRoutID;

ELSE

SIGNAL SQLSTATE '45000' SET message\_text = 'При удалении записи произошла неизвестная ошибка';

END IF;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.DISPATCHER\_ShowDepartures()

DETERMINISTIC

BEGIN

SELECT stops.Stop\_Name, routes.Rout\_Name, Train\_Number, TIME\_FORMAT(Departure\_Time, '%H:%i'), DATE\_FORMAT(Departure\_Date, '%d.%m.%Y') FROM departures

INNER JOIN stops ON departures.Stop\_ID = stops.Stop\_ID AND departures.Rout\_ID = stops.Rout\_ID

INNER JOIN routes ON departures.Rout\_ID = routes.Rout\_ID;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowRailcarTypesList()

BEGIN

SELECT DISTINCT Railcar\_Type FROM railcar\_types;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowTrainList()

BEGIN

SELECT DISTINCT Train\_Number FROM trains;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowAvailableRailcarList(IN Train\_Number\_IN VARCHAR(11))

BEGIN

DECLARE i int;

DECLARE MaxRailcarsCount int;

SET i = 1;

DROP TABLE IF EXISTS TMPRailcarListCount;

CREATE TEMPORARY table TMPRailcarListCount(

`RailcarCount\_ID` int AUTO\_INCREMENT,

`Railcar` int,

PRIMARY KEY (`RailcarCount\_ID`)

);

SELECT Railcar\_Count INTO MaxRailcarsCount FROM trains

WHERE Train\_Number = Train\_Number\_IN;

WHILE (i <> MaxRailcarsCount) DO

IF NOT EXISTS(SELECT \* FROM railcars

WHERE Train\_Number = Train\_Number\_IN AND Railcar\_Number = i) THEN

INSERT INTO TMPRailcarListCount (Railcar) VALUES (i);

END IF;

SET i = i + 1;

END WHILE;

SELECT Railcar FROM TMPRailcarListCount;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowMaxStopCount()

DETERMINISTIC

BEGIN

DECLARE i int;

SET i = 1;

DROP TABLE IF EXISTS TMPStopCount;

CREATE TEMPORARY TABLE TMPStopCount(

`Stop\_ID` int AUTO\_INCREMENT,

`Stop\_Number` int,

PRIMARY KEY (`Stop\_ID`)

);

WHILE (i <> 20 + 1) DO

INSERT INTO TMPStopCount (Stop\_Number) VALUES (i);

SET i = i + 1;

END WHILE;

SELECT Stop\_ID FROM TMPStopCount;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowRailcarListCount()

BEGIN

DECLARE i int;

SET i = 1;

DROP TABLE IF EXISTS TMPRailcarListCount;

CREATE TEMPORARY table TMPRailcarListCount(

`RailcarCount\_ID` int AUTO\_INCREMENT,

`Railcar` int,

PRIMARY KEY (`RailcarCount\_ID`)

);

WHILE (i <> 20 + 1) DO

INSERT INTO TMPRailcarListCount (Railcar) VALUES (i);

SET i = i + 1;

END WHILE;

SELECT RailcarCount\_ID FROM TMPRailcarListCount;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowRoutsNames()

BEGIN

SELECT Rout\_Name FROM routes;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.\_DISPATCHER\_ThrowTrainTypes()

BEGIN

SELECT DISTINCT Train\_Type FROM train\_types;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.ADMIN\_ThrowUsers()

BEGIN

SELECT user, default\_role FROM mysql.user

WHERE (is\_role = 'N') AND (default\_role <> '') AND (user <> 'RegMaster')

AND ((default\_role = 'Admin') or (default\_role = 'user') or (default\_role = 'RZD\_Dispatcher') or (default\_role = 'Blocked'));

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.ADMIN\_CreateUser(IN UserLogin varchar(30), IN UserPassword varchar(30), IN UserRole varchar(30))

BEGIN

SET @query1 = CONCAT('

CREATE USER "', UserLogin, '"@"localhost" IDENTIFIED BY "', UserPassword, '" '

);

PREPARE stmt FROM @query1;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

SET @query2 = CONCAT('grant "', UserRole, '" to "', UserLogin, '"@"localhost"');

PREPARE stmt FROM @query2;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

SET @query3 = CONCAT('set default role "', UserRole, '"for "', UserLogin, '"@"localhost"');

PREPARE stmt FROM @query3;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

FLUSH PRIVILEGES;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.ADMIN\_ThrowRoles()

BEGIN

SELECT DISTINCT default\_role FROM mysql.user

WHERE (default\_role = 'Blocked') or (default\_role = 'user') or (default\_role = 'RZD\_Dispatcher') or (default\_role = 'Admin');

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.ADMIN\_ChangeRole(IN Username varchar(30), IN Rolename varchar(30))

BEGIN

SET @query1 = CONCAT('REVOKE ALL PRIVILEGES ON \*.\* FROM "', Username,'"@"localhost"');

PREPARE stmt FROM @query1; EXECUTE stmt; DEALLOCATE PREPARE stmt;

set @query2 = CONCAT('grant "', Rolename, '" to "', Username,'"@"localhost"');

Execute immediate @query2;

PREPARE stmt FROM @query2; EXECUTE stmt; DEALLOCATE PREPARE stmt;

SET @query3 = CONCAT('set default role "', Rolename, '" FOR "', Username,'"@"localhost"');

execute immediate @query3;

PREPARE stmt FROM @query3; EXECUTE stmt; DEALLOCATE PREPARE stmt;

flush privileges;

END

CREATE DEFINER = 'root'@'localhost'

PROCEDURE rzd.ADMIN\_RefreshPrivileges()

DETERMINISTIC

BEGIN

GRANT EXECUTE ON PROCEDURE rzd.FindPassengerWithPersonalData TO user;

GRANT EXECUTE ON PROCEDURE rzd.ThrowAllStations TO user;

GRANT EXECUTE ON PROCEDURE rzd.Available\_Railcar\_Types TO user;

GRANT EXECUTE ON PROCEDURE rzd.TrainInfoTwo TO user;

GRANT EXECUTE ON PROCEDURE rzd.FindRout TO user;

GRANT EXECUTE ON PROCEDURE rzd.ThrowTrainNumbersList TO user;

GRANT EXECUTE ON PROCEDURE rzd.Available\_For\_Planting\_Seats TO user;

GRANT EXECUTE ON PROCEDURE rzd.newFindTrainList TO user;

GRANT EXECUTE ON PROCEDURE rzd.throwPassengerInfo TO user;

GRANT EXECUTE ON FUNCTION rzd.GetDepartureID TO user;

GRANT EXECUTE ON FUNCTION rzd.GetArrivalID TO user;

GRANT EXECUTE ON FUNCTION rzd.FindPassenger TO user;

GRANT EXECUTE ON FUNCTION rzd.PassengerAddToDB TO user;

GRANT EXECUTE ON FUNCTION rzd.ThrowRoutID TO user;

GRANT EXECUTE ON PROCEDURE rzd.throwCountAvailableTickets TO user;

GRANT EXECUTE ON PROCEDURE rzd.CancelTicket TO user;

GRANT EXECUTE ON PROCEDURE rzd.throwAvailableTicketsWithInfo TO user;

GRANT EXECUTE ON PROCEDURE rzd.EmployPlaces TO user;

GRANT EXECUTE ON PROCEDURE rzd.register\_createuser TO user;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_AddTrain TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_DropTrain TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_UpdateTrain TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_ShowTrains TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_ShowRailcars TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_ShowRoutes TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_ShowStops TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_ShowArrivals TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_ShowDepartures TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_DropRailcar TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_UpdateRailcar TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_UpdateRout TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_DropRout TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_UpdateStop TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_DropStop TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_UpdateArrival TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_DropArrival TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_UpdateDeparture TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_DropDeparture TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowRailcarListCount TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowTrainTypes TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowRoutsNames TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowRailcarTypesList TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowTrainList TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowAvailableRailcarList TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_AddRailcar TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowMaxStopCount TO RZD\_Dispatcher;

GRANT EXECUTE ON FUNCTION rzd.DISPATCHER\_AddRout TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowStopNumber TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_AddStop TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowAvailableStopsFindByRout TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.\_DISPATCHER\_ThrowAvailableTrainsFindByRout TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_AddArrival TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.DISPATCHER\_AddDeparture TO RZD\_Dispatcher;

GRANT EXECUTE ON PROCEDURE rzd.ADMIN\_ChangeRole TO Admin;

GRANT EXECUTE ON PROCEDURE rzd.ADMIN\_CreateUser TO Admin;

GRANT EXECUTE ON PROCEDURE rzd.ADMIN\_ThrowRoles TO Admin;

GRANT EXECUTE ON PROCEDURE rzd.ADMIN\_ThrowUsers TO Admin;

FLUSH PRIVILEGES;

END

ПРИЛОЖЕНИЕ Б  
Исходный код

Autorization.xaml

<Window x:Class="RZDKursovoy.Authorization"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:RZDKursovoy"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

Title="Билеты" MinHeight="550" MinWidth="800" Height="550" Width="800" ResizeMode="CanResize" WindowStartupLocation="CenterScreen">

<Grid x:Name="mainGrid">

<Grid x:Name="RegGrid" d:IsHidden="True">

<GroupBox x:Name="RegGB" Header="Регистрация">

<Grid>

<TextBox x:Name="RegLoginBox" PreviewTextInput="RegLoginBox\_PreviewTextInput" PreviewKeyUp="RegLoginBox\_PreviewKeyUp" MaxLength="30" Style="{StaticResource MaterialDesignFloatingHintTextBox}" HorizontalAlignment="Center" Margin="0,0,0,140" TextWrapping="Wrap" VerticalAlignment="Center" TabIndex="0" Height="40" Width="310" TextChanged="RegLoginBox\_TextChanged" >

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hintф>

<StackPanel Orientation="Horizontal" Margin="-2 0 0 0">

<materialDesign:PackIcon

Kind="Account" />

<TextBlock> Логин</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hintф>

</TextBox>

<TextBox x:Name="RegPassBox" PreviewTextInput="RegPassBox\_PreviewTextInput" PreviewKeyUp="RegPassBox\_PreviewKeyUp" Style="{StaticResource MaterialDesignFloatingHintTextBox}" HorizontalAlignment="Center" Margin="0,0,0,40" VerticalAlignment="Center" TabIndex="1" Height="40" Width="310" TextChanged="RegPassBox\_TextChanged">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<materialDesign:PackIcon

Kind="Key" />

<TextBlock>

Пароль

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<Button x:Name="RegRegButtonBEEP" Margin="0,120,0,0" Content="Зарегистрироваться" Click="RegRegButtonBEEP\_Click" HorizontalAlignment="Center" VerticalAlignment="Center" Width="310"/>

<CheckBox x:Name="CheckProcessingPersonalDataBox" Content="Я даю согласие на обработку персональных данных" Margin="0,40,0,0" HorizontalAlignment="Center" VerticalAlignment="Center"></CheckBox>

<Button x:Name="RegExit" Margin="0,0,0,10" Content="Выход" VerticalAlignment="Bottom" Click="RegExit\_Click"/>

</Grid>

</GroupBox>

</Grid>

<Grid x:Name="AuthGrid">

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="White" Offset="0"/>

<GradientStop Color="#FFD0F9D0" Offset="1"/>

<GradientStop Color="White"/>

</LinearGradientBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock x:Name="progNameText" Grid.Column="0" Grid.Row="1" Margin="0,0,0,0" TextWrapping="Wrap" Text="AVB Tickets. Авторизация" FontSize="20" TextAlignment="Center" FontFamily="Segoe UI Light" VerticalAlignment="Center"/>

<TextBox x:Name="loginBox" Grid.Column="0" Grid.Row="2" PreviewKeyUp="loginBox\_PreviewKeyUp" PreviewTextInput="loginBox\_PreviewTextInput" KeyDown="loginBox\_KeyDown" Style="{StaticResource MaterialDesignFloatingHintTextBox}" Margin="150,0" TextWrapping="Wrap" MaxLength="30" TabIndex="0" Height="40" Width="262" >

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<materialDesign:PackIcon

Kind="Account" />

<TextBlock>

Логин

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<PasswordBox x:Name="passBox" Grid.Column="0" Grid.Row="3" PreviewKeyUp="passBox\_PreviewKeyUp" PreviewTextInput="passBox\_PreviewTextInput" Style="{StaticResource MaterialDesignFloatingHintPasswordBox}" Margin="150,0" TabIndex="1" Width="262">

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<materialDesign:PackIcon

Kind="Key" />

<TextBlock> Пароль</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</PasswordBox>

<Button Grid.Column="0" Grid.Row="4" x:Name="authButton" Content="Авторизоваться" Margin="150,10" Height="Auto" TabIndex="2" IsDefault="True" Click="authButton\_Click" RenderTransformOrigin="0.494,0.5" Width="262"/>

<Button Grid.Column="0" Grid.Row="5" x:Name="regButton" Content="Регистрация" Click="regButton\_Click" Margin="150,10" Height="Auto" TabIndex="3" Width="262"/>

</Grid>

</Grid>

</Window>

Autorization.xaml.cs

using MySql.Data.MySqlClient;

using System.Windows;

using System.Windows.Controls;

namespace RZDKursovoy

{

public partial class Authorization : Window

{

private ApplicationLogic AL = new ApplicationLogic();

public string ThrowLogin { get; private set; } = "";

public MySqlConnection Connected { get; private set; }

public Authorization()

{

InitializeComponent();

RegGrid.Visibility = Visibility.Hidden;

AuthGrid.Visibility = Visibility.Visible;

}

private void authButton\_Click(object sender, RoutedEventArgs e)

{

try

{

ThrowLogin = loginBox.Text;

Connected = new MySqlConnection("Database ="+ Properties.PersonalData.Default.Database + "; " +

"DataSource = " + Properties.PersonalData.Default.DataSource + "; " +

"User Id = " + ThrowLogin + "; charset=cp866; Password =" + passBox.Password);

Connected.Open();

string CheckRole = "#####";

MySqlCommand checkrolecommand = new MySqlCommand("Select current\_role", Connected);

MySqlDataReader r = checkrolecommand.ExecuteReader();

r.Read();

try

{

CheckRole = r.GetString(0);

r.Close();

}

catch (System.Exception)

{}

if (CheckRole == "user")

{

Menu menu = new Menu();

menu.SetConnected = Connected;

menu.SetLogin = ThrowLogin;

mainGrid.Children.Clear();

mainGrid.Children.Add(menu);

menu.BoxesFiling();

return;

}

else if (CheckRole == "Blocked")

{

AL.MessageErrorShow("Ваш аккаунт заблокирован", "Ошибка");

return;

}

else if (CheckRole == "RZD\_Dispatcher")

{

Dispatcher\_Interface\_In\_To\_Face DITF = new Dispatcher\_Interface\_In\_To\_Face();

DITF.SetConnected = Connected;

DITF.SetLogin = ThrowLogin;

mainGrid.Children.Clear();

mainGrid.Children.Add(DITF);

AL.MessageShow("Привет, диспетчер " + ThrowLogin, "Привет!");

return;

}

else if (CheckRole == "Admin")

{

Admin\_Interface\_In\_To\_Face AIITF = new Admin\_Interface\_In\_To\_Face();

AIITF.SetConnected = Connected;

AIITF.SetLogin = ThrowLogin;

mainGrid.Children.Clear();

mainGrid.Children.Add(AIITF);

AL.MessageShow("Здравствуйте, администратор " + ThrowLogin, "Здравствуйте");

return;

}

else

{

AL.MessageErrorShow("Ваш аккаунт неправильно настроен. Обратитесь к администратору", "Ошибка");

return;

}

}

catch (MySqlException ex)

{

AL.MessageErrorShow("Вы ввели неправильный логин или пароль", "Ошибка");

return;

}

}

private void regButton\_Click(object sender, RoutedEventArgs e)

{

AuthGrid.Visibility = Visibility.Hidden;

RegGrid.Visibility = Visibility.Visible;

}

private void RegRegButtonBEEP\_Click(object sender, RoutedEventArgs e)

{

bool OK = false;

MySqlConnection FastConnect = new MySqlConnection("Database = " + Properties.PersonalData.Default.Database + "; " +

"DataSource = " + Properties.PersonalData.Default.DataSource + "; " +

"User Id = " + Properties.PersonalData.Default.RegLogin + "; charset=cp866; Password = "+ Properties.PersonalData.Default.RegPassword);

try

{

FastConnect.Open();

}

catch (MySqlException)

{

AL.MessageErrorShow("Сервер не отвечает", "Ошибка");

return;

}

if ((RegLoginBox.Text != "") && (RegPassBox.Text != ""))

{

if (CheckProcessingPersonalDataBox.IsChecked.Value)

{

try

{

var QueryString = "call register\_createuser";

string[] data = { RegLoginBox.Text, RegPassBox.Text };

var res = AL.MagicUniversalControlData(QueryString, data, "RegAdd", FastConnect);

poselki.BestErrors BE = new poselki.BestErrors();

BE.CatchError(res);

OK = true;

}

catch (MySqlException)

{

AL.MessageErrorShow("В настоящее время сервис регистрации недоступен. Пожалуйста, попробуйте позже", "Ошибка");

}

finally

{

FastConnect.Close();

}

}

else

{

AL.MessageErrorShow("Для продолжения работы с приложением вы должны дать согласие на обработку Ваших персональных данных", "Ошибка");

}

}

else

{

AL.MessageErrorShow("Вы не заполнили поля", "Ошибка");

}

if (OK)

{

RegGrid.Visibility = Visibility.Hidden;

AuthGrid.Visibility = Visibility.Visible;

}

}

private void RegExit\_Click(object sender, RoutedEventArgs e)

{

RegGrid.Visibility = Visibility.Hidden;

AuthGrid.Visibility = Visibility.Visible;

}

private void RegLoginBox\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.EN\_InputLoginWordsProtector(RegLoginBox, e);

}

private void loginBox\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.EN\_InputLoginWordsProtector(loginBox, e);

}

private void RegLoginBox\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RegLoginBox, e);

}

private void RegPassBox\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RegPassBox, e);

}

private void loginBox\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(loginBox, e);

}

}

}

ApplicationLogic.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Text.RegularExpressions;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Media;

namespace RZDKursovoy

{

class ApplicationLogic

{

/\*Переменные, предназначенные для защиты ввода\*/

private Regex OnlyLowCaseWordsChecker = new Regex("[а-я]");

private Regex OnlyUpCaseWordsChecker = new Regex("[А-Я]");

private Regex OnlyLowCaseWordsCheckerWithMinus = new Regex("[а-я -]");

private Regex OnlyUpCaseWordsCheckerWithMinus = new Regex("[А-Я -]");

private Regex EN\_OnlyLowCaseWordsChecker = new Regex("[a-z]");

private Regex EN\_OnlyUpCaseWordsChecker = new Regex("[A-Z]");

private Regex OnlyNumbersChecker = new Regex("[1-9 0]");

/\*Универсальные процедуры\*/

public string[] MagicUniversalControlData(string QueryString, string[] DataArgs, string userControl, MySqlConnection Connection)

{

if (userControl != "Delete")

{

QueryString += "(";

string[] ParameterArg = new string[DataArgs.Length];

for (int i = 0; i < DataArgs.Length; i++)

{

if (i != DataArgs.Length - 1)

QueryString += "@ARG" + i + ", ";

else

QueryString += "@ARG" + i;

ParameterArg[i] = "@ARG" + i;

}

QueryString += ")";

var BestCommand = new MySqlCommand(QueryString, Connection);

for (int i = 0; i < DataArgs.Length; i++)

{

BestCommand.Parameters.AddWithValue(ParameterArg[i], DataArgs[i]);

}

try

{

BestCommand.ExecuteNonQuery();

}

catch (MySqlException ex)

{

string[] Result = { ex.Message, ex.Number.ToString() };

return Result;

}

switch (userControl)

{

case "RegAdd":

MessageShow("Ваш аккаунт успешно зарегистрирован", "ОК");

break;

case "Reservation":

MessageShow("Спасибо за покупку! Билет доступен в личном кабинете", "ОК");

break;

case "DeleteTicket":

MessageShow("Резервирование отменено, деньги не вернутся на ваш счет (потому-что вы их не платили), а отмененный билет больше недействителен", "ОК");

break;

case "UpdateTrain":

MessageShow("Информация о поезде была отредактирована", "ОК");

break;

case "UpdateRailcar":

MessageShow("Информация о вагоне была отредактирована", "ОК");

break;

case "UpdateStop":

MessageShow("Информация об остановке была отредактирована", "ОК");

break;

case "UpdateRout":

MessageShow("Информация о маршруте была отредактирована", "ОК");

break;

case "DeleteTrain":

MessageShow("Информация о поезде была успешно удалена", "ОК");

break;

case "DeleteRailcar":

MessageShow("Информация о вагоне была успешно удалена", "ОК");

break;

case "DeleteStop":

MessageShow("Информация об остановке была успешно удалена", "ОК");

break;

case "DeleteRout":

MessageShow("Информация о маршруте была успешно удалена", "ОК");

break;

case "DeleteArrival":

MessageShow("Информация о прибытии поезда была успешно удалена", "ОК");

break;

case "UpdateArrival":

MessageShow("Информация о прибытии поезда была успешно изменена", "ОК");

break;

case "DeleteDeparture":

MessageShow("Информация об отправлении поезда была успешно удалена", "ОК");

break;

case "UpdateDeparture":

MessageShow("Информация об отправлении поезда была успешно изменена", "ОК");

break;

case "AddTrain":

MessageShow("Данные о поезде успешно добавлены", "ОК");

break;

case "AddRailcar":

MessageShow("Данные о вагоне успешно добавлены", "ОК");

break;

case "AddRout":

MessageShow("Данные о маршруте успешно добавлены", "ОК");

break;

case "AddStop":

MessageShow("Данные об остановке успешно добавлены", "ОК");

break;

case "AddArrival":

MessageShow("Данные прибытия поезда успешно добавлены", "ОК");

break;

case "AddDeparture":

MessageShow("Данные отправления поезда успешно добавлены", "ОК");

break;

case "CreateUser":

MessageShow("Пользователь успешно зарегистрирован", "ОК");

break;

case "UpdateUser":

MessageShow("Информация о пользователе была успешно обновлена", "ОК");

break;

}

}

else if (userControl == "Delete")

{

var DelCommand = new MySqlCommand(QueryString, Connection);

DelCommand.Parameters.AddWithValue("Num", DataArgs[0]);

try

{

DelCommand.ExecuteNonQuery();

}

catch (MySqlException ex)

{

string[] Result = { ex.Message, ex.Number.ToString() };

return Result;

}

}

string[] Result3 = { "OK", "1" };

return Result3;

}

public void MagicUniversalControlDataCatched(string QueryString, string[] DataArgs, string userControl, MySqlConnection Connection)

{

var res = MagicUniversalControlData(QueryString, DataArgs, userControl, Connection);

poselki.BestErrors BE = new poselki.BestErrors();

BE.CatchError(res);

}

public void MagicUserControl(MySqlConnection connected, DataRowView t, string Proc, string Control)

{

string[] args = new string[t.Row.ItemArray.Length];

for (int i = 0; i < t.Row.ItemArray.Length; i++)

args[i] = t.Row.ItemArray[i].ToString();

var res = MagicUniversalControlData(Proc, args, Control, connected);

poselki.BestErrors BE = new poselki.BestErrors();

BE.CatchError(res);

}

public List<string> CatchStringListResult(MySqlConnection con, string Query, string[] Args)

{

List<string> Result = new List<string>();

MySqlDataReader ResultReader = null;

Query += "(";

string[] ParameterArg = new string[Args.Length];

if (Args[0] != "null")

{

for (int i = 0; i < Args.Length; i++)

{

if (i != Args.Length - 1)

Query += "@ARG" + i + ", ";

else

Query += "@ARG" + i;

ParameterArg[i] = "@ARG" + i;

}

}

Query += ")";

var BestCommand = new MySqlCommand(Query, con);

if (Args[0] != "null")

{

for (int i = 0; i < Args.Length; i++)

{

BestCommand.Parameters.AddWithValue(ParameterArg[i], Args[i]);

}

}

try

{

ResultReader = BestCommand.ExecuteReader();

while (ResultReader.Read())

{

Result.Add(ResultReader.GetString(0));

}

}

catch (MySqlException e)

{

var trace = e.Message.ToString();

Result.Clear();

Result.Add("-1");

return Result;

}

finally

{

ResultReader.Close();

}

return Result;

}

public int CatchIntResult(MySqlConnection con, string Query, string[] Args)

{

int Result = new int();

MySqlDataReader ResultReader = null;

Query += "(";

string[] ParameterArg = new string[Args.Length];

if (Args[0] != "null")

{

for (int i = 0; i < Args.Length; i++)

{

if (i != Args.Length - 1)

Query += "@ARG" + i + ", ";

else

Query += "@ARG" + i;

ParameterArg[i] = "@ARG" + i;

}

}

Query += ")";

var BestCommand = new MySqlCommand(Query, con);

if (Args[0] != "null")

{

for (int i = 0; i < Args.Length; i++)

{

BestCommand.Parameters.AddWithValue(ParameterArg[i], Args[i]);

}

}

try

{

ResultReader = BestCommand.ExecuteReader();

ResultReader.Read();

Result = ResultReader.GetInt32(0);

}

catch (Exception)

{

Result = -1;

return Result;

}

finally

{

ResultReader.Close();

}

return Result;

}

public bool ConvertCheck(object sender, DataGridCellEditEndingEventArgs e, int[] itemarr)

{

if (itemarr[0] != -1)

{

for (int i = 0; i < itemarr.Length; i++)

{

if (e.Column.DisplayIndex == itemarr[i])

{

var t = e.EditingElement.ToString().Split(':');

try

{

Convert.ToInt32(t[1]);

}

catch (Exception)

{

MessageErrorShow("Вы ввели - " + t[1] + ". Это значение не является целым числом", "Ошибка");

return false;

}

}

}

}

return true;

}

public DataRowView BlockUpdate(object sender, DataGridCellEditEndingEventArgs e, int[] itemarr)

{

DataRowView TMPGridRow = null;

if (itemarr[0] != -1)

{

for (int i = 0; i < itemarr.Length; i++)

{

if (e.Column.DisplayIndex != i)

{

try

{

TMPGridRow = (DataRowView)(sender as DataGrid).CurrentItem;

}

catch (Exception)

{ }

}

else

{

TMPGridRow = null;

return TMPGridRow;

}

}

}

else

{

try

{

TMPGridRow = (DataRowView)(sender as DataGrid).CurrentItem;

}

catch (Exception)

{ }

}

return TMPGridRow;

}

/\*Уникальные процедуры\*/

public List<string> Available\_Railcar\_Types(MySqlConnection connection, string TrainNum)

{

List<string> SavedTypes = new List<string>();

string QueryString = "call Available\_Railcar\_Types(@Train\_Num)";

var BestCommand = new MySqlCommand(QueryString, connection);

BestCommand.Parameters.AddWithValue("Train\_Num", TrainNum);

var RailcarsTypesReader = BestCommand.ExecuteReader();

while (RailcarsTypesReader.Read())

{

SavedTypes.Add(RailcarsTypesReader.GetString(0) + " цена за билет - " + RailcarsTypesReader.GetString(1) + " руб.");

}

RailcarsTypesReader.Close();

return SavedTypes;

}

public List<string> TrainInfo(MySqlConnection connected, string TrainNumber, int Arrival\_ID\_IN, int Departure\_ID\_IN)

{

List<string> TrainInfoList = new List<string>();

var QueryString = "call TrainInfoTwo(@TrainNumber, @Arrival\_ID\_IN, @Departure\_ID\_IN)";

var BestCommand = new MySqlCommand(QueryString, connected);

BestCommand.Parameters.AddWithValue("TrainNumber", TrainNumber);

BestCommand.Parameters.AddWithValue("Arrival\_ID\_IN", Arrival\_ID\_IN);

BestCommand.Parameters.AddWithValue("Departure\_ID\_IN", Departure\_ID\_IN);

var r = BestCommand.ExecuteReader();

r.Read();

TrainInfoList.Add(r.GetString(0).ToString());

TrainInfoList.Add(r.GetString(1).ToString());

TrainInfoList.Add(r.GetString(2).ToString());

TrainInfoList.Add(r.GetString(3).ToString());

TrainInfoList.Add(r.GetString(4).ToString());

TrainInfoList.Add(r.GetString(5).ToString());

r.Close();

return TrainInfoList;

}

public List<string> FindPassengerWithPersonalData(MySqlConnection connection, int Passport\_Series\_IN, int Passport\_Number\_IN)

{

List<string> Result = new List<string>();

var QueryString = "call FindPassengerWithPersonalData(@Passport\_Series\_IN, @Passport\_Number\_IN, @KeySi)";

var BestCommand = new MySqlCommand(QueryString, connection);

BestCommand.Parameters.AddWithValue("Passport\_Series\_IN", Passport\_Series\_IN);

BestCommand.Parameters.AddWithValue("Passport\_Number\_IN", Passport\_Number\_IN);

BestCommand.Parameters.AddWithValue("KeySi", Properties.PersonalData.Default.KeySi);

var r = BestCommand.ExecuteReader();

r.Read();

for (int i = 0; i < 4; i++)

{ Result.Add(r.GetString(i)); }

r.Close();

return Result;

}

public List<string> throwPassengerInfo(MySqlConnection connected, int Reservation\_ID\_IN)

{

List<string> result = new List<string>();

var QueryString = "call throwPassengerInfo(@Reservation\_ID\_IN, @KeySi)";

var BestCommand = new MySqlCommand(QueryString, connected);

BestCommand.Parameters.AddWithValue("Reservation\_ID\_IN", Reservation\_ID\_IN);

BestCommand.Parameters.AddWithValue("KeySi", Properties.PersonalData.Default.KeySi);

var PassengerInfoRead = BestCommand.ExecuteReader();

while (PassengerInfoRead.Read())

{

for (int i = 0; i < 5; i++)

result.Add(PassengerInfoRead.GetString(i));

}

PassengerInfoRead.Close();

return result;

}

public List<string> throwRailcarInfo(MySqlConnection connected, string Train\_Number\_IN, int Railcar\_Number\_IN)

{

List<string> result = new List<string>();

var QueryString = "call throwRailcarInfo('" + Train\_Number\_IN + "', '" + Railcar\_Number\_IN + "')";

var BestCommand = new MySqlCommand(QueryString, connected);

var RailcarInfoRead = BestCommand.ExecuteReader();

while (RailcarInfoRead.Read())

{

for (int i = 0; i < 2; i++)

result.Add(RailcarInfoRead.GetString(i));

}

RailcarInfoRead.Close();

return result;

}

public string FindTrain(MySqlConnection connected, string RoutID, int Arrival\_Stop\_IN, string Arrival\_Date\_IN)

{

string QueryString = "SELECT FindTrain(@RoutID, @Arrival\_Stop\_IN, @Arrival\_Date\_IN)";

var BestCommand = new MySqlCommand(QueryString, connected);

BestCommand.Parameters.AddWithValue("RoutID", RoutID);

BestCommand.Parameters.AddWithValue("Arrival\_Stop\_IN", Arrival\_Stop\_IN);

BestCommand.Parameters.AddWithValue("Arrival\_Date\_IN", Arrival\_Date\_IN);

MySqlDataReader r = BestCommand.ExecuteReader();

string TrainID = "";

r.Read();

TrainID = r.GetString(0).ToString();

r.Close();

return TrainID;

}

/\*Процедуры для проверки ввода\*/

public void InputPersonalDataProtector(System.Windows.Controls.TextBox TB, System.Windows.Input.TextCompositionEventArgs e)

{

if (TB.Text.Length != 0)

{

Match matchLowCase = OnlyLowCaseWordsChecker.Match(e.Text);

if (!matchLowCase.Success)

e.Handled = true;

}

else

{

Match matchUpCase = OnlyUpCaseWordsChecker.Match(e.Text);

if (!matchUpCase.Success)

e.Handled = true;

}

}

public void InputNumbersDataProtector(System.Windows.Controls.TextBox TB, System.Windows.Input.TextCompositionEventArgs e)

{

Match match = OnlyNumbersChecker.Match(e.Text);

if (!match.Success)

e.Handled = true;

}

public void InputWordsProtector(System.Windows.Input.TextCompositionEventArgs e)

{

Match matchLowCase = OnlyLowCaseWordsChecker.Match(e.Text);

Match matchUpCase = OnlyUpCaseWordsChecker.Match(e.Text);

if ((!matchLowCase.Success) && (!matchUpCase.Success))

{

e.Handled = true;

}

}

public void EN\_InputWordsProtector(System.Windows.Input.TextCompositionEventArgs e)

{

Match matchLowCase = EN\_OnlyLowCaseWordsChecker.Match(e.Text);

Match matchUpCase = EN\_OnlyUpCaseWordsChecker.Match(e.Text);

if ((!matchLowCase.Success) && (!matchUpCase.Success))

{

e.Handled = true;

}

}

public void EN\_InputLoginWordsProtector(System.Windows.Controls.TextBox TB, System.Windows.Input.TextCompositionEventArgs e)

{

Match matchLowCase = EN\_OnlyLowCaseWordsChecker.Match(e.Text);

Match matchUpCase = EN\_OnlyUpCaseWordsChecker.Match(e.Text);

Match matchNumbersCase = OnlyNumbersChecker.Match(e.Text);

if ((!matchLowCase.Success) && (!matchUpCase.Success) && (!matchNumbersCase.Success))

{

e.Handled = true;

}

}

public void Input\_RoutNameProtector(System.Windows.Controls.TextBox TB, System.Windows.Input.TextCompositionEventArgs e)

{

Match CheckOne = OnlyLowCaseWordsCheckerWithMinus.Match(e.Text);

Match CheckTwo = OnlyUpCaseWordsCheckerWithMinus.Match(e.Text);

if ((!CheckOne.Success) && (!CheckTwo.Success))

e.Handled = true;

}

public bool DontCtrlVAndSpace(TextBox TB, KeyEventArgs e)

{

if (e.Key == Key.Space)

{

MessageErrorShow("Пробел запрещен", "Ошибка");

TB.Text = "";

return true;

}

if (e.Key == Key.V && (Keyboard.Modifiers & ModifierKeys.Control) == ModifierKeys.Control)

{

MessageErrorShow("Вставка символов запрещена", "Ошибка");

TB.Text = "";

return false;

}

if (e.Key == Key.Z && (Keyboard.Modifiers & ModifierKeys.Control) == ModifierKeys.Control)

{

MessageErrorShow("Возврат символов запрещён", "Ошибка");

TB.Text = "";

return false;

}

return false;

}

internal void DontCtrlVAndSpace(ComboBox CB, KeyEventArgs e)

{

if (e.Key == Key.Space)

{

MessageErrorShow("Пробел запрещен", "Ошибка");

CB.Text = "";

}

if (e.Key == Key.V && (Keyboard.Modifiers & ModifierKeys.Control) == ModifierKeys.Control)

{

MessageErrorShow("Вставка символов запрещена", "Ошибка");

CB.Text = "";

}

if (e.Key == Key.Z && (Keyboard.Modifiers & ModifierKeys.Control) == ModifierKeys.Control)

{

MessageErrorShow("Возврат символов запрещён", "Ошибка");

CB.Text = "";

}

}

/\*Вывод сообщения\*/

public void MessageErrorShow(string Message, string Title)

{

var msg = new BespokeFusion.CustomMaterialMessageBox

{

TxtMessage = { Text = Message, Foreground = Brushes.Black },

TxtTitle = { Text = Title, Foreground = Brushes.White },

BtnOk = { Content = "Ок", Background = Brushes.DarkRed, BorderBrush = Brushes.DarkRed},

BtnCancel = {Visibility = Visibility.Collapsed},

BtnCopyMessage = { Visibility = Visibility.Collapsed },

MainContentControl = { Background = Brushes.White},

TitleBackgroundPanel = { Background = Brushes.DarkRed },

BorderBrush = Brushes.DarkRed

};

msg.Show();

}

public void MessageShow(string Message, string Title)

{

var msg = new BespokeFusion.CustomMaterialMessageBox

{

TxtMessage = { Text = Message, Foreground = Brushes.Black },

TxtTitle = { Text = Title, Foreground = Brushes.White },

BtnOk = { Content = "Ок", Background = Brushes.LightSeaGreen, BorderBrush = Brushes.LightSeaGreen },

BtnCancel = { Visibility = Visibility.Collapsed },

BtnCopyMessage = { Visibility = Visibility.Collapsed},

MainContentControl = { Background = Brushes.White},

TitleBackgroundPanel = { Background = Brushes.LightSeaGreen },

BorderBrush = Brushes.LightSeaGreen

};

msg.Show();

}

/\*Dispatcher Procedures\*/

public bool KeyUpInside(MySqlConnection Connected, object sender, System.Windows.Input.KeyEventArgs e, DataGrid grid, DataRowView TMPRow, bool ConvertCheck, string DeleteCommand,

string UpdateCommand, string DeleteControl, string UpdateControl, string Error, string [] args, int [] activity )

{

try

{

var r = e.Key.ToString();

if (r == "Delete")

{

if (activity[0] == 0)

{

var res = MagicUniversalControlData(DeleteCommand, args, DeleteControl, Connected);

poselki.BestErrors BE = new poselki.BestErrors();

BE.CatchError(res);

}

else

{

MessageShow("Удаление временно недоступно. Следите за обновлениями приложения", "!");

}

return true;

}

else if (r == "Return")

{

if (activity[1] == 0)

{

var t1 = TMPRow;

if (t1 != null)

{

if (ConvertCheck)

{

MagicUserControl(Connected, t1, UpdateCommand, UpdateControl);

}

}

else

{

MessageErrorShow(Error, "Ошибка");

}

}

else

{

MessageShow("Редактирование временно недоступно. Следите за обновлениями приложения", "!");

}

return true;

}

else if (r == "Escape")

{

return true;

}

}

catch (Exception)

{ }

return false;

}

public bool FillTable(MySqlConnection connection, DataGrid DG, string query, string [] args)

{

try

{

MySqlDataAdapter ad = new MySqlDataAdapter();

ad.SelectCommand = new MySqlCommand(query, connection);

DataTable table = new DataTable();

ad.Fill(table);

for (int i = 0; i < args.Length; i++)

{

table.Columns[i].ColumnName = args[i];

}

DG.ItemsSource = table.DefaultView;

}

catch (Exception ex)

{

var e = ex.Message;

MessageErrorShow("При загрузке данных произошла ошибка", "Ошибка");

return false;

}

return true;

}

public bool ComboboxFiling(MySqlConnection connection, string query, ComboBox CB)

{

try

{

string[] args = { "null" };

var result = CatchStringListResult(connection, query, args);

for (int i = 0; i < result.Count; i++)

{ CB.Items.Add(result[i]); }

}

catch (Exception)

{ return false; }

return true;

}

public bool TextChecking(string [] args)

{

for (int i = 0; i < args.Length; i++)

{

if (string.IsNullOrEmpty(args[i]))

{

MessageErrorShow("Вы не заполнили одно или несколько полей, необходимых для добавления", "Ошибка");

return false;

}

}

return true;

}

public List<string> En\_To\_Ru\_Roles(List<string> EnArgs)

{

List<string> RuArgs = new List<string>();

for (int i = 0; i < EnArgs.Count; i++)

{

switch (EnArgs[i])

{

case "Blocked":

RuArgs.Add("Заблокированный");

break;

case "user":

RuArgs.Add("Пользователь");

break;

case "Admin":

RuArgs.Add("Администратор");

break;

case "RZD\_Dispatcher":

RuArgs.Add("Диспетчер");

break;

default:

MessageErrorShow("При загрузке ролей произошла ошибка", "Ошибка");

return RuArgs;

}

}

return RuArgs;

}

public string Ru\_To\_En\_Roles(string RuValue)

{

string Result = "";

switch (RuValue)

{

case "Заблокированный":

Result = "Blocked";

break;

case "Пользователь":

Result = "user";

break;

case "Администратор":

Result = "Admin";

break;

case "Диспетчер":

Result = "RZD\_Dispatcher";

break;

default:

MessageErrorShow("При загрузке ролей произошла ошибка", "Ошибка");

return "error";

}

return Result;

}

}

}

Admin\_AddUser.xaml

<Window x:Class="RZDKursovoy.Admin\_AddUser"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

Title="Работа с пользователями" Height="450" Width="600" WindowStartupLocation="CenterScreen" MinHeight="450" MinWidth="600">

<Grid x:Name="AddUserMenu" Loaded="AddUserMenu\_Loaded">

<GroupBox Header="Добавление пользователя">

<Grid>

<TextBox x:Name="Username\_BOX" MaxLength="30" PreviewTextInput="Username\_BOX\_PreviewTextInput" PreviewKeyUp="Username\_BOX\_PreviewKeyUp" Style="{StaticResource MaterialDesignFloatingHintTextBox}" VerticalAlignment="Center" Height="36" Width="289" HorizontalAlignment="Center" Margin="0,0,0,140">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel Orientation="Horizontal" Margin="-2 0 0 0">

<TextBlock>Логин</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<ComboBox materialDesign:HintAssist.Hint="Роль пользователя" x:Name="UserRole\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,70" Width="289">

</ComboBox>

<TextBox x:Name="UserPassword" MaxLength="30" PreviewTextInput="UserPassword\_PreviewTextInput" PreviewKeyUp="UserPassword\_PreviewKeyUp" Style="{StaticResource MaterialDesignFloatingHintTextBox}" VerticalAlignment="Center" Height="36" Width="289" HorizontalAlignment="Center" Margin="0,0,0,0">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel Orientation="Horizontal" Margin="-2 0 0 0">

<TextBlock>Пароль</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<Button x:Name="AddUser\_BUTTON" Click="AddUser\_BUTTON\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0,90,0,0" Width="289" Content="Добавить"></Button>

</Grid>

</GroupBox>

</Grid>

</Window>

Admin\_AddUser.xaml.cs

using MySql.Data.MySqlClient;

using System.Windows;

namespace RZDKursovoy

{

public partial class Admin\_AddUser : Window

{

public MySqlConnection Connected { get; private set; }

public MySqlConnection SetConnected { set { Connected = value; } }

public Admin\_Interface\_In\_To\_Face Admin\_Interface { get; private set; }

public Admin\_Interface\_In\_To\_Face SetInterface{ set { Admin\_Interface= value; } }

private ApplicationLogic AL = new ApplicationLogic();

public Admin\_AddUser()

{

InitializeComponent();

}

private void AddUserMenu\_Loaded(object sender, RoutedEventArgs e)

{

UserRole\_BOX.Items.Clear();

string[] args = { "null" };

var r = AL.CatchStringListResult(Connected, "call ADMIN\_ThrowRoles", args);

var TMPResult = AL.En\_To\_Ru\_Roles(r);

for (int i = 0; i < TMPResult.Count; i++)

{

UserRole\_BOX.Items.Add(TMPResult[i]);

}

}

private void AddUser\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

string[] args = { Username\_BOX.Text, UserPassword.Text, AL.Ru\_To\_En\_Roles(UserRole\_BOX.Text) };

if (AL.TextChecking(args))

{

AL.MagicUniversalControlDataCatched("call ADMIN\_CreateUser", args, "CreateUser", Connected);

Admin\_Interface.FillUserTable();

Admin\_Interface.LoadUserList();

}

}

private void Username\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(Username\_BOX, e);

}

private void UserPassword\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(UserPassword, e);

}

private void Username\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.EN\_InputLoginWordsProtector(Username\_BOX, e);

}

private void UserPassword\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.EN\_InputLoginWordsProtector(UserPassword, e);

}

}

}

Admin\_interface\_in\_to\_face.xaml

<UserControl x:Class="RZDKursovoy.Admin\_Interface\_In\_To\_Face"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:dragablz="http://dragablz.net/winfx/xaml/dragablz"

d:DesignHeight="450" MinHeight="450" d:DesignWidth="800" MinWidth="800">

<Grid x:Name="Reflector">

<dragablz:TabablzControl x:Name="ControlMenu" Loaded="ControlMenu\_Loaded">

<TabItem x:Name="Reservation\_TAB"

Header="Пользователи"

Margin="0" d:IsHidden="True">

<Grid>

<DataGrid CanUserAddRows="False" IsReadOnly="True" CanUserDeleteRows="False" CanUserReorderColumns="False" CanUserResizeColumns="False" CanUserResizeRows="False" x:Name="UserShow" Margin="0,0,0,34" ></DataGrid>

<Button x:Name="AddUserBUTTON" Click="AddUserBUTTON\_Click" Content="Добавить пользователя" VerticalAlignment="Bottom" Height="34" Margin="1"/>

</Grid>

</TabItem>

<TabItem x:Name="AccessTab" Header="Управление правами" Loaded="AccessTab\_Loaded">

<Grid>

<ComboBox materialDesign:HintAssist.Hint="Логин пользователя" x:Name="Username\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,140" Width="289">

</ComboBox>

<ComboBox materialDesign:HintAssist.Hint="Роль пользователя" x:Name="UserRole\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,50" Width="289">

</ComboBox>

<Button x:Name="UserEdit\_BUTTON" Click="UserEdit\_BUTTON\_Click" Content="Обновить доступ пользователя" VerticalAlignment="Center" HorizontalAlignment="Center" Width="289" Margin="0,60,0,0" ></Button>

</Grid>

</TabItem>

</dragablz:TabablzControl>

</Grid>

</UserControl>

Admin\_interface\_in\_to\_face.xaml.cs

using MySql.Data.MySqlClient;

using System.Windows.Controls;

namespace RZDKursovoy

{

/// <summary>

/// Логика взаимодействия для Admin\_Interface\_In\_To\_Face.xaml

/// </summary>

public partial class Admin\_Interface\_In\_To\_Face : UserControl

{

public MySqlConnection Connected { get; private set; }

public MySqlConnection SetConnected { set { Connected = value; } }

private string Login = "";

public string SetLogin { set { Login = value; } }

private ApplicationLogic AL = new ApplicationLogic();

public Admin\_Interface\_In\_To\_Face()

{

InitializeComponent();

}

public void FillUserTable()

{

string[] args = { "Логин", "Роль" };

AL.FillTable(Connected, UserShow, "call ADMIN\_ThrowUsers", args);

}

private void ControlMenu\_Loaded(object sender, System.Windows.RoutedEventArgs e)

{

FillUserTable();

}

private void AddUserBUTTON\_Click(object sender, System.Windows.RoutedEventArgs e)

{

Admin\_AddUser AAU = new Admin\_AddUser();

AAU.SetConnected = Connected;

AAU.SetInterface = this;

AAU.Show();

}

public void LoadUserList()

{

Username\_BOX.Items.Clear();

if (!AL.ComboboxFiling(Connected, "call ADMIN\_ThrowUsers", Username\_BOX))

{

AL.MessageErrorShow("При загрузке пользователей произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

UserRole\_BOX.Items.Clear();

string[] args = { "null" };

var r = AL.CatchStringListResult(Connected, "call ADMIN\_ThrowRoles", args);

var TMPResult = AL.En\_To\_Ru\_Roles(r);

for (int i = 0; i < TMPResult.Count; i++)

{

UserRole\_BOX.Items.Add(TMPResult[i]);

}

}

private void AccessTab\_Loaded(object sender, System.Windows.RoutedEventArgs e)

{

LoadUserList();

}

private void UserEdit\_BUTTON\_Click(object sender, System.Windows.RoutedEventArgs e)

{

string[] args = { Username\_BOX.Text, AL.Ru\_To\_En\_Roles(UserRole\_BOX.Text) };

AL.MagicUniversalControlDataCatched("call ADMIN\_ChangeRole", args, "UpdateUser", Connected);

FillUserTable();

}

}

}

Besterrors.cs

using System;

using System.IO;

using System.Windows;

namespace poselki

{

class BestErrors

{

public void ExceptionProtector()

{

MessageBox.Show("Ой, ой. А что это? Это ошибка, но, к сожалению, я о ней уже знал", "Митинг подавлен", MessageBoxButton.OK, MessageBoxImage.Information);

}

public string getError(string NumError)

{

switch (NumError)

{

case "1406":

NumError = "Вы заполнили слишком длинное значение для одного или нескольких полей";

break;

case "1452":

NumError = "He удается добавить или обновить дочернюю строку: ограничение внешнего ключа дает сбой";

break;

case "1062":

NumError = "Произошла ошибка. Дублирование информации запрещено";

break;

case "1366":

NumError = "Вы заполнили одно или несколько полей некорректно, извинитесь перед разработчиком и пообещайте ему что вы больше так не будете";

break;

case "1370":

NumError = "У вас нет необходимых прав, чтобы выполнить данное действие";

break;

case "1394":

NumError = "При работе с пользователями произошла ошибка. Возможно, что пользователь с таким логином уже существует";

break;

case "1396":

NumError = "При работе с пользователями произошла ошибка";

break;

case "1133":

NumError = "Такого пользователя не существует";

break;

case "1141":

NumError = "Такого пользователя не существует";

break;

case "1411":

NumError = "Произошла ошибка. Вы неправильно заполнили дату";

break;

case "1292":

NumError = "Произошла ошибка. Вы неправильно заполнили время или дату";

break;

default:

string ErrorToFile = "Произошла неизвестная ошибка. Номер ошибки - " + NumError;

if (LogMessageToFile(ErrorToFile))

NumError = "Произошла неизвестная ошибка. Номер ошибки - " + NumError + ". Отправьте разработчику лог - " + GetTempPath() + "AVBIncLogFile.txt";

else

NumError = "Произошла неизвестная ошибка. Номер ошибки - " + NumError;

break;

}

return NumError;

}

public void CatchError(string [] Error)

{

if (Error[1] != "1")

{

RZDKursovoy.ApplicationLogic AL = new RZDKursovoy.ApplicationLogic();

if (Error[1] != "1644")

{

AL.MessageErrorShow(getError(Error[1]), "Ошибка");

}

else

{

AL.MessageErrorShow(Error[0], "Ошибка");

}

}

}

private string GetTempPath()

{

string path = System.Environment.GetEnvironmentVariable("TEMP");

if (!path.EndsWith("\\")) path += "\\";

return path;

}

private bool LogMessageToFile(string msg)

{

StreamWriter sw = File.AppendText(

GetTempPath() + "AVBIncLogFile.txt");

try

{

string logLine = System.String.Format(

"{0:G}: {1}.", DateTime.Now, msg);

sw.WriteLine(logLine);

}

catch (Exception)

{

return false;

}

finally

{

sw.Close();

}

return true;

}

}

}

Dispatcher\_addarrival.xaml

<UserControl x:Class="RZDKursovoy.Dispatcher\_AddArrival"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<Grid>

<GroupBox Header="Добавление прибытия поезда">

<Grid>

<ComboBox LostFocus="RoutName\_BOX\_LostFocus" PreviewTextInput="RoutName\_BOX\_PreviewTextInput" PreviewKeyUp="RoutName\_BOX\_PreviewKeyUp" Loaded="RoutName\_BOX\_Loaded" materialDesign:HintAssist.Hint="Название маршрута" x:Name="RoutName\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,200" Width="289">

</ComboBox>

<ComboBox materialDesign:HintAssist.Hint="Название остановки" PreviewTextInput="StopName\_BOX\_PreviewTextInput" PreviewKeyUp="StopName\_BOX\_PreviewKeyUp" x:Name="StopName\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,130" Width="289">

</ComboBox>

<ComboBox materialDesign:HintAssist.Hint="Номер поезда" PreviewTextInput="Train\_Number\_BOX\_PreviewTextInput" PreviewKeyUp="Train\_Number\_BOX\_PreviewKeyUp" x:Name="Train\_Number\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,60" Width="289">

</ComboBox>

<materialDesign:TimePicker Language="ru-RU" HorizontalAlignment="Center" Margin="0,20,0,0" materialDesign:HintAssist.Hint="Время" x:Name="ArrivalTime" Width="289" VerticalAlignment="Center"></materialDesign:TimePicker>

<DatePicker x:Name="ArrivalDate" Language="ru-RU" VerticalAlignment="Center" materialDesign:HintAssist.Hint="Дата" HorizontalAlignment="Center" Width="289" Margin="0,80,0,0" ></DatePicker>

<Button x:Name="AddArrival\_BUTTON" Click="AddArrival\_BUTTON\_Click" VerticalAlignment="Center" Width="289" Content="Добавить" HorizontalAlignment="Center" Margin="0,150,0,0"></Button>

</Grid>

</GroupBox>

</Grid>

</UserControl>

Dispatcher\_addarrival.xaml.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace RZDKursovoy

{

public partial class Dispatcher\_AddArrival : UserControl

{

private MySqlConnection \_connected;

public MySqlConnection SetConnection

{

set { \_connected = value; }

}

private Dispatcher\_Interface\_In\_To\_Face DIITF;

private ApplicationLogic AL = new ApplicationLogic();

public Dispatcher\_Interface\_In\_To\_Face SetInterface

{

set { DIITF = value; }

}

public Dispatcher\_AddArrival()

{

InitializeComponent();

}

private void RoutName\_BOX\_LostFocus(object sender, RoutedEventArgs e)

{

if (RoutName\_BOX.Text != "")

{

StopName\_BOX.Items.Clear();

Train\_Number\_BOX.Items.Clear();

string[] args = { RoutName\_BOX.Text };

try

{

var r = AL.CatchStringListResult(\_connected, "call \_DISPATCHER\_ThrowAvailableStopsFindByRout", args);

for (int i = 0; i < r.Count; i++)

{

StopName\_BOX.Items.Add(r[i]);

}

}

catch (Exception)

{

AL.MessageErrorShow("При загрузке остановок произошла ошибка", "Ошибка");

this.IsEnabled = false;

return;

}

try

{

var r2 = AL.CatchStringListResult(\_connected, "call \_DISPATCHER\_ThrowAvailableTrainsFindByRout", args);

for (int i = 0; i < r2.Count; i++)

{

Train\_Number\_BOX.Items.Add(r2[i]);

}

}

catch (Exception)

{

AL.MessageErrorShow("При загрузке поездов произошла ошибка", "Ошибка");

this.IsEnabled = false;

return;

}

}

}

private void RoutName\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

RoutName\_BOX.Items.Clear();

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowRoutsNames", RoutName\_BOX))

{

AL.MessageErrorShow("При загрузке маршрутов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void AddArrival\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

string[] args = { RoutName\_BOX.Text, StopName\_BOX.Text, Train\_Number\_BOX.Text, ArrivalTime.Text, ArrivalDate.Text };

if (AL.TextChecking(args))

{

AL.MagicUniversalControlDataCatched("call DISPATCHER\_AddArrival", args, "AddArrival", \_connected);

DIITF.TryLoadingTables();

}

}

private void RoutName\_BOX\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (RoutName\_BOX.Text.Length < 30)

{

AL.InputWordsProtector(e);

}

else

e.Handled = true;

}

private void RoutName\_BOX\_PreviewKeyUp(object sender, KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RoutName\_BOX, e);

}

private void StopName\_BOX\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (StopName\_BOX.Text.Length < 30)

{

AL.InputWordsProtector(e);

}

else

e.Handled = true;

}

private void StopName\_BOX\_PreviewKeyUp(object sender, KeyEventArgs e)

{

AL.DontCtrlVAndSpace(StopName\_BOX, e);

}

private void Train\_Number\_BOX\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

}

private void Train\_Number\_BOX\_PreviewKeyUp(object sender, KeyEventArgs e)

{

AL.DontCtrlVAndSpace(Train\_Number\_BOX, e);

}

}

}

Dispatcher\_adddeparture.xaml

<UserControl x:Class="RZDKursovoy.Dispatcher\_AddDeparture"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<Grid>

<GroupBox Header="Добавление отправления поезда">

<Grid>

<ComboBox LostFocus="RoutName\_BOX\_LostFocus" PreviewTextInput="RoutName\_BOX\_PreviewTextInput" PreviewKeyUp="RoutName\_BOX\_PreviewKeyUp" Loaded="RoutName\_BOX\_Loaded" materialDesign:HintAssist.Hint="Название маршрута" x:Name="RoutName\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,200" Width="289">

</ComboBox>

<ComboBox materialDesign:HintAssist.Hint="Название остановки" PreviewTextInput="StopName\_BOX\_PreviewTextInput" PreviewKeyUp="StopName\_BOX\_PreviewKeyUp" x:Name="StopName\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,130" Width="289">

</ComboBox>

<ComboBox materialDesign:HintAssist.Hint="Номер поезда" PreviewTextInput="Train\_Number\_BOX\_PreviewTextInput" PreviewKeyUp="Train\_Number\_BOX\_PreviewKeyUp" x:Name="Train\_Number\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,60" Width="289">

</ComboBox>

<materialDesign:TimePicker Language="ru-RU" HorizontalAlignment="Center" Margin="0,20,0,0" materialDesign:HintAssist.Hint="Время" x:Name="DepartureTime" Width="289" VerticalAlignment="Center"></materialDesign:TimePicker>

<DatePicker x:Name="DepartureDate" Language="ru-RU" VerticalAlignment="Center" materialDesign:HintAssist.Hint="Дата" HorizontalAlignment="Center" Width="289" Margin="0,80,0,0" ></DatePicker>

<Button x:Name="AddDeparutre\_BUTTON" Click="AddDeparutre\_BUTTON\_Click" VerticalAlignment="Center" Width="289" Content="Добавить" HorizontalAlignment="Center" Margin="0,150,0,0"></Button>

</Grid>

</GroupBox>

</Grid>

</UserControl>

Dispatcher\_adddeparture.xaml.cs

using MySql.Data.MySqlClient;

using System;

using System.Windows;

using System.Windows.Controls;

namespace RZDKursovoy

{

public partial class Dispatcher\_AddDeparture : UserControl

{

private MySqlConnection \_connected;

public MySqlConnection SetConnection

{

set { \_connected = value; }

}

private Dispatcher\_Interface\_In\_To\_Face DIITF;

private ApplicationLogic AL = new ApplicationLogic();

public Dispatcher\_Interface\_In\_To\_Face SetInterface

{

set { DIITF = value; }

}

public Dispatcher\_AddDeparture()

{

InitializeComponent();

}

private void RoutName\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

RoutName\_BOX.Items.Clear();

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowRoutsNames", RoutName\_BOX))

{

AL.MessageErrorShow("При загрузке маршрутов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void RoutName\_BOX\_LostFocus(object sender, RoutedEventArgs e)

{

if (RoutName\_BOX.Text != "")

{

StopName\_BOX.Items.Clear();

Train\_Number\_BOX.Items.Clear();

string[] args = { RoutName\_BOX.Text };

try

{

var r = AL.CatchStringListResult(\_connected, "call \_DISPATCHER\_ThrowAvailableStopsFindByRout", args);

for (int i = 0; i < r.Count; i++)

{

StopName\_BOX.Items.Add(r[i]);

}

}

catch (Exception)

{

AL.MessageErrorShow("При загрузке остановок произошла ошибка", "Ошибка");

this.IsEnabled = false;

return;

}

try

{

var r2 = AL.CatchStringListResult(\_connected, "call \_DISPATCHER\_ThrowAvailableTrainsFindByRout", args);

for (int i = 0; i < r2.Count; i++)

{

Train\_Number\_BOX.Items.Add(r2[i]);

}

}

catch (Exception)

{

AL.MessageErrorShow("При загрузке поездов произошла ошибка", "Ошибка");

this.IsEnabled = false;

return;

}

}

}

private void AddDeparutre\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

string[] args = { RoutName\_BOX.Text, StopName\_BOX.Text, Train\_Number\_BOX.Text, DepartureTime.Text, DepartureDate.Text };

if (AL.TextChecking(args))

{

AL.MagicUniversalControlDataCatched("call DISPATCHER\_AddDeparture", args, "AddDeparture", \_connected);

DIITF.TryLoadingTables();

}

}

private void RoutName\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

if (RoutName\_BOX.Text.Length < 30)

{

AL.InputWordsProtector(e);

}

else

e.Handled = true;

}

private void RoutName\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RoutName\_BOX, e);

}

private void StopName\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

if (RoutName\_BOX.Text.Length < 30)

{

AL.InputWordsProtector(e);

}

else

e.Handled = true;

}

private void StopName\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(StopName\_BOX, e);

}

private void Train\_Number\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

}

private void Train\_Number\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(Train\_Number\_BOX, e);

}

}

}

Dispatcher\_AddForm.xaml

<Window x:Class="RZDKursovoy.Dispatcher\_AddForm"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

x:Name="BestWindow"

Title="Добавление" Height="350" Width="500" MinHeight="350" MinWidth="500">

<Grid>

</Grid>

</Window>

Dispatcher\_AddForm.xaml.cs

using MySql.Data.MySqlClient;

using System.Windows;

namespace RZDKursovoy

{

/// <summary>

/// Логика взаимодействия для Dispatcher\_AddForm.xaml

/// </summary>

public partial class Dispatcher\_AddForm : Window

{

private ApplicationLogic AL = new ApplicationLogic();

private MySqlConnection \_connected;

public MySqlConnection SetConnection

{

set { \_connected = value; }

}

private Dispatcher\_Interface\_In\_To\_Face DIITF;

public Dispatcher\_Interface\_In\_To\_Face SetInterface

{

set { DIITF = value; }

}

public void ShowControl(int ControlNumber)

{

switch(ControlNumber)

{

case 0:

Dispatcher\_AddTrain DAT = new Dispatcher\_AddTrain();

DAT.SetConnection = \_connected;

DAT.SetInterface = DIITF;

BestWindow.Content = DAT;

break;

case 1:

Dispatcher\_AddRailcar DAR = new Dispatcher\_AddRailcar();

DAR.SetConnection = \_connected;

DAR.SetInterface = DIITF;

BestWindow.Content = DAR;

break;

case 2:

Dispatcher\_AddRout \_DAR = new Dispatcher\_AddRout();

\_DAR.SetConnection = \_connected;

\_DAR.SetInterface = DIITF;

BestWindow.Content = \_DAR;

break;

case 3:

Dispatcher\_AddStop DAS = new Dispatcher\_AddStop();

DAS.SetConnection = \_connected;

DAS.SetInterface = DIITF;

BestWindow.Content = DAS;

break;

case 4:

Dispatcher\_AddArrival DAA = new Dispatcher\_AddArrival();

DAA.SetConnection = \_connected;

DAA.SetInterface = DIITF;

BestWindow.Content = DAA;

break;

case 5:

Dispatcher\_AddDeparture DAD = new Dispatcher\_AddDeparture();

DAD.SetConnection = \_connected;

DAD.SetInterface = DIITF;

BestWindow.Content = DAD;

break;

}

}

public Dispatcher\_AddForm()

{

InitializeComponent();

}

}

}

Dispatcher\_AddRailcar.xaml

<UserControl x:Class="RZDKursovoy.Dispatcher\_AddRailcar"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<Grid>

<GroupBox Header="Добавление вагона">

<Grid>

<ComboBox Loaded="TrainNumber\_BOX\_Loaded" LostFocus="TrainNumber\_BOX\_LostFocus" GotFocus="TrainNumber\_BOX\_GotFocus" PreviewTextInput="TrainNumber\_BOX\_PreviewTextInput" PreviewKeyUp="TrainNumber\_BOX\_PreviewKeyUp" materialDesign:HintAssist.Hint="Номер поезда" x:Name="TrainNumber\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,140" Width="289">

</ComboBox>

<ComboBox materialDesign:HintAssist.Hint="Номер вагона" PreviewTextInput="RailcarNumber\_BOX\_PreviewTextInput" PreviewKeyUp="RailcarNumber\_BOX\_PreviewKeyUp" x:Name="RailcarNumber\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,50" Width="289">

</ComboBox>

<ComboBox Loaded="RailcarType\_BOX\_Loaded" PreviewTextInput="RailcarType\_BOX\_PreviewTextInput" PreviewKeyUp="RailcarType\_BOX\_PreviewKeyUp" materialDesign:HintAssist.Hint="Тип вагона" x:Name="RailcarType\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,40,0,0" Width="289">

</ComboBox>

<Button x:Name="AddRailcar\_BUTTON" Click="AddRailcar\_BUTTON\_Click" Content="Добавить вагон" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0,130,0,0" Width="289"></Button>

</Grid>

</GroupBox>

</Grid>

</UserControl>

Dispatcher\_AddRailcar.xaml.cs

using MySql.Data.MySqlClient;

using System.Windows;

using System.Windows.Controls;

namespace RZDKursovoy

{

/// <summary>

/// Логика взаимодействия для Dispatcher\_AddRailcar.xaml

/// </summary>

public partial class Dispatcher\_AddRailcar : UserControl

{

private MySqlConnection \_connected;

public MySqlConnection SetConnection

{

set { \_connected = value; }

}

private Dispatcher\_Interface\_In\_To\_Face DIITF;

private ApplicationLogic AL = new ApplicationLogic();

public Dispatcher\_Interface\_In\_To\_Face SetInterface

{

set { DIITF = value; }

}

public Dispatcher\_AddRailcar()

{

InitializeComponent();

}

private void TrainNumber\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowTrainList", TrainNumber\_BOX))

{

AL.MessageErrorShow("При загрузке списка поездов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void RailcarType\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowRailcarTypesList", RailcarType\_BOX))

{

AL.MessageErrorShow("При загрузке типов вагонов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void AddRailcar\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

string[] args = { TrainNumber\_BOX.Text, RailcarNumber\_BOX.Text, RailcarType\_BOX.Text };

if (AL.TextChecking(args))

{

AL.MagicUniversalControlDataCatched("call DISPATCHER\_AddRailcar", args, "AddRailcar", \_connected);

DIITF.TryLoadingTables();

}

}

private void TrainNumber\_BOX\_LostFocus(object sender, RoutedEventArgs e)

{

if (TrainNumber\_BOX.Text != "")

{

RailcarNumber\_BOX.Items.Clear();

string[] args = { TrainNumber\_BOX.Text };

var r = AL.CatchStringListResult(\_connected, "call \_DISPATCHER\_ThrowAvailableRailcarList", args);

for (int i = 0; i < r.Count; i++)

{

RailcarNumber\_BOX.Items.Add(r[i]);

}

}

}

private void TrainNumber\_BOX\_GotFocus(object sender, RoutedEventArgs e)

{

RailcarNumber\_BOX.Text = "";

}

private void TrainNumber\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

}

private void TrainNumber\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(TrainNumber\_BOX, e);

}

private void RailcarNumber\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

}

private void RailcarNumber\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RailcarNumber\_BOX, e);

}

private void RailcarType\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

}

private void RailcarType\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RailcarType\_BOX, e);

}

}

}

Dispatcher\_AddRout.xaml

<UserControl x:Class="RZDKursovoy.Dispatcher\_AddRout"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<Grid>

<GroupBox Header="Добавление маршрута">

<Grid>

<TextBox PreviewTextInput="RoutName\_BOX\_PreviewTextInput" PreviewKeyUp="RoutName\_BOX\_PreviewKeyUp" x:Name="RoutName\_BOX" MaxLength="30" Style="{StaticResource MaterialDesignFloatingHintTextBox}" VerticalAlignment="Center" Height="36" Width="282" HorizontalAlignment="Center" Margin="0,0,0,120">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Название маршрута

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<ComboBox Loaded="TrainRout\_BOX\_Loaded" materialDesign:HintAssist.Hint="Количество остановок" x:Name="TrainRout\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,30" Width="289">

</ComboBox>

<Button x:Name="AddRout\_BUTTON" Click="AddRout\_BUTTON\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0,90,0,0" Width="289" Content="Добавить"></Button>

</Grid>

</GroupBox>

</Grid>

</UserControl>

Dispatcher\_AddRout.xaml.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace RZDKursovoy

{

/// <summary>

/// Логика взаимодействия для Dispatcher\_AddRout.xaml

/// </summary>

public partial class Dispatcher\_AddRout : UserControl

{

private MySqlConnection \_connected;

public MySqlConnection SetConnection

{

set { \_connected = value; }

}

private ApplicationLogic AL = new ApplicationLogic();

private Dispatcher\_Interface\_In\_To\_Face DIITF;

public Dispatcher\_Interface\_In\_To\_Face SetInterface

{

set { DIITF = value; }

}

public Dispatcher\_AddRout()

{

InitializeComponent();

}

private void TrainRout\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowMaxStopCount", TrainRout\_BOX))

{

AL.MessageErrorShow("При загрузке количества остановок произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void AddRout\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

string[] args = { TrainRout\_BOX.Text, RoutName\_BOX.Text};

if (AL.TextChecking(args))

{

AL.MagicUniversalControlDataCatched("SELECT DISPATCHER\_AddRout", args, "AddRout", \_connected);

DIITF.TryLoadingTables();

}

}

private void RoutName\_BOX\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

AL.Input\_RoutNameProtector(RoutName\_BOX, e);

}

private void RoutName\_BOX\_PreviewKeyUp(object sender, KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RoutName\_BOX, e);

}

}

}

Dispatcher\_AddStop.xaml

<UserControl x:Class="RZDKursovoy.Dispatcher\_AddStop"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

d:DesignHeight="450" d:DesignWidth="800">

<Grid>

<GroupBox Header="Добавление остановки">

<Grid>

<ComboBox LostFocus="RoutName\_BOX\_LostFocus" Loaded="RoutName\_BOX\_Loaded" materialDesign:HintAssist.Hint="Название маршрута" x:Name="RoutName\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,150" Width="289">

</ComboBox>

<ComboBox materialDesign:HintAssist.Hint="Номер остановки" x:Name="StopID"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,70" Width="289">

</ComboBox>

<TextBox PreviewTextInput="StopName\_PreviewTextInput" PreviewKeyUp="StopName\_PreviewKeyUp" x:Name="StopName" MaxLength="20" Style="{StaticResource MaterialDesignFloatingHintTextBox}" VerticalAlignment="Center" Height="36" Width="282" HorizontalAlignment="Center" Margin="0,10,0,0">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Название остановки

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<TextBox PreviewTextInput="Train\_Station\_Name\_PreviewTextInput" PreviewKeyUp="Train\_Station\_Name\_PreviewKeyUp" x:Name="Train\_Station\_Name" MaxLength="20" Style="{StaticResource MaterialDesignFloatingHintTextBox}" VerticalAlignment="Center" Height="36" Width="282" HorizontalAlignment="Center" Margin="0,80,0,0">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Название вокзала

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<Button x:Name="AddStop\_BUTTON" Click="AddStop\_BUTTON\_Click" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="0,160,0,0" Width="289" Content="Добавить"></Button>

</Grid>

</GroupBox>

</Grid>

</UserControl>

Dispatcher\_AddStop.xaml.cs

using MySql.Data.MySqlClient;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

namespace RZDKursovoy

{

public partial class Dispatcher\_AddStop : UserControl

{

private MySqlConnection \_connected;

public MySqlConnection SetConnection

{

set { \_connected = value; }

}

private Dispatcher\_Interface\_In\_To\_Face DIITF;

private ApplicationLogic AL = new ApplicationLogic();

public Dispatcher\_Interface\_In\_To\_Face SetInterface

{

set { DIITF = value; }

}

public Dispatcher\_AddStop()

{

InitializeComponent();

}

private void RoutName\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowRoutsNames", RoutName\_BOX))

{

AL.MessageErrorShow("При загрузке маршрутов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void RoutName\_BOX\_LostFocus(object sender, RoutedEventArgs e)

{

if (RoutName\_BOX.Text != "")

{

StopID.Items.Clear();

string[] args = { RoutName\_BOX.Text };

var r = AL.CatchStringListResult(\_connected, "call \_DISPATCHER\_ThrowStopNumber", args);

for (int i = 0; i < r.Count; i++)

{

StopID.Items.Add(r[i]);

}

}

}

private void AddStop\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

string[] args = { RoutName\_BOX.Text, StopID.Text, StopName.Text, Train\_Station\_Name.Text };

string[] RequredArgs = { RoutName\_BOX.Text, StopID.Text, StopName.Text };

if (AL.TextChecking(RequredArgs))

{

AL.MagicUniversalControlDataCatched("call DISPATCHER\_AddStop", args, "AddStop", \_connected);

DIITF.TryLoadingTables();

}

}

private void StopName\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

AL.InputWordsProtector(e);

}

private void StopName\_PreviewKeyUp(object sender, KeyEventArgs e)

{

AL.DontCtrlVAndSpace(StopName, e);

}

}

}

Dispatcher\_AddTrain.xaml

<UserControl x:Class="RZDKursovoy.Dispatcher\_AddTrain"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:RZDKursovoy"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<Grid>

<GroupBox Header="Добавление поезда">

<Grid x:Name="TrainGrid">

<TextBox x:Name="TrainNumber\_BOX" MaxLength="5" PreviewKeyUp="TrainNumber\_BOX\_PreviewKeyUp" Style="{StaticResource MaterialDesignFloatingHintTextBox}" VerticalAlignment="Center" Height="36" Width="282" HorizontalAlignment="Center" Margin="0,0,0,190">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Номер поезда

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<ComboBox Loaded="RailcarCount\_BOX\_Loaded" materialDesign:HintAssist.Hint="Количество вагонов" x:Name="RailcarCount\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,100" Width="289">

</ComboBox>

<ComboBox Loaded="TrainType\_BOX\_Loaded" PreviewTextInput="TrainType\_BOX\_PreviewTextInput" PreviewKeyUp="TrainType\_BOX\_PreviewKeyUp" materialDesign:HintAssist.Hint="Тип поезда" x:Name="TrainType\_BOX"

IsEditable="True"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,10" Width="289">

<ComboBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</ComboBox.ContextMenu>

</ComboBox>

<ComboBox Loaded="TrainRout\_BOX\_Loaded" materialDesign:HintAssist.Hint="Маршрут" x:Name="TrainRout\_BOX"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,80,0,0" Width="289">

</ComboBox>

<Button x:Name="AddTrain\_BUTTON" Click="AddTrain\_BUTTON\_Click" VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0,170,0,0" Width="289" Content="Добавить"></Button>

</Grid>

</GroupBox>

</Grid>

</UserControl>

Dispatcher\_AddTrain.xaml.cs

using MySql.Data.MySqlClient;

using System.Windows;

using System.Windows.Controls;

namespace RZDKursovoy

{

/// <summary>

/// Логика взаимодействия для Dispatcher\_AddTrain.xaml

/// </summary>

public partial class Dispatcher\_AddTrain : UserControl

{

private MySqlConnection \_connected;

private ApplicationLogic AL = new ApplicationLogic();

public MySqlConnection SetConnection

{

set { \_connected = value; }

}

private Dispatcher\_Interface\_In\_To\_Face DIITF;

public Dispatcher\_Interface\_In\_To\_Face SetInterface

{

set { DIITF = value; }

}

public Dispatcher\_AddTrain()

{

InitializeComponent();

}

private void RailcarCount\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowRailcarListCount", RailcarCount\_BOX))

{

AL.MessageErrorShow("При загрузке количества вагонов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void AddTrain\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

string[] args = { TrainNumber\_BOX.Text, RailcarCount\_BOX.Text, TrainType\_BOX.Text, TrainRout\_BOX.Text };

if (AL.TextChecking(args))

{

AL.MagicUniversalControlDataCatched("call DISPATCHER\_AddTrain", args, "AddTrain", \_connected);

DIITF.TryLoadingTables();

}

}

private void TrainType\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowTrainTypes", TrainType\_BOX))

{

AL.MessageErrorShow("При загрузке типов вагонов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void TrainRout\_BOX\_Loaded(object sender, RoutedEventArgs e)

{

if (!AL.ComboboxFiling(\_connected, "call \_DISPATCHER\_ThrowRoutsNames", TrainRout\_BOX))

{

AL.MessageErrorShow("При загрузке маршрутов произошла ошибка", "Ошибка");

this.IsEnabled = false;

}

}

private void TrainNumber\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(TrainNumber\_BOX, e);

}

private void TrainType\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

if (TrainType\_BOX.Text.Length < 30)

{

AL.InputWordsProtector(e);

}

else

e.Handled = true;

}

private void TrainType\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(TrainType\_BOX, e);

}

}

}

Dispatcher\_Interface\_In\_To\_Face.xaml

<UserControl x:Class="RZDKursovoy.Dispatcher\_Interface\_In\_To\_Face"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:RZDKursovoy"

xmlns:dragablz="http://dragablz.net/winfx/xaml/dragablz"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800" Background="White">

<Grid x:Name="ExecutableGrid" Loaded="TryLoading">

<dragablz:TabablzControl x:Name="ControlMenu">

<TabItem Header="Поезда">

<Grid>

<DataGrid CanUserAddRows="False" CanUserDeleteRows="False" CanUserReorderColumns="False" x:Name="ShowTrains" KeyUp="ShowTrains\_KeyUp" CellEditEnding="ShowTrains\_CellEditEnding" Margin="0,0,0,30"></DataGrid>

<Button x:Name="OpenAddTrainMenu" Click="OpenAddTrainMenu\_Click" Height="29" Content="Добавить поезд" Margin="2" VerticalAlignment="Bottom" />

</Grid>

</TabItem>

<TabItem Header="Вагоны">

<Grid>

<DataGrid CanUserAddRows="False" IsReadOnly="True" CanUserDeleteRows="False" CanUserReorderColumns="False" CellEditEnding="ShowRailcars\_CellEditEnding" KeyUp="ShowRailcars\_KeyUp" x:Name="ShowRailcars" Margin="0,0,0,30"></DataGrid>

<Button x:Name="OpenAddRailcarMenu" Click="OpenAddRailcarMenu\_Click" Height="29" Content="Добавить вагон" Margin="2" VerticalAlignment="Bottom" />

</Grid>

</TabItem>

<TabItem Header="Маршруты">

<Grid>

<DataGrid CanUserAddRows="False" CanUserDeleteRows="False" CanUserReorderColumns="False" CellEditEnding="ShowRoutes\_CellEditEnding" KeyUp="ShowRoutes\_KeyUp" x:Name="ShowRoutes" Margin="0,0,0,30"></DataGrid>

<Button x:Name="OpenAddRoutMenu" Click="OpenAddRoutMenu\_Click" Height="29" Content="Добавить маршрут" Margin="2" VerticalAlignment="Bottom" />

</Grid>

</TabItem>

<TabItem Header="Остановки">

<Grid>

<DataGrid CanUserAddRows="False" CanUserDeleteRows="False" CanUserReorderColumns="False" CellEditEnding="ShowStops\_CellEditEnding" KeyUp="ShowStops\_KeyUp" x:Name="ShowStops" Margin="0,0,0,30"></DataGrid>

<Button x:Name="OpenAddStopsMenu" Click="OpenAddStopsMenu\_Click" Height="29" Content="Добавить остановку поезда" Margin="2" VerticalAlignment="Bottom" />

</Grid>

</TabItem>

<TabItem Header="Прибытия">

<Grid>

<DataGrid CanUserAddRows="False" CanUserDeleteRows="False" CanUserReorderColumns="False" CellEditEnding="ShowArrivals\_CellEditEnding" KeyUp="ShowArrivals\_KeyUp" x:Name="ShowArrivals" Margin="0,0,0,30"></DataGrid>

<Button x:Name="OpenAddArrivalMenu" Click="OpenAddArrivalMenu\_Click" Height="29" Content="Добавить прибытие поезда" Margin="2" VerticalAlignment="Bottom" />

</Grid>

</TabItem>

<TabItem Header="Отправления">

<Grid>

<DataGrid CanUserAddRows="False" CanUserDeleteRows="False" CanUserReorderColumns="False" CellEditEnding="ShowDepartures\_CellEditEnding" KeyUp="ShowDepartures\_KeyUp" x:Name="ShowDepartures" Margin="0,0,0,30"></DataGrid>

<Button x:Name="OpenAddDepartureMenu" Click="OpenAddDepartureMenu\_Click" Height="29" Content="Добавить отправление поезда" Margin="2" VerticalAlignment="Bottom" />

</Grid>

</TabItem>

</dragablz:TabablzControl>

</Grid>

</UserControl>

Dispatcher\_Interface\_In\_To\_Face.xaml.cs

using MySql.Data.MySqlClient;

using System;

using System.Data;

using System.Security;

using System.Windows.Controls;

namespace RZDKursovoy

{

public partial class Dispatcher\_Interface\_In\_To\_Face : UserControl

{

public MySqlConnection Connected { get; private set; }

public MySqlConnection SetConnected { set { Connected = value; } }

private string Login = "";

public string SetLogin { set { Login = value; } }

private ApplicationLogic AL = new ApplicationLogic();

private DataRowView TMPGridRow = null;

private bool ConvertCheck = true;

Dispatcher\_AddForm DAF = null;

public Dispatcher\_Interface\_In\_To\_Face()

{

InitializeComponent();

}

/\*Key Up Events\*/

private void ShowTrains\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

int[] activity = { 0, 0 };

var t = (DataRowView) ShowTrains.CurrentItem;

string[] args = { t[0].ToString() };

if (AL.KeyUpInside(Connected, sender, e, ShowTrains, TMPGridRow, ConvertCheck, "call DISPATCHER\_DropTrain", "call DISPATCHER\_UpdateTrain", "DeleteTrain", "UpdateTrain",

"При редактировании произошла ошибка. Редактировать номер поезда запрещено", args, activity))

TryLoadingTables();

}

private void ShowRailcars\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

int[] activity = { 0, 1 };

var t = (DataRowView)ShowRailcars.CurrentItem;

string[] args = { t[0].ToString(), t[1].ToString() };

if (AL.KeyUpInside(Connected, sender, e, ShowRailcars, TMPGridRow, ConvertCheck, "call DISPATCHER\_DropRailcar", "-1", "DeleteRailcar", "DontWork",

"При редактировании произошла ошибка. Редактировать номер поезда запрещено", args, activity))

TryLoadingTables();

}

private void ShowRoutes\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

int[] activity = { 0, 0 };

var t = (DataRowView)ShowRoutes.CurrentItem;

string[] args = { t[0].ToString() };

if (AL.KeyUpInside(Connected, sender, e, ShowRoutes, TMPGridRow, ConvertCheck, "call DISPATCHER\_DropRout", "call DISPATCHER\_UpdateRout", "DeleteRout", "UpdateRout",

"При редактировании произошла ошибка. Редактировать идентификатор маршрута запрещено", args, activity))

TryLoadingTables();

}

private void ShowStops\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

int[] activity = { 0, 0 };

var t = (DataRowView)ShowStops.CurrentItem;

string[] args = { t[0].ToString(), t[1].ToString() };

if (AL.KeyUpInside(Connected, sender, e, ShowStops, TMPGridRow, ConvertCheck, "call DISPATCHER\_DropStop", "call DISPATCHER\_UpdateStop", "DeleteStop", "UpdateStop",

"При редактировании произошла ошибка. Редактировать идентификатор маршрута запрещено", args, activity))

TryLoadingTables();

}

private void ShowArrivals\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

int[] activity = { 0, 0 };

var t = (DataRowView)ShowArrivals.CurrentItem;

string[] args = { t[1].ToString(), t[0].ToString(), t[2].ToString() };

if (AL.KeyUpInside(Connected, sender, e, ShowArrivals, TMPGridRow, ConvertCheck, "call DISPATCHER\_DropArrival", "call DISPATCHER\_UpdateArrival", "DeleteArrival", "UpdateArrival",

"При редактировании произошла неизвестная ошибка", args, activity))

TryLoadingTables();

}

private void ShowDepartures\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

int[] activity = { 0, 0 };

var t = (DataRowView)ShowDepartures.CurrentItem;

string[] args = { t[1].ToString(), t[0].ToString(), t[2].ToString() };

if (AL.KeyUpInside(Connected, sender, e, ShowDepartures, TMPGridRow, ConvertCheck, "call DISPATCHER\_DropDeparture", "call DISPATCHER\_UpdateDeparture", "DeleteDeparture", "UpdateDeparture",

"При редактировании произошла неизвестная ошибка", args, activity))

TryLoadingTables();

}

/\*Cell Edit Ending Events\*/

private void ShowTrains\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)

{

ConvertCheck = true;

int[] arr = { 1 };

ConvertCheck = AL.ConvertCheck(sender, e, arr);

int[] itemarr = { 0 };

TMPGridRow = AL.BlockUpdate(sender, e, itemarr);

}

private void ShowRailcars\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)

{

ConvertCheck = true;

int[] arr = { 1 };

ConvertCheck = AL.ConvertCheck(sender, e, arr);

int[] itemarr = { 0 };

TMPGridRow = AL.BlockUpdate(sender, e, itemarr);

}

private void ShowRoutes\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)

{

ConvertCheck = true;

int[] arr = { 1 };

ConvertCheck = AL.ConvertCheck(sender, e, arr);

int[] itemarr = { 0 };

TMPGridRow = AL.BlockUpdate(sender, e, itemarr);

}

private void ShowStops\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)

{

ConvertCheck = true;

int[] arr = { 1 };

ConvertCheck = AL.ConvertCheck(sender, e, arr);

int[] itemarr = { -1 };

TMPGridRow = AL.BlockUpdate(sender, e, itemarr);

}

private void ShowArrivals\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)

{

ConvertCheck = true;

int[] arr = { -1 };

ConvertCheck = AL.ConvertCheck(sender, e, arr);

int[] itemarr = { -1 };

TMPGridRow = AL.BlockUpdate(sender, e, itemarr);

}

private void ShowDepartures\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)

{

ConvertCheck = true;

int[] arr = { -1 };

ConvertCheck = AL.ConvertCheck(sender, e, arr);

int[] itemarr = { -1 };

TMPGridRow = AL.BlockUpdate(sender, e, itemarr);

}

/\*Click Button Events\*/

private void OpenAddTrainMenu\_Click(object sender, System.Windows.RoutedEventArgs e)

{

SetControl(0);

}

private void OpenAddRailcarMenu\_Click(object sender, System.Windows.RoutedEventArgs e)

{

SetControl(1);

}

private void OpenAddRoutMenu\_Click(object sender, System.Windows.RoutedEventArgs e)

{

SetControl(2);

}

private void OpenAddStopsMenu\_Click(object sender, System.Windows.RoutedEventArgs e)

{

SetControl(3);

}

private void OpenAddArrivalMenu\_Click(object sender, System.Windows.RoutedEventArgs e)

{

SetControl(4);

}

private void OpenAddDepartureMenu\_Click(object sender, System.Windows.RoutedEventArgs e)

{

SetControl(5);

}

private void SetControl(int ControlID)

{

if (DAF != null)

{

if (!DAF.IsVisible)

DAF = null;

}

if (DAF == null)

{

DAF = new Dispatcher\_AddForm();

DAF.SetInterface = this;

DAF.SetConnection = Connected;

DAF.ShowControl(ControlID);

DAF.Show();

}

else

{

DAF.SetConnection = Connected;

DAF.SetInterface = this;

DAF.ShowControl(ControlID);

}

}

/\*Загрузка данных из таблиц\*/

public void TryLoadingTables()

{

string[] argsTrains = { "Номер поезда", "Кол-во вагонов", "Тип поезда", "Название маршрута" };

string[] argsRailcars = { "Номер поезда", "Номер вагона", "Тип вагона" };

string[] argsRoutes = { "Идентификатор маршрута", "Количество остановок", "Название маршрута" };

string[] argsStops = { "Название маршрута", "Номер остановки", "Название остановки", "Название вокзала" };

string[] argsArrivals = { "Название остановки", "Название маршрута", "Номер поезда", "Время прибытия", "Дата прибытия" };

string[] argsDepartures = { "Название остановки", "Название маршрута", "Номер поезда", "Время отправления", "Дата отправления" };

try

{

if (!AL.FillTable(Connected, ShowTrains, "call DISPATCHER\_ShowTrains", argsTrains))

throw new ApplicationException();

if (!AL.FillTable(Connected, ShowRailcars, "call DISPATCHER\_ShowRailcars", argsRailcars))

throw new ApplicationException();

if (!AL.FillTable(Connected, ShowRoutes, "call DISPATCHER\_ShowRoutes", argsRoutes))

throw new ApplicationException();

if (!AL.FillTable(Connected, ShowStops, "call DISPATCHER\_ShowStops", argsStops))

throw new ApplicationException();

if (!AL.FillTable(Connected, ShowArrivals, "call DISPATCHER\_ShowArrivals", argsArrivals))

throw new ApplicationException();

if (!AL.FillTable(Connected, ShowDepartures, "call DISPATCHER\_ShowDepartures", argsDepartures))

throw new ApplicationException();

}

catch (ApplicationException)

{

ExecutableGrid.IsEnabled = false;

AL.MessageErrorShow("Соединение сервером было внезапно разорвано", "Ошибка");

}

}

private void TryLoading(object sender, System.Windows.RoutedEventArgs e)

{

TryLoadingTables();

}

}

}

Menu.xaml

<UserControl x:Class="RZDKursovoy.Menu"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:RZDKursovoy"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:dragablz="http://dragablz.net/winfx/xaml/dragablz"

mc:Ignorable="d"

d:DesignHeight="450" MinHeight="450" MinWidth="800" d:DesignWidth="800">

<Grid Background="White" x:Name="PerfectReflectionGRID">

<Grid x:Name="Reflector">

<dragablz:TabablzControl x:Name="ControlMenu" Loaded="ControlMenu\_Loaded">

<TabItem x:Name="Reservation\_TAB"

Header="Резервирование"

Margin="0" d:IsHidden="True">

<Grid x:Name="ChooseRout" d:IsHidden="True">

<ComboBox PreviewTextInput="Arrival\_BOX\_PreviewTextInput" PreviewKeyUp="Arrival\_BOX\_PreviewKeyUp" x:Name="Arrival\_BOX" materialDesign:HintAssist.Hint="Точка отправления"

IsEditable="True"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,140" Width="210">

<ComboBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</ComboBox.ContextMenu>

</ComboBox>

<ComboBox PreviewTextInput="Departure\_BOX\_PreviewTextInput" PreviewKeyUp="Departure\_BOX\_PreviewKeyUp" x:Name="Departure\_BOX" materialDesign:HintAssist.Hint="Точка назначения"

IsEditable="True"

Style="{StaticResource MaterialDesignFloatingHintComboBox}" HorizontalAlignment="Center" Margin="0,0,0,60" Width="210">

<ComboBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</ComboBox.ContextMenu>

</ComboBox>

<Button x:Name="FindTrain\_BUTTON" Content="Найти" Margin="0,100,0,0" Height="Auto" Width="240" VerticalAlignment="Center" HorizontalAlignment="Center" Click="FindTrain\_BUTTON\_Click"/>

<DatePicker x:Name="Arrival\_Date" HorizontalAlignment="Center" Width="210" materialDesign:HintAssist.Hint="Выберите дату"

Style="{StaticResource MaterialDesignFloatingHintDatePicker}" Language="Ru-ru" VerticalAlignment="Center" Margin="0,30,0,0" AutomationProperties.IsOffscreenBehavior="Onscreen" />

</Grid>

</TabItem>

<TabItem x:Name="LK\_Tab" Header="Личный кабинет">

<Grid>

<Grid x:Name="LK\_Empty" d:IsHidden="True">

<Label Content="У вас пока что нет приобретенных билетов" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="20" FontFamily="Segoe UI Semilight"></Label>

</Grid>

<Grid x:Name="LK\_NotEmpty">

<DataGrid SelectionChanged="ShowBuyedTickets\_SelectionChanged" IsReadOnly="True" x:Name="ShowBuyedTickets" Margin="0,0,0,30"></DataGrid>

<Grid VerticalAlignment="Bottom" Height="34">

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Button Grid.Column="0" x:Name="CancelTripBUTTON" Click="CancelTripBUTTON\_Click" Content="Отменить поездку" VerticalContentAlignment="Center" Height="34"/>

<Button Grid.Column="1" x:Name="PrintTicketBUTTON" Click="PrintTicketBUTTON\_Click" IsDefault="True" Content="Сохранить билет" Height="34"/>

</Grid>

</Grid>

</Grid>

</TabItem>

</dragablz:TabablzControl>

</Grid>

</Grid>

</UserControl>

Menu.xaml.cs

using MySql.Data.MySqlClient;

using PdfSharp.Pdf;

using System.Collections.Generic;

using System.Data;

using System.Windows;

using System.Windows.Controls;

using TheArtOfDev.HtmlRenderer.PdfSharp;

namespace RZDKursovoy

{

/// <summary>

/// Логика взаимодействия для Menu.xaml

/// </summary>

public partial class Menu : UserControl

{

public MySqlConnection Connected { get; private set; }

public MySqlConnection SetConnected { set { Connected = value; } }

private ApplicationLogic AL = new ApplicationLogic();

private string Login = "";

public string SetLogin { set { Login = value; } }

private int AvailableTicketsCount = new int();

private int CurrentTicketID = new int();

private string \_CurrentTrainNumber = "";

private int \_Current\_Railcar\_Number = new int();

private int \_Current\_Place\_Number = new int();

private string \_Current\_Arrival\_Time = "";

private string \_Current\_Arrival\_Date = "";

private string \_Current\_Arrival\_Stop\_Name = "";

private string \_Current\_Departure\_Time = "";

private string \_Current\_Departure\_Date = "";

private string \_Current\_Departure\_Stop\_Name = "";

public Menu()

{

InitializeComponent();

}

public void CheckActivateCabinet()

{

string[] args = { Login + "@localhost" };

AvailableTicketsCount = AL.CatchIntResult(Connected, "call throwCountAvailableTickets", args);

if (AvailableTicketsCount == 0)

{

LK\_Empty.Visibility = Visibility.Visible;

LK\_NotEmpty.Visibility = Visibility.Collapsed;

}

else

{

LK\_Empty.Visibility = Visibility.Collapsed;

LK\_NotEmpty.Visibility = Visibility.Visible;

MySqlDataAdapter ad = new MySqlDataAdapter();

string Query = "call throwAvailableTicketsWithInfo('" + Login + "@localhost')";

ad.SelectCommand = new MySqlCommand(Query, Connected);

System.Data.DataTable table = new System.Data.DataTable();

ad.Fill(table);

table.Columns[0].ColumnName = "Номер билета";

table.Columns[1].ColumnName = "Номер поезда";

table.Columns[2].ColumnName = "Номер вагона";

table.Columns[3].ColumnName = "Номер места";

table.Columns[4].ColumnName = "Время отправления";

table.Columns[5].ColumnName = "Дата отправления";

table.Columns[6].ColumnName = "Станция отправления";

table.Columns[7].ColumnName = "Время прибытия";

table.Columns[8].ColumnName = "Дата прибытия";

table.Columns[9].ColumnName = "Станция прибытия";

ShowBuyedTickets.ItemsSource = table.DefaultView;

}

}

public void BoxesFiling()

{

Arrival\_BOX.Items.Clear();

Departure\_BOX.Items.Clear();

string[] param = { "null" };

var a = AL.CatchStringListResult(Connected, "call ThrowAllStations", param); // AL.TrowAllStations(Connected);

for (int i = 0; i < a.Count; i++)

{

Arrival\_BOX.Items.Add(a[i]);

Departure\_BOX.Items.Add(a[i]);

}

CheckActivateCabinet();

}

private void FindTrain\_BUTTON\_Click(object sender, RoutedEventArgs e)

{

if ((Arrival\_BOX.Text != "") && (Departure\_BOX.Text != "") && (Departure\_BOX.Text != ""))

{

try

{

if (Arrival\_BOX.Text != Departure\_BOX.Text)

{

ApplicationLogic AL = new ApplicationLogic();

string[] Data = { Arrival\_BOX.Text, Departure\_BOX.Text, Arrival\_Date.Text };

List<string> Routs = AL.CatchStringListResult(Connected, "call FindRout", Data);

if (Routs.Count > 0)

{

List<string> TrainsList = new List<string>();

for (int i = 0; i < Routs.Count; i++)

{

string[] args = { Routs[i], Arrival\_BOX.Text, Arrival\_Date.Text, Departure\_BOX.Text };

var TMPInputer = AL.CatchStringListResult(Connected, "call newFindTrainList", args);

for (int j = 0; j < TMPInputer.Count; j++)

if (TMPInputer[j] != "-1")

TrainsList.Add(TMPInputer[j]);

}

if (TrainsList.Count > 0)

{

ReservationControl RCN = new ReservationControl();

RCN = new ReservationControl();

RCN.SetConnection = Connected;

RCN.SetArrival = Arrival\_BOX.Text;

RCN.SetDeparture = Departure\_BOX.Text;

RCN.SetDate = Arrival\_Date.Text;

RCN.SetRouts = Routs;

RCN.SetTrainsList = TrainsList;

RCN.SetMenu = this;

PerfectReflectionGRID.Children.Add(RCN);

Reflector.IsEnabled = false;

}

else

{

throw new System.Exception();

}

}

else

{

throw new System.Exception();

}

}

else

{

throw new System.Exception();

}

}

catch (System.Exception)

{

AL.MessageShow("К сожалению, поездов по нужному Вам маршруту в данное время нет. Попробуйте выбрать другой день", "=(");

}

}

else

{

AL.MessageErrorShow("Вы не заполнили данные", "Ошибка");

}

}

private void ControlMenu\_Loaded(object sender, RoutedEventArgs e)

{

}

private void Arrival\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

if (Arrival\_BOX.Text.Length < 30)

{

AL.InputWordsProtector(e);

}

else

e.Handled = true;

}

private void Departure\_BOX\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

if (Departure\_BOX.Text.Length < 30)

{

AL.InputWordsProtector(e);

}

else

e.Handled = true;

}

private string throwTicketCode(string[] args)

{

string result =

"<table style='height: auto; width: auto;' border='1'>" +

"<tbody>" +

"<tr style = 'height: auto;'> " +

"<td style = 'width: auto; height: auto; text-align: center;'> (c)2018.AVB Inc.</td>" +

"<td style = 'width: auto; height: auto;'>" +

"<p style = 'text-align: center;' >Номер электронного билета:</p>" +

"<p style = 'text-align: center;'>&nbsp;<strong>" + args[0] + "</strong></p>" +

"</td>" +

"</tr>" +

"<tr style = 'height: auto;'>" +

"<td style = 'width: auto; height: auto; text-align: right;'> Маршрут следования: " + args[1] + "-&gt; &nbsp;</td>" +

"<td style = 'width: auto; height: auto;'>" + args[2] + "</td>" +

"</tr>" +

"<tr style = 'height: auto;'>" +

"<td style = 'width: auto; height: auto; text-align: center;' > Отправление: " + args[3] + " " + args[4] + "</td>" +

"<td style = 'width: auto; height: auto; text-align: center;' > Прибытие: " + args[5] + " " + args[6] + "</td>" +

"</tr>" +

"<tr style = 'height: auto;'>" +

"<td style = 'width: auto; height: auto; text-align: center;'>" +

"<p> Пассажир - " + args[7] + " " + args[8] + " " + args[9] + " " + "&nbsp;</p>" +

"<p> Паспортные данные - " + args[10] + " " + args[11] + "</p>" +

"</td>" +

"<td style = 'width: auto; height: auto; text-align: center;'>" +

"<p> Номер поезда: " + args[12] + "&nbsp;</p>" +

"<p> Номер вагона: " + args[13] + "&nbsp;</p>" +

"<p> Тип вагона: " + args[14] + "</p>" +

"<p> Номер места: " + args[15] +

"&nbsp;</p>" +

"<p> Цена за билет: " + args[16] + "</p>" +

"</td>" +

"</tr>" +

"</tbody>" +

"</table>";

return result;

}

private void PrintTicketBUTTON\_Click(object sender, RoutedEventArgs e)

{

if (CurrentTicketID != 0)

{

var RailcarInfo = AL.throwRailcarInfo(Connected, \_CurrentTrainNumber, \_Current\_Railcar\_Number);

var PassPrivateInfo = AL.throwPassengerInfo(Connected, CurrentTicketID);

var t = new HtmlAgilityPack.HtmlDocument();

string[] data = { CurrentTicketID.ToString(), \_Current\_Arrival\_Stop\_Name, \_Current\_Departure\_Stop\_Name, \_Current\_Arrival\_Date, \_Current\_Arrival\_Time, \_Current\_Departure\_Date, \_Current\_Departure\_Time,

PassPrivateInfo[0], PassPrivateInfo[1], PassPrivateInfo[2], PassPrivateInfo[3], PassPrivateInfo[4], \_CurrentTrainNumber.ToString(), \_Current\_Railcar\_Number.ToString(), RailcarInfo[0], \_Current\_Place\_Number.ToString(), RailcarInfo[1] };

Microsoft.Win32.SaveFileDialog SFD = new Microsoft.Win32.SaveFileDialog();

SFD.Filter = "PDF файл (\*.pdf)|\*.pdf";

if (SFD.ShowDialog() == true)

{

PdfDocument pdf = new PdfDocument();

PdfGenerateConfig PGC = new PdfGenerateConfig();

PGC.PageSize = PdfSharp.PageSize.A4;

PGC.PageOrientation = PdfSharp.PageOrientation.Landscape;

pdf = PdfGenerator.GeneratePdf(throwTicketCode(data), PGC);

try

{

pdf.Save(SFD.FileName);

}

catch (System.Exception)

{ AL.MessageErrorShow("При сохранении билета произошла ошибка", "Ошибка"); }

}

}

else

AL.MessageShow("Вы должны выбрать билет, необходимый для сохранения", "Предупреждение");

}

private void CancelTripBUTTON\_Click(object sender, RoutedEventArgs e)

{

if (CurrentTicketID != 0)

{

try

{

string[] args = { CurrentTicketID.ToString() };

var res = AL.CatchStringListResult(Connected, "call CancelTicket", args);

AL.MessageShow(res[0], "Отменено");

CheckActivateCabinet();

}

catch (System.Exception)

{

AL.MessageErrorShow("Во время отмены произошла ошибка", "Ошибка");

return;

}

}

else

{

AL.MessageShow("Для начала вы должны выбрать билет, который хотите отменить", "Ошибка");

return;

}

}

private void ShowBuyedTickets\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

try

{

var t = (DataRowView)ShowBuyedTickets.CurrentItem;

CurrentTicketID = System.Convert.ToInt32(t[0].ToString());

\_CurrentTrainNumber = t[1].ToString();

\_Current\_Railcar\_Number = System.Convert.ToInt32(t[2].ToString());

\_Current\_Place\_Number = System.Convert.ToInt32(t[3].ToString());

\_Current\_Arrival\_Time = t[4].ToString();

\_Current\_Arrival\_Date = t[5].ToString();

\_Current\_Arrival\_Stop\_Name = t[6].ToString();

\_Current\_Departure\_Time = t[7].ToString();

\_Current\_Departure\_Date = t[8].ToString();

\_Current\_Departure\_Stop\_Name = t[9].ToString();

}

catch (System.Exception)

{ }

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

}

private void Departure\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(Departure\_BOX, e);

}

private void Arrival\_BOX\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(Arrival\_BOX, e);

}

}

}

ReservationControl.xaml

<UserControl x:Class="RZDKursovoy.ReservationControl"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:RZDKursovoy"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

d:DesignHeight="450" MinHeight="450" MinWidth="800" d:DesignWidth="800">

<Grid Background="White">

<GroupBox x:Name="ChooseTrainBox" Header="Выберите поезд" Margin="0" d:IsHidden="True">

<Grid>

<Button x:Name="ChooseTrainNextButton" VerticalAlignment="Bottom" Content="Далее" Height="29" Margin="0" Click="ChooseTrainNextButton\_Click" RenderTransformOrigin="0.497,1.172"/>

<DataGrid AutoGenerateColumns="True" AutoGeneratingColumn="ChooseTrainGRID\_AutoGeneratingColumn" SelectionChanged="ChooseTrainGRID\_SelectionChanged" IsReadOnly="True" x:Name="ChooseTrainGRID" Margin="0,0,0,30" Loaded="ChooseTrainGRID\_Loaded"/>

</Grid>

</GroupBox>

<GroupBox x:Name="ChooseTrainTypeBox" Header="Выбор места" Margin="0,0,0,0" d:IsHidden="True">

<Grid>

<Grid x:Name="ChooseTrainTypeGRID">

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid x:Name="StepOneGrid" Grid.Row="0" Grid.Column="0">

<ComboBox x:Name="ChooseRailcarType" HorizontalAlignment="Center" Margin="0,26,0,0" VerticalAlignment="Top" Width="240"/>

<Label x:Name="RailcarTypeLab" Content="Выберите тип вагона:&#xD;&#xA;" HorizontalAlignment="Center" Margin="0,0,0,0" VerticalAlignment="Top" FontWeight="Bold" Height="26" Width="240" HorizontalContentAlignment="Center"/>

<Button x:Name="ChooseRailcarTypeButton" Content="Выбрать" HorizontalAlignment="Center" Margin="0,68,0,0" VerticalAlignment="Top" Click="ChooseRailcarTypeButton\_Click" Width="240"/>

</Grid>

<Grid x:Name="StepTwoGrid" Grid.Row="0" Grid.Column="1" HorizontalAlignment="Center" VerticalAlignment="Top" Width="242" Margin="0,0,0,0">

<Label x:Name="RailcarChooseNumberLab" Content="Выберите номер вагона:" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,0,0,0" Width="215" FontWeight="Bold" HorizontalContentAlignment="Center"></Label>

<ComboBox x:Name="ChooseRailcarNumberBox" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,26,0,0" Width="215"></ComboBox>

<Button x:Name="ChooseRailcarNumberButton" Content="Выбрать" HorizontalAlignment="Center" Margin="0,68,0,0" Click="ChooseRailcarNumberButton\_Click" VerticalAlignment="Top" Width="215"/>

</Grid>

<Grid x:Name="StepThreeGrid" Grid.Row="1" Grid.ColumnSpan="3" Grid.Column="0" Margin="0,0,0,0">

<Image Grid.Row="1" Grid.Column="0" x:Name="TrainCardImage" Margin="0,0,0,10"></Image>

</Grid>

<Grid x:Name="StepThreePlusGrid" Grid.Row="0" Grid.Column="2">

<Label x:Name="SeatChooseNumberLab" Content="Выберите номер места:" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,0,0,0" Width="225" FontWeight="Bold" HorizontalContentAlignment="Center"/>

<ComboBox x:Name="SeatChooseNumberBox" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,26,0,0" Width="225"/>

<Button x:Name="SeatChooseNumberButton" Content="Выбрать" HorizontalAlignment="Center" Margin="0,68,0,0" VerticalAlignment="Top" Width="225" Click="SeatChooseNumberButton\_Click"/>

</Grid>

</Grid>

</Grid>

</GroupBox>

<GroupBox x:Name="InputPersonalDataBox" Margin="0" BorderBrush="Black" Header="Ввод данных" Foreground="Black" Background="{x:Null}">

<Grid x:Name="test">

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition Height="40"></RowDefinition>

<RowDefinition Height="40"></RowDefinition>

<RowDefinition Height="40"></RowDefinition>

<RowDefinition Height="40"></RowDefinition>

<RowDefinition Height="40"></RowDefinition>

<RowDefinition Height="40"></RowDefinition>

<RowDefinition Height="40"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="50"></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition Width="50"></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBox x:Name="RegFamBox" Grid.Column="1" Grid.Row="1" PreviewKeyUp="RegFamBox\_PreviewKeyUp" PreviewTextInput="RegFamBox\_PreviewTextInput" MaxLength="30" Style="{StaticResource MaterialDesignFloatingHintTextBox}" Margin="180,0" VerticalAlignment="Center" Height="36" Width="282">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Фамилия\*

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<TextBox MaxLength="30" Grid.Column="1" Grid.Row="2" x:Name="RegNameBox" PreviewKeyUp="RegNameBox\_PreviewKeyUp" PreviewTextInput="RegNameBox\_PreviewTextInput" Style="{StaticResource MaterialDesignFloatingHintTextBox}" Margin="180,0" Height="36" VerticalAlignment="Center" Width="282">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Имя\*

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<TextBox x:Name="RegPathrBox" Grid.Column="1" Grid.Row="3" PreviewTextInput="RegPathrBox\_PreviewTextInput" PreviewKeyUp="RegPathrBox\_PreviewKeyUp" Style="{StaticResource MaterialDesignFloatingHintTextBox}" Margin="180,0" VerticalAlignment="Center" MaxLength="30" Height="40" Width="282">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Отчество

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<materialDesign:PackIcon Grid.Column="1" Grid.Row="4" VerticalAlignment="Center" HorizontalAlignment="Center" Kind="Phone" Margin="0,0,300,0" />

<xctk:MaskedTextBox materialDesign:HintAssist.Hint="test" Grid.Column="1" Grid.Row="4" x:Name="\_maskedTextBox" LostFocus="\_maskedTextBox\_LostFocus" VerticalAlignment="Center" Margin="180,0" Height="26" GotFocus="\_maskedTextBox\_GotFocus" Width="282"/>

<TextBox x:Name="RegPassSeries" Grid.Column="1" Grid.Row="5" PreviewTextInput="RegPassSeries\_PreviewTextInput" PreviewKeyUp="RegPassSeries\_PreviewKeyUp" Style="{StaticResource MaterialDesignFloatingHintTextBox}" Margin="180,0" VerticalAlignment="Center" MaxLength="4" Height="36" Width="282">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Серия паспорта\*

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<TextBox x:Name="RegPassNumber" Grid.Column="1" Grid.Row="6" PreviewTextInput="RegPassNumber\_PreviewTextInput" PreviewKeyUp="RegPassNumber\_PreviewKeyUp" Style="{StaticResource MaterialDesignFloatingHintTextBox}" Margin="180,0" MaxLength="6" VerticalAlignment="Center" Height="36" Width="282">

<TextBox.ContextMenu>

<ContextMenu>

</ContextMenu>

</TextBox.ContextMenu>

<materialDesign:HintAssist.Hint>

<StackPanel

Orientation="Horizontal"

Margin="-2 0 0 0">

<TextBlock>

Номер паспорта\*

</TextBlock>

</StackPanel>

</materialDesign:HintAssist.Hint>

</TextBox>

<Button Grid.Column="1" Grid.Row="7" x:Name="InputData" VerticalAlignment="Bottom" Content="Отправить данные" Margin="180,0" Click="InputData\_Click" Width="282"/>

<Label Grid.Column="1" Grid.Row="8" Content="AVB Inc PRODUCTION" Margin="0" VerticalAlignment="Center" IsEnabled="False" Height="25" HorizontalContentAlignment="Center"/>

</Grid>

</GroupBox>

</Grid>

</UserControl>

ReservationControl.xaml.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Reflection;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Imaging;

namespace RZDKursovoy

{

/// <summary>

/// Логика взаимодействия для ReservationControl.xaml

/// </summary>

public partial class ReservationControl : UserControl

{

/\*Переменные\*/

private Menu \_menu;

private string ArrivalStation = "###";

private string DepartureStation = "###";

private string ArrivalDate = "###";

private List<string> Routs = new List<string>();

private List<string> TrainsList = new List<string>();

private MySqlConnection \_connection;

private int Arrival\_ID = new int();

private int Departure\_ID = new int();

private int Railcar\_Number = new int();

private int CurrentRoutID = new int();

private string CurrentTrainNumber = "";

private string ChoosedRailcarType = "";

private int ChoosedSeatNumber = new int();

private int Passenger\_Number = new int();

private ApplicationLogic AL = new ApplicationLogic();

private poselki.BestErrors Errors = new poselki.BestErrors();

/\*Проперти\*/

public string SetArrival { set { ArrivalStation = value; } }

public string SetDeparture { set { DepartureStation = value; } }

public string SetDate { set { ArrivalDate = value; } }

public List<string> SetRouts { set { Routs = value; } }

public List<string> SetTrainsList { set { TrainsList = value; } }

public MySqlConnection SetConnection { set { \_connection = value; } }

public Menu SetMenu { set { \_menu = value; } }

/\*Процедуры\*/

public ReservationControl()

{

InitializeComponent();

ChooseTrainBox.Visibility = Visibility.Visible;

ChooseTrainTypeBox.Visibility = Visibility.Hidden;

InputPersonalDataBox.Visibility = Visibility.Hidden;

StepTwoGrid.IsEnabled = false;

StepThreePlusGrid.IsEnabled = false;

StepThreeGrid.IsEnabled = false;

}

private void ThrowTrainListToTable()

{

var test = new List<TableFillKostil>();

for (int i = 0; i < Routs.Count; i++)

{

for (int j = 0; j < TrainsList.Count; j++)

{

if (TrainsList[j] != "-1")

{

string[] Args = { ArrivalStation, Routs[i], TrainsList[j] };

Arrival\_ID = AL.CatchIntResult(\_connection, "select GetArrivalID", Args);

string[] args = { DepartureStation, Routs[i], TrainsList[j] };

Departure\_ID = AL.CatchIntResult(\_connection, "select GetDepartureID", args);

if ((Arrival\_ID != -1) && (Departure\_ID != -1))

{

var TrainData = AL.TrainInfo(\_connection, TrainsList[j], Arrival\_ID, Departure\_ID);

test.Add(new TableFillKostil(TrainData[0], TrainData[1], TrainData[2], TrainData[3], TrainData[4], TrainData[5]));

}

}

}

}

if (test.Count != 0)

{

ChooseTrainGRID.ItemsSource = test;

}

else

{

AL.MessageErrorShow("При загрузке поездов произошла ошибка. Сообщите об этом администратору", "Ошибка");

}

}

private void ChooseTrainNextButton\_Click(object sender, RoutedEventArgs e)

{

if (CurrentTrainNumber != "")

{

string[] Args = { ArrivalStation, CurrentRoutID.ToString(), CurrentTrainNumber };

Arrival\_ID = AL.CatchIntResult(\_connection, "select GetArrivalID", Args);

string[] args = { DepartureStation, CurrentRoutID.ToString(), CurrentTrainNumber };

Departure\_ID = AL.CatchIntResult(\_connection, "select GetDepartureID", args);

ChooseTrainBox.Visibility = Visibility.Hidden;

ChooseTrainTypeBox.Visibility = Visibility.Visible;

AddToRailcarTypesBox();

}

else

{

AL.MessageShow("Вы должны выбрать поезд", "Предупреждение");

}

}

private void AddToRailcarTypesBox()

{

var Data = AL.Available\_Railcar\_Types(\_connection, CurrentTrainNumber);

for (int i = 0; i < Data.Count; i++)

{

ChooseRailcarType.Items.Add(Data[i]);

}

}

private void ChooseRailcarTypeButton\_Click(object sender, RoutedEventArgs e)

{

try

{

ChooseRailcarNumberBox.Items.Clear();

SeatChooseNumberBox.Items.Clear();

var r = ChooseRailcarType.SelectedItem.ToString().Split(' ');

ChoosedRailcarType = r[0];

string[] args = { CurrentTrainNumber, ChoosedRailcarType };

var Data = AL.CatchStringListResult(\_connection, "call ThrowTrainNumbersList", args);//ThrowTrainNumbersList(\_connection, CurrentTrainNumber, ChoosedRailcarType);

for (int i = 0; i < Data.Count; i++)

{

ChooseRailcarNumberBox.Items.Add(Data[i]);

}

if (ChoosedRailcarType == "Плацкартный")

{

TrainCardImage.Source = new BitmapImage(new Uri(@"/Images/PlackartScheme.png", UriKind.Relative));

}

else if (ChoosedRailcarType == "Купе")

{

TrainCardImage.Source = new BitmapImage(new Uri(@"/Images/CupeScheme.png", UriKind.Relative));

}

else if (ChoosedRailcarType == "СВ")

{

TrainCardImage.Source = new BitmapImage(new Uri(@"/Images/SvScheme.png", UriKind.Relative));

}

StepTwoGrid.IsEnabled = true;

StepThreeGrid.IsEnabled = true;

}

catch (Exception)

{

AL.MessageShow("Вы должны выбрать тип вагона", "Предупреждение");

}

}

private void ChooseRailcarNumberButton\_Click(object sender, RoutedEventArgs e)

{

try

{

SeatChooseNumberBox.Items.Clear();

Railcar\_Number = Convert.ToInt32(ChooseRailcarNumberBox.SelectedItem.ToString());

StepThreePlusGrid.IsEnabled = true;

string[] data = { CurrentTrainNumber, Railcar\_Number.ToString(), Arrival\_ID.ToString(), Departure\_ID.ToString() };

var k = AL.CatchStringListResult(\_connection, "call Available\_For\_Planting\_Seats", data); //AL.Available\_For\_Planting\_Seats(\_connection, CurrentTrainNumber, Railcar\_Number, Arrival\_ID, Departure\_ID);

for (int i = 0; i < k.Count; i++)

SeatChooseNumberBox.Items.Add(k[i]);

}

catch (Exception)

{

AL.MessageShow("Вы должны выбрать номер вагона", "Предупреждение");

}

}

private void SeatChooseNumberButton\_Click(object sender, RoutedEventArgs e)

{

try

{

ChoosedSeatNumber = Convert.ToInt32(SeatChooseNumberBox.SelectedItem.ToString());

ChooseTrainTypeBox.Visibility = Visibility.Hidden;

InputPersonalDataBox.Visibility = Visibility.Visible;

}

catch (Exception)

{

AL.MessageShow("Вы должны выбрать место", "Предупреждение");

}

}

private void InputData\_Click(object sender, RoutedEventArgs e)

{

try

{

var QueryString = "call EmployPlaces";

string[] args = { RegPassSeries.Text, RegPassNumber.Text, Properties.PersonalData.Default.KeySi };

Passenger\_Number = AL.CatchIntResult(\_connection, "SELECT FindPassenger", args); //FindPassenger(\_connection, Convert.ToInt32(RegPassSeries.Text), Convert.ToInt32(RegPassNumber.Text));

if ((RegNameBox.Text != "") && (RegFamBox.Text != "") && (RegPassSeries.Text != "") && (RegPassNumber.Text != ""))

{

if (Passenger\_Number == -1)

{

if ((RegPassSeries.Text.Length == 4) && (RegPassNumber.Text.Length == 6))

{

string[] Args = { RegFamBox.Text, RegNameBox.Text, RegPathrBox.Text, RegPassSeries.Text, RegPassNumber.Text, \_maskedTextBox.Text, Properties.PersonalData.Default.KeySi };

var PasNewNum = AL.CatchIntResult(\_connection, "select PassengerAddToDB", Args);

string[] data = { CurrentTrainNumber, Railcar\_Number.ToString(), ChoosedSeatNumber.ToString(), PasNewNum.ToString(), Arrival\_ID.ToString(), Departure\_ID.ToString() };

var res = AL.MagicUniversalControlData(QueryString, data, "Reservation", \_connection);

poselki.BestErrors BE = new poselki.BestErrors();

BE.CatchError(res);

}

else

{

AL.MessageErrorShow("Вы неправильно заполнили паспортные данные", "Ошибка");

return;

}

}

else

{

var ExistsData = AL.FindPassengerWithPersonalData(\_connection, Convert.ToInt32(RegPassSeries.Text), Convert.ToInt32(RegPassNumber.Text));

if ((RegFamBox.Text == ExistsData[0]) && (RegNameBox.Text == ExistsData[1]) && (RegPathrBox.Text == ExistsData[2]))

{

if ((RegPassSeries.Text.Length == 4) && (RegPassNumber.Text.Length == 6))

{

string[] data = { CurrentTrainNumber, Railcar\_Number.ToString(), ChoosedSeatNumber.ToString(), Passenger\_Number.ToString(), Arrival\_ID.ToString(), Departure\_ID.ToString() };

var res = AL.MagicUniversalControlData(QueryString, data, "Reservation", \_connection);

poselki.BestErrors BE = new poselki.BestErrors();

BE.CatchError(res);

}

else

{

AL.MessageErrorShow("Вы неправильно заполнили паспортные данные", "Ошибка");

return;

}

}

else

{

AL.MessageErrorShow("Информация о существующем в базе пассажире заполнена неверно. Полиция уже рядом", "Ошибка");

return;

}

}

}

else

{

AL.MessageErrorShow("Вы не заполнили одно или несколько полей, необходимых для регистрации", "Ошибка");

return;

}

}

catch (Exception)

{

AL.MessageErrorShow("Вы неверно заполнили серию или номер паспорта", "Ошибка");

return;

}

\_menu.CheckActivateCabinet();

\_menu.PerfectReflectionGRID.Children.Remove(this);

\_menu.Reflector.IsEnabled = true;

InputData.Focusable = false;

}

private void \_maskedTextBox\_GotFocus(object sender, RoutedEventArgs e)

{

\_maskedTextBox.Mask = "+7(000)000-0000";

}

private void ChooseTrainGRID\_Loaded(object sender, RoutedEventArgs e)

{

ThrowTrainListToTable();

}

private void ChooseTrainGRID\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

try

{

TableFillKostil TFK = (TableFillKostil)ChooseTrainGRID.SelectedItem;

CurrentTrainNumber = TFK.Par1;

string[] args = { CurrentTrainNumber };

CurrentRoutID = AL.CatchIntResult(\_connection, "select ThrowRoutID", args);//ThrowRoutID(\_connection, CurrentTrainNumber);

}

catch (Exception)

{

AL.MessageErrorShow("При выборе поезда произошла ошибка. Попробуйте выбрать другой поезд.", "Ошибка");

}

}

public static string GetPropertyDisplayName(object descriptor)

{

var pd = descriptor as PropertyDescriptor;

if (pd != null)

{

// Check for DisplayName attribute and set the column header accordingly

var displayName = pd.Attributes[typeof(DisplayNameAttribute)] as DisplayNameAttribute;

if (displayName != null && displayName != DisplayNameAttribute.Default)

{

return displayName.DisplayName;

}

}

else

{

var pi = descriptor as PropertyInfo;

if (pi != null)

{

Object[] attributes = pi.GetCustomAttributes(typeof(DisplayNameAttribute), true);

for (int i = 0; i < attributes.Length; ++i)

{

var displayName = attributes[i] as DisplayNameAttribute;

if (displayName != null && displayName != DisplayNameAttribute.Default)

{

return displayName.DisplayName;

}

}

}

}

return null;

}

private void ChooseTrainGRID\_AutoGeneratingColumn(object sender, DataGridAutoGeneratingColumnEventArgs e)

{

var displayName = GetPropertyDisplayName(e.PropertyDescriptor);

if (!string.IsNullOrEmpty(displayName))

{

e.Column.Header = displayName;

}

}

private void RegFamBox\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.InputPersonalDataProtector(RegFamBox, e);

}

private void RegPassSeries\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.InputNumbersDataProtector(RegPassSeries, e);

}

private void RegNameBox\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.InputPersonalDataProtector(RegNameBox, e);

}

private void RegPathrBox\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.InputPersonalDataProtector(RegPathrBox, e);

}

private void RegPassNumber\_PreviewTextInput(object sender, System.Windows.Input.TextCompositionEventArgs e)

{

AL.InputNumbersDataProtector(RegPassNumber, e);

}

private void \_maskedTextBox\_LostFocus(object sender, RoutedEventArgs e)

{

bool error = new bool();

for (int i = 0; i < \_maskedTextBox.Text.Length; i++)

if (\_maskedTextBox.Text[i] == '\_')

error = true;

if (error)

{

\_maskedTextBox.Text = "";

\_maskedTextBox.Mask = "";

}

}

private void RegFamBox\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RegFamBox, e);

}

private void RegNameBox\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RegNameBox, e);

}

private void RegPathrBox\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RegPathrBox, e);

}

private void RegPassSeries\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RegPassSeries, e);

}

private void RegPassNumber\_PreviewKeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

AL.DontCtrlVAndSpace(RegPassNumber, e);

}

}

}

TableFillKostil.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace RZDKursovoy

{

class TableFillKostil

{

public TableFillKostil(string Par1, string Par2, string Par3, string Par4, string Par5, string Par6)

{

this.Par1 = Par1;

this.Par2 = Par2;

this.Par3 = Par3;

this.Par4 = Par4;

this.Par5 = Par5;

this.Par6 = Par6;

}

[DisplayName("Номер поезда")]

public string Par1 { get; set; }

[DisplayName("Время отправления")]

public string Par2 { get; set; }

[DisplayName("Время прибытия")]

public string Par3 { get; set; }

[DisplayName("Дата прибытия")]

public string Par4 { get; set; }

[DisplayName("Начальная остановка")]

public string Par5 { get; set; }

[DisplayName("Конечная остановка")]

public string Par6 { get; set; }

}

}