МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 «Программное обеспечение информационной

безопасности мобильных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

Разработка программного средства «Автовокзал»

Выполнил студент Анкудович Виталий Геннадьевич

(Ф.И.О.)

Руководитель проекта ассистент Гладкий М.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Заведующий кафедрой д.к.т.н., доц. Смелов В.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Консультанты ассистент Гладкий М.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Нормоконтролер ассистент Гладкий М.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Минск 2017

**Реферат**

АВТОВОКЗАЛ, БАЗА ДАННЫЙ, ООП, С#, WPF.

Основной целью курсового проекта является разработка wpf-приложения «Автовокзал».

Пояснительная записка состоит из введения, 4 разделов и заключения.

Во введении представлена общая информация, дающая представление о предстоящей работе.

В первом разделе представлена информация о средствах разработки приложения.

Во втором разделе рассматривается архитектура базы данных.

В третьем разделе представлена информация о архитектуре приложения

В четвертом разделе представлено руководство пользователя

Заключение состоит из итогов курсового проекта и задач, решенных в ходе проектирования и разработки приложения.

Пояснительная записка курсового проекта содержит 26 страниц, 36 рисунков, 4 источников литературы и 1 приложения.

**Оглавление**

[Введение 6](#_Toc483201879)

[1. Постановка задачи 7](#_Toc483201880)

[2.Разработка функциональной модели и модели данных ПС 10](#_Toc483201881)

[3.Архитектура приложения 12](#_Toc483201882)

[4.Руководство пользователя 13](#_Toc483201883)

[5.Тестирование 20](#_Toc483201884)

[Заключение 23](#_Toc483201885)

[Список используемых источников 24](#_Toc483201886)

[Приложение 25](#_Toc483201887)

# **Введение**

Путешествие — то, что приносит нам радость. Особенно в XXI веке, когда умные гаджеты и программы помогают бронировать билеты, не выходя из дому. Также не терять свое личное время в очередях ради покупки билетов, если их еще не распродали. А также зайти и посмотреть во сколько отправление, сколько времени ехать в той или иной пункт назначения, а вдруг нужно еще купить билет для друга или подруги. Безусловно, выход есть.

А что, если просматривать отправления и бронировать билеты лично в своем аккаунте. Все же станет намного проще, не нужно искать билет, вспоминать во сколько отправление автобуса, какие билеты забронировал. Теперь, все ваши забронированные билеты в сохранности – о них позаботиться приложение «Автовокзал». Остается лишь зайти в свой аккаунт и нажать кнопку «Забронированные билеты».

Целью моего курсового проекта является создание программного средства «Автовокзал». Особенностями данного приложения являются:

* интуитивно-понятный интерфейс;
* регистрация и авторизация пользователей;
* процедура бронирования билетов;
* просмотр забронированных билетов;
* изменение данных авторизированного пользователя.

Пользователь с легкостью сможет разобраться в приложении. Именно поэтому программное средство «Автовокзал» будет полезно для пользователей и значительно упростит бронирование билетов и сэкономит их время.

# **Постановка задачи**

Функционально ПС должно выполнять следующие задачи:

* сохранять рабочую информацию в централизованной базе данных;
* процедура бронирования билетов;
* регистрация и авторизация пользователей;
* предоставление удобного интерфейс пользователя.
  1. **Средства разработки**

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Когда говорят C#, нередко имеют в виду технологии платформы .NET (WPF, ASP.NET). И, наоборот, когда говорят .NET, нередко имеют в виду C#. Однако, хотя эти понятия связаны, отождествлять их неверно. Язык C# был создан специально для работы с фреймворком .NET, однако само понятие .NET несколько шире. Также еще следует отметить такую особенность языка C# и фреймворка .NET, как автоматическая сборка мусора. А это значит, что нам в большинстве случаев не придется, в отличие от С++, заботиться об освобождении памяти. Вышеупомянутая общеязыковая среда CLR сама вызовет сборщик мусора и очистит память.

Технология WPF (Windows Presentation Foundation) является часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов. Если при создании традиционных приложений на основе WinForms за отрисовку элементов управления и графики отвечали такие части ОС Windows, как User32 и GDI+, то приложения WPF основаны на DirectX. В этом состоит ключевая особенность рендеринга графики в WPF: используя WPF, значительная часть работы по отрисовке графики, как простейших кнопочек, так и сложных 3D-моделей, ложиться на графический процессор на видеокарте, что также позволяет воспользоваться аппаратным ускорением графики.

Одной из важных особенностей является использование языка декларативной разметки интерфейса XAML, основанного на XML: вы можете создавать насыщенный графический интерфейс, используя или декларативное объявление интерфейса, или код на управляемых языках C# и VB.NET, либо совмещать и то, и другое.

ADO.NET предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология представляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Причем важно отметить, что систем управления баз данных может быть множество. В своей сущности они могут различаться. MS SQL Server, например, для создания запросов использует язык T-SQL, а MySQL и Oracle применяют язык PL-SQL. Разные системы баз данных могут иметь разные типы данных. Также могут различаться какие-то другие моменты. Однако функционал ADO.NET построен таким образом, чтобы предоставить разработчикам унифицированный интерфейс для работы с самыми различными СУБД.

Основу интерфейса взаимодействия с базами данных в ADO.NET представляет ограниченный круг объектов: Connection, Command, DataReader, DataSet и DataAdapter. С помощью объекта Connection происходит установка подключения к источнику данных. Объект Command позволяет выполнять операции с данными из БД. Объект DataReader считывает полученные в результате запроса данные. Объект DataSet предназначен для хранения данных из БД и позволяет работать с ними независимо от БД. И объект DataAdapter является посредником между DataSet и источником данных. Главным образом, через эти объекты и будет идти работа с базой данных.

Однако, чтобы использовать один и тот же набор объектов для разных источников данных, необходим соответствующий провайдер данных. Собственно, через провайдер данных в ADO.NET и осуществляется взаимодействие с базой данных. Причем для каждого источника данных в ADO.NET может быть свой провайдер, который собственно и определяет конкретную реализацию вышеуказанных классов.

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

**1.2 Обзор прототипов**

Прототипов на данный момент не существует, поэтому ниже будет продемонстрирован прототип web-приложения.

Интерфейс одного такого прототипа представлен на рисунке 1.1.



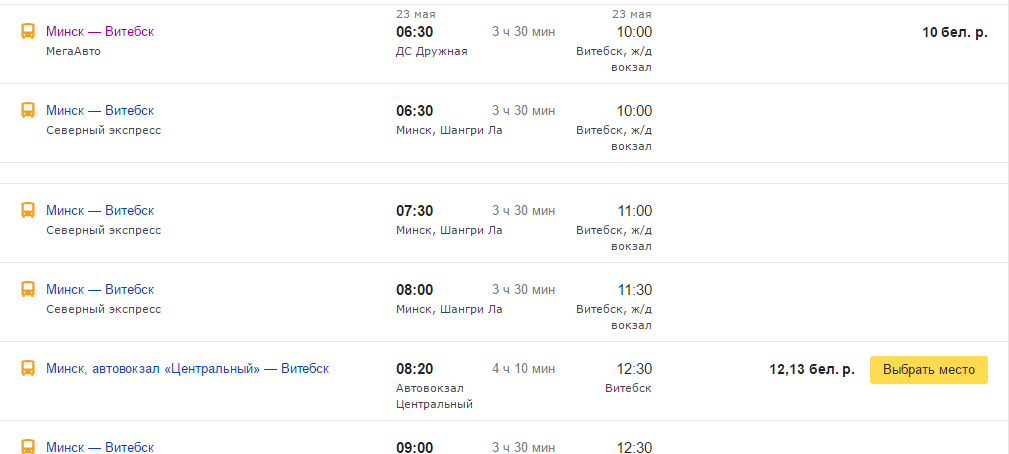


Рисунок 1.1 – просмотр маршрута из одной отправления в точку прибытия

Данный сайт предоставляет выбор точки отправления, точки прибытия и дня отправления. В результате поиска, нажав на кнопку «Найти», появляется список автобусов, которые разбиты на разное время отправление, а также время в пути и цену билета.

После нажатия на «Минск-Витебск», предоставляется информция и времени отправлении и прибытии, а также о времени в пути, предоставленное на рисунке 1.2.

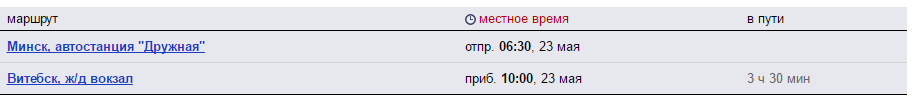


Рисунок 1.2 – Дополнительная информация о маршруте

Данный сайт подходит для тех пользователей, которые хотя узнать информации о времени отправления и прибытия.

Таким образом, в связи с отсутствием просмотра количества свободных мест, а также возможности бронирования билетов, была поставлена задача реализовать приложение с недостающим функционалом.

# **Разработка функциональной модели и модели данных ПС**

Модель данных строилась при помощи системы управления реляционными базами данных Microsoft SQL Server.

База данных – это совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ.

База данных Kursach программного средства «Автовокзал» состоит из 3 таблиц, а именно «Ticket», «Route», «yuzer», которые связаны между собой. Структура таблиц изображена на рисунке 2.1:

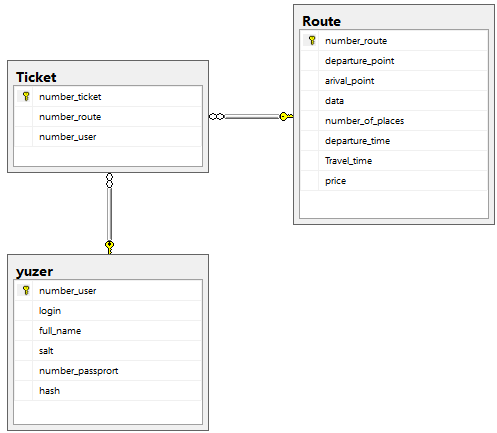


Рисунок 2.1 – схема базы данных Kursach

Таблица «Route» предназначена для хранения информации о номере маршрута. Она содержит следующие поля:

* number\_route – идентификационных номер каждого маршрута, первичный ключ (primary key), таблица имеет тип int и генерируется автоматически с помощью свойства identity;
* departure\_point - пункт отправления;
* arrival\_point - пункт назначения;
* data - день недели отправления;
* number\_of\_places - информация о количестве свободных мест;
* departure\_time - время отправления;
* Travel\_time - время в пути;
* price – стоимость.

Таблица «yuzer» предназначена для хранения информации о пользователе. Она содержит следующие поля:

* number\_user - идентификационных номер каждого пользователя, первичный ключ (primary key), таблица имеет тип int и генерируется автоматически с помощью свойства identity;
* login - хранит логин пользователя;
* full\_name – хранит ФИО пользователя;
* salt - хранит соль, для хеширования пароля;
* number\_passport - хранит номер паспорта пользователя;
* hash – хранит хеш пароля пользователя.

Таблица «Ticket» предназначена для хранения информации о номере билета, входящих в бронирование. Она имеет следующие поля:

* number\_ticket - идентификационных номер каждого билета, первичный ключ (primary key), таблица имеет тип int и генерируется автоматически с помощью свойства identity;
* number\_route - идентификационный номер маршрута;
* number\_user - идентификационный номер пользователя;

# **Архитектура приложения**

Общая диаграмма продемонстрирована на рисунке 3.1



Рисунок 3.1 – диаграмма программы

В данной структуре показаны классы, которые взаимодействуют с базой данный. Также есть классы, которые относятся к какому-то определенному функционалу.

* класс «db» - класс, предназначенный для обращения к базе данных;
* классы «Validate», «Registr», «Login», «UpdateLogin», «UpdateName», «UpdatePass», «RemovePass», «Bron1», «Bron2» - классы, за счет которых осуществляется валидация;
* классы «Settings», «DeletAkk», «RemovePass», «EE», «UpName», «UpLog» - классы, в которых осуществляется изменение данных пользователя;
* классы «Login», «Registr», «SaltedHash» - относятся к авторизации и регистрации пользователей;
* класс «MainWindow» - основное окно программы;
* класс «Bron1», «Bron2», «Booking0», «Booking1», «Booking2» - классы, отвечающие за бронирование и вывод информации о билетах.

# **Руководство пользователя**

При запуске приложения «Автовокзал» появляется окно. На данном окне присутствуют 2 кнопки, изображенное на рисунке 5.1:

* кнопа «Вход» для входа в свой аккаунт;
* кнопка «Регистрация» для регистрации новых пользователей.

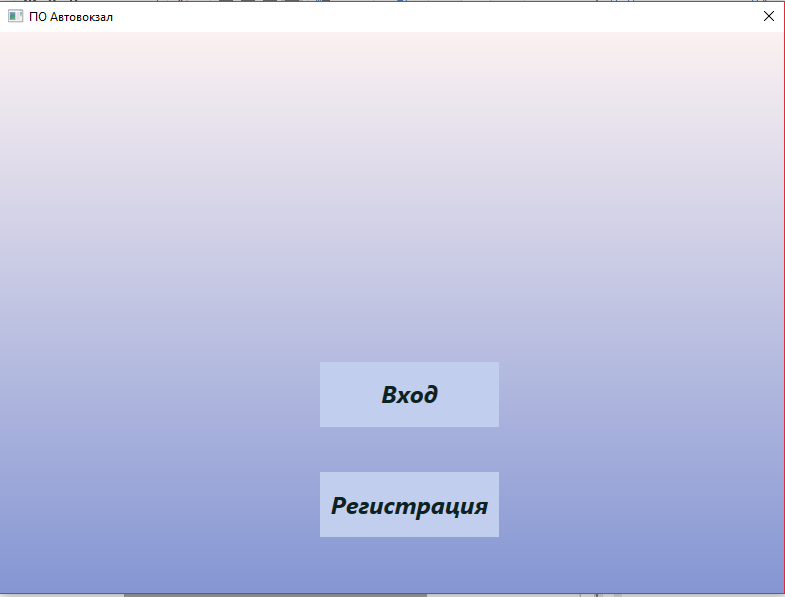
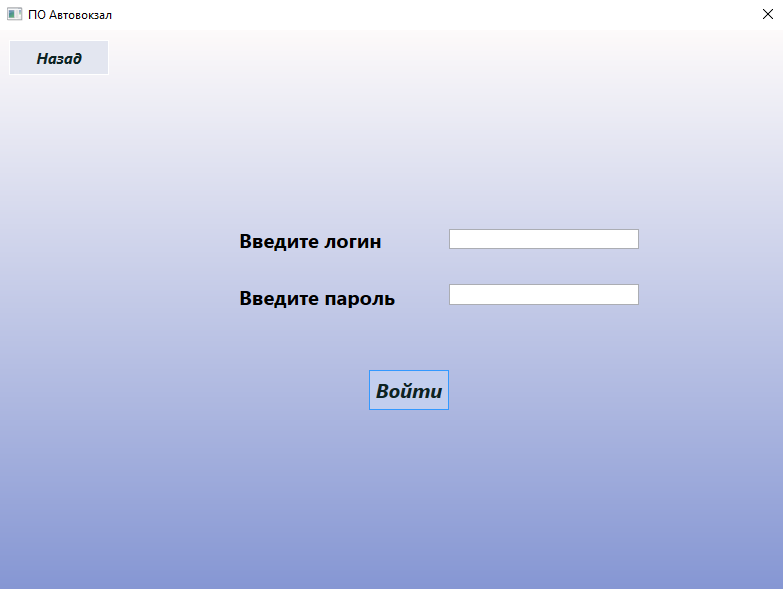


Рисунок 5.1 – окно при запуске программы

Нажав на кнопку «Вход», пользователь переходит к авторизации, где изображены 2 поля для вода данных, которые слева имеют название своего поля «Введите логин» и «Введите пароль» соответственно, кнопка для выхода обратно «Назад», а также кнопка для авторизации «Войти» как показано на рисунке 5.2



Рисукнок 5.2 – Авторизация

Нажав на кнопку «Регистрация», пользователь переходит к регистрации, где изображены 5 полей для вода данных, которые слева имеют название своего поля «Логин», «ФИО», «Номер паспорта», «Пароль», «Повторите пароль» соответственно, кнопка для выхода обратно «Назад», а также кнопка для регистрации «Зарегистрироваться» как показано на рисунке 5.3

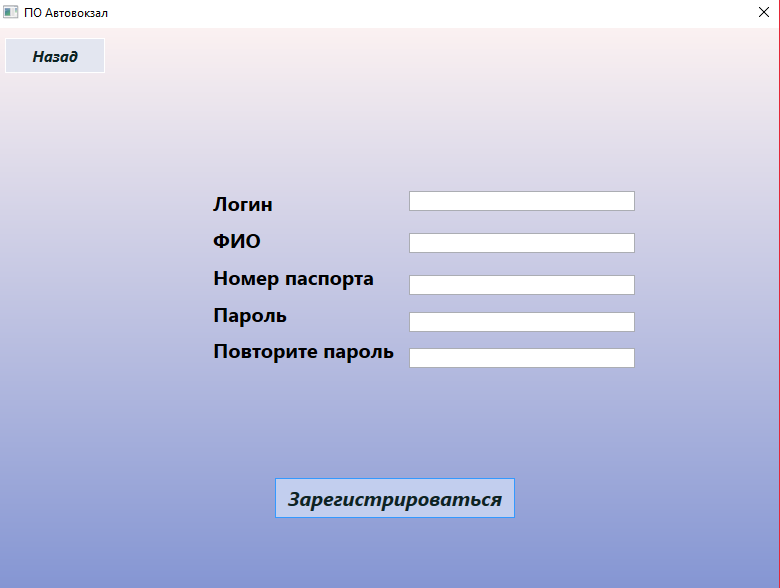


Рисунок 5.3 – Регистрация

После правильно заполненных полех и нажатия на кнопку регистрации, переходим на Авторизацию, что было выше описано. После успешной авторизации переходим на главное меню пользователя, где находятся 5 кнопок (рис. 5.4):

* «Информация» - просмотр информации о прибытии и отбытии автобусов;
* «Забронировать билет» - процесс бронирование билетов;
* «Забронированные билеты» - просмотр забронированных билетов;
* «Настройка» - пользовательские настройки;
* «Выйти» - выход с аккаунта.



Рисунок 5.4 – главное меню

Кнопка «Информация» имеет 2 кнопки, при нажатии на которых открывается информация о маршрутах, времени отправлении, дате, количестве свободных мест и времени в пути, как показано на рисунке 5.5.1 и 5.5.2.

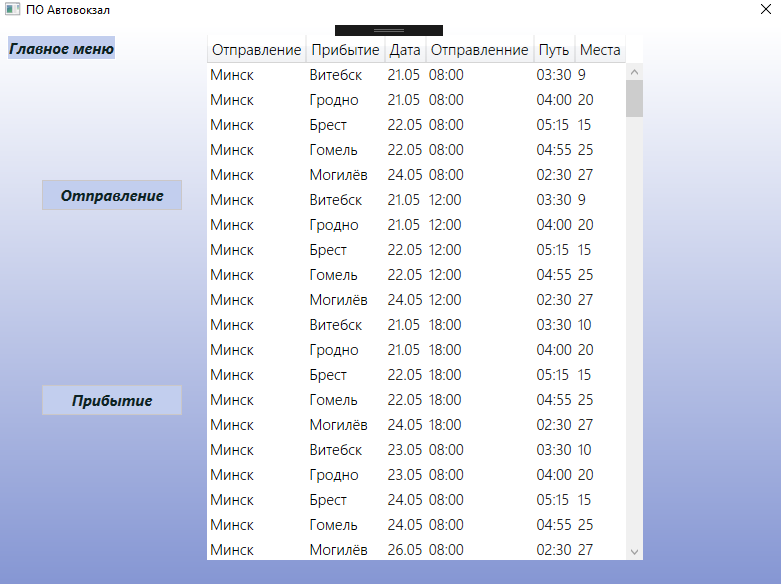


Рисунок 5.5.1 – информация

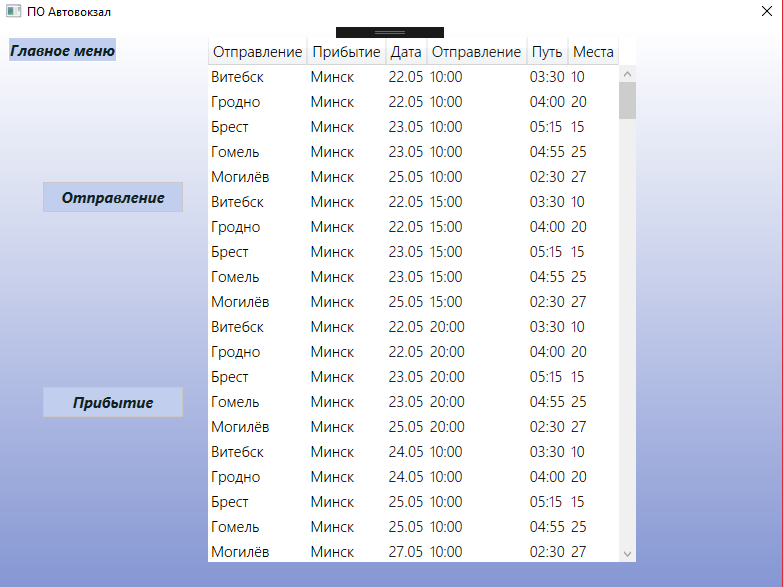


Рисунок 5.5.2 – информация

Кнопка «Забронировать билет» содержит выборку по городу, в который будет осуществляться бронирование билета и кнопка далее (Рис. 5.6).

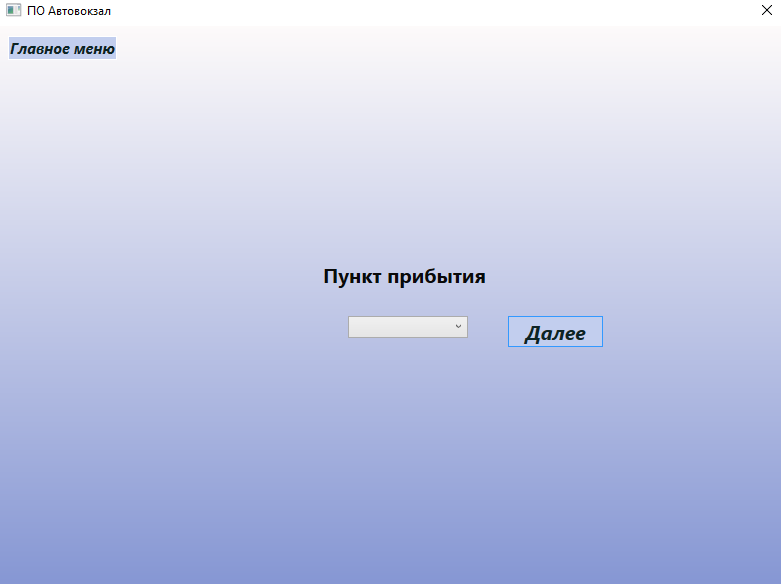


Рисунок 5.6 – 1-й этап бронирования

Кнопка «Далее», последний этап бронирования, где выбирается день недели и время отправления, а также информация в таблице чуть выше о дате и времени, кнопка «Забронировать», «Назад», «Главное меню» (Рис. 5.6).

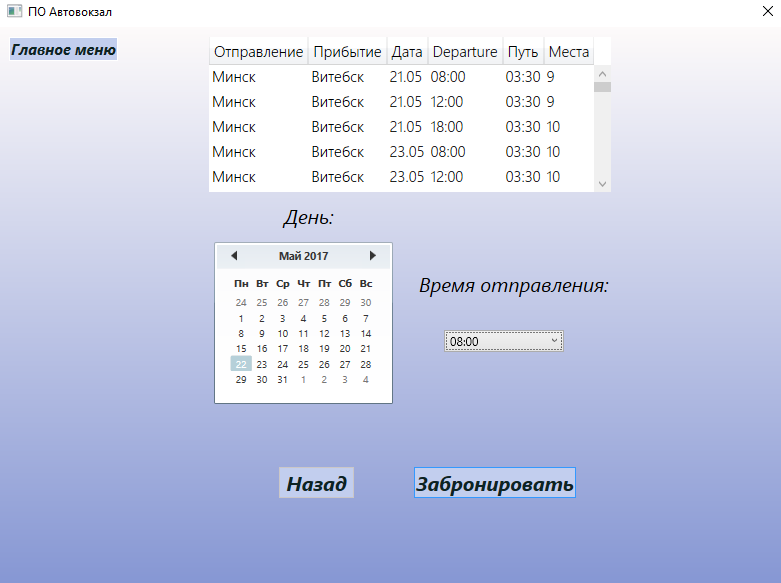


Рисунок 5.7 – последний этап бронирования билета

Кнопка «Забронировать» бронирует, если есть свободные места и правильно выбран дата и время.

Кнопка «Настройки» имеет 6 кнопок (Рис. 5.8):

* «Главное меню» - переход на главное меню;
* «Изменить имя» - изменение ФИО пользователя;
* «Изменить логин» - изменение логина пользователя;
* «Изменить пароль» - изменение пароля пользователя;
* «Удаление аккаунта» - удаление аккаунта пользователя;
* «Посмотреть ФИО и логин» - появляется ФИО и логин пользователя.



Рисунок 5.8 – пользовательские настройки

Кнопка ««Изменить имя», выпадает окно с полем, куда вводиться новое ФИО (Рис. 5.9).

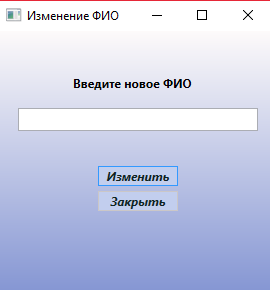


Рисунок 5.9 – изменение ФИО

Кнопка «Изменить логин», выпадает окно с полем для ввода нового логина (Рис. 5.10).

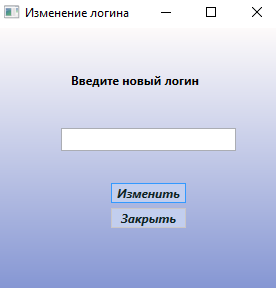


Рисунок 5.10 – изменение логина

Кнопка «Изменить пароль», выпадает с вводом старого пароля и нового (Рис.5.11).

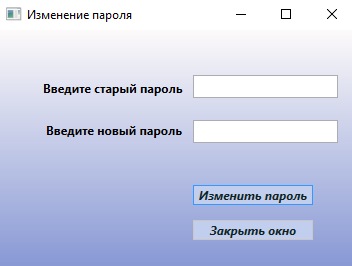


Рисунок 5.11 – изменение пароля

Кнопка «Удаление аккаунта», поле для подтверждения пароля (Рис. 5.12) и правильного ввода и нажатия на кнопку удалить, открывается окно, изображенное на рисунке 5.1.

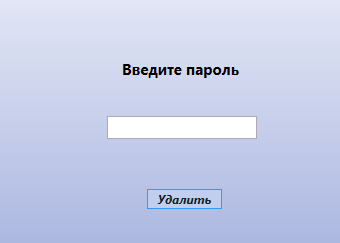


Рисунок 5.12 – удаление аккаунта

Кнопка «Забронированные билеты» показывает таблицу с забронированными билетами (Рис 5.13).

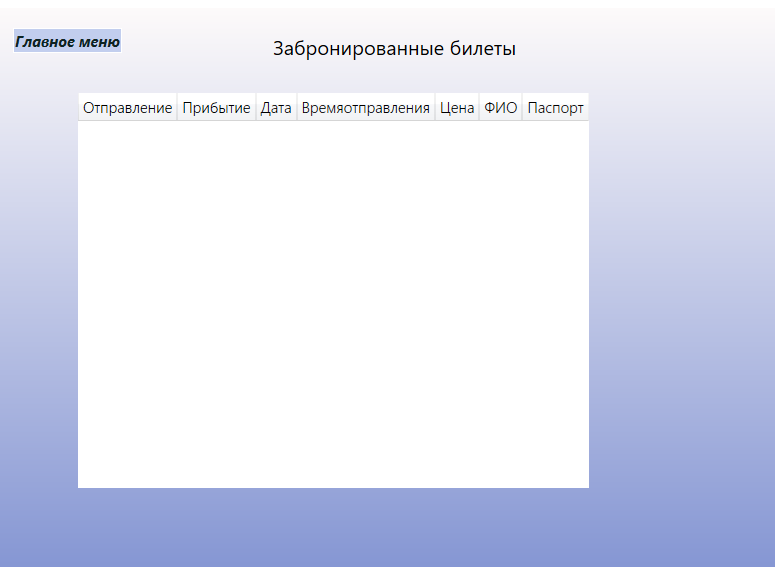


Рисунок 5.13 – забронированные билеты

# **Тестирование**

Объектом испытания является программа «Автовокзал».

Испытания проводятся с целью проверки правильности функционирования программы и выявления имеющих недостатков.

Приложение работает в среде Windows 10.

Программа разработана на языке C#.

Для проведения испытания необходимо запустить приложение и проверить работу всех вариантов использования системы. Разработанное приложение может использоваться на персональном компьютере с операционной системой Windows.

Тестирование проводится на допустимые значения и на правильно выбранные значение в выпадающем меню.

Первая проверка на вводимы данные будет происходить при регистрации (Рис. 5.1, 5.2, 5.3, 5.4, 5.5).

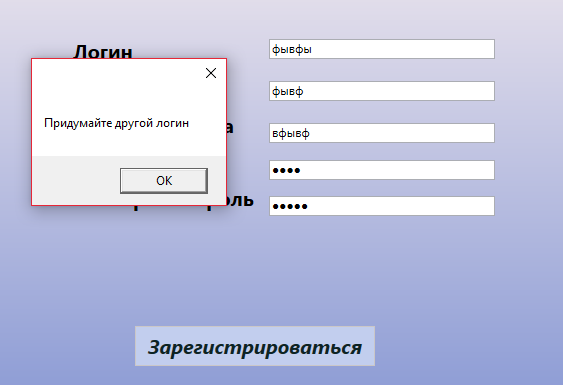


Рисунок 5.1 – проверка на правильно введенный логин

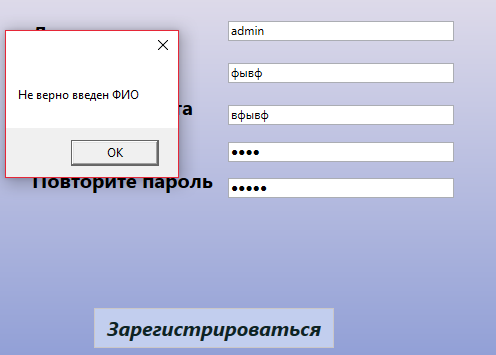


Рисунок 5.2 – проверка на правильно введеный ФИО

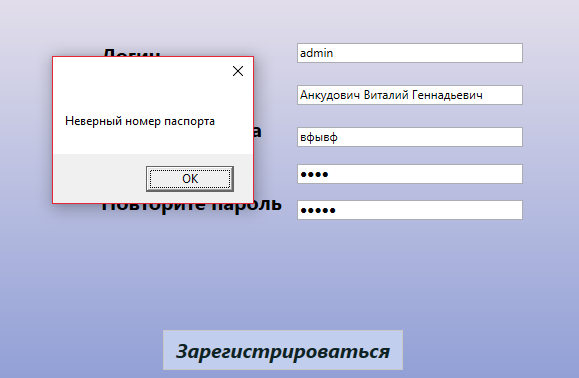


Рисунок 5.3 – проверка на правильно введенный номер паспорта

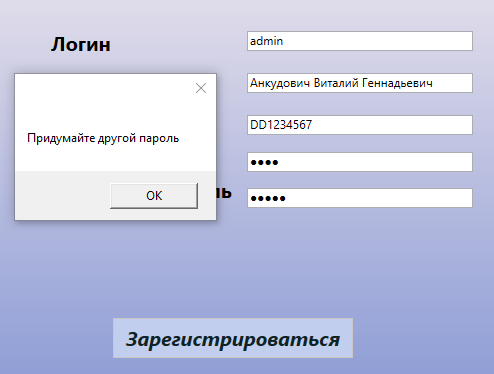


Рисунок 5.4 – проверка на правильно введенный пароль

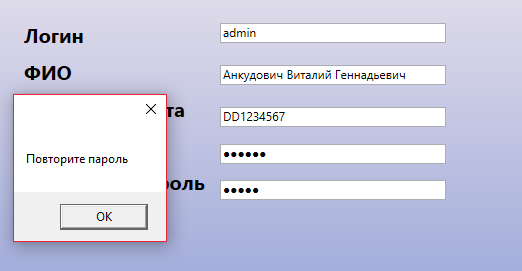


Рисунок 5.5 – проверка на соответствие паролей

Вторая проверка на авторизацию (Рис. 5.6).

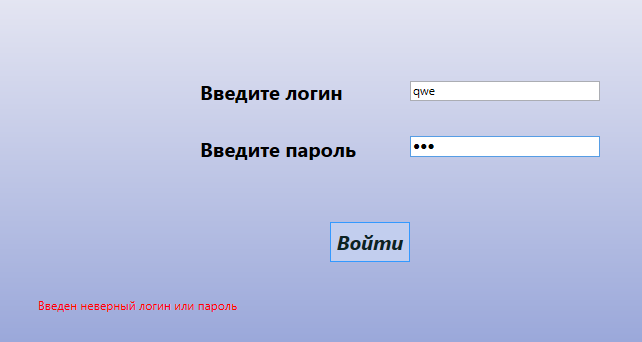


Рисунок 5.6 – проверка на подлинного пользователя и

неправильно введенный логин или пароль

Третья проверка на выбор пункта прибытия (Рис. 5.7).

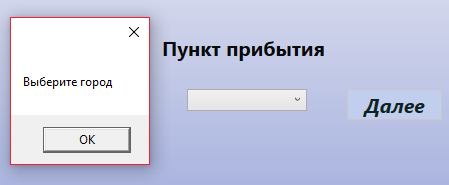


Рисунок 5.7 – проверка на 1-й этап бронирования билетов

Четвертая проверка на выбор дня и времени (Рис. 5.8).

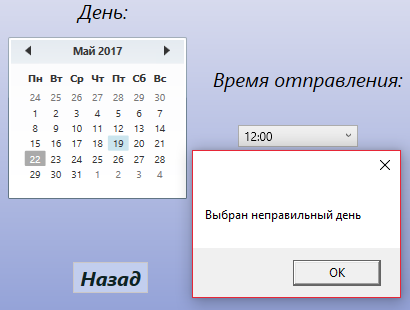


Рисунок 5.8 – проверка бронирования билетов

Данное программное средство тестировалось вручную, были выполнены все технические моменты, предусмотренные реализацией. В ходе ручного тестирования ошибок не выявлено.

# **Заключение**

В данном курсовом проекте было разработано программное средство «Автовокзал» при помощи C# WPF. При разработке данного программного средства использовалась базы данных MS Sql. Данное приложение соответствует поставленной задаче и отвечает всем требованиям, необходимым для пользования данным программным средством «Автовокзал»

В ходе выполнения курсового проекта мной были разработаны:

* Приложение, позволяющее пользователям бронировать билеты.
* База данных, а также программный код для взаимодействия с базой данных.

В результате приложение обладает следующими особенностями:

* Изменение данных пользователя.
* Процедура бронирование билетов.
* Предоставление удобного интерфейс пользования.
* Регистрация и авторизация пользователей.

# **Список используемых источников**

1. Пацей, Н.В. Курс лекций по языку программирования C# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с.

2. Руководство по Visual Studio [Электронный ресурс]/ msdn.microsoft.com – Режим доступа: http://msdn.microsoft.com. Дата доступа 08.03.2016.

3. <http://www.realcoding.net/article/view/2611>

4. https://habrahabr.ru/post/52173/

# **Приложение**

