Утверждаю

Руководитель предприятия

\_\_\_\_\_\_\_\_

(Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, печать предприятия)

« \_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Учреждение образования

«Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

ОТЧЕТ

о производственной технологической практике

на \_ЧУО «БЦПКРРиС «Большая Медведица» в период с «27» июня 2022 г. по «22» июля 2022 г.\_

(наименование предприятия, сроки практики)

Исполнитель

студент 3 курса \_\_1\_\_ группы Божко Д. В. \_\_\_

(подпись, дата) (Ф.И.О.)

Руководитель практики

от предприятия

\_\_Чуль А. И.\_\_\_\_\_

(должность, печать предприятия) (подпись, дата) (Ф.И.О.)

Руководитель практики

от университета

старший преподаватель\_\_\_ \_\_Берников В. О.\_ \_\_\_\_

(должность, уч. звание) (подпись, дата) (Ф.И.О.)

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Минск 2022

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc109302918)

[1 Описание предприятия ЧУО «БЦПКРРиС «Большая Медведица» 4](#_Toc109302919)

[1.1 История создания компании и основные виды деятельности 4](#_Toc109302920)

[1.2 Организационная структура ЧУО «БЦПКРРиС «Большая Медведица» 5](#_Toc109302921)

[1.3 Информационные технологии ЧУО «БЦПКРРиС «Большая Медведица» 5](#_Toc109302922)

[2 Описание используемых технологий 7](#_Toc109302923)

[2.1 Microsoft Visual Studio 7](#_Toc109302924)

[2.2 Microsoft SQL Server 7](#_Toc109302925)

[3 Индивидуальное задание 8](#_Toc109302926)

[3.1 Постановка задачи 8](#_Toc109302927)

[3.2 Разработка функциональных требований 8](#_Toc109302928)

[3.3 Проектирование базы данных 8](#_Toc109302929)

[3.4 Проектирование и реализация классов приложения 10](#_Toc109302930)

[3.5 Реализация программного средства 11](#_Toc109302931)

[3.6 Тестирование программного средства 16](#_Toc109302932)

[ЗАКЛЮЧЕНИЕ 21](#_Toc109302933)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 22](#_Toc109302934)

# ВВЕДЕНИЕ

В данный момент мир информационных технологий характеризуется быстрым ростом количества различной информации. Развивается множество языков программирования, различные области программирования. Появляется множество новых компаний, некоторые компании становятся компаниями-лидерами, а другие не связаны с технологическим сектором, но применяют информационные технологии для улучшения производства и оптимизации своих услуг. Одним из таких предприятий является ЧУО «БЦПКРРиС «Большая Медведица».

Прохождение производственной практики является важным этапом обучения. Это специфический вид учебного процесса, в ходе которого осуществляется связь обучения с производством, где студент сам может увидеть производство и что-то попробовать сделать, а также выполнить свое индивидуальное задание. Практика должна способствовать формированию у студентов профессиональных практических знаний, умений и навыков, необходимых для будущей работы на предприятии.

Прохождение практики осуществлялось в департаменте одного из филиалов ЧУО «БЦПКРРиС «Большая Медведица».

Целью производственной практики является ознакомление студентов с реальными условиями работы на предприятии. Проходя практику, студенты имеют возможность выполнить задания и показать себя, и, кроме того, успешно справившиеся с заданием студенты могут быть приглашены на работу в данное предприятие после прохождения производственной практики.

В этот период было применено на практике ПО, которое используется сотрудниками организации. Было выполнено индивидуальное задание по разработке заданного приложения.

Также при прохождении практики были поставлены следующие задачи:

* ознакомиться с историей создания и деятельностью предприятия;
* ознакомиться с организационной структурой предприятия;
* ознакомиться с информационными технологиями, используемыми на предприятии;
* ознакомиться с методами информационной безопасности на предприятии;
* сформировать план работ на время производственной практики;
* проанализировать и обобщить полученную информацию;
* результат выполнения включить в отчет по практике.

# 1 Описание предприятия ЧУО «БЦПКРРиС «Большая Медведица»

## История создания компании и основные виды деятельности

Учреждение представляет собой предприятие со специальной материальной базой, которое осуществляет теоретическую и практическую подготовку водителей категории «А», «B», переподготовку с категории «B» или «C» на «А», с «B» или «C» на «D», с «B» на «BE», с «C» на «CE», с «В» на «С» [1].

Автошкола была создана в 1997 году. За это время на предприятии сложился прекрасный коллектив преподавателей и мастеров производственного обучения управлению механическим транспортным средством. Занятия по теории проводятся в кабинетах, оборудованных согласно требованиям программы подготовки и переподготовки водителей. Практическое вождение осуществляется на автомобилях марки Рено Логан, а также грузовых автомобилях ЗИЛ, ГАЗ, автобусах ПАЗ 32050, МАЗ 2662 00, Мерседес Спринтер, мотоцикл ЗиД. Ученики сами составляют удобный для них график вождения. Каждому курсанту предоставляется рабочее место для тренировки и контроля знаний по ПДД, основам управления транспортным средством и безопасности движения, устройству и техническому обслуживанию автомобилей, управлению автомобилем.

Автошкола оказывает дополнительные услуги по вождению автомобиля, и предоставляет компьютеры для самостоятельной подготовки. Также в организации имеется собственный автодром и помещение для проведения практических занятий. Также в организации имеется собственная автостоянка и шиномонтаж.

Все дополнительные услуги:

* подготовка водителей для работы на таксомоторных перевозках пассажиров;
* подготовка водителей для эксплуатации автомобилей, работающих на газовом топливе;
* подготовка и повышение квалификации водителей для осуществления перевозки опасных грузов;
* подготовка и переподготовка водителей для международных перевозок пассажиров и грузов;
* подготовка по специальности «Аккумуляторщик»;
* подготовка по специальности «Водитель погрузчика».

## Организационная структура ЧУО «БЦПКРРиС «Большая Медведица»

«Большая Медведица» является предприятием с определенной материальной и технической базой, для поддержки которой необходим определенный кадровый набор. Организационная структура предприятия представлена на рисунке 1.1.



Рисунок 1.1 – Организационная структура предприятия

## 1.3 Информационные технологии ЧУО «БЦПКРРиС «Большая Медведица»

Каждому сотруднику компании предоставляется стандартный пакет программ, а также дополнительный в соответствии с занимаемой им должностью. Стандартный пакет программ включает:

* *Skype for Business* для ведения деловой переписки, проведения совещаний и переговоров. Данное ПО также позволяет компании отказаться от использования стационарных телефонов.
* *The Bat!* – почтовый сервис, который также используется для ведения деловой переписки.
* Пакет *Microsoft Office* для работы с документами, составления презентаций и др.
* *Total Commander* – альтернативный менеджер файлов.
* *WinRAR* – инструмент для работы с архивами.
* *1C:Бухгалтерия* – набор программ для бухгалерского учета.

В дополнительный пакет программ входит ПО, которое необходимо сотруднику для непосредственного выполнения своих обязанностей. Это могут быть среды интегрированной разработки и др.

В зависимости от технологий, используемых на проекте, специалисту может предоставляться следующее ПО:

* *Microsoft SQL Server Management Studio* – инструмент для написания *SQL*-запросов к базе данных *Microsoft SQL Server* и выполнения данных запросов.
* *Oracle SQL Developer* – инструмент для написания *SQL*-запросов к базе данных Oracle и выполнения данных запросов.
* *PyCharm IDE*– интегрированная среда разработки на языке *Python*.
* *Microsoft Visual Studio* – среда для разработки на других различных языках (например, C#).

При необходимости сотрудник может скачать лицензированное ПО со внутреннего сетевого хранилища.

# Описание используемых технологий

## Microsoft Visual Studio

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов [2]. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight. После покупки компании Xamarin корпорацией Microsoft появилась возможность разработки IOS и Android программ.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

## Microsoft SQL Server

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft [3]. 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.

# Индивидуальное задание

## Постановка задачи

Целью данной курсовой работы будет являться разработка программного средства «Автошкола» для ЧУО «БЦПКРРиС «Большая Медведица». Реализация проекта будет выполняться на языке программирования C# с использованием базы данных SQL Server.

## Разработка функциональных требований

Разработаем примерный список функций, которые сможет выполнять тот или иной пользователь в зависимости от его роли.

**Администратор.** Это пользователь, функционал которого обладает наиболее широкими возможностями. Его основная задача — это управление всеми областями данных приложения. В данном случае администратор будет иметь возможность создавать, удалять, изменять и просматривать в удобном табличном представлении данные о таких областях как группы, автомобили, автодромы, школы, учителя, инструкторы, а также иметь доступ к данным об аккаунтах студентов и преподавателей. При необходимости администратор может выполнить экспорт любых данных в Excel.

**Преподаватель.** Это один из основных пользователей приложения. В основной функционал преподавателя должны входить функции для управления редактирования информации о группах и выставления студентам зачетов. Также будет предусмотрен экспорт данных о студентах в Excel.

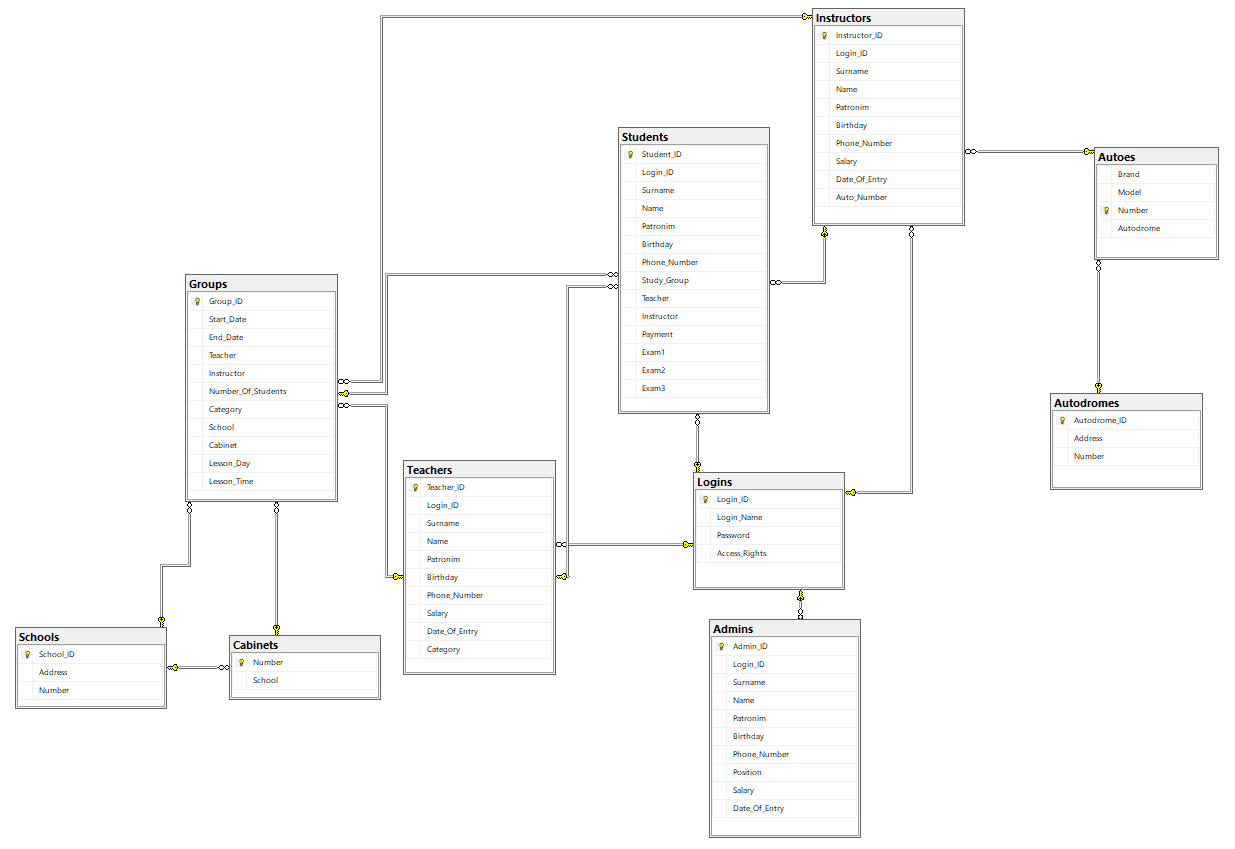
**Инструктор.** Также один из основных пользователей приложения. Будет выполнять схожие с преподавателем функции, однако больше специализироваться будет на автомобилях: ему будет разрешено редактировать данные о машинах и просматривать информацию по группе. Также будет предусмотрен экспорт данных об автомобилях в Excel.

**Студент.** Это пользователь с наименьшим функционалом. Для данной категории пользователей предусмотрена лишь возможность просмотра данных о группе, студентах и автомобилях в табличном виде.

В данном приложении не предусмотрена регистрация для каждого пользователя лично. Эту роль будет выполнять администратор путем создания аккаунтов для студентов, преподавателей и инструкторов.

## Проектирование базы данных

В данном индивидуальном задании будет использована база данных SQL Server. Разработка всего приложения начинается именно с проектирования и реализации базы данных. В процессе написания логики взаимодействия таблиц между собой база данных может потерпеть некоторые изменения, такие как изменение ключевых полей, изменение типов данных полей, а также абсолютное их удаление и добавление новых. Прежде всего будут созданы и заполнены основные таблицы. Итоговым решением стала база данных, логическая схема которой представлена на рисунке 3.1.

Рисунок 3.1 –Логическая схема базы данных

База данных для данной курсовой работы состоит из 10 таблиц, взаимосвязанных между собой.

**LOGINS.** Содержит всю необходимую информацию для входа в приложение, а именно: логин, пароль и права доступа.

**ADMINS.** Содержит всю информацию об администраторском составе автошколы. Здесь будут находиться директора, бухгалтеры, секретари, заместители и прочие управляющие лица. Поля будут описывать данные о персоне: имя, дату рождения, должность и др.

**INSTRUCTORS**, **TEACHERS** и **STUDENTS**. Таблицы, отвечающие за инструкторов, преподавателей и студентов соответственно. Данные таблицы очень схожи и имеют набор полей, способных описать самые важные и необходимые данные о персоне: имя, дату рождения, телефонный номер, принадлежность к группе и прочее.

**SCHOOLS**, **CABINETS** и **AUTODROMES**. Таблицы, отвечающие за места, где проходит обучение. Являются небольшими и не особо важными. Содержат имена, адреса и уникальные идентификаторы.

**AUTOES.** Таблица, содержащая информацию об автомобилях, принадлежащих автошколе и на которых происходит обучение. Содержит информацию о марке автомобиля, его модели и идентификационном номере.

**GROUPS**. Важная таблица, которая содержит информацию о группах, в которых обучаются студенты. Набор полей показывает все данные о группе: номер, даты начала и конца обучения, привязанных к группе преподавателей, категорию, на которую идет обучение, число студентов и др.

## Проектирование и реализация классов приложения

В данном индивидуальном задании мы будем использовать технологию WPF - аналог WinForms, систему для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML [4]. Большинство классов данного приложения были сгенерированы автоматически при помощи технологии Entity Framework, основываясь на модели базы данных, которую мы используем. Однако в приложении существует еще некоторое количество классов, которые выделились в свои отдельные группы.

Первая категория – классы, которые формируют окна приложения. Данные классы являются самым важным составляющим приложения. В них задается вся логика взаимодействия соответствующих им xaml-разметок и пользователя. В них описываются обработчики событий для кнопок и других событий, задаются контексты и источники данных.

Вторая категория – классы-модели. Они содержат инициализацию классов элементов, с которыми мы взаимодействуем в процессе выполнения приложения. Хранятся они в папке classes нашего проекта (рисунок 3.2).

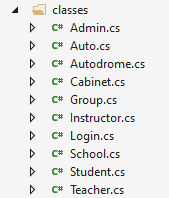


Рисунок 3.2 –Папка classes

Третья категория – вспомогательные классы. Необходимы для того, чтобы мы могли реализовывать какие-либо дополнительные функции, которые не смогли бы реализовать с имеющимися классами. Здесь находится только один класс StaticClassForLogin, которой необходим для корректного отображения имени пользователя по его логину.

## Реализация программного средства

Как уже упоминалось ранее, данное программное средство реализовано с помощи технологии Windows Presentation Foundation. Это система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем. Для данного приложения она подходит лучше всего.

Реализована страничная система с помощью класса System.Windows.Controls.Page, так как именно она позволяет обеспечить в нашем приложении удобную навигацию.

Перед тем, как начать программировать все страницы, необходимо задать источник для стилей (в нашем случае это шрифты и дизайн кнопок). Все стили прописаны в файле StyleDictionary.xaml, листинг кода которого представлен в приложении Б данной записки.

На начальной странице MainWindow.xaml будем использовать объект Frame, который будет находиться внутри данного окна, что позволит нам сразу переключиться на страничную систему (xaml-код окна MainWindow.xaml представлен на листинге 3.1).

|  |
| --- |
| <Window x:Class="BigDipper.MainWindow"  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:BigDipper"  mc:Ignorable="d"  Title="Автошкола 'Большая Медведица'" Height="700" Width="1200">  <Grid>  <Frame x:Name="MainPageFrame" NavigationUIVisibility="Hidden" Source="pages/LoginPage.xaml"/>  </Grid>  </Window> |

Листинг 3.1 –MainWindow.xaml, xaml-код

Как видно из кода выше, при запуске приложения (а именно MainWindow.xaml запускается после запуска приложения) нас перенаправляет на страницу LoginPage.xaml. Это страница авторизации пользователя. На ней пользователь сможет вводить свои логин и пароль и авторизироваться в само приложение.

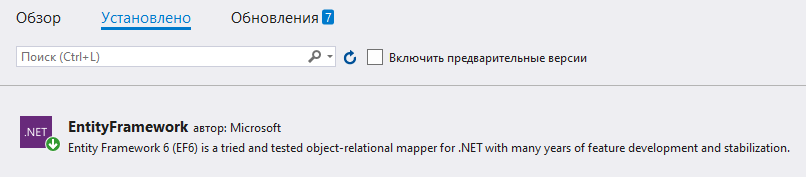
Небольшим дополнительным элементом этой и некоторых последующих страниц является пользовательский элемент Clock. Проще говоря, это часы, которые обновляются с интервалом в секунду и показывают пользователю текущую дату и время.

Первое, что происходит после ввода правильных логина и пароля и нажатия на кнопку «Войти», - отправка администратору программного средства электронного сообщения, которое приходит ему на почту. Оно содержит краткую информацию о пользователе, который только что авторизировался: его логин, ФИО, уровень доступа и время доступа. Код функции, которая реализовывает данную возможность, представлен на листинге 3.2.

|  |
| --- |
| private static async Task SendEmailAsync(string login, string access, string surname, string name, string patronim) // в функцию идет передача логина авторизированного пользователя, его ФИО и уровень доступа  {  try  {  MailAddress from = new MailAddress("bigdipperbstu@gmail.com", "Администратор автошколы 'Большая медведица'"); // адрес отправителя  MailAddress to = new MailAddress("denisxelizait@gmail.com"); // адрес получателя  MailMessage m = new MailMessage(from, to);  string accessrus = null;  if (access == "admin") accessrus = "администратор";  if (access == "student") accessrus = "студент";  if (access == "instructor") accessrus = "инструктор";  if (access == "teacher") accessrus = "преподаватель";  m.Subject = "Авторизация в систему автошкола 'Большая медведица'"; // тема письма  m.Body = String.Format("Произошла авторизация в программное средство для управления базой данных автошколы 'Большая медведица'\n" + // текст письма  "Логин пользователя: {0}\n" +  "Уровень доступа: {1}\n" +  "Фамилия: {2}\n" +  "Имя: {3}\n" +  "Отчество: {4}\n" +  "Время доступа: {5}\n\n" +  "Программное средство\nАвтошкола 'Большая медведица'", login, accessrus, surname, name, patronim, DateTime.Now);  SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587); // smtp-сервер для отправки  smtp.Credentials = new NetworkCredential("bigdipperbstu@gmail.com", "1234QwerAsdf"); // пароль от почты отправителя  smtp.EnableSsl = true;  await smtp.SendMailAsync(m);  }  catch  {  MessageBox.Show("Не удалось отправить сообщение на почту");  }  }  } |

Листинг 3.2 –Функция отправки электронного сообщения на почту

Далее происходит сам процесс авторизации. Для того, чтобы обращаться к базе данных, где содержится таблица с логинами и паролями, используем технологию Entity Framework. Это определенная технология на базе фреймворка ADO.NET для работы с данными. Подключили мы ее к проекту с помощью службы управления пакетами NuGet (рис. 3.3).

Рисунок 3.3 –Подключение Entity Framework

Для взаимодействия с базой данных через Entity Framework нам нужен контекст данных, поэтому добавим класс EntityContext. В своем конструкторе он передает в конструктор базового класса название строки подключения из файла App.config. Также в контексте данных определяется свойство по типу DbSet - через него мы будем взаимодействовать с таблицей, которая хранит объекты таблиц из базы данных. Код класса EntityContext представлен на листинге 3.3.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Data.Entity;  using BigDipper.classes;  namespace BigDipper  {  public class EntityContext : DbContext  {  public EntityContext() : base("DefaultConnection")  {  }  public DbSet<Admin> Admins { get; set; }  public DbSet<Auto> Autoes { get; set; }  public DbSet<Autodrome> Autodromes { get; set; }  public DbSet<Cabinet> Cabinets { get; set; }  public DbSet<Group> Groups { get; set; }  public DbSet<Instructor> Instructors { get; set; }  public DbSet<Login> Logins { get; set; }  public DbSet<School> Schools { get; set; }  public DbSet<Student> Students { get; set; }  public DbSet<Teacher> Teachers { get; set; }  }  } |

Листинг 3.3 –EntityContext.cs

Далее необходимо определить классы-модели, которые будем использовать для инициализации объектов и взаимодействия с ними.

После мы обращаемся к уровню доступа авторизующегося пользователя, дабы понять, на какую страницу его перенаправлять. Также нам на странице, на которую попадет пользователь, будет необходимо отображать его имя. Для этого в поле SignInLogin статического класса StaticClassForLogin будем записывать имя пользователя. Пример авторизации для пользователя с уровнем доступа «Администратор» представлен в листинге 3.4.

|  |
| --- |
| if (access == "admin")  {  var login = from s in db.Admins  join sa in db.Logins on s.Login\_ID equals sa.Login\_ID  where s.Login\_ID == loginid  select s;  var user = login.First();  StaticClassForLogin.SignInLogin = user.Name;  SendEmailAsync(rightlogin.Login\_Name, access, user.Surname, user.Name, user.Patronim).GetAwaiter();  this.NavigationService.Navigate(new Uri("pages/MainAdmin.xaml", UriKind.Relative));  } |

Листинг 3.4 –Авторизация для пользователя «Администратор»

В итоге мы переходим на одну из четырех главных страниц в зависимости от нашего уровня доступа: администратор, преподаватель, инструктор и студент. Главная страница будет содержать кнопки для перехода на те разделы, которые доступны для пользователя, для выхода из аккаунта, а также часы, которые были описаны ранее.

Далее в качестве примера будем рассматривать страницу «Студенты» пользователя «Администратор», так как на ней имеется самое большое количество интерактивных элементов и в большинстве своем для каждой страницы функционал будет похожий. Во-первых, необходимо было в xaml-коде реализовать технологию DataGrid, с помощью которой мы сможем отображать пользователю таблицы.

Для более понятного пользователю отображения поля с зачетами для студентов, которые содержат значения passed или not passed, будем использовать цветовую раскраску с помощью триггеров. Пример xaml-кода для ячейки DataGrid-a, раскрашенной в определенный цвет, представлен в листинге 3.5.

|  |
| --- |
| <DataGridTextColumn Header="Зачет 1" Binding="{Binding Exam1}" Width="100">  <DataGridTextColumn.ElementStyle>  <Style TargetType="{x:Type TextBlock}">  <Style.Triggers>  <Trigger Property="Text" Value="passed">  <Setter Property="Background" Value="LightGreen"/>  </Trigger>  <Trigger Property="Text" Value="not passed">  <Setter Property="Background" Value="LightCoral"/>  </Trigger>  </Style.Triggers>  </Style>  </DataGridTextColumn.ElementStyle>  </DataGridTextColumn> |

Листинг 3.5 –Окрашивание ячейка DataGrid-a в определенный цвет

Очевидно, что необходимо загрузить данные с базы данных в наш DataGrid. Для этого используем свойство ItemSource нашего грида (листинг 3.6).

|  |
| --- |
| db.Students.Load();  AdminStudentsGrid.ItemsSource = db.Students.Local.ToBindingList(); |

Листинг 3.6 –Использование свойства ItemsSource

Для того, чтобы использовать больше возможностей WPF, такие как, например, MaskedTextBox, используем стороннюю библиотеку Extended WPF Toolkit, которую будем устанавливать через NuGet (рис. 3.4).

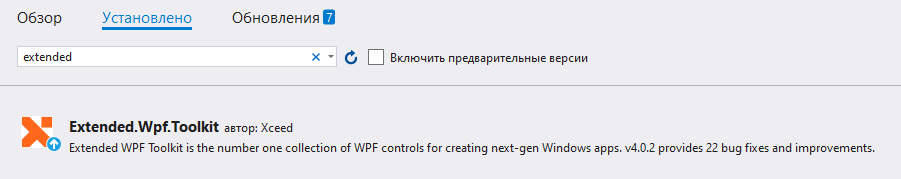


Рисунок 3.4 –Подключение Extended WPF Toolkit

Затем будем реализовывать различные методы взаимодействия с таблицей, такие как обновление, удаление элемента, добавление элемента в базу данных и фильтрация.

Также будем использовать функцию экспорта текущей таблицы в Excel-документ. Для этого необходимо подключить стороннюю библиотеку Microsoft Office Interop Excel, которая позволит нам использовать взаимодействие с приложением Microsoft Excel (рис. 3.5).

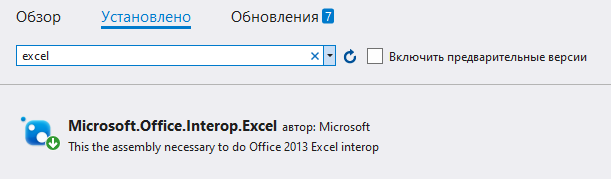


Рисунок 3.5 –Подключение Microsoft Office Interop Excel

В итоге мы получаем готовое программное средство для автошколы. Остается лишь провести тестирование на предмет поиска возможных ошибок.

## Тестирование программного средства

Тестирование программы является важнейшим этапом в процессе разработки. Для этого привлекаются различные люди и конкретно направленные специалисты. Так как в этой программе присутствует небольшое количество потенциально опасных мест, тестирование будет проводиться собственными силами.

Данное приложение будет тестироваться по нескольким принципам: проверка на нулевые примеры, проверка на исключительные ситуации, проверка на неправильные типы данных.

Приложение от многочисленных вылетов помогает спасти конструкция try-catch, благодаря которой даже после очевидных ошибок программное средство может находиться в рабочем состоянии.

Обработку ошибок начнем со страницы входа в приложение. Очевидно, что пользователь может ввести неверные логин или пароль. В такой ситуации будет выведено сообщение об ошибке (рис. 3.6), но приложение будет продолжать работать. Нулевые данные будут также обрабатываться с помощью сообщений об ошибке, но с другим содержимым (рис. 3.7).

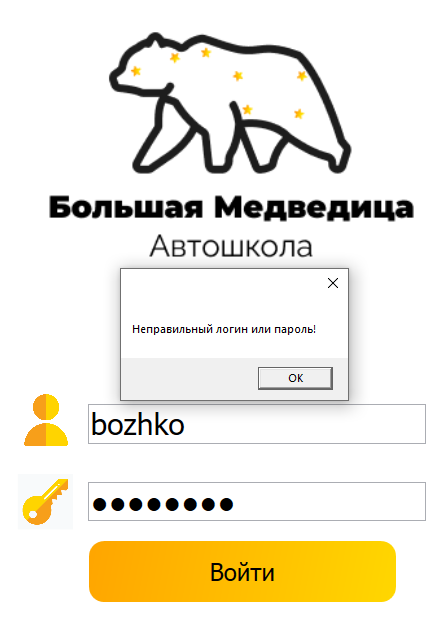


Рисунок 3.6 –Ошибка при аутентификации

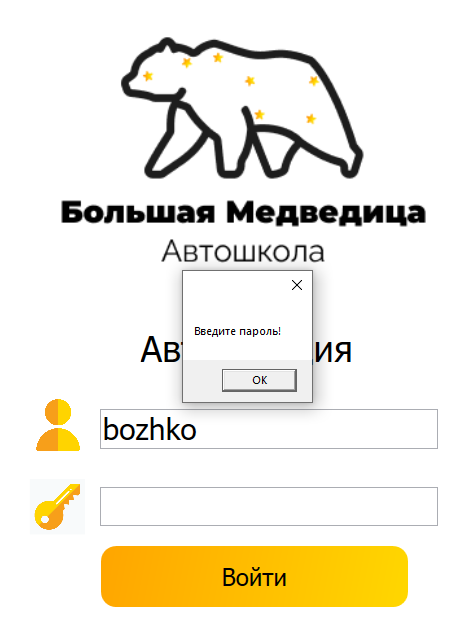


Рисунок 3.7 –Ошибка отсутствия пароля

Следующим местом, где может ошибиться пользователь, является форма добавления объекта в базу данных. Например, пользователь может ничего не ввести в какое-нибудь важное поле или ввести числовое значение в поле, которое требует только текст. Эта проблема актуальна для каждого пользователя, за исключением студента, так как у него нет возможности редактировать базу данных. В случае ввода пустого значения будет выведено сообщение об ошибке (рис. 3.8), а в случае ввода неверного значения по формату будет выведено такое же сообщение, но с другим содержимым (рис. 3.9). В обоих случаях поле, в котором была допущена ошибка, будет выделяться цветом.

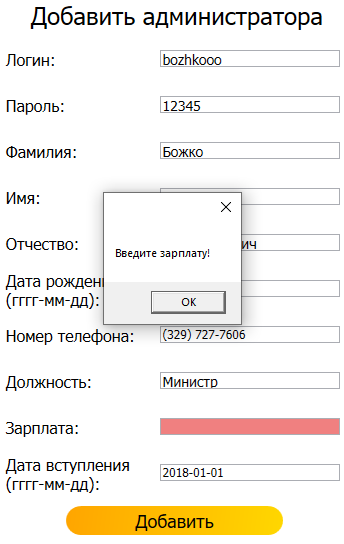


Рисунок 3.8 –Ошибка пустого поля

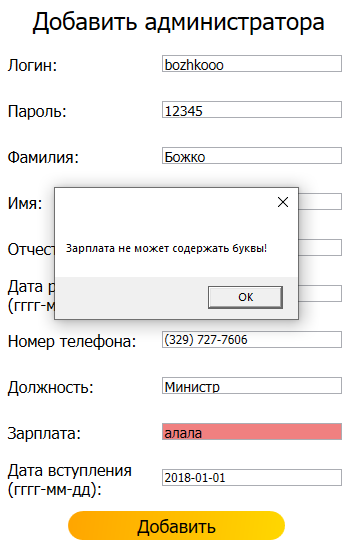


Рисунок 3.9 –Ошибка неверного формата данных

Приложение подразумевает, что для того, чтобы администратор мог добавить определенный элемент, который ссылается на другой, то пользователь должен точно вводить данные, чтобы приложение не вылетело. Этот момент, безусловно, нужно предусмотреть, то есть предостеречь пользователя от возможных ошибок. Для этого будем использовать вывод сообщения об ошибке и цветовое выделение (рис. 3.10).

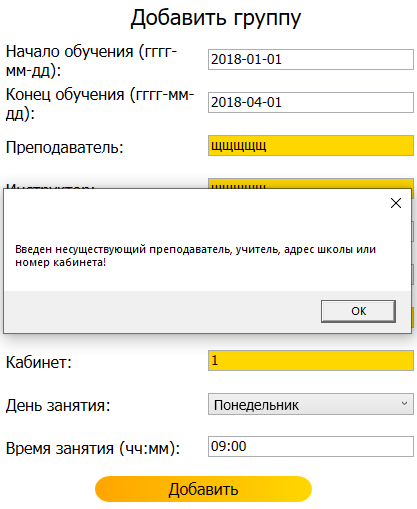


Рисунок 3.10 –Ошибка неверного ссылочного значения

В приложении предусмотрена возможность экспорта данных из таблицы в Excel. Конечно, здесь тоже могут возникнуть технические ошибки. Поэтому будем выводить пользователю сообщение об ошибке в случае неудачного экспорта (рис. 3.11).

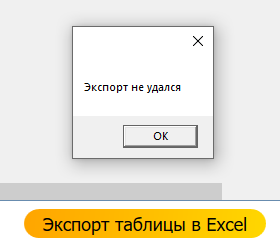


Рисунок 3.11 –Ошибка экспорта в Excel

Так как пользователь может редактировать имеющиеся у него данные непосредственно через DataGrid, то может возникнуть такая ситуация, что он не введет правильные данные в ячейку. В таком случае он также получит предупреждение (рис. 3.12).

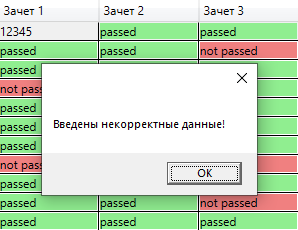


Рисунок 3.12 –Ошибка неверного ввода данных в DataGrid

# ЗАКЛЮЧЕНИЕ

ЧУО «Большая Медведица» - одна из крупнейших организаций по переподготовке и обучению специалистов в Республике Беларусь. Деятельность учреждения осуществляется по многим направлениям:

* подготовка водителей категории «А»;
* подготовка водителей категории «В»;
* переподготовка водителей с категории «В» на категорию «С»;
* переподготовка водителей с категории «В» на категорию «Е»;
* переподготовка водителей с категории «В» на категорию «D»;
* подготовка водителей для работы на таксомоторных перевозках пассажиров;
* подготовка водителей для эксплуатации автомобилей, работающих на газовом топливе;
* подготовка и повышение квалификации водителей для осуществления перевозки опасных грузов;
* подготовка и переподготовка водителей для международных перевозок пассажиров и грузов;
* подготовка по специальности «Аккумуляторщик»;
* подготовка по специальности «Водитель погрузчика».

Во время прохождения производственной практики подробно изучена деятельность компании. Углубленно изучены методики разработки приложений, изучено ПО, используемое сотрудниками департамента для выполнения своих обязанностей.

В качестве индивидуального задания разработано приложение «Автошкола».

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Официальный сайт ЧУО «Большая Медведица» [Электронный ресурс]. Режим доступа: http://medvedica.by/company.php#ru. Дата доступа: 19.07.2022.
2. Visual Studio: IDE и редактор кода [Электронный ресурс]. Режим доступа: https://visualstudio.microsoft.com/ru/. Дата доступа: 19.07.2022.
3. SQL Server Downloads [Электронный ресурс]. Режим доступа: https://www.microsoft.com/en-us/sql-server/sql-server-downloads. Дата доступа: 19.07.2022.
4. Windows Presentation Foundation – Википедия [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Windows\_Presentation\_Foundation. Дата доступа: 19.07.2022.