**УО «Белорусский государственный технологический университет»**

Факультет **информационных технологий**

Кафедра **программной инженерии**

Специальность **1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОЙ РАБОТЫ**

**по дисциплине «**Объектно-ориентированное программирование»

**тема «**Программное средство “Автошкола” »

|  |  |  |
| --- | --- | --- |
| **Исполнитель** |  | |
| Студент 2 курса группы1 | подпись, дата | Е.В. Гончаревич  инициалы и фамилия |
|  |  |  |
| **Руководитель** |  |  |
| Ассистент | подпись, дата | А.В. Годун  инициалы и фамилия |

|  |  |
| --- | --- |
| **Курсовая работа защищена с оценкой** | **\_\_\_\_\_\_\_\_\_\_** |
| **Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  (подпись) | А.В. Годун  инициалы и фамилия |

**Минск 2022**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра программной инженерии

Утверждаю

Заведующий кафедрой ПИ

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ Н.В. Пацей

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

**ЗАДАНИЕ**

**к курсовому проектированию**

**по дисциплине** "Объектно-ориентированное программирование"

|  |  |
| --- | --- |
| Специальность: 1-40 05 01-03 Информационные системы и технологии  Студент: Гончаревич Е. В. | Группа:\_\_1\_\_ |
| **Тема: Программное средство «**Автошкола**»** | |

**1. Срок сдачи студентом законченной работы**: "20 мая 2022 г."

**2. Исходные данные к проекту:**

**2.1**. Функционально ПС поддерживает следующие функции:

* Вход от имени администратора/пользователя;
* Администрирование базы данных:
  + Создание и изменение профиля пользователя
  + Редактирование тестов, экзамена и т.д.
* Прохождение тестов и экзамена.
* Получение информации о ПДД.

**2.2.** При выполнении курсового проекта необходимо использовать принципы проектирования ООП. Приложение разрабатывается под ОС Windows и представляет собой настольное приложение (desktop). Отображение, бизнес логика должны быть максимально независимы друг от друга для возможности расширения. Язык разработки проекта – C#. Управление программой должно быть интуитивно понятным и удобным. При разработке использовать несколько наиболее подходящих шаблонов проектирования ПО.

**3. Содержание расчетно-пояснительной записки**

(перечень вопросов подлежащих разработке)

* Введение
* Постановка задачи и обзор литературы (алгоритмы решения, обзор прототипов, актуальность задачи)
* Проектирование архитектуры проекта (структура модулей, классов).
* Разработка функциональной модели и модели данных ПС (выполняемые функции)
* Тестирование
* Заключение
* Список используемых источников
* Приложения

**4. Форма представления выполненной курсовой работы:**

* + Теоретическая часть курсового проекта должны быть представлены в формате docx. Оформление записки должно быть согласно выданным правилам.
  + Листинги программы представляются частично в приложении.
  + Пояснительную записку, листинги, проект (инсталляцию проекта) необходимо загрузить на диск, указанный преподавателем.

***Календарный план***

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Введение | 19.02.2022 |  |
| 2 | Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования | 12.03.2022 |  |
| 3 | Анализ и проектирование архитектуры приложения (построение диаграмм, проектирование бизнес-слоя, представления и данных) | 26.03.2022 |  |
| 4 | Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса | 2.04.2022 |  |
| 5 | Кодирование программного средства | 23.04.2022 |  |
| 6 | Тестирования и отладка программного средства | 30.04.2022 |  |
| 7 | Оформление пояснительной записки | 7.05.2022 |  |
| 8 | Сдача проекта | 20.05.2022 |  |

**5. Дата выдачи задания \_\_\_\_**12.02.2022**\_\_\_\_**

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *А.В. Годун*

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата и подпись студента)

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ** 4

**1.** **Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования** 6

**1.1.** **Обзор и анализ аналогов** 6

**2.** **Анализ и проектирование архитектуры приложения** 11

**2.1.** **Анализ требований к программному средству** 11

**2.2.** **Разработка функциональных требований** 11

**3.** **Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса** 13

**3.1.** **Общая структура проекта** 13

**3.2.** **Проектирование базы данных** 15

**3.3.** **Проектирование и реализация классов приложения** 16

**3.4.** **Диаграмма последовательности** 17

**4.** **Реализация программного средства** 19

**5.** **Тестирование программного средства** 24

**6.** **Руководство пользователя** 29

**ЗАКЛЮЧЕНИЕ** 34

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 35

**ПРИЛОЖЕНИЕ А** 36

**ПРИЛОЖЕНИЕ Б** 38

**ПРИЛОЖЕНИЕ В** 39

**ПРИЛОЖЕНИЕ Г** 40

**ПРИЛОЖЕНИЕ Д** 42

**ПРИЛОЖЕНИЕ Е** 46

**ПРИЛОЖЕНИЕ Ж** 47

# **ВВЕДЕНИЕ**

21 век – век информационных технологий. Вполне очевидно, что одним из важнейших ресурсов в текущий период является информация. А значит, хранение, передача и обработка информации занимает одно из наиболее важных мест в нашей жизни.

Еще не так много лет назад все взаимодействие с информацией производилось вручную с использованием примитивных методов. Однако все больше и больше мелких и крупных предприятий, интернациональных корпораций, государственных, образовательных и медицинских учреждений внедряют современные информационные технологии в свои сферы для более качественной и быстрой работы с данными. Во многом это подразумевает создание программных средств с использованием баз данных.

Актуальной темой в наши дни является получение образования, так как ценность и востребованность высококвалифицированных кадров намного больше. Для улучшения качества образования учащихся автошколы было придумано электронное программное средство, позволяющее самостоятельно изучать правила ПДД и проходить пробные тесты и экзамены. Данное приложение может использоваться в рамках автошколы для самостоятельного обучения учащихся.

Целью данной курсовой работы будет являться разработка программного средства «Автошкола». Реализация проекта будет выполняться на языке программирования C# с использованием базы данных SQL Server[1]. Главными задачами работы будут обзор и анализ различных аналогов, разбор требований к программному средству, его реализация, тестирование и составление руководства пользователя.

# **Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования**

## **Обзор и анализ аналогов**

На данном этапе рассмотрим несколько примеров интерфейсов на основе существующих аналогов.

Рассмотрим два аналога, которые являются учебными программами Правил Дорожного Движения. В процессе разработки будет необходимо постараться объединить два интерфейса.

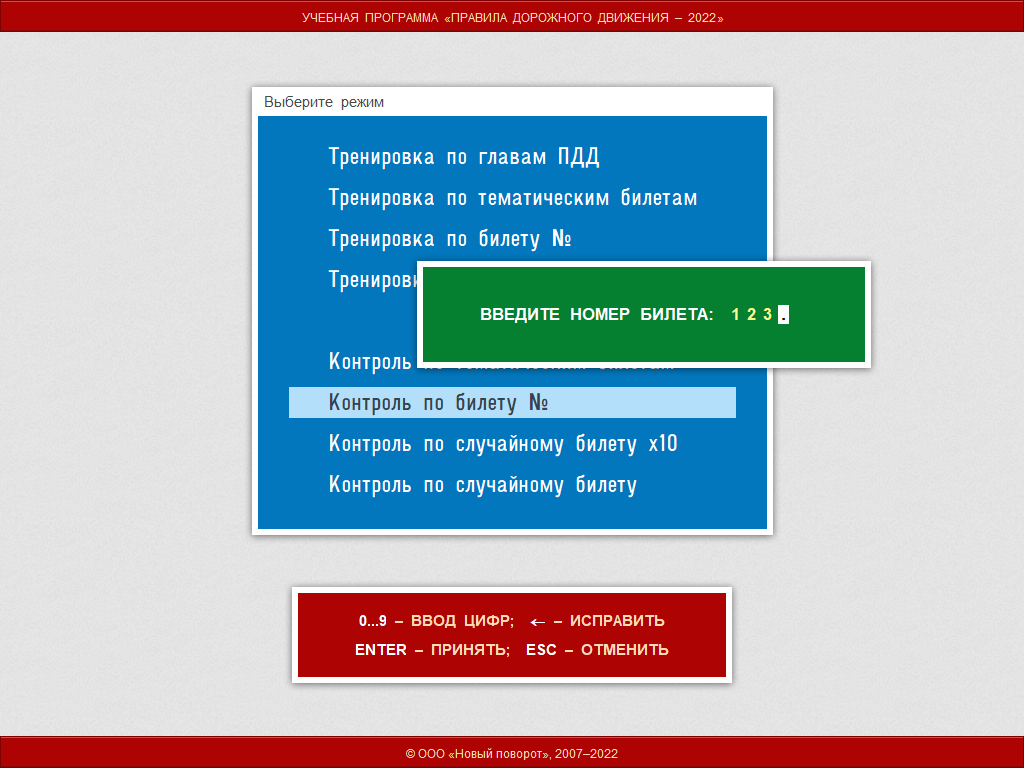
Первый пример – Учебная программа ПДД (рис. 1.1) [2].

Рисунок 1.1 –Главная страница примера №1

Интерфейс этого электронного приложения довольно простой и интуитивно понятный, не требует дополнительных инструкций по использованию. Проект выглядит немного устаревшим – обновление дизайна интерфейса пошло бы ему на пользу.

Приложение имеет обширный функционал, который позволяет проходить различные тесты вплоть до отдельного билета. Имеются удобные функции для выбора билетов.

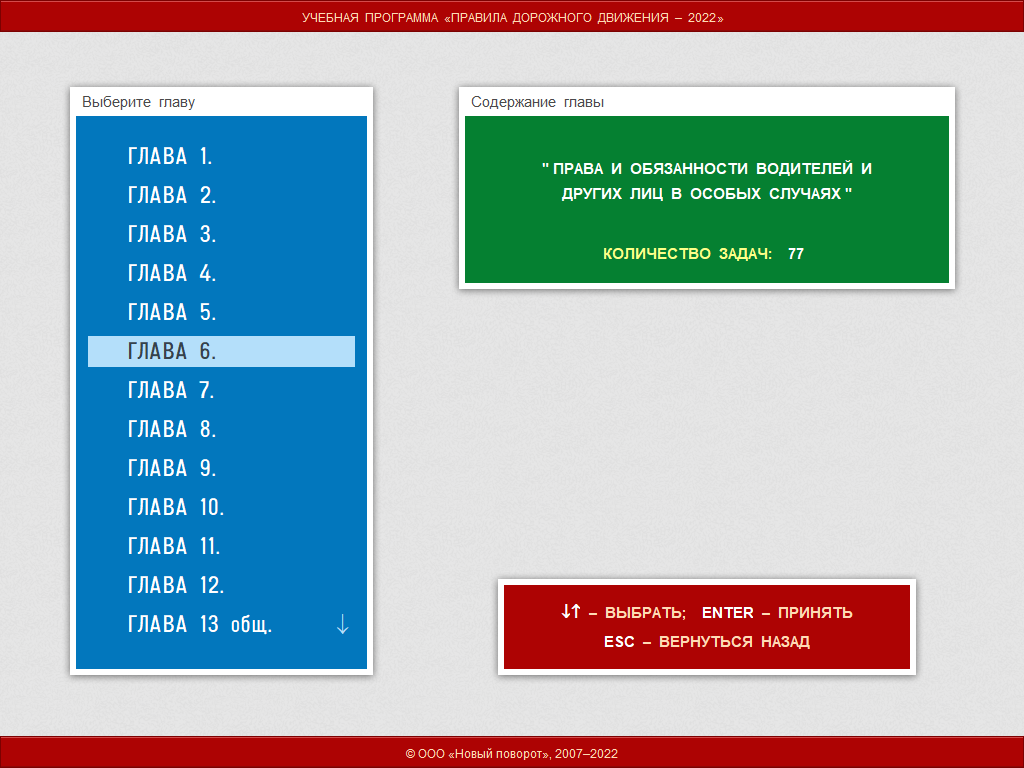


Рисунок 1.2 – Выбор главы

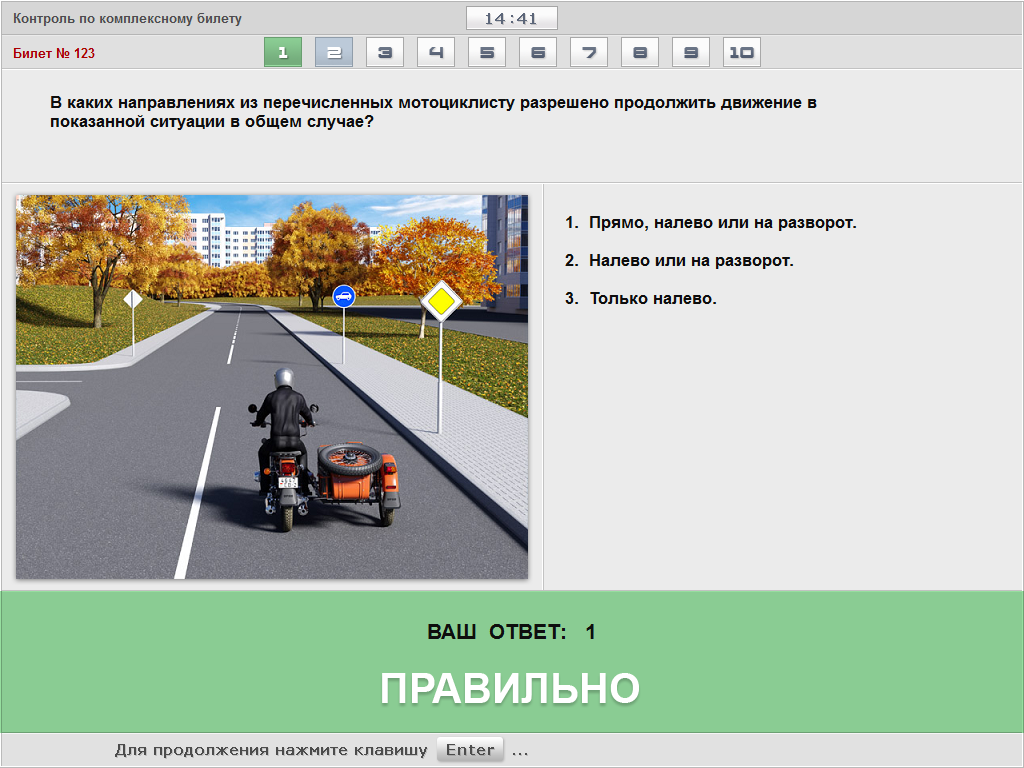
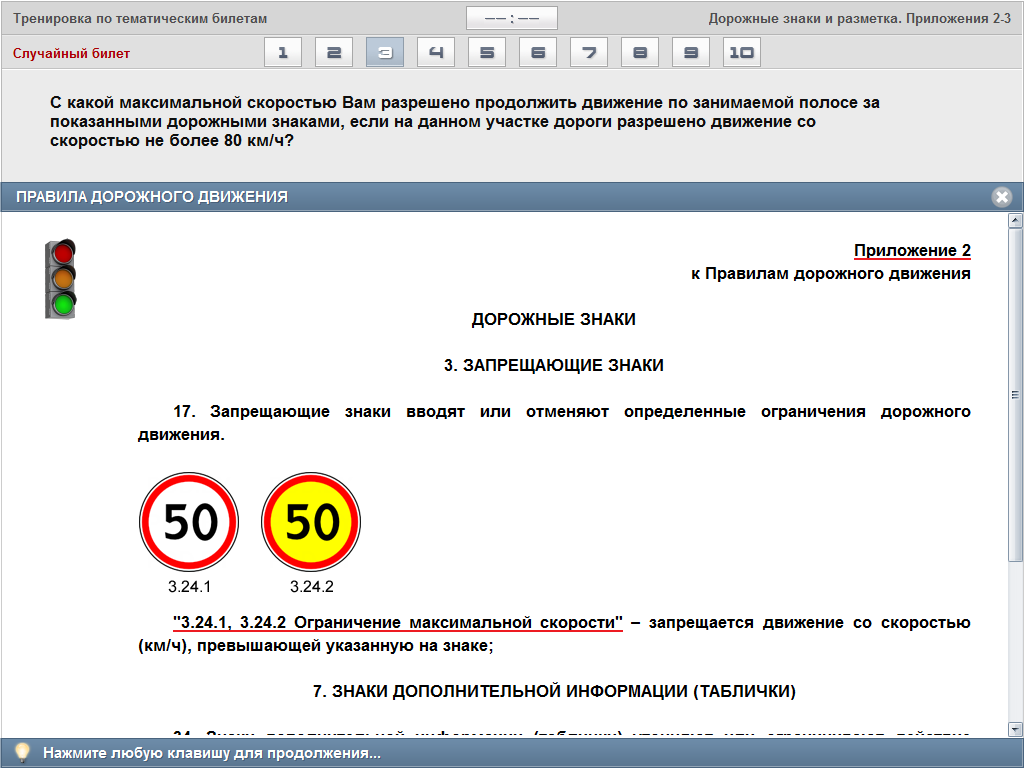


Рисунок 1.3 – Пример теста

Самым важным плюсом этого приложения является то, что можно посмотреть пояснение к задаче.

Рисунок 1.4 –Пример второго теста

По итогу можно сказать, что данное приложение для тестирование является хорошим программным средством для контроля знаний, однако имеет немного устаревший дизайн, который было бы неплохо модернизировать.

Второй пример – Тесты ПДД Беларусь (рис. 1.5) [3].

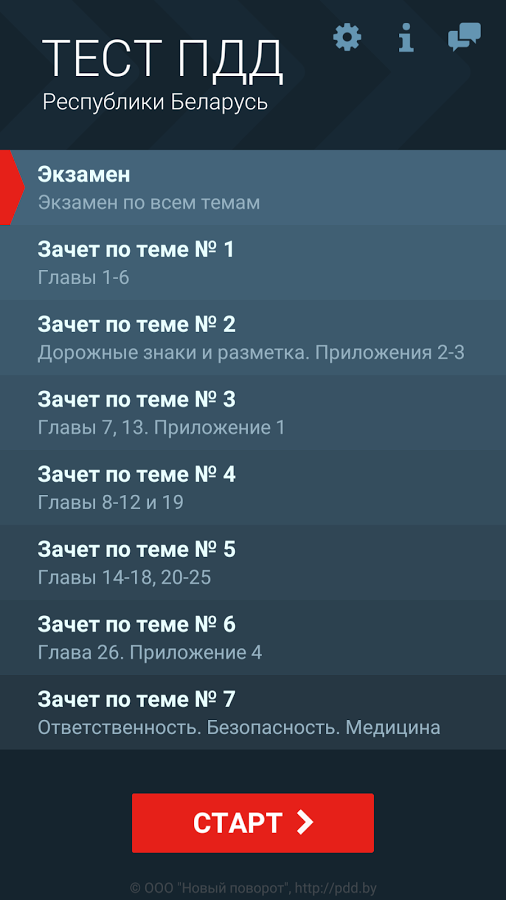


Рисунок 1.5 –Тест ПДД Беларусь

Интерфейс этого программного средства удобный для пользователя, так как содержит все необходимое, чтобы спокойно и беспрепятственно организовывать работу. Проект не выглядит устаревшим – темные тона не напрягают глаза, а на главной странице имеются все средства первой необходимости. Система имеет обширный функционал, который позволяет осуществлять все необходимые операции.

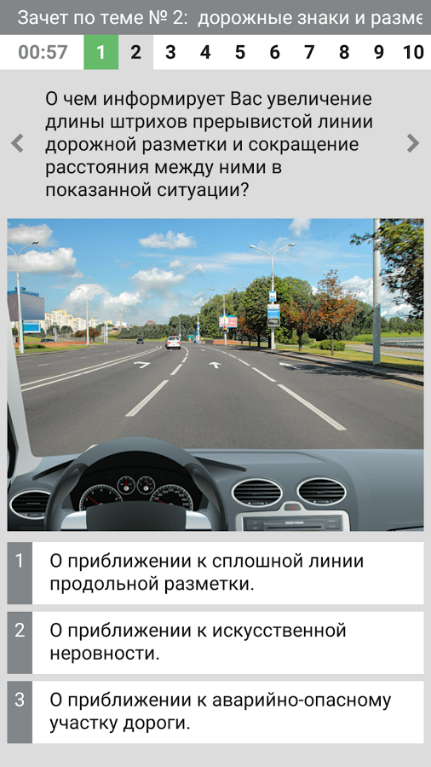


Рисунок 1.6 –Пример теста

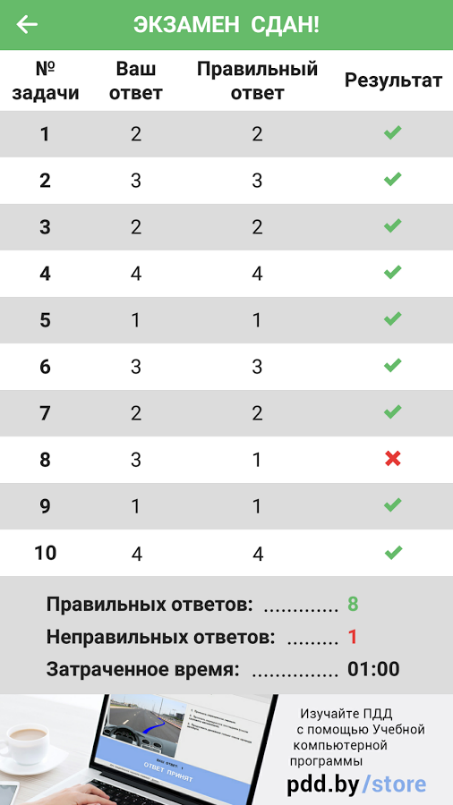


Рисунок 1.7 – Результаты теста

Плюсы данного приложения в том, что функционал понятен даже ребенку и так же не малый плюс в том, что в конце теста выводится результат на одной странице.

Минусом данной системы является то, что нет возможности просмотреть правила дорожного движения внутри приложения.

По итогу можно сказать, что данное программное средство является очень удобным приложением, хоть и не без недостатков.

# **Анализ и проектирование архитектуры приложения**

## **Анализ требований к программному средству**

На предыдущем этапе курсового проекта были рассмотрены несколько аналогов программных средств для обучения правилам. Функционал нашего приложения будет включать в себя совокупность основных возможностей аналогов с вариативной доработкой и расширением. Стоит учесть все излишки и проблемы рассмотренных аналогов, а также добавить некоторые дополнительные возможности.

Благодаря анализу похожих по функционалу приложений, было принято решение проектировки функционала для четырех основных категорий пользователей: администраторы и студенты. Представление общего функционала для всех групп можно увидеть на Use Case диаграмме (рис. 2.1).

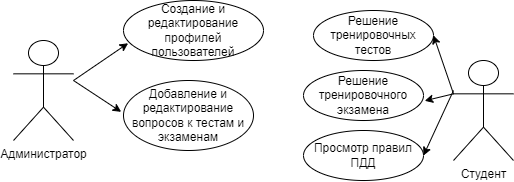


Рисунок 2.1 –Use Case диаграмма

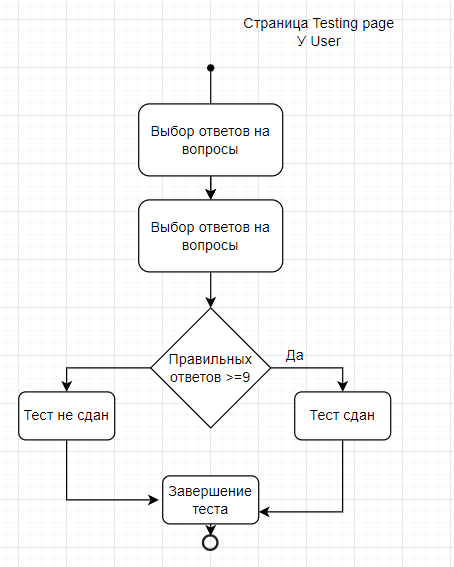


Рисунок 2.2 – Диаграмма активности

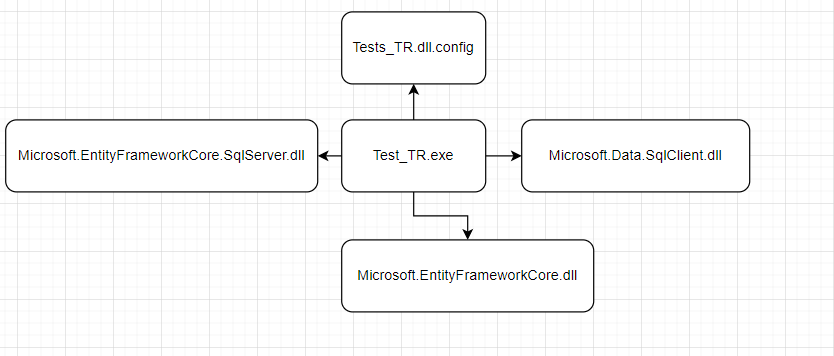


Рисунок 2.3 – диаграмма компонентов

## **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 студентах и редактировать вопросы к тестам.

**Студент.** Для данной категории пользователей предусмотрена возможность прохождения тренировочных тестов, экзамена, а также просмотр правил ПДД.

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

# **Проектирование структуры базы данных. Разработка дизайна пользовательского интерфейса**

## **Общая структура проекта**

В данной главе будут описаны все шаги проектирования и разработки приложения, в результате которых, получилась определенная структура решения (рис. 3.1).

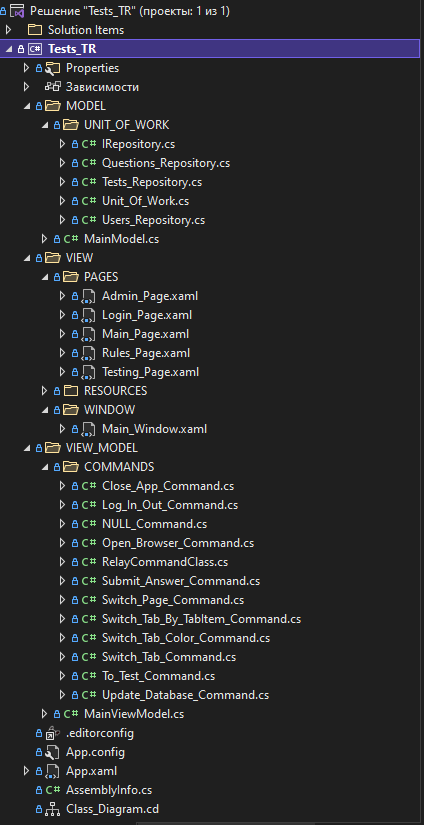


Рисунок 3.1 –Структура проекта

В решении присутствуют разные виды классов и ресурсов. Опишем их более подробно:

– файл Main\_Window.xaml основной файл, окно приложения. В нем отображаются все остальные страницы.MainWindow содержит в себе фрейм, перенаправляющий нас на Login\_Page.

* файлы Login\_Page.xaml, Main\_Page.xaml, Admin\_Page.xaml, Testing\_Page.xaml, Rules\_Page.xaml являются основными классами, реализующими окна WPF приложения, и содержат xaml разметку. Их аналоги с расширением xaml.cs содержат привязку данных ViewModel к DataContext, описанную на языке C#. MainWindow содержит в себе фрейм, перенаправляющий нас на Login\_Page – страницу авторизации пользователя. Main\_Page – страница с выбором тестов. Rules\_Page – страница, где можно почитать правила. Testing\_Page – форма теста, куда подгружаются вопросы. Admin\_Page – страница администратора;
* папка RESOURCES содержит несколько подпапок и фалов. В папке DICTIONARY находится словарь Rules.xaml, в котором находятся правила ПДД. Папка STYLES содержит файлы стилей для некоторых страниц. Файл favicon.ico – иконка проекта. LoadScreen.png – SplashScreen. Questions.sql – набор дефолтных данных для приложения. Используется при инициализации базы данных впервые. Tesla.png – картинка главной страницы. (рис. 3.2);

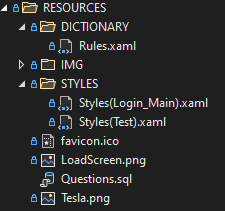


Рисунок 3.2 –Папка RESOURCES

* папка IMG содержит всю графическую информацию, необходимую для приложения, такую как изображения (рис. 3.3);

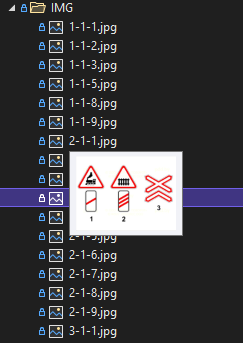


Рисунок 3.3 –Папка IMG

* Class\_Diagram.cd – диаграммы классов;
* класс MainModel.cs является классом контекста, который наследуется от класса DbContext. В нем реализованы базовые методы для работы с бд, такие как загрузка данных при подключении и там же реализованы сущности бд и их ограничения.
* В папке UNIT\_OF\_WORK находятся файлы с расширением .cs, реализующие паттерны Repository и Unit of Work для работы с бд.
* App.config – файл конфигураций приложения.
* App.xaml – файл для подключения ресурсов (словари и стили).
* В папке VIEW\_MODEL находятся файл MainViewModel.cs, в котором реализована вся бизнес-логика, и папка COMMANDS, в которой реализованы различные команды и реализация класса RelayCommand в файле RelayCommandClass.cs.

Разработка приложения выполнялась в следующей последовательности:

* проектирование макета и реализация xaml разметки окон приложения;
* реализация бизнес-логики приложения;
* проектирование и реализация базы данных;
* подключение базы данных к проекту;
* генерация модели данных;
* создание дополнительных классов для обеспечения работы функционала;
* улучшение внешнего вида программы.

Все этапы будут подробно рассмотрены в главе 4.

## **Проектирование базы данных**

В данной курсовой работе будет использована база данных SQL Server. 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 рис. 3.4.

База данных для данной курсовой работы состоит из 3 таблиц: двух взаимосвязанных между собой и одной отдельной.

**USERS.** Содержит всю необходимую информацию о пользователях для входа в приложение, а именно: логин, хэш пароля, роль, ФИО пользователя. Доступна для изменения администратором.

**TESTS.** Содержит всю информацию о главах теста, такую как ID теста, глава, и название. Заполняется один раз при инициализации бд и недоступна для изменения.

**QUESTIONS**. Таблица, хранящая вопросы, ответы на тест и прочее. При инициализации заполняется данными по умолчанию, доступна для изменения администратором.

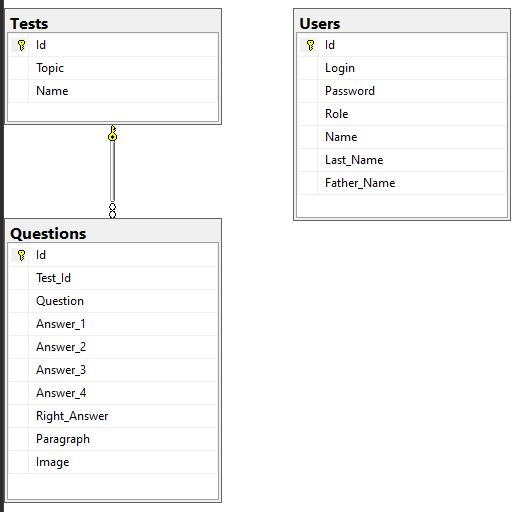


Рисунок 3.4 –Логическая схема базы данных

## **Проектирование и реализация классов приложения**

В приложении выделилось несколько категорий классов.

Первая категория – класс, который реализует бизнес логику(диаграмма представлена в приложении А). Данный класс является самым важным составляющим приложения. В нем задается вся логика взаимодействия xaml-разметок и пользователя. В нем описываются обработчики событий для кнопок и других событий, задаются контексты и источники данных.

Вторая категория – классы-модели (диаграмма представлена в приложении А). Они содержат инициализацию классов элементов, с которыми мы взаимодействуем в процессе выполнения приложения.

Третья категория – вспомогательные классы (представлен в приложении А). Необходимы для того, чтобы мы могли реализовывать какие-либо дополнительные функции, которые не смогли бы реализовать с имеющимися классами. Здесь находятся классы Repository и Unit\_Of\_Work, которые необходимы для упрощения работы с бд.

## **Диаграмма последовательности**

Диаграмма последовательности входа в приложение представлена на рис. 3.6. Как можно заметить на схеме, процесс состоит из следующих этапов: ввод логина и пароля, нажатие кнопки «Войти», валидация введенных данных, проверка соответствия логина и пароля в базе данных, и соответственно ответные реакции системы.

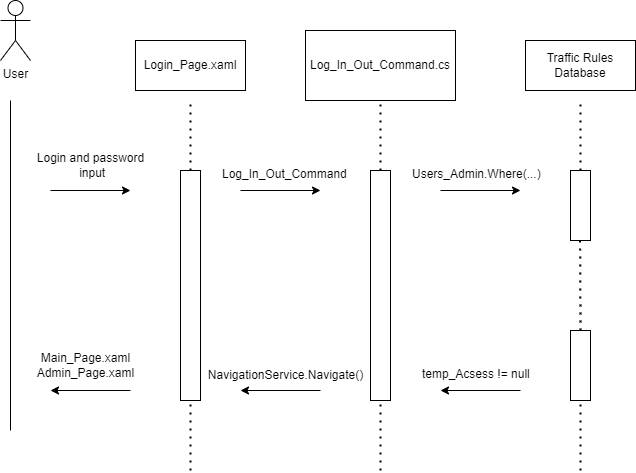


Рисунок 3.6 –Диаграмма последовательности входа в приложение

# **Реализация программного средства**

Данное программное средство реализовано с помощью 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 [5].

Перед тем, как начать программировать все страницы, необходимо задать источник для стилей (в нашем случае это дизайн страниц). Все стили прописаны в файлах Styles(Login\_Main).xaml и Styles(Test).xaml, листинги кода которых представлены в приложении Б данной записки.

На начальной странице Main\_Window.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 4.1).

|  |
| --- |
| <Window  x:Class="Tests\_TR.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:local="clr-namespace:Tests\_TR"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  Title="Тесты ПДД 2077"  Width="1000"  Height="650"  d:DataContext="{d:DesignInstance Type=local:PagesViewModel}"  BorderThickness="0"  Icon="/VIEW/RESOURCES/favicon.ico"  ResizeMode="NoResize"  WindowStartupLocation="CenterScreen"  WindowStyle="None"  mc:Ignorable="d">  <Window.InputBindings>  <KeyBinding  Key="Left"  Command="{Binding Switch\_Tab}"  CommandParameter="false" />  <KeyBinding  Key="Right"  Command="{Binding Switch\_Tab}"  CommandParameter="true" />  </Window.InputBindings>  <Grid>  <Frame  Name="MainFrame"  Content="{Binding Current\_Page}"  NavigationUIVisibility="Hidden" />  </Grid>  </Window> |

Листинг 4.1 –Main\_Window.xaml, xaml-код

Как видно из кода выше, при запуске приложения (а именно Main\_Window.xaml запускается после запуска приложения) нас перенаправляет на страницу Login\_Page.xaml. Это страница авторизации пользователя, xaml-код которой представлен в приложении В. На ней пользователь сможет вводить свои логин и пароль и авторизоваться в само приложение.

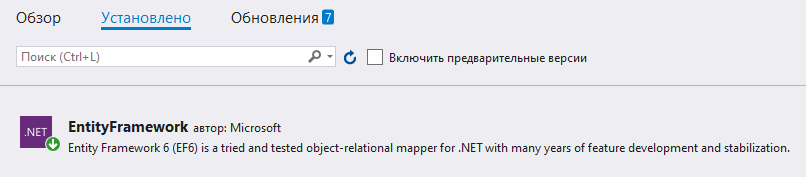
Небольшим дополнительным элементом этой и некоторых последующих страниц является InputBindings, который позволяет обрабатывать сочетания клавиш.

Первое, что происходит после ввода правильных логина и пароля и нажатия на кнопку «Войти», - переход на страницу Main\_Page или Admin\_Page, в зависимости от роли пользователя. Код функции, которая реализует данную возможность, представлен на листинге 4.2.

|  |
| --- |
| if (next\_page == login\_Page) //Logout  {  var temp\_Result = MessageBox.Show("Вы действительно хотите выйти из аккаунта?", "Выход из аккаунта", MessageBoxButton.YesNo);  if (temp\_Result == MessageBoxResult.No) return;  real\_Password = "";  fake\_Password\_Length = 0;  login\_Password[0] = "";  login\_Page.Password.Text = "";  NavService.Navigate(next\_page);  selectedIndex[0] = 0; current\_Page = (Page)next\_page;  }  if (current\_Page == login\_Page) //Login  {  if (login\_Password[0] == "" || real\_Password == "") return;  if (login\_Password[0] == "supasupauserlogin" && real\_Password == "supasupauserpassword") //supauser  {  NavService.Navigate((Page)pages\_List[3]);  current\_Page = (Page)pages\_List[3];  return;  }  var temp\_Acsess = Users\_Admin.Where(x => x.Login == login\_Password[0] && x.Password == ComputeHash(real\_Password)).FirstOrDefault();  if (temp\_Acsess != null)  {  var temp\_Role = unit\_Of\_Work.Users.GetAll()  .Where(p => p.Login == login\_Password[0] && p.Password == ComputeHash(real\_Password))  .Where(p => p.Role.ToLower() == "admin").FirstOrDefault();  MessageBox.Show(temp\_Role != null ? "Вы вошли как администратор" : "Вы вошли как студент", "Авторизация прошла успешно!");  if (temp\_Role != null)  {  NavService.Navigate((Page)pages\_List[3]);  current\_Page = (Page)pages\_List[3];  }  else  {  NavService.Navigate(next\_page);  selectedIndex[0] = 0; current\_Page = (Page)next\_page;  }  }  else  {  MessageBox.Show("Неверно введен логин или пароль", "Ошибка авторизации!");  }  ;  } |

Листинг 4.2 –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 [7]. Это определенная технология на базе фреймворка ADO.NET для работы с данными. Подключили мы ее к проекту с помощью службы управления пакетами NuGet (рис. 4.1).

Рисунок 4.1 –Подключение Entity Framework

Для взаимодействия с базой данных через Entity Framework нам нужен контекст данных, поэтому добавим класс EntityContext. В своем методе OnConfiguring он передает 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 4.3.

|  |
| --- |
| public class DatabaseContext : DbContext  {  public DbSet<Test> Tests { get; set; }  public DbSet<Questions> Questions { get; set; }  public DbSet<User> Users { get; set; }  public DatabaseContext()  {  var temp\_Result = Database.EnsureCreated();  if (temp\_Result)  {  string sql\_Raw = "";  var taskA = Task.Run(() => sql\_Raw = File.ReadAllText(@"./VIEW/RESOURCES/Questions.sql"));  string Server\_Name = ConfigurationManager.ConnectionStrings["NameOfDatabase"].ConnectionString;  taskA.Wait();  Database.ExecuteSqlRaw(string.Format(sql\_Raw, Server\_Name));  }  Load\_Data();  }  protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)  {  optionsBuilder.UseSqlServer(ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString);  }  protected override void OnModelCreating(ModelBuilder modelBuilder)  {  modelBuilder.Entity<User>(User\_Builder);  modelBuilder.Entity<Test>(Test\_Builder);  modelBuilder.Entity<Questions>(Questions\_Builder);  }  private void User\_Builder(EntityTypeBuilder<User> e)  {  e.HasKey(i => i.Id).HasName("PK\_User");  e.Property(i => i.Id).ValueGeneratedOnAdd();  e.HasIndex(i => i.Login).IsUnique();  e.Property(l => l.Login).HasMaxLength(30).IsRequired();  e.HasIndex(i => i.Password).IsUnique();  e.Property(p => p.Password).IsRequired();  e.Property(r => r.Role).HasMaxLength(6).IsRequired();  e.Property(n => n.Name).HasMaxLength(30).IsRequired();  e.Property(ln => ln.Last\_Name).HasMaxLength(30).IsRequired();  e.Property(fn => fn.Father\_Name).HasMaxLength(30);  e.HasCheckConstraint("CK\_User\_Login", "len(login) > 4");  e.HasCheckConstraint("CK\_User\_Password", "len(password) > 4");  e.HasCheckConstraint("CK\_User\_Role", "role = 'admin' or role = 'user'");  e.HasCheckConstraint("CK\_User\_Name", "len(name) > 0");  e.HasCheckConstraint("CK\_User\_Last\_Name", "len(last\_name) > 0");  }  private void Test\_Builder(EntityTypeBuilder<Test> e)  {  e.HasKey(i => i.Id).HasName("PK\_Test");  e.Property(i => i.Id).ValueGeneratedNever();  e.HasIndex(i => i.Id).IsUnique();  e.Property(n => n.Name).HasMaxLength(50).IsRequired();  e.Property(t => t.Topic).IsRequired();  e.HasMany(t => t.Questions)  .WithOne(q => q.Test)  .HasForeignKey(q => q.Test\_Id)  .HasConstraintName("FK\_Test\_Questions\_Test\_Id ")  .OnDelete(DeleteBehavior.Cascade);  }  private void Questions\_Builder(EntityTypeBuilder<Questions> e)  {  e.HasKey(i => i.Id).HasName("PK\_Questions");  e.Property(q => q.Question).HasMaxLength(400).IsRequired();  e.Property(a => a.Answer\_1).HasMaxLength(300).IsRequired();  e.Property(a => a.Answer\_2).HasMaxLength(300).IsRequired();  e.Property(a => a.Answer\_3).HasMaxLength(300);  e.Property(a => a.Answer\_4).HasMaxLength(300);  e.Property(ra => ra.Right\_Answer).HasMaxLength(1).IsRequired();  e.Property(p => p.Paragraph).HasMaxLength(60);  e.Property(i => i.Image).HasMaxLength(250);  e.Property(t => t.Test\_Id).IsRequired();  e.HasCheckConstraint("CK\_Questions\_Id", "Test\_Id >= 0 and Test\_Id <= 7");  e.HasCheckConstraint("CK\_Questions\_Question", "len(Question) > 0");  e.HasCheckConstraint("CK\_Questions\_Answer\_1", "len(Answer\_1) > 0");  e.HasCheckConstraint("CK\_Questions\_Answer\_2", "len(Answer\_2) > 0");  }  public void Load\_Data()  {  Users.Load();  Questions.Load();  Tests.Load();  }  } |

Листинг 4.3 –MainModel.cs

Далее необходимо определить классы-модели, которые будем использовать для инициализации объектов и взаимодействия с ними. Листинг всех классов представлен в приложении Д данной записки.

После мы обращаемся к уровню доступа авторизующегося пользователя, дабы понять, на какую страницу его перенаправлять. Пример авторизации представлен в листинге 4.4.

|  |
| --- |
| if (login\_Password[0] == "" || real\_Password == "") return;  if (login\_Password[0] == "supasupauserlogin" && real\_Password == "supasupauserpassword") //supauser  {  NavService.Navigate((Page)pages\_List[3]);  current\_Page = (Page)pages\_List[3];  return;  }  var temp\_Acsess = Users\_Admin.Where(x => x.Login == login\_Password[0] && x.Password == ComputeHash(real\_Password)).FirstOrDefault();  if (temp\_Acsess != null)  {  var temp\_Role = unit\_Of\_Work.Users.GetAll()  .Where(p => p.Login == login\_Password[0] && p.Password == ComputeHash(real\_Password))  .Where(p => p.Role.ToLower() == "admin").FirstOrDefault();  MessageBox.Show(temp\_Role != null ? "Вы вошли как администратор" : "Вы вошли как студент", "Авторизация прошла успешно!");  if (temp\_Role != null)  {  NavService.Navigate((Page)pages\_List[3]);  current\_Page = (Page)pages\_List[3];  }  else  {  NavService.Navigate(next\_page);  selectedIndex[0] = 0; current\_Page = (Page)next\_page;  }  }  else  {  MessageBox.Show("Неверно введен логин или пароль", "Ошибка авторизации!");  } |

Листинг 4.4 –Авторизация для пользователя

В итоге мы переходим на одну из двух главных страниц в зависимости от нашего уровня доступа: администратор или 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.

# **Тестирование программного средства**

Итогом предыдущих этапов курсовой работы получилось программное средство «ПДД Учебная программа».

Тестирование программы является важнейшим этапом в процессе ее 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. 5.1), но приложение будет продолжать работать. При пустых полях кнопка не срабатывает. (рис. 5.2).

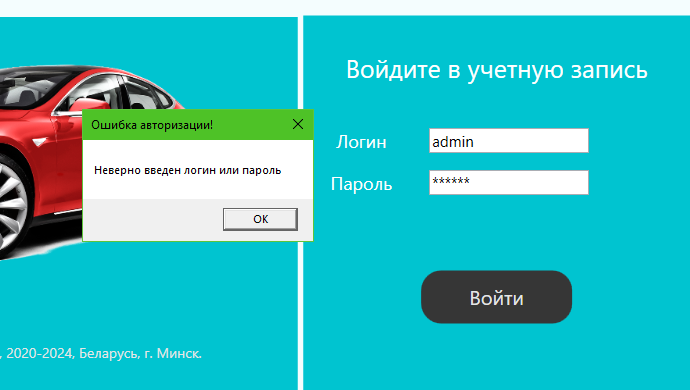


Рисунок 5.1 –Ошибка при аутентификации

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 администратора. В случае ввода неправильных данных выводится предупреждающее окно с требованиями корректности данных. (рис. 5.4).

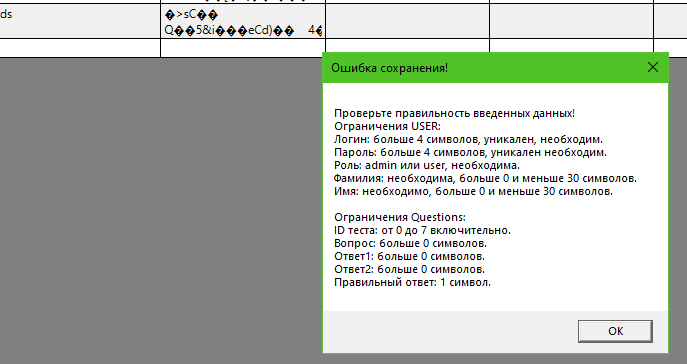


Рисунок 5.4 –Ошибка неверного ввода данных

# **Руководство пользователя**

После запуска приложения «ПДД Учебная программа» перед пользователем появляется окно аутентификации (рис. 6.1). Она содержит поля для ввода логина и пароля, кнопку входа. При запуске приложения в данной системе первый раз – доступен только один “вшитый” пользователь (admin) с логином “supasupauserlogin” и паролем “supasupauserpassword”.

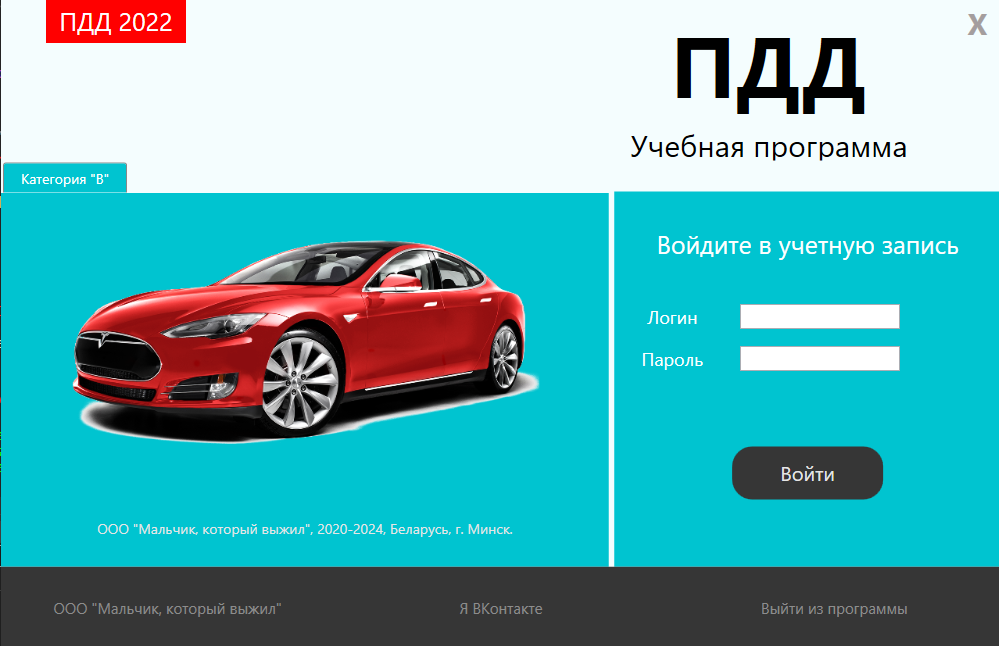


Рисунок 6.1 –Окно аутентификации

В зависимости от уровня доступа пользователя, он будет перенаправлен на соответствующую этому уровню доступа страницу.

Администратор увидит окно, представленное на рис. 5.3. Данное окно представляет стартовое меню администратора, с помощью которого он может открыть окна для контроля других областей приложения.

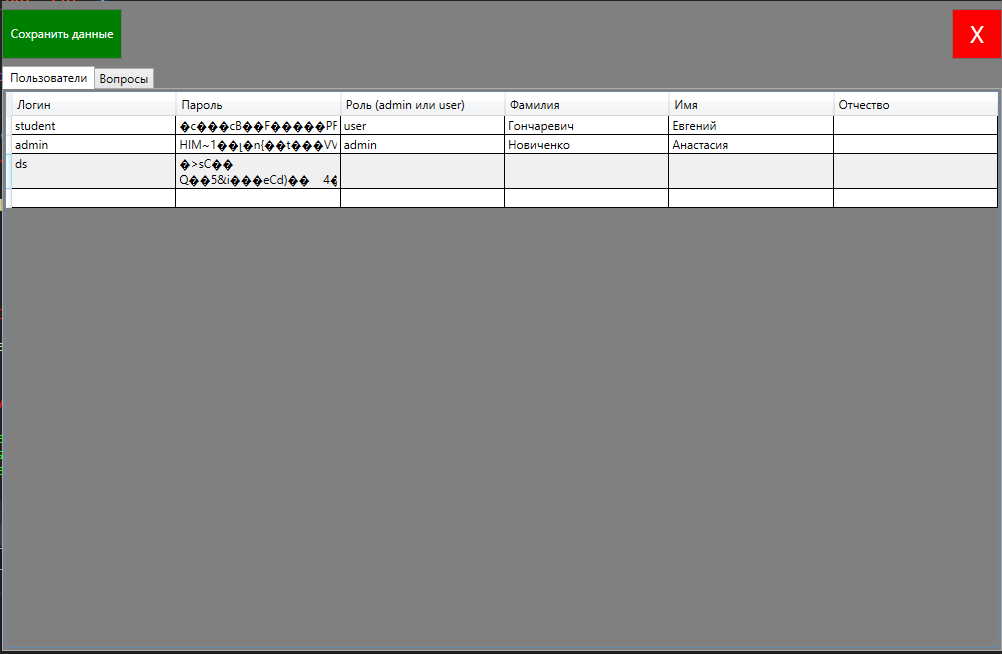


Рисунок 6.3 –Меню администратора

После клика по одной из вкладок администратору представляется, соответственно, одно из двух окон. Например, окно настройки логинов и паролей представлено на рис. 6.3. Вопросы к тесту на рис. 6.4.

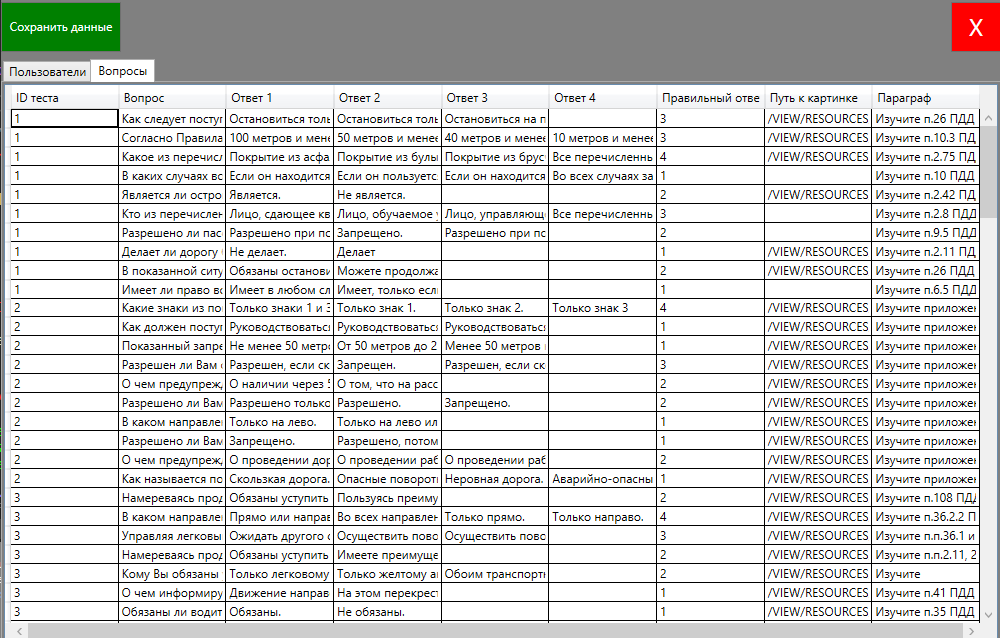


Рисунок 6.4 –Настройка логинов и паролей пользователей

Стартовая страница студентов представлена на рис. 6.6.

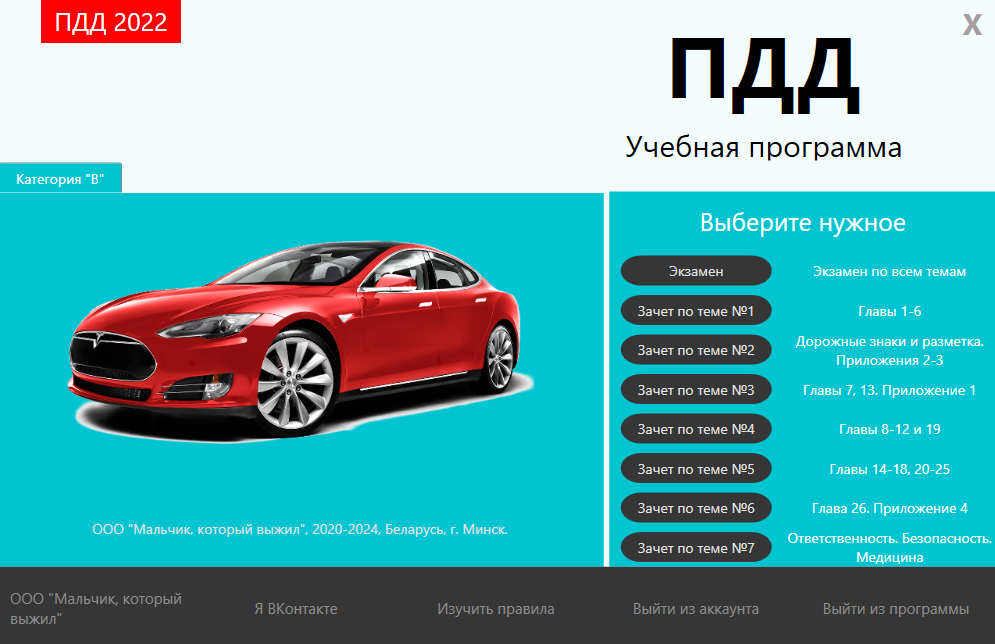


Рисунок 6.6 –Меню инструктора

Так же у студентов присутствует страница теста и правил пдд на рис. 6.7 и 6.8 соответственно.

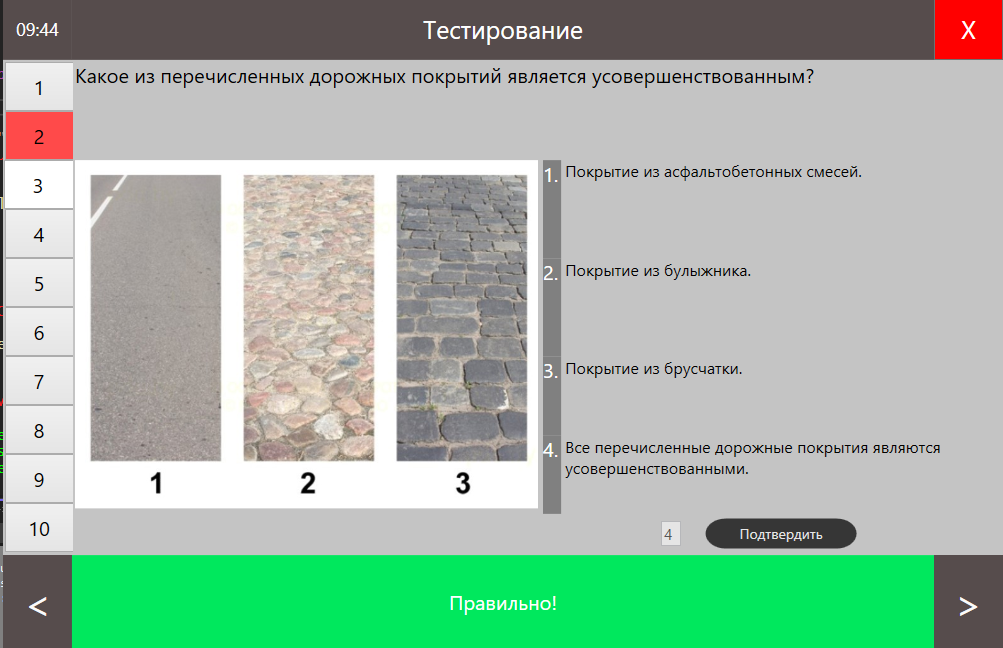


Рисунок 6.7 –Страница теста

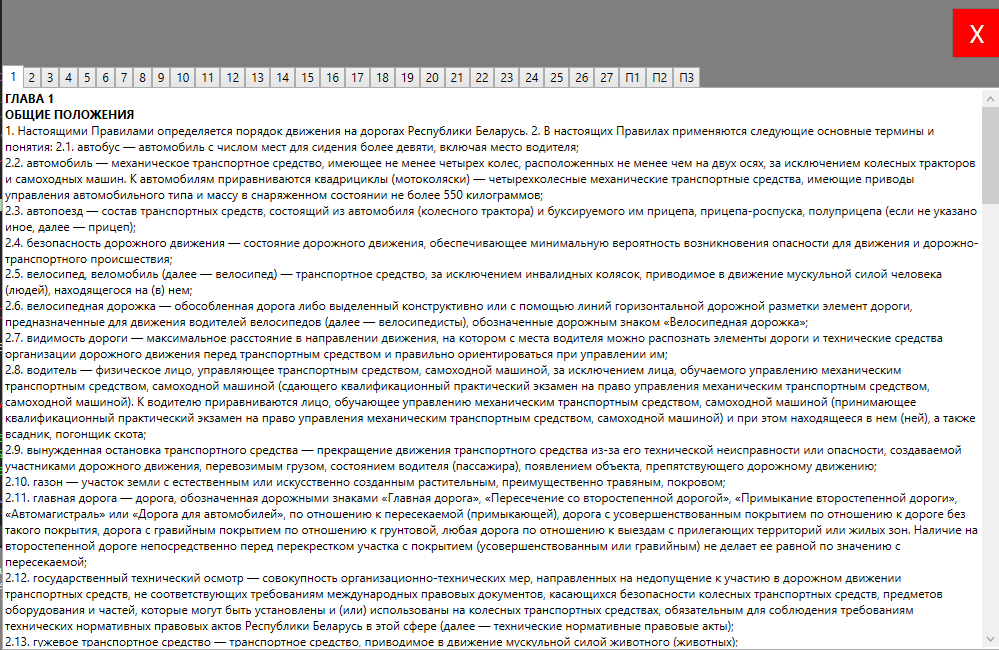


Рисунок 6.8 –Страница теста

# **ЗАКЛЮЧЕНИЕ**

Итогом курсовой работы стало программное средство «ПДД учебная программа». В ходе выполнения был проведен тщательный анализ, который включал в себя обзор аналогов, выявление положительных и отрицательных сторон конкурентов, проверка актуальности данного ПО, постановка целей и задач, формирование технического задания. Было проведено проектирование, включающее разработку структуры базы данных, разработку архитектуры приложения, разработка пользовательского интерфейса, разработку классов и других функциональных частей.

Благодаря подготовительным этапам получилась стабильная реализация приложения. Тщательно спроектированная база данных позволит в будущем без значительных её изменений добавить огромную функциональную составляющую, позволяющую превратить программное средство для автошколы в полноценное приложение, управляющее всеми областями структуры.

На последних этапах разработки программное средство “ПДД учебная программа” было протестировано для выявления неточностей в работе и для всех категорий пользователей было предложено руководство по пользованию.

Для выполнения курсовой работы одним из важнейших этапов стало изучение материала по таким направлениям, как «Объектно-ориентированное программирование на языке С#», «Реляционные базы данных SQL на основе SQL Server», «Система для построения клиентских приложений WPF» и «Entity Framework».

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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

[2] Журнал. Английский язык, 10 «А». Демонстрационная школа №1 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://demo.schools.by/journal/100121> - Время доступа: 15.03.2021

[3] Программа управления предприятием leadertask.ru [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.leadertask.ru/content/view/programma-upravleniya-predpriyatiem> - Время доступа: 15.03.2021

[4] C# и .NET | Классы и объекты [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/sharp/tutorial/3.1.php> Время доступа: 01.05.2021

[5] WPF | Страничные интерфейсы [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://professorweb.ru/my/WPF/UI_WPF/level24/24_2.php> Время доступа: 01.05.2021

[6] Работа с электронной почтой в C# и .NET [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/sharp/net/8.1.php> Время доступа: 03.05.2021

[7] Введение в Entity Framework 6 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/sharp/entityframework/1.1.php> Время доступа: 03.05.2021

[8] Extended WPF Toolkit - How to add reference to project? [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://stackoverflow.com/questions/14809446/extended-wpf-toolkit-how-to-add-reference-to-project> Время доступа: 04.05.2021

[9] Microsoft.Office.Interop.Excel Пространство имен [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/microsoft.office.interop.excel?view=excel-pia> Время доступа: 06.05.2021

# **ПРИЛОЖЕНИЕ А**

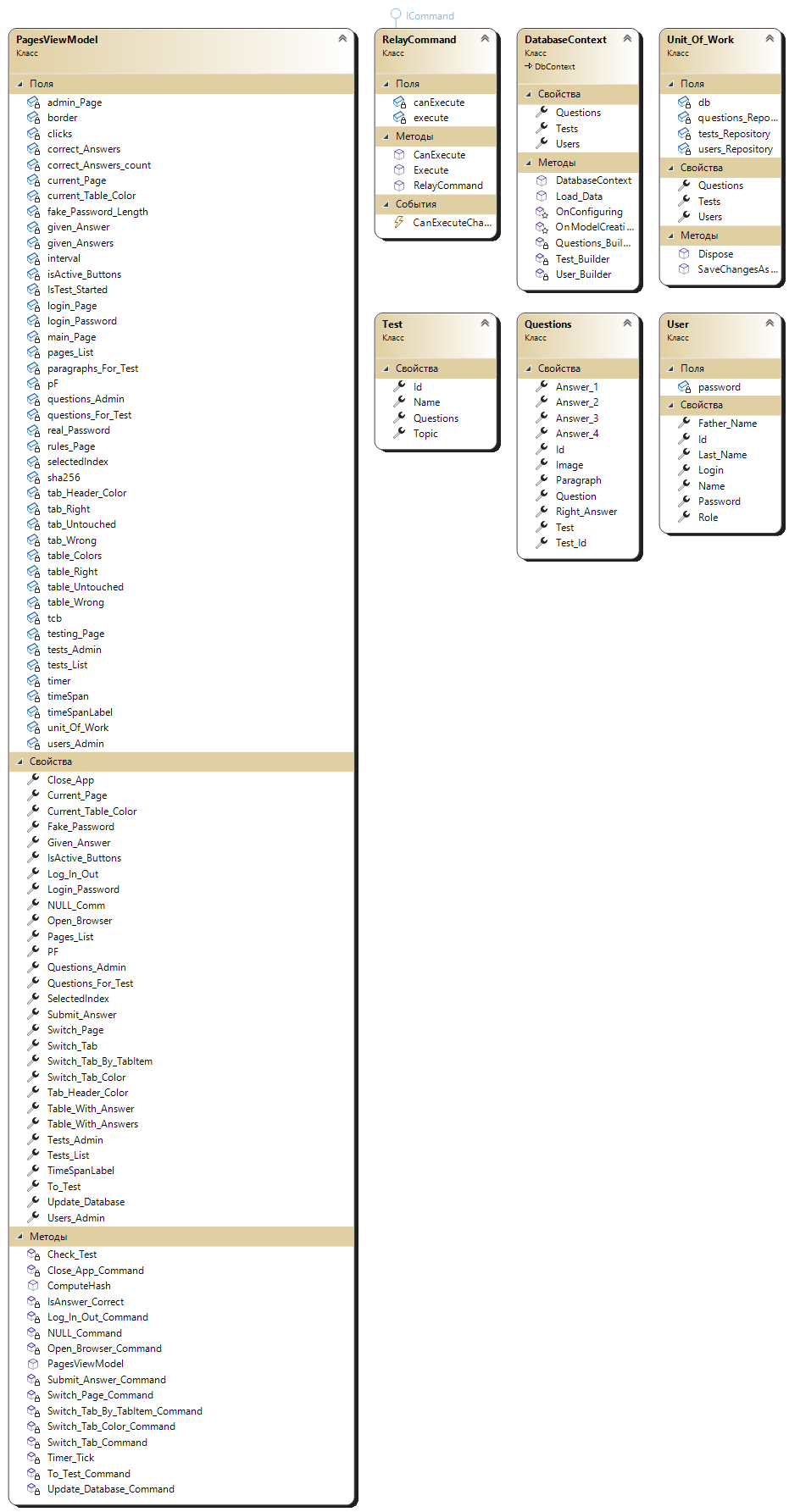


Рисунок 1 –Диаграмма классов ViewModel и классов-моделей приложения

# **ПРИЛОЖЕНИЕ Б**

Файл Styles(Login\_Main).xaml, xaml-код

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

<SolidColorBrush x:Key="RedBrush" Color="Red" />

<SolidColorBrush x:Key="SolidBorderBrush" Color="#888" />

<SolidColorBrush x:Key="GreenBrush" Color="#00C4D0" />

<SolidColorBrush x:Key="DisabledBackgroundBrush" Color="#EEE" />

<SolidColorBrush x:Key="DisabledBorderBrush" Color="#AAA" />

<SolidColorBrush x:Key="DisabledForegroundBrush" Color="#888" />

<Style x:Key="Blue\_Tab" TargetType="{x:Type TabItem}">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type TabItem}">

<Grid>

<Border

Name="Border"

Margin="0,0,-4,0"

Background="{StaticResource RedBrush}"

BorderBrush="{StaticResource SolidBorderBrush}"

BorderThickness="0"

CornerRadius="2,2,0,0">

<ContentPresenter

x:Name="ContentSite"

Margin="12,2,12,2"

HorizontalAlignment="Center"

VerticalAlignment="Center"

ContentSource="Header"

RecognizesAccessKey="True" />

</Border>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Panel.ZIndex" Value="100" />

<Setter TargetName="Border" Property="Background" Value="{StaticResource GreenBrush}" />

<Setter TargetName="Border" Property="BorderThickness" Value="1,1,1,0" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="Button\_Classic" TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}" CornerRadius="20">

<ContentPresenter HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Width" Value="151" />

<Setter Property="Height" Value="53" />

<Setter Property="Background" Value="#363636" />

<Setter Property="FontSize" Value="20" />

<Setter Property="Foreground" Value="#e6e6e6" />

</Style>

<Style x:Key="Button\_Classic\_Mini" TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}" CornerRadius="20">

<ContentPresenter HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Width" Value="151" />

<Setter Property="Height" Value="30" />

<Setter Property="Background" Value="#363636" />

<Setter Property="FontSize" Value="14" />

<Setter Property="Foreground" Value="#e6e6e6" />

</Style>

<Style x:Key="Description" TargetType="TextBlock">

<Setter Property="Foreground" Value="White" />

<Setter Property="FontSize" Value="14" />

<Setter Property="TextAlignment" Value="Center" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="TextWrapping" Value="Wrap" />

</Style>

<Style x:Key="Page\_Size" TargetType="Page">

<Setter Property="Height" Value="650" />

<Setter Property="Width" Value="1000" />

</Style>

<Style x:Key="White\_Line" TargetType="Grid">

<Setter Property="Background" Value="#F4FDFE" />

</Style>

<Style x:Key="Red\_Rect" TargetType="Grid">

<Setter Property="HorizontalAlignment" Value="Left" />

<Setter Property="VerticalAlignment" Value="Top" />

</Style>

<Style x:Key="PDD2022" TargetType="TextBox">

<Setter Property="Width" Value="140" />

<Setter Property="Height" Value="45" />

<Setter Property="Margin" Value="45,0,0,0" />

<Setter Property="HorizontalContentAlignment" Value="Center" />

<Setter Property="VerticalContentAlignment" Value="Center" />

<Setter Property="Background" Value="Red" />

<Setter Property="BorderThickness" Value="0" />

<Setter Property="FontSize" Value="25" />

<Setter Property="Foreground" Value="White" />

</Style>

<Style x:Key="BIGPDD" TargetType="TextBlock">

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="FontSize" Value="85" />

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="TextAlignment" Value="Center" />

</Style>

<Style x:Key="Stud\_Prog" TargetType="TextBlock">

<Setter Property="FontSize" Value="30" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="TextAlignment" Value="Center" />

</Style>

<Style x:Key="Close\_App\_X" TargetType="Button">

<Setter Property="FontSize" Value="30" />

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="Foreground" Value="#A49E9E" />

<Setter Property="Background" Value="Transparent" />

<Setter Property="BorderThickness" Value="0" />

<Setter Property="Margin" Value="30,-80,0,0" />

</Style>

<Style x:Key="Blue\_Tab\_Control" TargetType="TabControl">

<Setter Property="Background" Value="#00C4D0" />

<Setter Property="BorderThickness" Value="0" />

</Style>

<Style x:Key="Company\_White" TargetType="TextBlock">

<Setter Property="FontSize" Value="14" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="TextAlignment" Value="Center" />

<Setter Property="Foreground" Value="#E6E6E6" />

</Style>

<Style x:Key="Blue\_Line" TargetType="Grid">

<Setter Property="Background" Value="#00C4D0" />

<Setter Property="Margin" Value="0,31,0,0" />

</Style>

<Style x:Key="Come\_In" TargetType="TextBlock">

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="FontSize" Value="25" />

<Setter Property="Foreground" Value="White" />

</Style>

<Style x:Key="Text\_Label" TargetType="TextBlock">

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="FontSize" Value="18" />

<Setter Property="Foreground" Value="White" />

</Style>

<Style x:Key="Input\_Box" TargetType="TextBox">

<Setter Property="HorizontalAlignment" Value="Left" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="VerticalContentAlignment" Value="Center" />

<Setter Property="FontSize" Value="15" />

<Setter Property="Width" Value="160" />

<Setter Property="Height" Value="25" />

<Setter Property="Margin" Value="10,0,0,0" />

</Style>

<Style x:Key="Bottom\_Button" TargetType="Button">

<Setter Property="FontSize" Value="15" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="Foreground" Value="#8D8D8D" />

<Setter Property="Background" Value="Transparent" />

<Setter Property="BorderThickness" Value="0" />

</Style>

<Style x:Key="Company\_Gray" TargetType="TextBlock">

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="FontSize" Value="15" />

<Setter Property="Foreground" Value="#8D8D8D" />

<Setter Property="TextWrapping" Value="Wrap" />

</Style>

</ResourceDictionary>

Файл Styles(Test).xaml, xaml-код

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

<!--#region universal-->

<Style x:Key="Background" TargetType="Grid">

<Setter Property="Background" Value="#c4c4c4" />

</Style>

<!-- endregion -->

<!--#region Testing\_Page-->

<Style x:Key="Timer\_Container" TargetType="Grid">

<Setter Property="Background" Value="#564C4C" />

</Style>

<Style x:Key="Timer\_Text" TargetType="TextBlock">

<Setter Property="FontSize" Value="18" />

<!--<Setter Property="FontWeight" Value="500" />-->

<Setter Property="Foreground" Value="White" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="HorizontalAlignment" Value="Center" />

</Style>

<Style x:Key="Topic" TargetType="TextBlock">

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="Foreground" Value="White" />

<Setter Property="FontSize" Value="25" />

</Style>

<Style x:Key="Topic\_Container" TargetType="Grid">

<Setter Property="Background" Value="#564C4C" />

<Setter Property="VerticalAlignment" Value="Stretch" />

</Style>

<Style x:Key="Tab\_Control" TargetType="TabControl">

<Setter Property="Background" Value="#c4c4c4" />

<Setter Property="TabStripPlacement" Value="Left" />

</Style>

<Style x:Key="Tab\_Header\_Untouched" TargetType="TabItem">

<Setter Property="Width" Value="68" />

<Setter Property="Height" Value="49" />

<Setter Property="FontSize" Value="20" />

</Style>

<Style

x:Key="Tab\_Header\_Wrong"

BasedOn="{StaticResource Tab\_Header\_Untouched}"

TargetType="TabItem">

<Setter Property="Background" Value="#FF4A4A" />

</Style>

<Style

x:Key="Tab\_Header\_Right"

BasedOn="{StaticResource Tab\_Header\_Untouched}"

TargetType="TabItem">

<Setter Property="Background" Value="#00E85D" />

</Style>

<Style x:Key="Table\_Wrong" TargetType="Grid">

<Setter Property="Background" Value="#FF4A4A" />

</Style>

<Style x:Key="Table\_Untouched" TargetType="Grid">

<Setter Property="Background" Value="#564C4C" />

</Style>

<Style x:Key="Table\_Right" TargetType="Grid">

<Setter Property="Background" Value="#00E85D" />

</Style>

<Style x:Key="Table\_Text" TargetType="TextBlock">

<Setter Property="HorizontalAlignment" Value="Center" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="Foreground" Value="White" />

<Setter Property="FontSize" Value="20" />

</Style>

<Style x:Key="Exit\_Button" TargetType="Button">

<Setter Property="Background" Value="Red" />

<Setter Property="Foreground" Value="White" />

<Setter Property="FontSize" Value="25" />

</Style>

<Style x:Key="Question\_Image" TargetType="Image">

<Setter Property="HorizontalAlignment" Value="Stretch" />

<Setter Property="VerticalAlignment" Value="Top" />

</Style>

<Style x:Key="Question" TargetType="TextBlock">

<Setter Property="TextWrapping" Value="Wrap" />

</Style>

<Style x:Key="Answer" TargetType="TextBlock">

<Setter Property="TextWrapping" Value="Wrap" />

<Setter Property="Padding" Value="4,0,4,0" />

<Setter Property="FontSize" Value="16" />

</Style>

<Style x:Key="Answer\_Num" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

</Style>

<Style x:Key="Input\_Box" TargetType="TextBox">

<Setter Property="MaxLength" Value="1" />

<Setter Property="VerticalContentAlignment" Value="Center" />

<Setter Property="HorizontalContentAlignment" Value="Center" />

</Style>

<Style x:Key="Button\_Submit" TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}" CornerRadius="20">

<ContentPresenter HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Width" Value="151" />

<Setter Property="Height" Value="30" />

<Setter Property="Background" Value="#363636" />

<Setter Property="FontSize" Value="14" />

<Setter Property="Foreground" Value="#e6e6e6" />

</Style>

<!--#region Labels' styles-->

<Style x:Key="Third\_Answer" TargetType="TextBlock">

<Setter Property="Margin" Value="5,0,0,0" />

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock1, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer" TargetType="TextBlock">

<Setter Property="Margin" Value="5,0,0,0" />

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock2, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

<Setter Property="Background" Value="Red" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer2" TargetType="TextBlock">

<Setter Property="Margin" Value="5,0,0,0" />

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock3, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer2" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock4, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer3" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock5, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer3" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock6, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer4" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock7, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer4" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock8, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer5" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock9, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer5" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock10, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer6" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock11, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer6" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock12, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer7" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock13, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer7" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock14, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer8" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock15, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer8" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock16, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer9" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock17, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer9" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock18, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Third\_Answer10" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock19, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<Style x:Key="Fourth\_Answer10" TargetType="TextBlock">

<Setter Property="Background" Value="Gray" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Margin" Value="5,0,0,0" />

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=TextBlock20, Path=Text}" Value="">

<Setter Property="Visibility" Value="Collapsed" />

</DataTrigger>

</Style.Triggers>

</Style>

<!--#endregion-->

<!--#endregion-->

</ResourceDictionary>

# **ПРИЛОЖЕНИЕ В**

Файл Login\_Page.xaml, xaml-код

<Page

x:Class="Tests\_TR.VIEW.PAGES.Login\_Page"

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:tests\_tr="clr-namespace:Tests\_TR"

d:DataContext="{d:DesignInstance Type=tests\_tr:PagesViewModel}"

Style="{StaticResource Page\_Size}"

mc:Ignorable="d">

<Grid Style="{StaticResource White\_Line}">

<!--#region Definitions-->

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.55\*" />

<ColumnDefinition Width="0.005\*" />

<ColumnDefinition Width="0.35\*" />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="0.4\*" />

<RowDefinition Height="\*" />

<RowDefinition Height="0.2\*" />

</Grid.RowDefinitions>

<!--#endregion-->

<Grid

Grid.Row="0"

Grid.Column="0"

Style="{StaticResource Red\_Rect}">

<TextBox Style="{StaticResource PDD2022}" Text="ПДД 2022" />

</Grid>

<Grid Grid.Row="0" Grid.Column="2">

<!--#region Definitions-->

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.80\*" />

<ColumnDefinition Width="0.20\*" />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="0.8\*" />

<RowDefinition Height="0.2\*" />

</Grid.RowDefinitions>

<!--#endregion-->

<TextBlock

Grid.Row="0"

Grid.Column="0"

Style="{StaticResource BIGPDD}"

Text="ПДД" />

<TextBlock

Grid.Row="1"

Grid.Column="0"

Style="{StaticResource Stud\_Prog}"

Text="Учебная программа" />

<Button

Grid.Row="0"

Grid.Column="1"

Command="{Binding Close\_App}"

Content="X"

Style="{StaticResource Close\_App\_X}" />

</Grid>

<TabControl

Grid.Row="1"

Grid.Column="0"

Style="{StaticResource Blue\_Tab\_Control}">

<TabItem

Width="120"

Height="30"

Background="#00C4D0"

FontSize="14"

Foreground="White"

Header="Категория &quot;B&quot;"

IsSelected="True"

Style="{StaticResource Blue\_Tab}">

<Grid>

<!--#region Definitions-->

<Grid.RowDefinitions>

<RowDefinition Height="0.8\*" />

<RowDefinition Height="0.2\*" />

</Grid.RowDefinitions>

<!--#endregion-->

<Image

Grid.Row="0"

Grid.Column="0"

Source="/VIEW/RESOURCES/Tesla.png" />

<TextBlock

Grid.Row="1"

Grid.Column="0"

Style="{StaticResource Company\_White}"

Text="ООО &quot;Мальчик, который выжил&quot;, 2020-2024, Беларусь, г. Минск." />

</Grid>

</TabItem>

</TabControl>

<Grid

Grid.Row="1"

Grid.Column="3"

Style="{StaticResource Blue\_Line}">

<!--#region Definitions-->

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.3\*" />

<ColumnDefinition Width="0.7\*" />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="0.25\*" />

<RowDefinition Height="0.1\*" />

<RowDefinition Height="0.1\*" />

<RowDefinition Height="0.45\*" />

</Grid.RowDefinitions>

<!--#endregion-->

<TextBlock

Grid.Row="0"

Grid.Column="0"

Grid.ColumnSpan="2"

Style="{StaticResource Come\_In}"

Text="Войдите в учетную запись" />

<TextBlock

Grid.Row="1"

Grid.Column="0"

Style="{StaticResource Text\_Label}"

Text="Логин" />

<TextBox

x:Name="Login"

Grid.Row="1"

Grid.Column="1"

Style="{StaticResource Input\_Box}"

Text="{Binding Login\_Password[0], Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />

<TextBlock

Grid.Row="2"

Grid.Column="0"

Style="{StaticResource Text\_Label}"

Text="Пароль" />

<TextBox

x:Name="Password"

Grid.Row="2"

Grid.Column="1"

Style="{StaticResource Input\_Box}"

Text="{Binding Fake\_Password, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />

<Button

x:Name="LogIn"

Grid.Row="3"

Grid.Column="0"

Grid.ColumnSpan="2"

Command="{Binding Path=Log\_In\_Out}"

CommandParameter="{Binding Pages\_List[1]}"

Content="Войти"

Style="{StaticResource Button\_Classic}" />

</Grid>

<Grid

Grid.Row="2"

Grid.Column="0"

Grid.ColumnSpan="3"

Background="#363636">

<!--#region Definitions-->

<Grid.ColumnDefinitions>

<ColumnDefinition />

<ColumnDefinition />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition />

</Grid.RowDefinitions>

<!--#endregion-->

<TextBlock

Grid.Row="0"

Grid.Column="0"

Style="{StaticResource Company\_Gray}"

Text="ООО &quot;Мальчик, который выжил&quot;" />

<Button

Grid.Row="0"

Grid.Column="1"

Command="{Binding Open\_Browser}"

Content="Я ВКонтакте"

Style="{StaticResource Bottom\_Button}" />

<Button

Grid.Row="0"

Grid.Column="2"

Command="{Binding Close\_App}"

Content="Выйти из программы"

Style="{StaticResource Bottom\_Button}" />

</Grid>

</Grid>

</Page>

# **ПРИЛОЖЕНИЕ Г**

Файл Update\_Database\_Command.cs, код C#

namespace Tests\_TR

{

public partial class PagesViewModel

{

private static RelayCommand Update\_Database\_Command()

{

return new(cap => unit\_Of\_Work.SaveChangesAsync());

}

}

}

# **ПРИЛОЖЕНИЕ Д**

Исходный код класса MainModel.cs

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Metadata.Builders;

using System.Collections.Generic;

using System.Configuration;

using System.IO;

using System.Threading.Tasks;

#nullable disable

namespace Tests\_TR.MODEL

{

public class DatabaseContext : DbContext

{

public DbSet<Test> Tests { get; set; }

public DbSet<Questions> Questions { get; set; }

public DbSet<User> Users { get; set; }

public DatabaseContext()

{

var temp\_Result = Database.EnsureCreated();

if (temp\_Result)

{

string sql\_Raw = "";

var taskA = Task.Run(() => sql\_Raw = File.ReadAllText(@"./VIEW/RESOURCES/Questions.sql"));

string Server\_Name = ConfigurationManager.ConnectionStrings["NameOfDatabase"].ConnectionString;

taskA.Wait();

Database.ExecuteSqlRaw(string.Format(sql\_Raw, Server\_Name));

}

Load\_Data();

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString);

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<User>(User\_Builder);

modelBuilder.Entity<Test>(Test\_Builder);

modelBuilder.Entity<Questions>(Questions\_Builder);

}

private void User\_Builder(EntityTypeBuilder<User> e)

{

e.HasKey(i => i.Id).HasName("PK\_User");

e.Property(i => i.Id).ValueGeneratedOnAdd();

e.HasIndex(i => i.Login).IsUnique();

e.Property(l => l.Login).HasMaxLength(30).IsRequired();

e.HasIndex(i => i.Password).IsUnique();

e.Property(p => p.Password).IsRequired();

e.Property(r => r.Role).HasMaxLength(6).IsRequired();

e.Property(n => n.Name).HasMaxLength(30).IsRequired();

e.Property(ln => ln.Last\_Name).HasMaxLength(30).IsRequired();

e.Property(fn => fn.Father\_Name).HasMaxLength(30);

e.HasCheckConstraint("CK\_User\_Login", "len(login) > 4");

e.HasCheckConstraint("CK\_User\_Password", "len(password) > 4");

e.HasCheckConstraint("CK\_User\_Role", "role = 'admin' or role = 'user'");

e.HasCheckConstraint("CK\_User\_Name", "len(name) > 0");

e.HasCheckConstraint("CK\_User\_Last\_Name", "len(last\_name) > 0");

}

private void Test\_Builder(EntityTypeBuilder<Test> e)

{

e.HasKey(i => i.Id).HasName("PK\_Test");

e.Property(i => i.Id).ValueGeneratedNever();

e.HasIndex(i => i.Id).IsUnique();

e.Property(n => n.Name).HasMaxLength(50).IsRequired();

e.Property(t => t.Topic).IsRequired();

e.HasMany(t => t.Questions)

.WithOne(q => q.Test)

.HasForeignKey(q => q.Test\_Id)

.HasConstraintName("FK\_Test\_Questions\_Test\_Id ")

.OnDelete(DeleteBehavior.Cascade);

}

private void Questions\_Builder(EntityTypeBuilder<Questions> e)

{

e.HasKey(i => i.Id).HasName("PK\_Questions");

e.Property(q => q.Question).HasMaxLength(400).IsRequired();

e.Property(a => a.Answer\_1).HasMaxLength(300).IsRequired();

e.Property(a => a.Answer\_2).HasMaxLength(300).IsRequired();

e.Property(a => a.Answer\_3).HasMaxLength(300);

e.Property(a => a.Answer\_4).HasMaxLength(300);

e.Property(ra => ra.Right\_Answer).HasMaxLength(1).IsRequired();

e.Property(p => p.Paragraph).HasMaxLength(60);

e.Property(i => i.Image).HasMaxLength(250);

e.Property(t => t.Test\_Id).IsRequired();

e.HasCheckConstraint("CK\_Questions\_Id", "Test\_Id >= 0 and Test\_Id <= 7");

e.HasCheckConstraint("CK\_Questions\_Question", "len(Question) > 0");

e.HasCheckConstraint("CK\_Questions\_Answer\_1", "len(Answer\_1) > 0");

e.HasCheckConstraint("CK\_Questions\_Answer\_2", "len(Answer\_2) > 0");

}

public void Load\_Data()

{

Users.Load();

Questions.Load();

Tests.Load();

}

}

public class Test

{

public int Id { get; set; }

public int Topic { get; set; }

public string Name { get; set; }

public List<Questions> Questions { get; set; } = new();

}

public class Questions

{

public int Id { get; set; }

public int Test\_Id { get; set; }

public string Question { get; set; }

public string Answer\_1 { get; set; }

public string Answer\_2 { get; set; }

public string Answer\_3 { get; set; }

public string Answer\_4 { get; set; }

public string Right\_Answer { get; set; }

public string Paragraph { get; set; }

public string Image { get; set; }

public Test Test { get; set; }

}

public class User

{

public int Id { get; set; }

public string Login { get; set; }

public string Password

{

get => password;

set => password = PagesViewModel.ComputeHash(value);

}

public string Role { get; set; }

public string Name { get; set; }

public string Last\_Name { get; set; }

public string Father\_Name { get; set; }

private string password;

}

}

Исходный код класса MainViewModel.cs

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Security.Cryptography;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using Tests\_TR.MODEL;

using Tests\_TR.VIEW.PAGES;

namespace Tests\_TR

{

public partial class PagesViewModel //Props

{

private static string real\_Password = "";

private static byte fake\_Password\_Length = 0;

public static string Fake\_Password

{

get

{

string temp\_string = "";

for (byte i = 0; i < real\_Password.Length; i++) temp\_string += "\*";

return temp\_string;

}

set

{

real\_Password = value.Length < fake\_Password\_Length ? real\_Password[..value.Length] : real\_Password + value[fake\_Password\_Length..];

fake\_Password\_Length = (byte)value.Length;

}

}

public static ObservableCollection<string> Given\_Answer

{

get => given\_Answer;

set => given\_Answer = value;

}

public static string PF

{

get { return pF; }

set

{

if (value == "1" || value == "2" || value == "3" || value == "4" || value == "")

{

pF = value;

Given\_Answer[selectedIndex[0]] = pF;

}

else return;

}

}

public static Page Current\_Page { get => current\_Page; set => current\_Page = value; }

public static List<Page> Pages\_List { get => pages\_List; set => pages\_List = value; }

#region Test\_Region

public static string[] Tests\_List { get => tests\_List; }

public static ObservableCollection<Questions> Questions\_For\_Test { get => questions\_For\_Test; set => questions\_For\_Test = value; }

public static ObservableCollection<string> Login\_Password { get => login\_Password; set => login\_Password = value; }

public static ObservableCollection<string> IsActive\_Buttons { get => isActive\_Buttons; set => isActive\_Buttons = value; }

public static ObservableCollection<sbyte> SelectedIndex { get => selectedIndex; set => selectedIndex = value; }

public static ObservableCollection<Style> Tab\_Header\_Color { get => tab\_Header\_Color; set => tab\_Header\_Color = value; }

public static ObservableCollection<Style> Current\_Table\_Color { get => current\_Table\_Color; set => current\_Table\_Color = value; }

public static ObservableCollection<string> TimeSpanLabel { get => timeSpanLabel; set => timeSpanLabel = value; }

public static ObservableCollection<string> Table\_With\_Answer { get; set; } = new() { "" };

public static ObservableCollection<string> Table\_With\_Answers { get; set; } = new() { "", "", "", "", "", "", "", "", "", "", };

#endregion

#region Admin\_Region

public static ObservableCollection<User> Users\_Admin { get => users\_Admin; set => users\_Admin = value; }

public static ObservableCollection<Test> Tests\_Admin { get => tests\_Admin; set => tests\_Admin = value; }

public static ObservableCollection<Questions> Questions\_Admin { get => questions\_Admin; set => questions\_Admin = value; }

#endregion

#region Command\_Region

public static RelayCommand Log\_In\_Out { get; } = Log\_In\_Out\_Command();

public static RelayCommand Switch\_Page { get; } = Switch\_Page\_Command();

public static RelayCommand To\_Test { get; } = To\_Test\_Command();

public static RelayCommand Switch\_Tab { get; } = Switch\_Tab\_Command();

public static RelayCommand Switch\_Tab\_Color { get; } = Switch\_Tab\_Color\_Command();

public static RelayCommand Submit\_Answer { get; } = Submit\_Answer\_Command();

public static RelayCommand Update\_Database { get; } = Update\_Database\_Command();

public static RelayCommand Switch\_Tab\_By\_TabItem { get; } = Switch\_Tab\_By\_TabItem\_Command();

public static RelayCommand NULL\_Comm { get; } = NULL\_Command();

public static RelayCommand Close\_App { get; } = Close\_App\_Command();

public static RelayCommand Open\_Browser { get; } = Open\_Browser\_Command();

#endregion

public PagesViewModel()

{

}

}

public partial class PagesViewModel //Fields

{

private static readonly Unit\_Of\_Work unit\_Of\_Work = new();

private static readonly Login\_Page login\_Page = new(); //Awailable page

private static readonly Main\_Page main\_Page = new(); //Awailable page

private static Testing\_Page testing\_Page = new(); //Awailable page

private static readonly Admin\_Page admin\_Page = new(); //Awailable page

private static readonly Rules\_Page rules\_Page = new(); //Awailable page

private static Page current\_Page = login\_Page;

#region Test\_Region

private static readonly Style tab\_Wrong = (Style)Application.Current.FindResource("Tab\_Header\_Wrong");

private static readonly Style tab\_Untouched = (Style)Application.Current.FindResource("Tab\_Header\_Untouched");

private static readonly Style tab\_Right = (Style)Application.Current.FindResource("Tab\_Header\_Right");

private static readonly Style table\_Wrong = (Style)Application.Current.FindResource("Table\_Wrong");

private static readonly Style table\_Untouched = (Style)Application.Current.FindResource("Table\_Untouched");

private static readonly Style table\_Right = (Style)Application.Current.FindResource("Table\_Right");

private static ObservableCollection<Style> current\_Table\_Color = new() { };

private static readonly List<Style> table\_Colors = new(10);

private static ObservableCollection<string> given\_Answer = new() { "" };

private static readonly string[] tests\_List = { "0", "1", "2", "3", "4", "5", "6", "7", }; //list of chapters (0=exam

private static ObservableCollection<sbyte> selectedIndex = new() { 0 }; //Index of tab

private static ObservableCollection<Style> tab\_Header\_Color = new();

private static ObservableCollection<string> login\_Password = new() { "", Fake\_Password };

private static ObservableCollection<string> isActive\_Buttons = new();

private static ObservableCollection<Questions> questions\_For\_Test = new();

private static List<Page> pages\_List = new() { login\_Page, main\_Page, testing\_Page, admin\_Page, rules\_Page };//0, 1, 2, 3, 4

private static readonly List<string> correct\_Answers = new();

private static List<string> given\_Answers = new(10);

private static ObservableCollection<string> timeSpanLabel = new() { "10:00" };

private static readonly List<string> paragraphs\_For\_Test = new();

private static bool IsTest\_Started;

private static string pF = "";

#endregion

#region Admin\_Region

private static ObservableCollection<User> users\_Admin = unit\_Of\_Work.Users.GetAllToObservableCollection();

private static ObservableCollection<Questions> questions\_Admin = unit\_Of\_Work.Questions.GetAllToObservableCollection();

private static ObservableCollection<Test> tests\_Admin = unit\_Of\_Work.Tests.GetAllToObservableCollection();

private static readonly SHA256 sha256 = SHA256.Create();

#endregion

}

public partial class PagesViewModel //ComputeHash

{

public static string ComputeHash(string str)

{

return Encoding.UTF8.GetString(sha256.ComputeHash(Encoding.UTF8.GetBytes(str)));

}

}

}

# **ПРИЛОЖЕНИЕ Е**

Файл Admin\_Page.xaml, код xaml

<Page

x:Class="Tests\_TR.VIEW.PAGES.Admin\_Page"

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:tests\_tr="clr-namespace:Tests\_TR"

Title="Admin\_Page"

d:DataContext="{d:DesignInstance Type=tests\_tr:PagesViewModel}"

d:DesignHeight="450"

d:DesignWidth="800"

mc:Ignorable="d">

<Page.InputBindings>

<KeyBinding

Key="Backspace"

Command="{Binding NULL\_Comm}"

CommandParameter="0" />

</Page.InputBindings>

<Grid Background="Gray">

<Grid.ColumnDefinitions>

<ColumnDefinition />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="0.1\*" />

<RowDefinition Height="0.9\*" />

</Grid.RowDefinitions>

<Button

Width="120"

Height="50"

HorizontalAlignment="Left"

VerticalAlignment="Center"

Background="Green"

Command="{Binding Update\_Database}"

CommandParameter="Прив"

Content="Сохранить данные"

Foreground="White" />

<Button

Grid.Row="0"

Grid.Column="1"

Width="50"

Height="50"

HorizontalAlignment="Right"

VerticalAlignment="Center"

VerticalContentAlignment="Center"

Background="Red"

Command="{Binding Path=Log\_In\_Out}"

CommandParameter="{Binding Pages\_List[0]}"

Content="X"

FontSize="24"

Foreground="White" />

<TabControl

Grid.Row="1"

Grid.Column="0"

Grid.ColumnSpan="2"

Background="Gray">

<TabItem Header="Пользователи">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<DataGrid

AutoGenerateColumns="False"

Background="Gray"

ColumnWidth="\*"

ItemsSource="{Binding Users\_Admin, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Path=Login}" Header="Логин" />

<DataGridTextColumn Binding="{Binding Path=Password}" Header="Пароль" />

<DataGridTextColumn Binding="{Binding Path=Role}" Header="Роль (admin или user)" />

<DataGridTextColumn Binding="{Binding Path=Last\_Name}" Header="Фамилия" />

<DataGridTextColumn Binding="{Binding Path=Name}" Header="Имя" />

<DataGridTextColumn Binding="{Binding Path=Father\_Name}" Header="Отчество" />

</DataGrid.Columns>

</DataGrid>

</Grid>

</TabItem>

<!--<TabItem Header="Тесты">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<DataGrid

AutoGenerateColumns="False"

ItemsSource="{Binding Tests\_Admin, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Path=Id, Mode=TwoWay}" Header="Id" />

<DataGridTextColumn Binding="{Binding Path=Topic, Mode=TwoWay}" Header="Глава" />

<DataGridTextColumn Binding="{Binding Path=Name, Mode=TwoWay}" Header="Название" />

</DataGrid.Columns>

</DataGrid>

<StackPanel

Grid.Row="1"

HorizontalAlignment="Center"

Orientation="Horizontal">

<Button

Command="{Binding Update\_Database}"

CommandParameter=""

Content="Обновить" />

</StackPanel>

</Grid>

</TabItem>-->

<TabItem Header="Вопросы">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<DataGrid

AutoGenerateColumns="False"

ColumnWidth="\*"

ItemsSource="{Binding Questions\_Admin, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Path=Test\_Id, Mode=TwoWay}" Header="ID теста" />

<DataGridTextColumn Binding="{Binding Path=Question, Mode=TwoWay}" Header="Вопрос" />

<DataGridTextColumn Binding="{Binding Path=Answer\_1, Mode=TwoWay}" Header="Ответ 1" />

<DataGridTextColumn Binding="{Binding Path=Answer\_2, Mode=TwoWay}" Header="Ответ 2" />

<DataGridTextColumn Binding="{Binding Path=Answer\_3, Mode=TwoWay}" Header="Ответ 3" />

<DataGridTextColumn Binding="{Binding Path=Answer\_4, Mode=TwoWay}" Header="Ответ 4" />

<DataGridTextColumn Binding="{Binding Path=Right\_Answer, Mode=TwoWay}" Header="Правильный ответ" />

<DataGridTextColumn Binding="{Binding Path=Image, Mode=TwoWay}" Header="Путь к картинке" />

<DataGridTextColumn Binding="{Binding Path=Paragraph, Mode=TwoWay}" Header="Параграф" />

</DataGrid.Columns>

</DataGrid>

</Grid>

</TabItem>

</TabControl>

</Grid>

</Page>

# **ПРИЛОЖЕНИЕ Ж**

Файл Unit\_Of\_Work.cs, код C#

using System.Windows;

namespace Tests\_TR.MODEL

{

internal class Unit\_Of\_Work

{

private readonly DatabaseContext db = new();

private Questions\_Repository? questions\_Repository;

private Tests\_Repository? tests\_Repository;

private Users\_Repository? users\_Repository;

public Questions\_Repository Questions { get => questions\_Repository ??= new(db); }

public Tests\_Repository Tests { get => tests\_Repository ??= new(db); }

public Users\_Repository Users { get => users\_Repository ??= new(db); }

public void Dispose() => db.Dispose();

public async void SaveChangesAsync()

{

try

{

await db.SaveChangesAsync();

MessageBox.Show("Данные успешно добавлены!", "Успех!");

}

catch (System.Exception)

{

MessageBox.Show($"Проверьте правильность введенных данных!" +

$"\nОграничения USER:"

+ "\nЛогин: больше 4 символов, уникален, необходим."

+ "\nПароль: больше 4 символов, уникален необходим."

+ "\nРоль: admin или user, необходима."

+ "\nФамилия: необходима, больше 0 и меньше 30 символов."

+ "\nИмя: необходимо, больше 0 и меньше 30 символов."

+ "\n"

+ "\nОграничения Questions:"

+ "\nID теста: от 0 до 7 включительно."

+ "\nВопрос: больше 0 символов."

+ "\nОтвет1: больше 0 символов."

+ "\nОтвет2: больше 0 символов."

+ "\nПравильный ответ: 1 символ.", "Ошибка сохранения!"

);

}

}

}

}