|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
|  |
| по междисциплинарному курсу: МДК.02.02. Технология разработки и защиты баз данных |
|  |
| на тему: Разработка генератора экзаменационных билетов |
|  |
| студента группы 090203-9o-20/1  специальности 09.02.03 Программирование в компьютерных системах |
| Смирнова Константина Вадимовича |

|  |  |  |
| --- | --- | --- |
| Студент |  | К.В. Смирнов |
| Руководитель курсового проекта |  | Е.А. Ларионова |
| Председатель ПЦК специальности 09.02.03 Программирование в компьютерных системах |  | А.И. Глускер |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. | | |
| Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Заведующий отделением № 1 |  | И.А. Миланова | |

Москва

2023

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc134709767)

[1 ОБЩАЯ ЧАСТЬ 6](#_Toc134709768)

[1.1 Назначения и цели создания системы 6](#_Toc134709769)

[1.2 Обзор и анализ предметной области 8](#_Toc134709770)

[1.3 Жизненный цикл базы данных 9](#_Toc134709771)

[1.4 Выбор и характеристика СУБД 10](#_Toc134709772)

[1.5 Выбор и характеристика среды разработки приложения 11](#_Toc134709773)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 14](#_Toc134709774)

[2.1 Постановка задачи 14](#_Toc134709775)

[2.2 Архитектура информационной системы 15](#_Toc134709776)

[2.3 Логическая модель базы данных 16](#_Toc134709777)

[2.4 Нормализация таблиц 16](#_Toc134709778)

[2.5 Описание таблиц 17](#_Toc134709779)

[2.6 Другие объекты базы данных (представления, хранимые процедуры, триггеры, имена входа, пользователи) 20](#_Toc134709780)

[2.7 Разработка приложения 21](#_Toc134709781)

[2.7.1 Диаграмма вариантов использования 22](#_Toc134709782)

[2.7.2 Решение главной задачи проекта 22](#_Toc134709783)

[2.7.3 Тестирование приложения 31](#_Toc134709784)

[2.7.4 Защита информационной системы 37](#_Toc134709785)

[2.8 Инструкция пользователю 37](#_Toc134709786)

[2.8.1 Общие сведения об информационной системе 38](#_Toc134709787)

[2.8.2 Требования к техническим средствам 38](#_Toc134709788)

[2.8.3 Требования к программным средствам 39](#_Toc134709789)

[2.8.4 Настройка информационной системы 39](#_Toc134709790)

[2.8.5 Формы ввода 40](#_Toc134709791)

[2.8.6 Отчёты 44](#_Toc134709792)

[ЗАКЛЮЧЕНИЕ 46](#_Toc134709793)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 48](#_Toc134709794)

[ПРИЛОЖЕНИЕ A.SQL скрипты на создание и заполнение базы данных, на создание учётных записей пользователей, на назначение привилегий 49](#_Toc134709795)

[ПРИЛОЖЕНИЕ Б. Код программы 72](#_Toc134709796)

ВВЕДЕНИЕ

Когда заходит речь о Word документах, то обычному пользователю или сотруднику, который работает с документацией, приходится заполнять документы самостоятельно, тратя человеко-часы на рутину, уставая и направляя своё внимание на излишние действия, что сказывается на усталости человека. Такой подход может привести к наличию сотрудников, которых при автоматизации заполнения документов можно заменить, сэкономив, направив бюджет и силы компании на более полезные проекты или повысить зарплату других сотрудников.

Использование приложения, которое позволяет взаимодействовать через интуитивно понятный и простой интерфейс поможет избежать лишней работы, в следствие повышения скорости и удобства заполнения документации, используя шаблоны с тегами, вместо которых будут подставляться записи из базы данных.

Основная задач проекта исходя основываясь на предметной области - создание базы данных для автоматизированной генерации экзаменационных билетов. В базе данных имеется информации и билетах, вопросах в билетах, учителях, дисциплинах и специальности, от выбора которых зависят перечисленные таблицы. Так же имеется промежуточная таблица, в которой содержатся первичные ключи от курсов, семестров, протоколов и председателей пцк. Предусмотрена таблица пользователей для реализации ограничений в зависимости от роли.

Актуальность курсовой работы обусловлена тем, что при получении навыков в создание Word документов по средству языка c#, можно автоматизировать создание необходимой документации, что положительно скажется на продуктивности, скорости и качестве производимых документов. В данном курсовом проекте осуществляется комплексное решение задач, направленных на достижение этих целей.

Цель исследования: систематизировать теоретические знания создания базы данных, взаимодействия с Entity Framework, Word.

Задачи курсовой работы:

1. исследовать, изучить, проанализировать и использовать источники по работе Word и базы данных
2. Применение полученной информации на практике для получения практических навыков

Результатом выполнения курсовой работы представлены в виде c# приложения и теоретический подготовки студента. При разработке приложения были рассмотрены существующие приложения в данной предметной области.

1. ОБЩАЯ ЧАСТЬ
   1. Назначения и цели создания системы

Назначение: Система предназначена для генерации экзаменационных билетов

Целью создания системы является получение опыта в работе с базой данных, реализация учебного плана, C#, WPF, Entity Framework и библиотеками Word.

* 1. Обзор и анализ предметной области

Основная задача проектируемого приложения состоит в создание экзаменационного билета на основе выбранных пользователем данных. Система должна предусматривать авторизацию пользователей и администраторов, регистрацию, формирование билетов, редактирование таблиц вопросов, пользователей и дисциплин.

Формирование билета в приложение состоит из выбора: специальности, дисциплины, семестра, количества билетов, преподавателя, председателя цикловой комиссии, курса, протокола и сложности вопроса. Соответственно возможно наличие нескольких билетов в базе данных, которые отличаются только вопросами. В соответствии с предметной областью система строится с учётом следующих особенностей:

1. Предоставить пользователю выбор составляющих для формирования билетов.
2. Формировать билеты в формате Word документ
3. Пользователь, у которого роль является администратор, может:
4. Просматривать таблицы пользователей, дисциплин, вопросов и билетов
5. Добавлять записи в таблицы пользователей, дисциплин, вопросов и билетов
6. Редактировать записи в таблице пользователей, дисциплин, вопросов и билетов

При открытии страницы, на которой осуществляется выбор данных для формирования билета, должны выводиться следующие характеристики:

1. Специальность
2. Дисциплина
3. Семестр
4. Количество билетов, от числа которых зависит сколько будет сформировано билетов
5. Преподаватель
6. Председатель цикловой комиссии
7. Курс
8. Протокол
9. Сложность вопроса

Базовые сущности предметной области:

Пользователи. Атрибуты пользователей – номер пользователя, логин, пароль, роль. Для хранения данных о логинах и паролей администраторов используется таблица Пользователи. Разница между функционалом приложения для администратора и клиента определяется полем роль. Администраторы имеют доступ к просмотру таблиц, добавлять и изменять данные в таблицах.

Комплект билетов. Атрибуты комплекта билетов – курс, семестр, протокол, идентификатор председателя пцк.

Вопросы. Атрибуты вопросов – идентификатор вопросов, идентификатор дисциплины, вопрос, тип вопроса, сложность. Вопросы напрямую связаны с Билетами.

Билеты. Атрибуты билетов – идентификатор билетов, номер комплект, идентификатор вопроса 1, идентификатор вопроса 2, идентификатор вопроса 3, идентификатор учителя. Билеты являются главной таблицей, на основе которых формируется Word документ.

Логическая модель базы данных представлена на Рисунке 2.2.

Возможности приложения определялись исходя из рассмотрения уже существующих вариантов.

Рассмотренные приложения:

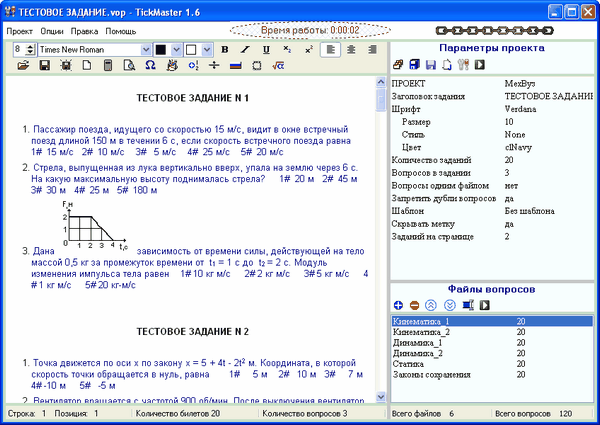


Рисунок 1.1. Приложение TickMaster

Основное назначение программы - генератор билетов - создание билетов из предварительно подготовленной базы вопросов - RTF файлов, содержащих вопросы.



Рисунок 1.2. Логотип приложения EasyTestMaker

EasyTestMaker - программа, которая автоматически создает экзаменационные билеты с различными типами вопросов, включая множественный выбор, короткие ответы и эссе. Она также позволяет генерировать ответы на вопросы, тестовые листы и ключи.

* 1. Жизненный цикл базы данных

Жизненный цикл базы данных — это процесс создания, сопровождения и удаления базы данных. Жизненный цикл включает в себя несколько этапов:

1. Анализ и проектирование базы данных: на этом этапе определяются требования к базе данных и создается концептуальная модель данных. Затем проектируется логическая модель данных и создаются схемы базы данных.
2. Разработка базы данных: на этом этапе создаются таблицы, индексы и другие объекты базы данных.
3. Тестирование и отладка: на этом этапе выполняется тестирование базы данных для выявления ошибок и багов.
4. Реализация и развертывание: база данных развертывается на реальной платформе и настраивается для работы.
5. Мониторинг и оптимизация: на этом этапе производится мониторинг базы данных для определения производительности и выявления проблем. В случае необходимости производится оптимизация базы данных.
6. Сопровождение и обслуживание: на этом этапе выполняются задачи по управлению базой данных, включая резервное копирование, восстановление данных в случае сбоев и обновление версии базы данных.
7. Удаление: на этом этапе база данных полностью удаляется из системы и выполняется очистка всех связанных с ней объектов.
   1. Выбор и характеристика СУБД

Во время разработки информационной системы использовалась концепция клиент-серверной архитектуры, которая включает в себя наличие серверов и клиентских устройств, которые используют определённые сервисы.

Данная архитектура была использована из-за того, что серверы и клиенты имеют возможность одновременно работать на разных системах в сети, что позволяет распределять нагрузку и обеспечивать более высокую доступность и производительность приложений.

Разделение приложения на клиентскую и серверную части позволяет изменять или обновлять каждую часть независимо от другой.

Серверная часть приложения может обеспечивать более высокий уровень безопасности с помощью механизмов аутентификации, авторизации и шифрования данных.

Клиент-серверная архитектура позволяет централизованно управлять хранилищами данных, масштабировать приложения по мере необходимости, добавляя дополнительные серверы или клиенты, оптимизировать производительность приложения, разделяя задачи между клиентом и сервером. обеспечивает возможность добавления новых функций и модулей без необходимости изменения всего приложения.

Для реализации клиент-серверной архитектуры была выбрана СУБД MS SQL Server, ибо предоставляется бесплатный доступ. Также ПО MS SQL Server Management Studio для работы с MS SQL Server упрощает разработку, так как предоставляет удобный и интуитивно понятный интерфейс, а также имеет встроенный компилятор для кода на SQL языке.

Сравнение SQL Server с другими СУБД представлено на Рисунке 1.4.



Рисунок 1.3. Сравнение СУБД

* 1. Выбор и характеристика среды разработки приложения

ADO.NET и Entity Framework являются двумя разными подходами к решению задач, связанных с доступом к данным в приложениях, использующих технологии .NET.

ADO.NET - это набор библиотек для работы с данными в .NET, который включает в себя классы, используемые для подключения к базам данных, выполнения запросов и получения результатов. ADO.NET предоставляет более прямой доступ к данным, но с его использованием не всегда удобно и быстро писать код.

Entity Framework - это ORM фреймворк для .NET, который использует подход Code First (код первым). Он предоставляет набор инструментов, с помощью которых можно создавать сущности данных, отображающие таблицы в базе данных и хранить их в коде приложения. Entity Framework позволяет работать с данными в объектно-ориентированном стиле, что упрощает и ускоряет разработку приложений.

Однако существует множество факторов, влияющих на выбор между ADO.NET и Entity Framework, такие как опыт разработчиков, сложность приложения, требования к быстродействию и т.д. В каждом конкретном случае необходимо выбрать тот инструмент, который позволяет достичь наилучших результатов при оптимальных затратах времени и усилий.

Для разработки приложения был выбран Entity Framework в связи с удобством выполнения запросов в языке программирования c#, позволяя представить данные как реальные объекты, что позволяет абстрагироваться от базы данных и её таблиц, работая с данными как с объектами.

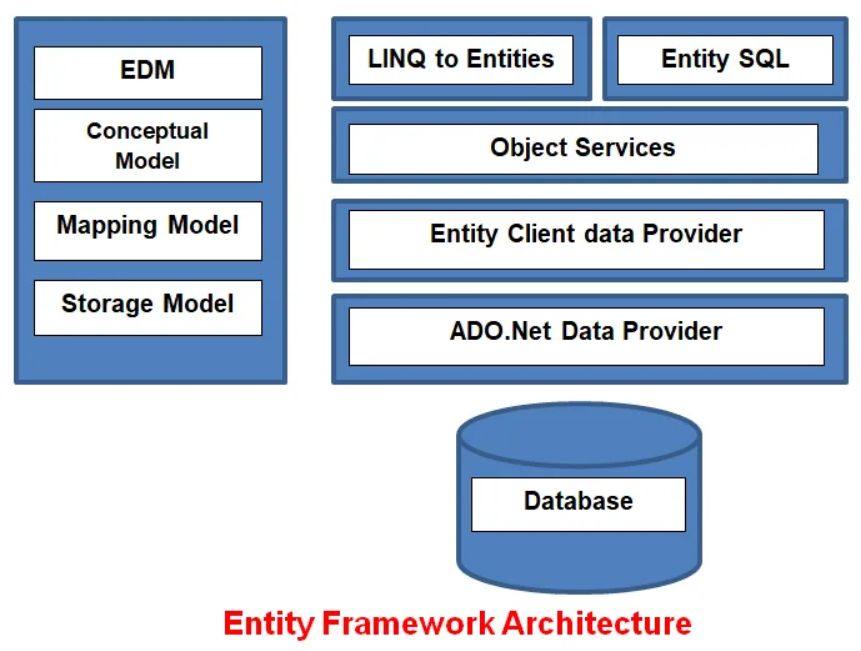


Рисунок 1.4. Схема работы Entity Framework

Entity Framework - инструмент ORM (Object-Relational Mapping), позволяющий взаимодействовать с базами данных, используя объектно-ориентированный подход.

Минусами Entity Framework являются:

1. Большой объем данных, который нужно хранить в памяти при использовании внутренних объектов Entity Framework

Плюсами Entity Framework являются:

1. Простота в использовании и повышение уровня абстракции
2. Можно работать с данными на уровне объектов вместо работы непосредственно с запросами SQL

ADO.NET позволяет взаимодействовать с базами данных с помощью языка программирования C# и языка SQL

Минусами ADO.NET являются:

1. Неудобство работы с объектами
2. Много повторяющихся блоков кода
3. Необходимость написания SQL-запросов

Плюсами ADO.NET являются:

1. Простота в использовании, для работы достаточно хорошо знать SQL язык
2. Основной нужный функционал включен в ядро .NET Framework
3. Стабильность и быстрота работы
4. Простота в использовании, для работы достаточно хорошо знать SQL язык

Исходя из вышеперечисленного, выбор был сделан в пользу Entity Framework.

В качестве среды разработки была выбрана Microsoft Visual Studio.

Visual Studio содержит все необходимые инструменты и функции, такие как редактор кода, отладчик, компилятор. Имеет мощный отладчик, который позволяет находить и исправлять ошибки в коде. Он также поддерживает множество отладочных функций, таких как точки остановок, просмотр переменных и значения, трассировка стека вызовов и др.

Интерфейс Visual Studio разработан, чтобы сделать работу программиста простой и удобной. Инструменты и функции доступны благодаря простой структуре меню и панелей инструментов.

1. **Специальная часть**
   1. Постановка задачи

Спроектировать средствами MS SQL SERVER базу данных и создать приложение, позволяющее ввести вопросы   по дисциплинам для проведения комплексного экзамена и сформировать по ним билеты.  Предусмотреть, чтобы

1. На  одной странице  помещается  1 билет.
2. Число билетов выбирается пользователем.
3. В одном билете были вопросы по одинаковой дисциплине.
4. Ни один вопрос не должен дублироваться.
5. Число вопросов равняется 3
6. Билеты формируются с учётом сложности вопросов.

Предусмотреть возможность добавления, удаления, редактирования записей. Билеты должны быть представлены в виде отчёта или текстового файла. Обеспечить защиту базы данных и приложения.

В дополнение к вышеперечисленным этапам курсовая работа включает следующие задачи:

1. Разработка графического интерфейса для приложения, чтобы сделать его более удобным для пользователя.
2. Использование администраторов и пользователей для создания градации по функциональным возможностям при работе программы.
3. Просмотр, добавление и редактирование записей в таблицах.
4. Обеспечить защиту базы данных и приложения.

Курсовая работа будет включает в себя сочетание навыков программирования и знаний по управлению базами данных для создания практического и полезного программного приложения.

* 1. Архитектура информационной системы

Архитектура информационной системы для генератора экзаменационных билетов включает разработку и внедрение системы программного обеспечения, которую можно использовать для эффективного производства экзаменационных билетов.

База данных вопросов: хранит набор всех доступных вопросов вместе с дисциплиной, типом вопроса и уровнем сложности.

Компоненты, входящие в состав архитектуры информационной системы книжного магазина, будут включать клиентское приложение C# WPF, серверную часть и базу данных MS SQL Server. Клиентское приложение обеспечит пользователю взаимодействие с системой, предоставив пользователям удобный вид приложения для ввода параметров билета, таких как специальность, дисциплина, количество вопросов и другие, а также отобразит сгенерированный экзаменационный документ в формате Word для дальнейшего использования пользователем. В приложение предусмотрена авторизация и регистрация пользователей.

Генератор экзаменационных билетов реализует алгоритм, основанный на пользовательском вводе, который будет случайным образом выбирать билеты, которые доступны после выбора пользователем.

Систем разработана с использованием комбинации языков программирования, таких как c#, и системой управления базами данных MSSQL. Пользовательский интерфейс разработан с использованием таких библиотек wpf.

Архитектура информационной системы для генератора экзаменационных билетов требует сочетания принципов проектирования программного обеспечения, знаний по управлению базами данных и понимания процесса создания билета для разработки эффективной и полезной программной системы.

* 1. Логическая модель базы данных

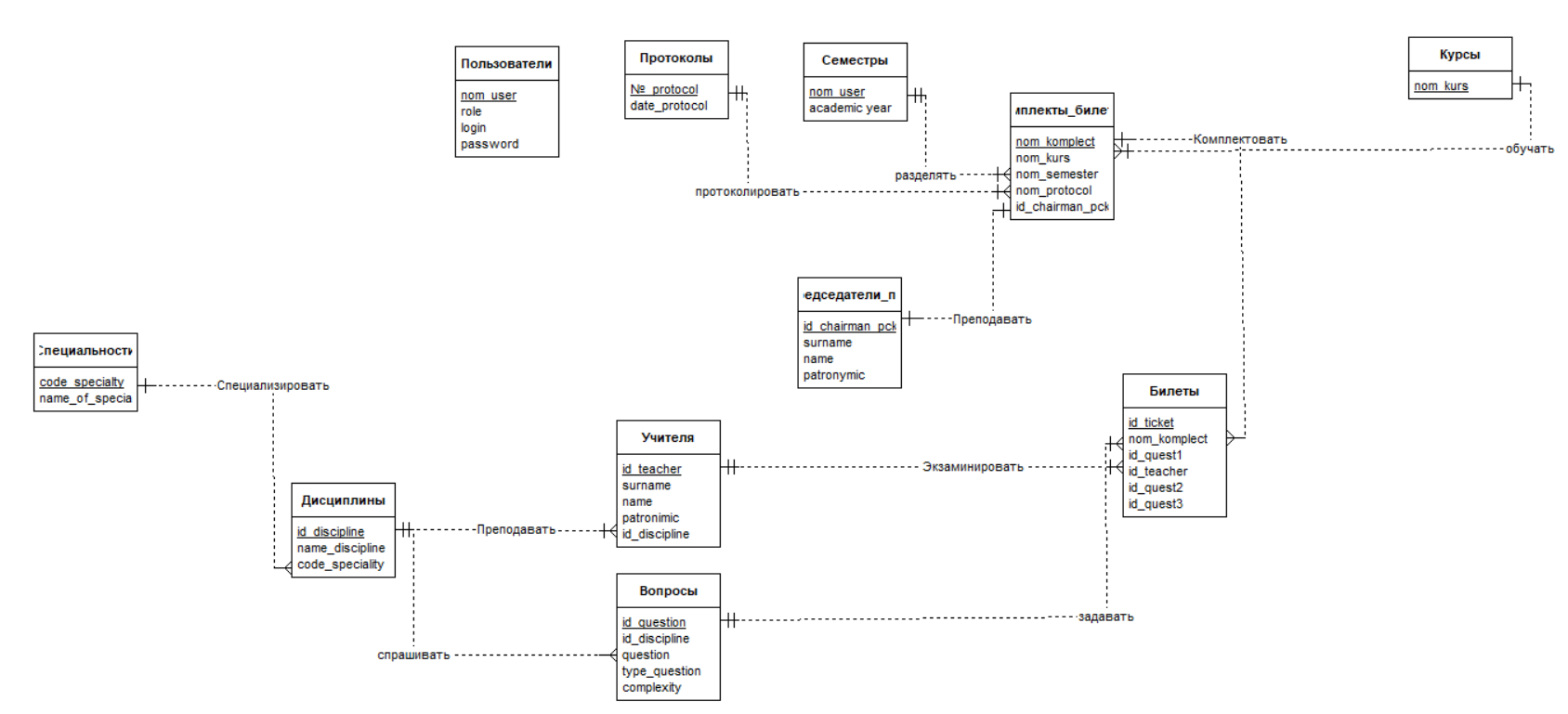


Рисунок 2.1. Логическая модель данных

* 1. Нормализация таблиц

Нормализация таблиц — это процесс проектирования базы данных с целью устранения повторений данных и установления связей между таблицами.

Как правило, нормализация таблиц проходит через несколько стадий, от первой нормальной формы до третьей нормальной формы. На каждой стадии устраняются определенные типы повторений данных и устанавливается определенный тип связи между таблицами.

Таблицы базы данных были приведены к третьей нормальной форме, в которой минимизировано повторение данных, устраняется аномалия данных, обеспечивается ссылочная целостность и относительно простое управление данными. Сущности в таблицах, соответствующих третьей нормальной форме, зависят только от первичного ключа и элементы таблиц являются неделимыми.

Все таблицы в базе данных приведены к третьей нормальной форме так как:

1. Каждое поле содержит только одно значение и что все поля уникальны, вся информация разделена на более мелкие и более точные единицы, что обеспечивает более эффективное и точное управление базой данных. (Первая нормальная форма)
2. Каждое поле таблиц зависит от первичного ключа. Это позволяет обеспечить более эффективное управление базой данных, так как можно избежать дублирования информации и других аномалий данных. (Вторая нормальная форма)
3. Отсутствует транзитивная зависимость. Транзитивная зависимость - неключевые столбцы зависят от значений других неключевых столбцов. Это позволяет избежать аномалий данных. (Третья нормальная форма)
   1. Описание таблиц

Таблица 1 – Users (Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| nom\_user | Номер пользователя | Int | FK |
| login\_ | Логин | nvarchar(50) |  |
| password\_ | Пароль | nvarchar(50) |  |
| role\_ | Роль | nvarchar(5) |  |

Таблица 2 – Kurs (Курсы)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| nom\_kurs | Номер курса | Int | FK |

Таблица 3 – Semesters (Семестры)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| nom\_semester | Номер семестра | Int | FK |
| academic\_year | Академический год | Date |  |

Таблица 4 – Protocols (Протоколы)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| nom\_protocol | Номер протокола | Int | FK |
| date\_protocol | Дата протокола | Date |  |

Таблица 5 – Chairman\_pck (Председатель\_пцк)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| id\_chairman\_pck | Идентификатор председателя ПЦК | Int | FK |
| Surname | Фамилия | nvarchar(100) |  |
| name\_ | Имя | nvarchar(100) |  |
| patronymic | Отчество | nvarchar(100) |  |

Таблица 6 – Komplect\_tickets (Комплекты\_билетов)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| nom\_komplect | Идентификатор комплекта билетов | Int | FK |
| nom\_kurs | Номер курса | Int |  |
| nom\_semester | Номер семестра | Int |  |
| nom\_protocol | Номер протокола | Int |  |
| id\_chairman\_pck | Идентификатор Предсседателя\_пцк | Int |  |

Таблица 7 – Speciality (Специальности)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| code\_speciality | Код специальности | nvarchar(100) | FK |
| name\_of\_speciality | Название специальности | nvarchar(100) |  |

Таблица8 – Disciplines (Дисциплины)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| id\_discipline | Код специальности | int | FK |
| name\_discipline | Название специальности | nvarchar(500) |  |
| code\_speciality | Код специальности | nvarchar(100) |  |

1. Таблица 9 – Disciplines (Дисциплины)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| id\_question | Идентификатор вопроса | int | FK |
| id\_discipline | Идентификатор дисциплины | Int |  |
| Question | Вопрос | nvarchar(500) |  |
| type\_question | Тип вопроса | nvarchar(14) |  |
| Complexity | Сложность вопроса | nvarchar(8) |  |

Таблица 10 – Teacher (Преподаватель)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| id\_teacher | Идентификатор учителя | int | FK |
| Surname |  | nvarchar(100) |  |
| name\_ | Вопрос | nvarchar(100) |  |
| Patronymic | Тип вопроса | nvarchar(100) |  |
| id\_discipline | Идентификатор дисциплины | int |  |

Таблица 11 – Tickets (Билеты)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| id\_ticket | Идентификатор билета | int | FK |
| id\_quest1 |  | int |  |
| id\_quest1 | Вопрос | int |  |
| id\_quest1 | Тип вопроса | int |  |
| nom\_komplect | Идентификатор дисциплины | int |  |
| id\_teacher | Идентификатор учителя | int |  |

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

Другие объекты базы данных не используются.

* 1. Разработка приложения

Разработано приложение На основе предметной области. Создание происходило в несколько этапов.

На первом этапе был определён функционал приложения и была изучена информация о среде разработки Microsoft Visual Studio, языке C#, системы WPF и Entity Framework.

В функционал приложения вошли:

* Добавление записей в таблицы;
* Удаление записей в таблицах;
* Редактирование записей в таблицах;
* Просмотр информации из таблиц.

На следующем этапе был разработан относительно удобный и интуитивно понятный интерфейс, для простоты взаимодействия пользователя с приложением.

Далее, после преодоления проблем, связанных с проектированием, была начата работа над реализацией приложения.

Для работы с базой данных были реализованы запросы на языке LINQ, которые вызываются в приложении при выборе пользователем данных.

На следующем этапе было проведено тестирование приложения и отладка.

* + 1. Диаграмма вариантов использования

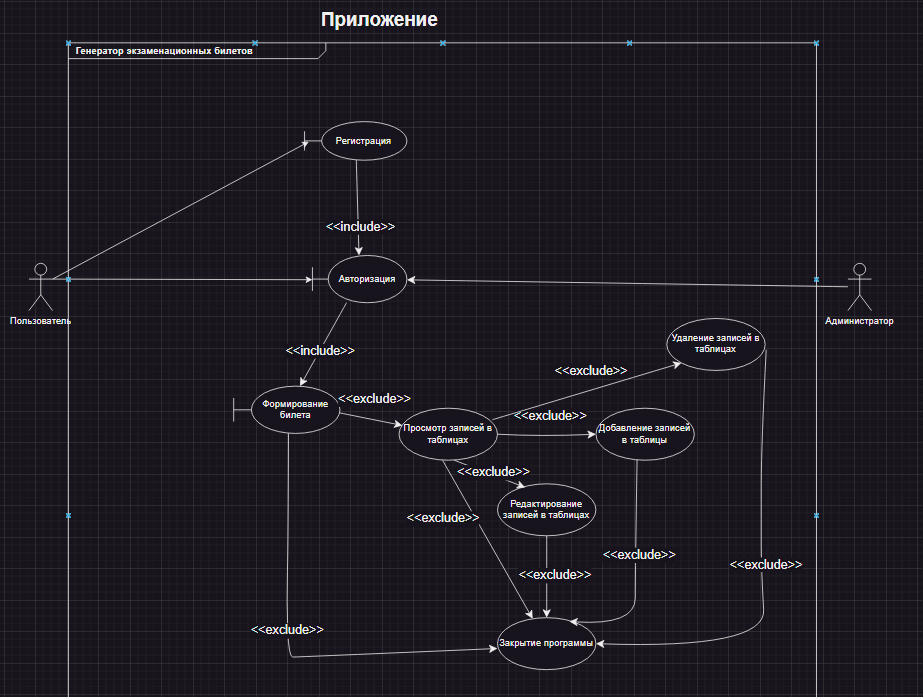


Рисунок 2.3. Диаграмма вариантов использования.

2.7.2 Решение главной задачи проекта

Главная задача генератора экзаменационных билетов состоит в создании приложения, которое будет позволять эффективно управлять всеми процессами: авторизация пользователя, просмотр записи в таблице, редактирование записи в таблице, добавление записи в таблице, удаление записи из таблицы.

1. Реализовать вход пользователя в аккаунт

Код на c#:

/// <summary>

/// авторизация пользователя

/// </summary>

private void But\_authorization(object sender, RoutedEventArgs e)

{

string loginUser = textBox\_login.Text;

string roleUser = Role.Text;

string passUser = textBox\_password.Password.ToString();

if ((loginUser != "") && (roleUser != "") && (passUser != ""))

{

if (roleUser == "Admin" || roleUser == "User") //работает Admin

{

//сверяем введённые данные и данные в бд

if (db.Users.Any(o => (o.login\_ == loginUser) && (o.password\_ == passUser) && (o.role\_ == roleUser)))

{

MessageBox.Show("Успешная авторизация");

NavigationService.Navigate(new Choice\_admin(roleUser));

}

else

{

MessageBox.Show("Неправильный логин или пароль");

}

}

else

{

MessageBox.Show("Неверно указана роль пользователя");

}

}

else MessageBox.Show("Для продолжения заполните все поля");

}

Выполнение в приложении указано на Рисунке 2.8.

1. Реализовать формирование билета

Код на c#:

private void But\_Click\_Form\_Ticket(object sender, RoutedEventArgs e)

{

if (FindSpecId(FindSpecialityId()).ToString() == "")

{

MessageBox.Show("Выберите Дисциплину");

}

Initialize\_questions();

string disca\_content = Disca.Text;

var helper = new WordHelper("Ex\_Ticket\_Prac.docx");

string count\_tickets = count\_of\_tickets.Text;

string teacher\_content = Teacher.Text;

string Chairman\_pck\_content = Chairman.Text;

string kurs\_content = Kurs.Text;

string semester\_content = Semester.Text;

string speciality\_content = Spec.Text;

string protocol\_content = Protocol.Text;

string protocol\_date\_content = GetProtDateString();

string sem\_year\_content = GetSemYearString();

int nom\_ticket = 1;

int t;

if (int.TryParse(count\_tickets, out t))

{

if (t < 1)

{

MessageBox.Show("Число билетов должно быть >= 1");

return;

}

}

else

{

MessageBox.Show("Неверный формат");

return;

}

try

{

for (int i = 0; i < Convert.ToInt32(count\_tickets); i++)

{

var ticket = new List<string>(FindQuestions(NextTicketId(FindKomplectId(GetKursId(), GetSemesterId(), GetProtocolsId()))));

string quest1 = ticket[0];

string quest2 = ticket[1];

string quest3 = ticket[2];

var Items = new Dictionary<string, string>

{

{"<DISC>", disca\_content},

{"<PCK>", Chairman\_pck\_content},

{"<PREP>", teacher\_content},

{"<KURS>", kurs\_content},

{"<SEM>", semester\_content},

{"<SPEC>", speciality\_content},

{"<NOMPROT>", protocol\_content},

{"<DATEPROT>", protocol\_date\_content},

{"<YEARSEM> ", sem\_year\_content},

{"<NOMTICK>", nom\_ticket.ToString()},

{"<TEO1>", quest1},

{"<TEO2>", quest2},

{"<PRAC1>", quest3},

};

helper.Process(Items);

nom\_ticket++;

MessageBox.Show($"Билет №{nom\_ticket} сформирован");

}

}

catch (Exception ex)

{

MessageBox.Show("Билеты закончились");

}

}

/// <summary>

/// формирование word документа

/// </summary>

internal bool Process(Dictionary<string, string> items)

{

using (var doc = WordprocessingDocument.Create("FilePath", WordprocessingDocumentType.Document))

{

MainDocumentPart mainPart = doc.AddMainDocumentPart();

mainPart.Document = new Document();

Body body = mainPart.Document.AppendChild(new Body());

Paragraph para = body.AppendChild(new Paragraph());

Run run = para.AppendChild(new Run());

run.AppendChild(new Text("this new text for test"));

}

Word.Application app = null;

try

{

app = new Word.Application();

Object file = fileinfo\_.FullName;

Object missing = Type.Missing;

app.Documents.Open(file);

foreach (var item in items)

{

Word.Find find = app.Selection.Find;

find.Text = item.Key;

find.Replacement.Text = item.Value;

Object wrap = Word.WdFindWrap.wdFindContinue;

Object replace = Word.WdReplace.wdReplaceAll;

find.Execute(FindText: Type.Missing,

MatchCase: false,

MatchWholeWord: false,

MatchWildcards: false,

MatchSoundsLike: missing,

MatchAllWordForms: false,

Forward: true,

Wrap: wrap,

Format: false,

ReplaceWith: missing, Replace: replace

);

}

Object newFileName = Path.Combine("C:\\VS Projects\\Commit\\Kursach\\Kursach\\WpfApp1\\Tickets\\",

DateTime.Now.ToString("HHmmss") + fileinfo\_.Name);

app.ActiveDocument.SaveAs2(newFileName);

app.ActiveDocument.Close();

return true;

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

return false;

}

finally

{

if (app != null)

{

app.Quit();

}

}

}

1. Реализовать просмотр информации таблицы

Код на c#:

private void ViewingTable\_admin\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

RandomTicketGenerator.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridQuestion.ItemsSource = RandomTicketGenerator.GetContext().Questions.ToList();

}

}

1. Реализовать добавление записи в таблицу

Код на c#:

private void But\_Click\_Save\_Question(object sender, RoutedEventArgs e)

{

var currentQuest = GetQuestions();

if (string.IsNullOrWhiteSpace(currentQuest.question))

{

MessageBox.Show("Корректно напишите вопрос");

return;

}

RandomTicketGenerator.GetContext().Questions.Add(currentQuest);

try

{

RandomTicketGenerator.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

1. Изменения записи в таблице

Код на c#:

private void UpdateQuestions()

{

\_selectedQuestion.id\_discipline = GetDisciplineId();

\_selectedQuestion.question = question\_textbox.Text;

\_selectedQuestion.type\_question = Type\_question.Text;

\_selectedQuestion.complexity = Complexity\_question.Text;

}

private void But\_Click\_Save\_Question(object sender, RoutedEventArgse)

{

if (string.IsNullOrWhiteSpace(\_selectedQuestion.question))

{

MessageBox.Show("Корректно напишите вопрос");

return;

}

UpdateQuestions();

try

{

RandomTicketGenerator.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

2.7.3 Тестирование приложения

Таблица 3 – Метод проверки требований к приложению

1. Тестирование клиентской части

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действие | Результат | Ожидаемый результат |
| 1 | Запуск приложения | Приложение подключается к серверу и базе данных | Приложение подключается к серверу и базе данных |
| 2 | Пользователь нажимает на кнопку «Авторизация» | Пользователю показывается уведомление о том, что не все поля заполнены | Пользователю показывается уведомление о том, что не все поля заполнены |
| 3 | Пользователь вводит данные и нажимает кнопку «Авторизация» | Система проверяет, есть ли в базе данных пользователь с таким логином и паролем. Если нет, то выводится уведомление, что логин или пароль не верны. Если да -открывается страница формирования билета | Система проверяет, есть ли в базе данных пользователь с таким логином и паролем. Если нет - уведомление, что логин или пароль не верны. Если да -открывается страница формирования билета |
| 4 | Пользователь нажимает на кнопку «Регистрация» | Пользователю показывается уведомление о том, что не все поля заполнены | Пользователю показывается уведомление о том, что не все поля заполнены |
| 5 | Пользователь вводит данные и нажимает кнопку «Регистрация» | Регистрация пользователя и переход на страницу авторизации | Регистрация пользователя и переход на страницу авторизации |
| 4 | Пользователь выбирает доступные специальности | Система отправляет на сервер запрос списка специальностей и предоставляет его пользователю | Система отправляет на сервер запрос списка специальностей и предоставляет его пользователю |
| 5 | Пользователь нажимает на кнопку подтверждение специальности | Система отображает доступные дисциплины | Система отображает доступные дисциплины |
| 6 | Пользователь выбирает доступные дисциплины | Система отправляет на сервер запрос списка специальностей и предоставляет его пользователю | Система отправляет на сервер запрос списка специальностей и предоставляет его пользователю |
|  | Пользователь выбирает доступные семестры | Система отправляет на сервер запрос списка семестров и предоставляет его пользователю | Система отправляет на сервер запрос списка курсов и предоставляет его пользователю |
| 7 | Пользователь выбирает количество билетов | Система проверяет, правильно ли указаны введённые данные. Если нет, то выводится уведомление о неверном формате. Если да – в переменную сохраняется число формируемых билетов | Система проверяет, правильно ли указаны введённые данные. Если нет, то выводится уведомление о неверном формате. Если да – в переменную сохраняется число формируемых билетов |
| 8 | Пользователь выбирает доступных преподавателей | Система отправляет на сервер запрос списка преподавателей и предоставляет его пользователю | Система отправляет на сервер запрос списка преподавателей и предоставляет его пользователю |
| 9 | Пользователь выбирает доступных председателей пцк | Система открывает пользователю окно корзины и отображает товары в ней | Система открывает пользователю окно корзины и отображает товары в ней |
| 10 | Пользователь выбирает доступные семестры | Система предлагает пользователю сохранить чек и после этого приложение завершает работу | Система предлагает пользователю сохранить чек и после этого приложение завершает работу |
|  | Пользователь выбирает доступные протоколы | Система предлагает пользователю сохранить чек и после этого приложение завершает работу | Система предлагает пользователю сохранить чек и после этого приложение завершает работу |
|  | Пользователь выбирает сложность | Система предлагает пользователю сохранить чек и после этого приложение завершает работу | Система предлагает пользователю сохранить чек и после этого приложение завершает работу |

1. Тестирование части администратора

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действие | Результат | Ожидаемый результат |
| 1 | Запуск приложения | Приложение подключается к серверу и базе данных | Приложение подключается к серверу и базе данных |
| 2 | Пользователь нажимает на кнопку «Войти» | Пользователю высвечивается уведомление о том, что не все поля заполнены | Пользователю высвечивается уведомление о том, что не все поля заполнены |
| 3 | Пользователь вводит данные и нажимает кнопку «Войти» | Система проверяет, есть ли в базе данных администратор с таким логином и паролем. Если нет, то пользователю выводится уведомление о том, что логин или пароль не верны. Если да, то открывается страница администратора | Система проверяет, есть ли в базе данных администратор с таким логином и паролем. Если нет, то пользователю выводится уведомление о том, что логин или пароль не верны. Если да, то открывается страница администратора |
| 4 | Пользователь нажимает на кнопку «Регистрация» | Пользователю показывается уведомление о том, что не все поля заполнены | Пользователю показывается уведомление о том, что не все поля заполнены |
| 5 | Пользователь вводит данные и нажимает кнопку «Регистрация» | Пользователь вводит данные и нажимает кнопку «Регистрация» | Регистрация пользователя и переход на страницу авторизации |
| 6 | Пользователь выбирает доступные специальности | Система отправляет на сервер запрос списка специальностей и предоставляет его пользователю | Система отправляет на сервер запрос списка специальностей и предоставляет его пользователю |
| 7 | Пользователь нажимает на кнопку подтверждение специальности | Система отображает доступные дисциплины | Система отображает доступные дисциплины |
|  | Адмнистратор нажимает на кнопку “Просмотр база дынных” | Переход на страницу вопросов и просмотр таблицы вопросов | Переход на страницу вопросов и просмотр таблицы вопросов |
|  | Администратор нажимает на кнопку “Редактировать” | Редактирование таблицы вопросов | Редактирование таблицы вопросов |
|  | Администратор нажимает на кнопку “Добавить” | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет |
|  | Администратор нажимает на кнопку “Удалить” | Удаление вопроса | Удаление вопроса |
|  | Пользователь нажимает на кнопку “Таблица дисциплин” | Переход на страницу “Просмотр дисциплин” | Переход на страницу “Просмотр дисциплин” |
|  | Администратор нажимает на кнопку “Редактировать” | Редактирование таблицы дисциплин | Редактирование таблицы дисциплин |
|  | Администратор нажимает на кнопку “Добавить” | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет |
|  | Администратор нажимает на кнопку “Удалить” | Удаление дисциплины | Удаление дисциплины |
|  | Пользователь нажимает на кнопку “Таблица Пользователей” | Переход на страницу “Просмотр пользователей” | Переход на страницу “Просмотр пользователей ” |
|  | Администратор нажимает на кнопку “Редактировать” | Редактирование таблицы пользователей | Редактирование таблицы пользователей |
|  | Администратор нажимает на кнопку “Добавить” | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет |
|  | Администратор нажимает на кнопку “Удалить” | Удаление пользователей | Удаление пользователей |
|  | Пользователь нажимает на кнопку “Таблица Билетов” | Переход на страницу “Просмотр билетов” | Переход на страницу “Просмотр билетов ” |
|  | Администратор нажимает на кнопку “Редактировать” | Редактирование таблицы билетов | Редактирование таблицы билетов |
|  | Администратор нажимает на кнопку “Добавить” | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет | Система проверяет, заполнено ли поле ввода.  Если да, система добавляет |
|  | Администратор нажимает на кнопку “Удалить” | Удаление билетов | Удаление билетов |

* + 1. Защита информационной системы

Для обеспечения безопасности в базах данных использовались средства защиты в MS SQL Server и Entity Framework.

Использованные средства:

Защита от инъекций: Entity Framework позволяет использовать LINQ запросы, чтобы предотвратить ошибки при работе с данными.

* 1. Инструкция пользователю

Инструкция для пользователя:

1. Для начала использования приложения необходимо авторизоваться. Для этого введите логин и пароль в соответствующие поля на странице авторизации и нажмите кнопку "Войти".
2. Если у вас нет аккаунта, нажмите кнопку "Регистрация" и заполните поля Роль, Логин и Пароль. После этого вы сможете авторизоваться в приложении.
3. После авторизации переход на главную страницу приложения. На этой странице представлены составляющие билета. Выберите данные для формирования билета. Первоначально нужно выбрать специальность, чтобы получить доступ к выбору дисциплины и преподавателя. Также можно нажать кнопку "Авторизация", чтобы перейти к выбору пользователя.
4. После корректного выбора составляющих билета, при нажатии кнопки “Формирование билета” происходит формирование экзаменационного билета.
5. При нажатии на кнопку “Просмотр базы данных”, если при авторизации была указана роль Admin, то происходит переход на страницу просмотра вопросов. Можно просмотреть имеющиеся вопросы, добавить, редактировать или добавить записи.
6. При нажатии на кнопку “Таблица дисциплин происходит переход на страницу просмотра дисциплин. Можно просмотреть имеющиеся дисциплины, добавить, редактировать или добавить записи.
7. При нажатии на кнопку “Таблица пользователей” происходит переход на страницу просмотра пользователей. Можно просмотреть имеющихся пользователей, добавить, редактировать или добавить записи.
8. При нажатии на кнопку “Таблица билетов” происходит переход на страницу просмотра билетов. Можно просмотреть имеющиеся билеты, добавить, редактировать или добавить записи.
   * 1. Общие сведения об информационной системе

Генератор экзаменационных билетов– это комплект программных средств, которые помогают автоматизировать составление билетов.

Основными элементами информационной системы являются:

1. Компьютер с установленным на него соответствующим программным обеспечением, предназначенным для учета заказов и хранения информации о пользователях.
2. Приложение, написанное на C# WPF с использованием технологии Entity Framework, которое реализует подключение к базе данных генератора экзаменационного билета и позволяет получать из неё данные и изменять их.
3. База данных MS SQL Server, которая хранит в себе данные

Информационной системы позволяет повысить эффективность работы, что позволяет экономить время и ресурсы. Также, система упрощает работу и обработку данных о билетах.

* + 1. Требования к техническим средствам

Для корректной работы приложения требуется соблюдать данные технические характеристики:

1. Минимальный объём оперативной памяти: 4 ГБ
2. Минимальные требования к монитору: Super VGA с разрешением 800x600 пикселей или более высоким.
3. Свободное место на диске: 30 МБ (Без установки SQL Server)
4. Доступ в Интернет
5. Тип процессора: AMD Opteron, AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T, Intel Pentium IV с поддержкой EM64T.
6. Быстродействие процессора: частота 2,0 ГГц и выше
   * 1. Требования к программным средствам

Для корректной работы приложения, программное обеспечение должно соответствовать данному списку:

1. Операционная система: Windows 10 или Windows 11
2. MS SQL Server Management Studio
3. База данных в MS SQL Server
4. .NET Framework
5. Последние драйверы для системы
   * 1. Настройка информационной системы

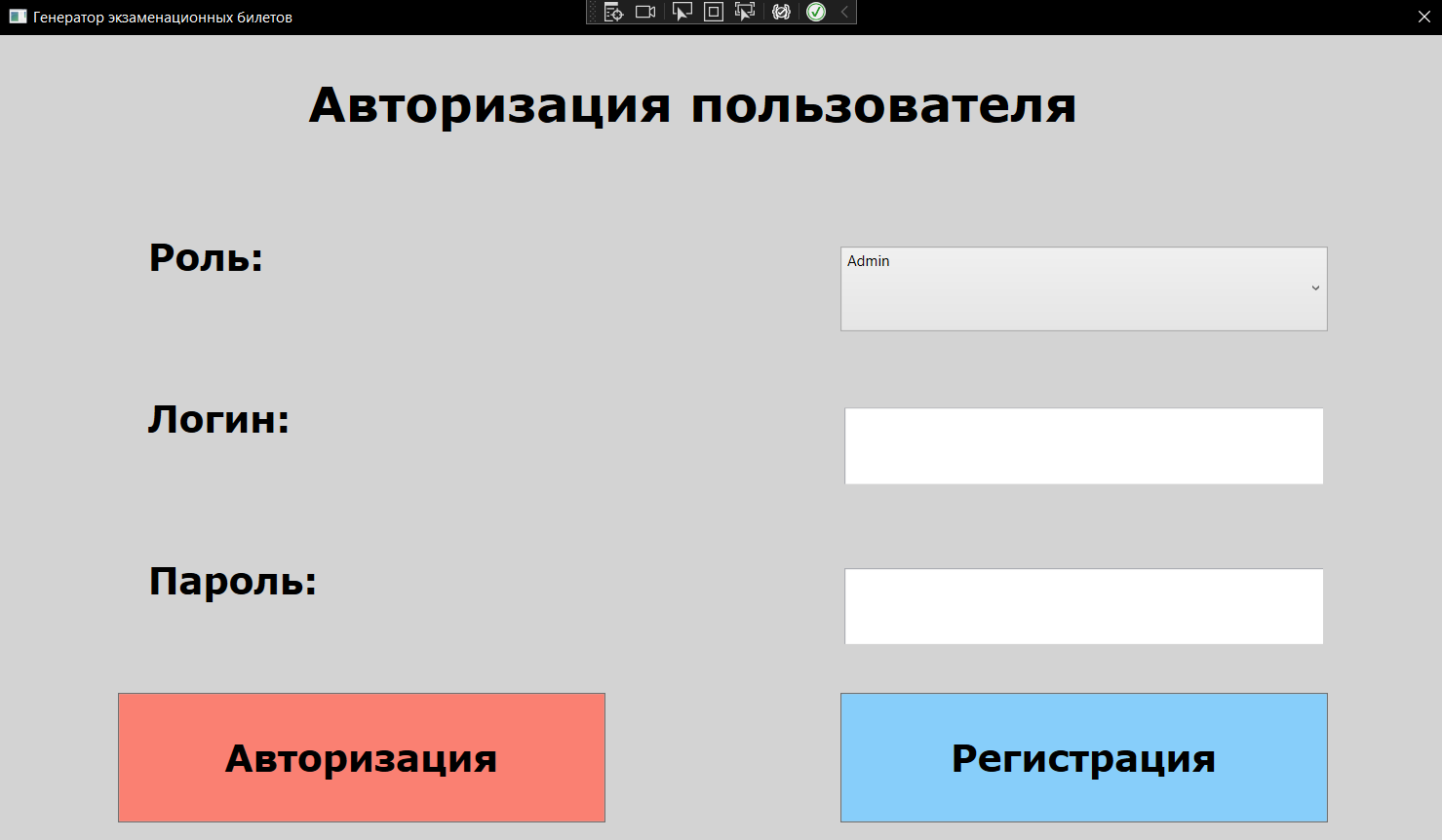
Для интеграции информационной системы

Генератора экзаменационных билетов для работы понадобится:

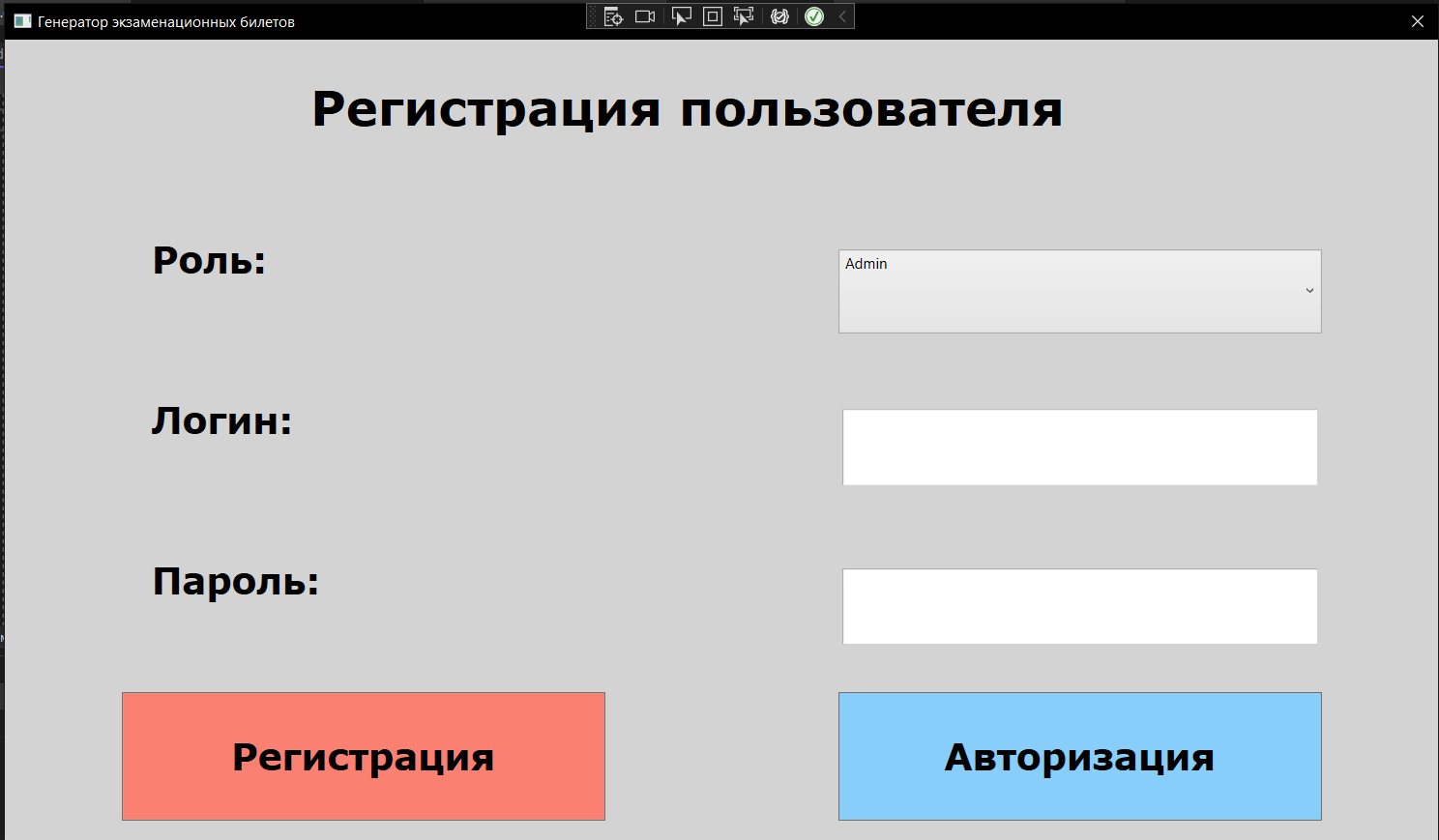
1. Проверить, что система соответствует минимальным системным требованиям.
2. Установить всё необходимое программное обеспечение.
3. Настроить MS SQL Server.
4. Выполнить скрипты на создание базы данных.
5. Заменить строку подключения в приложении.
6. Раздать администраторам логины и пароли для доступа к базе данных.
7. Распространить копии приложения преподавателям или другим пользователям.
   * 1. Формы ввода

В информационной системе генератора экзаменационных билетов использовались следующие формы ввода:

1. Форма авторизации.

Рисунок 2.4. Форма авторизации

1. Форма регистрации, на которую пользователь переходи в случае отсутствия данных для авторизации. После регистрации переход на страницу авторизации.

Рисунок 2.5. Форма регистрации

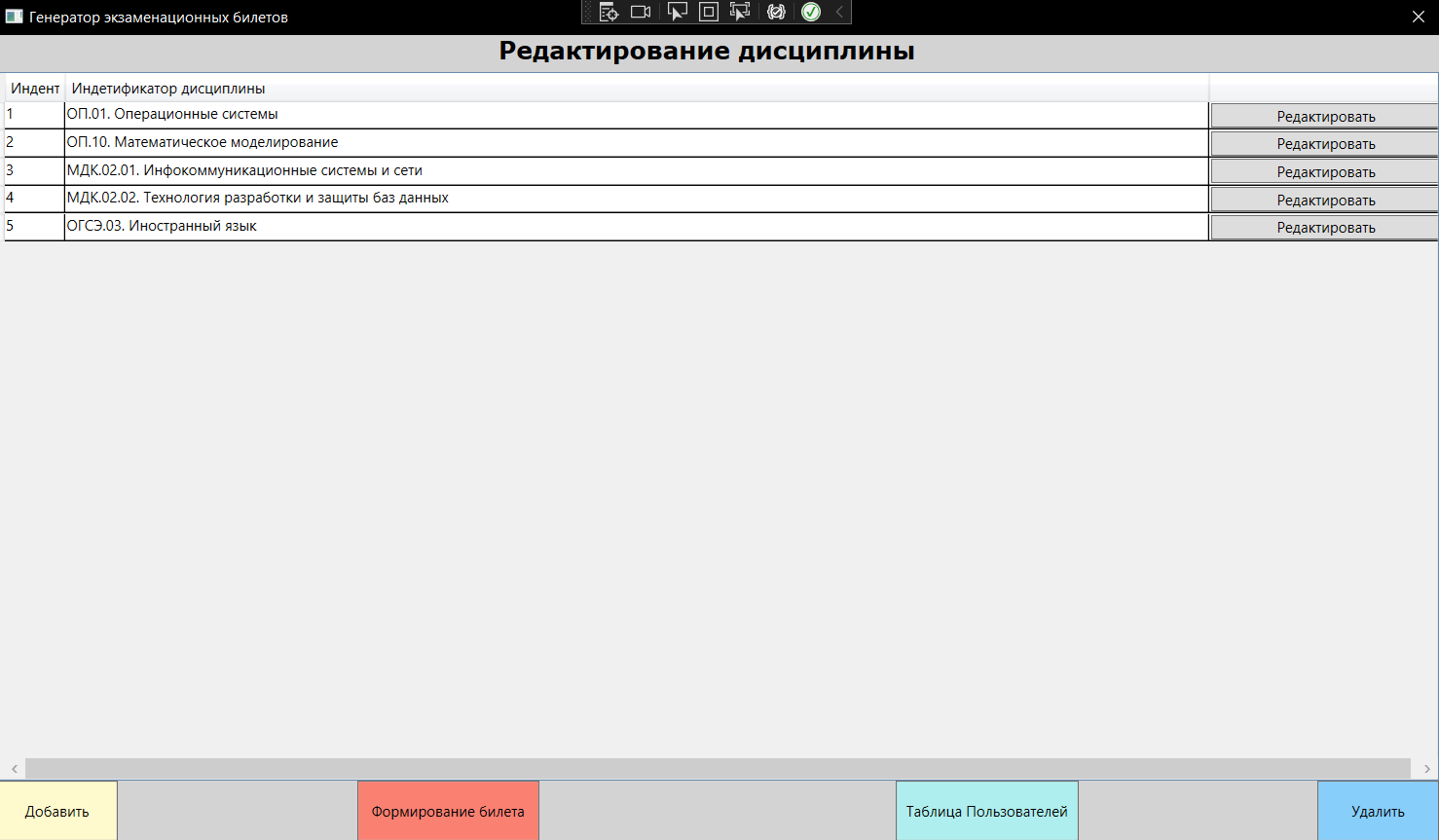
1. Форма формирования билета, в которой пользователь может выбрать данные, которые будут в билете.

Рисунок 2.6. Форма формирования билета

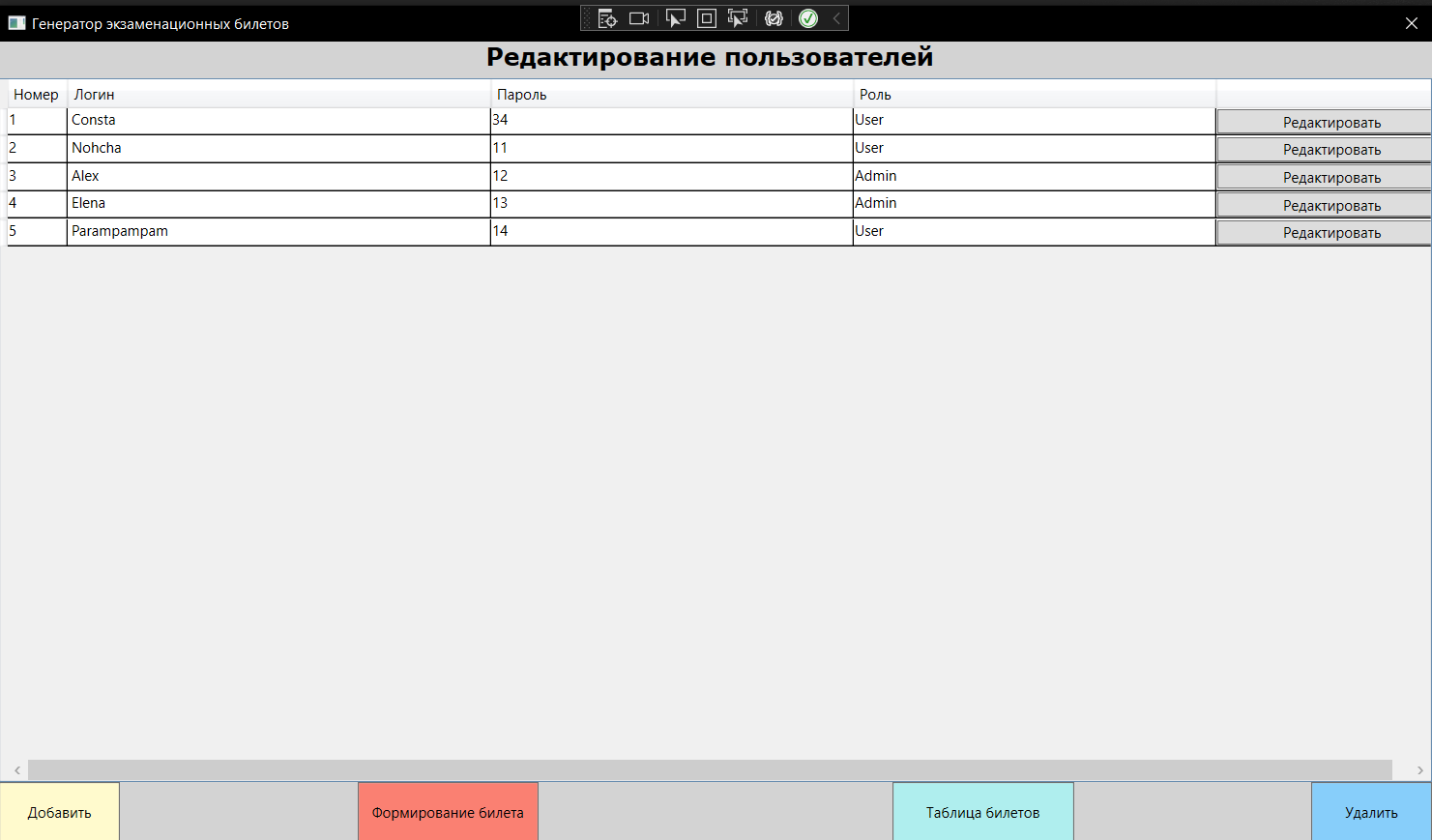
1. Форма просмотра вопросов, в которой можно просмотреть, добавить, редактировать, удалить данные из таблицы вопросов.

Рисунок 2.6. Форма просмотра вопросов

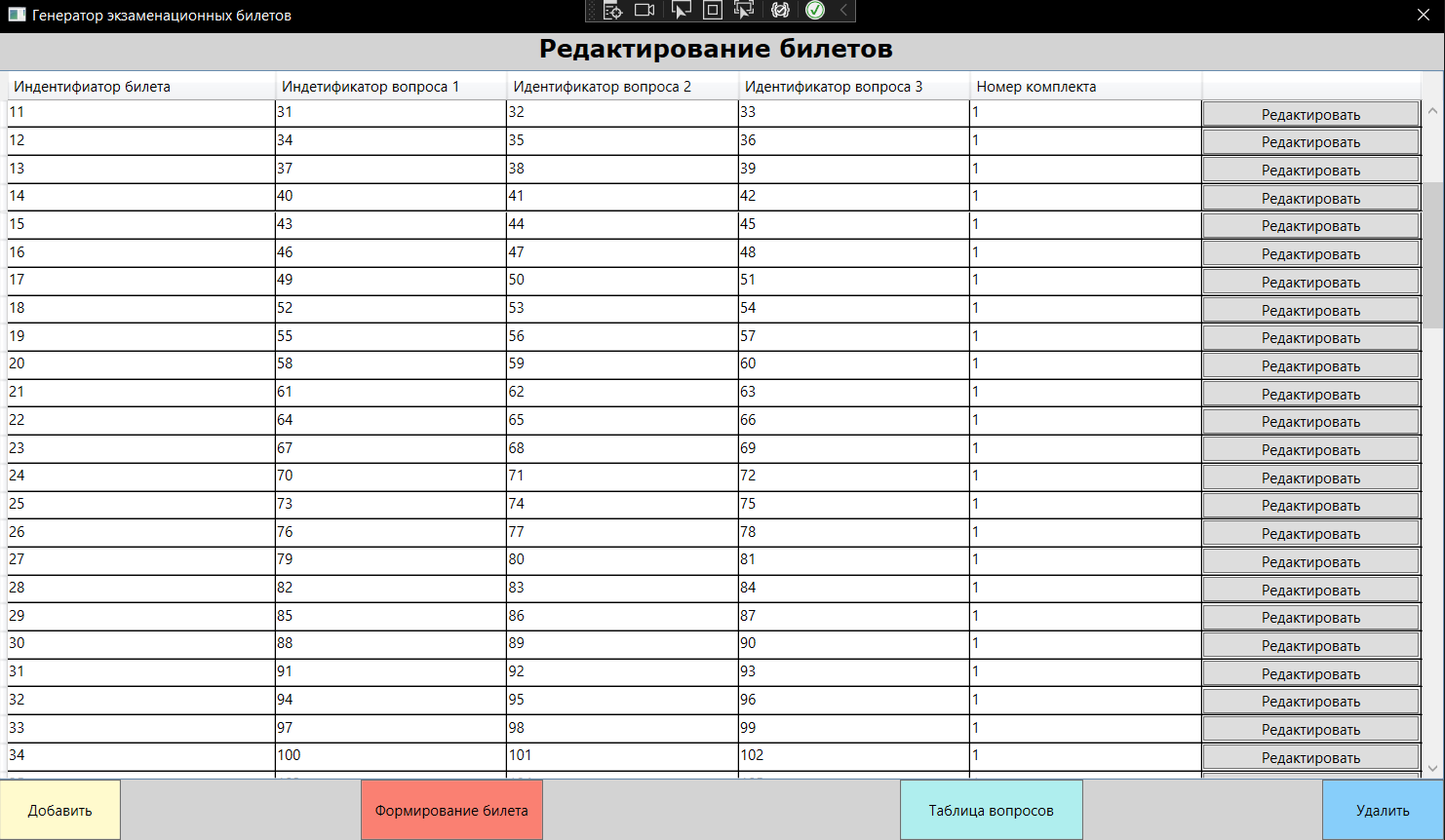
1. Форма просмотра дисциплин, в которой можно просмотреть, добавить, редактировать, удалить данные из таблицы дисциплин.

Рисунок 2.6. Форма просмотра дисциплин

1. Форма просмотра пользователей, в которой можно просмотреть, добавить, редактировать, удалить данные из таблицы дисциплин.

Рисунок 2.6. Форма просмотра пользователей

1. Форма просмотра билетов, в которой можно просмотреть, добавить, редактировать, удалить данные из таблицы пользователей.

Рисунок 2.6. Форма просмотра билетов

2.8.6 Отчёты

При нажатие на кнопку “Формирование билета” создаётся word документ с указанными пользователем данными

Шаблон билета:

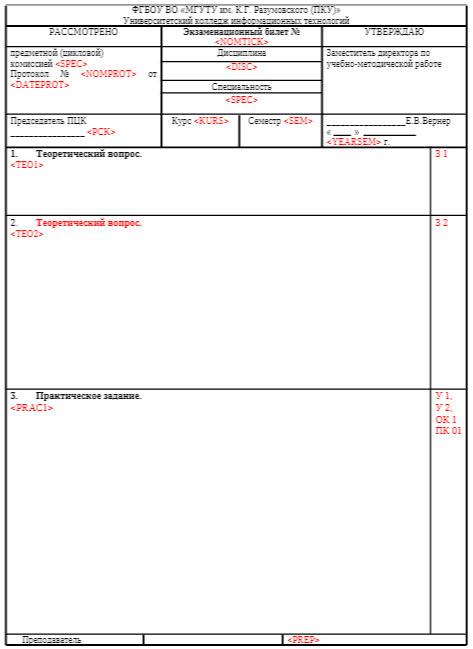
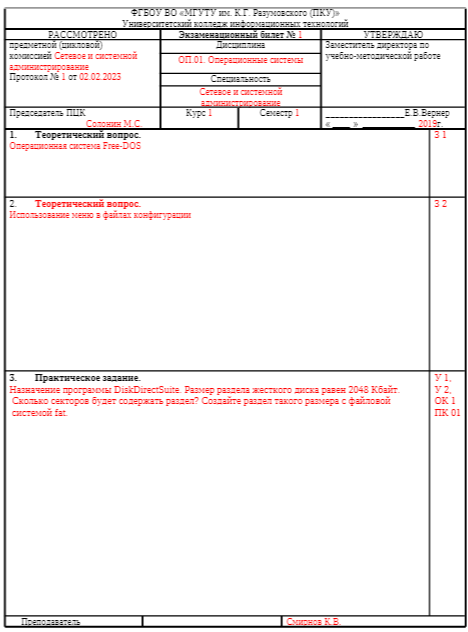


Рисунок 2.13. Шаблон билета

Сформированный билет:

Рисунок 2.14. Сформированный билет



ЗАКЛЮЧЕНИЕ

В заключении курсовой работы по созданию генератора экзаменационных билетов на языке C# с использованием технологии Entity Framework и базы данных MSSQL, можно сказать, что данная работа была успешно выполнена. В процессе работы были изучены основные принципы работы с технологией Entity Framework и инструментами, используемыми для работы с ней, включая LINQ-запросы. Была разработана программа, которая позволяет генерировать экзаменационные билеты для преподавателей на основе заданных критериев, таких как специализация, курс, дисциплина, семестр, преподаватель, председатель пцк, курс, протокол, сложность вопросов и количество вопросов в билете. Были использованы современные методы программирования, такие как C# и база данных MSSQL.

Были преодолены сложности с проблемами формирования word документа, корректного отображения элементов таблиц, значений по умолчанию, с добавлением, удалением и редактированием записей, авторизацией, регистрацией, и с запросами.

Получен опыт в проектирование и сделан вывод, что понимание на раннем этапе того, из чего должна состоять предметная область – способствует более быстрой реализации поставленной задачи.

Результаты работы демонстрируют важность и актуальность использования технологии Entity Framework для упрощения работы с базами данных и выполнения сложных запросов. В итоге, генератор экзаменационных билетов на языке C# с использованием технологии Entity Framework и базы данных MSSQL является полезным инструментом для любой организации, проводящей массовую проверку знаний студентов или сдающих сертификационные экзамены.

В будущем планируется усовершенствование дизайна и предоставление возможности выбора количества вопросов.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Программные решения для бизнеса. Информационные и коммуникационные технологии: <https://nationalteam.worldskills.ru/skills/programmnye-resheniya-dlya-biznesa/>
2. Основы LINQ: <https://metanit.com/sharp/tutorial/15.1.php>
3. Руководство по WPF: <https://metanit.com/sharp/wpf/>
4. SQL запросы быстро: <https://habr.com/ru/articles/480838/>
5. Microsoft.Office.Interop.Word Пространство имен: <https://learn.microsoft.com/ru-ru/dotnet/api/microsoft.office.interop.word?view=word-pia>
6. Работа с Word файлом через Interop.Word. Замена слов в шаблоне: <https://www.youtube.com/watch?v=vQ7uW6g0z-U&t=938s>
7. Уроки C# – LINQ – Where, Select, GroupBy, AsParallel, x.Key – C#: <https://www.youtube.com/watch?v=gF4X3yr0nsA&t=886s>
8. Элемент ComboBox: <https://metanit.com/sharp/windowsforms/4.7.php>
9. C# WPF: <https://www.youtube.com/watch?v=LkYB9gLzKxc&t=291s>
10. Урок 3: Подключение базы к приложению, аутентификация пользователей | WPF + MSSQL: <https://www.youtube.com/watch?v=7SoudUnDk5k&t=677s>
11. Getting selected value of a combobox: <https://stackoverflow.com/questions/6901070/getting-selected-value-of-a-combobox>
12. ЧТО ТАКОЕ ENTITY FRAMEWORK? <https://www.youtube.com/watch?v=QPuFSSVxM4I&t=1087s>

**ПРИЛОЖЕНИЕ SQL скрипты на создание и заполнение базы данных.**

USE master;

IF EXISTS (SELECT \* FROM SYS.DATABASES WHERE NAME = 'Base')

DROP DATABASE Base

GO

CREATE DATABASE Base

ON

( NAME = Base\_data,

FILENAME = 'C:\VS Projects\Commit\Kursach\Create\_base\_script\All\Basedata.mdf',

SIZE = 8,

MAXSIZE = 100,

FILEGROWTH = 10 )

LOG ON

( NAME = Base\_log,

FILENAME = 'C:\VS Projects\Commit\Kursach\Create\_base\_script\All\Baselog.ldf',

SIZE = 5MB,

MAXSIZE = 25MB,

FILEGROWTH = 5MB );

GO

USE Base;

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Users')

DROP TABLE Users

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Kurs')

DROP TABLE Kurs

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Semesters')

DROP TABLE Semesters

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Protocols')

DROP TABLE Protocols

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Chairman\_pck')

DROP TABLE Chairman\_pck

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Speciality')

DROP TABLE Speciality

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Disciplines')

DROP TABLE Disciplines

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Teacher')

DROP TABLE Teacher

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Komplect\_tickets')

DROP TABLE Komplect\_tickets

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Questions')

DROP TABLE Questions

IF EXISTS (SELECT \* FROM SYSOBJECTS WHERE NAME = 'Tickets')

DROP TABLE Tickets

CREATE TABLE Users

(

nom\_user INT IDENTITY PRIMARY KEY,

login\_ nvarchar(50),

password\_ nvarchar(50),

role\_ nvarchar(5)

);

INSERT INTO Users (login\_, password\_, role\_)

VALUES ('Consta', 34, 'User'),

('Nohcha', 11,'User'),

('Alex', 12,'Admin'),

('Elena', 13,'Admin'),

('Parampampam', 14,'User');

CREATE TABLE Kurs

(

nom\_kurs INT PRIMARY KEY

);

INSERT INTO Kurs (nom\_kurs)

VALUES (1),

(2),

(3),

(4),

(5);

CREATE TABLE Semesters

(

nom\_semester INT IDENTITY PRIMARY KEY,

academic\_year DATE

);

INSERT INTO Semesters (academic\_year)

VALUES ('01.01.2019'),

('05.05.2020'),

('01.01.2021'),

('05.05.2022'),

('01.01.2023');

CREATE TABLE Protocols

(

nom\_protocol INT IDENTITY PRIMARY KEY,

date\_protocol date,

);

--DELETE FROM Protocols;

INSERT INTO Protocols (date\_protocol)

VALUES ('02.02.2023'),

('03.03.2023'),

('04.04.2023'),

('05.05.2023'),

('06.06.2023');

CREATE TABLE Chairman\_pck

(

id\_chairman\_pck INT IDENTITY PRIMARY KEY,

surname nvarchar(100),

name\_ nvarchar(100),

patronymic nvarchar(100)

);

INSERT INTO Chairman\_pck (surname, name\_, patronymic)

VALUES

('Солонин', 'Марк', 'Семёнович'),

('Суворов', 'Виктор', 'Богданович'),

('Ларионова', 'Елена', 'Анатольевна'),

('Ленин', 'Владимир', 'Ильич'),

('Керенский', 'Александр', 'Фёдорович');

CREATE TABLE Speciality

(

code\_speciality nvarchar(100) PRIMARY KEY,

name\_of\_speciality nvarchar(100),

);

INSERT INTO Speciality (code\_speciality, name\_of\_speciality)

VALUES ('09.02.06', 'Сетевое и системной администрирование'),

('09.02.07', 'Информационные системы и программирование'),

('10.02.05', 'Обеспечение информационной безопасности автоматизированных систем'),

('21.02.19', 'Землеустройство'),

('42.02.01', 'Реклама');

CREATE TABLE Disciplines

(

id\_discipline int IDENTITY PRIMARY KEY,

name\_discipline nvarchar(500),

code\_speciality nvarchar(100),

FOREIGN KEY (code\_speciality) REFERENCES Speciality (code\_speciality) ON DELETE CASCADE ON UPDATE CASCADE,

);

INSERT INTO Disciplines (name\_discipline, code\_speciality)

VALUES ('ОП.01. Операционные системы', '09.02.06'),

('ОП.10. Математическое моделирование', '09.02.07'),

('МДК.02.01. Инфокоммуникационные системы и сети', '10.02.05'),

('МДК.02.02. Технология разработки и защиты баз данных', '21.02.19'),

('ОГСЭ.03. Иностранный язык', '42.02.01');

CREATE TABLE Teacher

(

id\_teacher INT IDENTITY PRIMARY KEY,

surname nvarchar(100),

name\_ nvarchar(100),

patronymic nvarchar(100),

id\_discipline int,

FOREIGN KEY (id\_discipline) REFERENCES Disciplines (id\_discipline)

ON DELETE CASCADE ON UPDATE CASCADE

);

INSERT INTO Teacher (surname, name\_, patronymic, id\_discipline)

VALUES

('Смирнов', 'Константин', 'Вадимович', 1),

('Ларионова', 'Елена', 'Анатольевна', 1),

('Мурашов', 'Анатолий', 'Алексеевич', 1),

('Глускер', 'Александр', 'Игоревич', 1),

('Храбров', 'Илья', 'Николаевич', 1),

('Марцинкевич', 'Максим', 'Сергееевич', 2),

('Сталин', 'Иосиф', 'Виссарионович', 2),

('Троцкий', 'Лев', 'Давидович', 3),

('Бухарин', 'Николай', 'Иванович', 4),

('Зиновье', 'Григорий', 'Евсеевич', 5);

CREATE TABLE Komplect\_tickets

(

nom\_komplect INT IDENTITY PRIMARY KEY,

nom\_kurs INT,

nom\_semester INT,

nom\_protocol INT,

id\_chairman\_pck INT,

id\_teacher INT,

FOREIGN KEY (nom\_kurs) REFERENCES Kurs (nom\_kurs) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (nom\_semester) REFERENCES Semesters (nom\_semester) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (nom\_protocol) REFERENCES Protocols (nom\_protocol) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_chairman\_pck) REFERENCES Chairman\_pck (id\_chairman\_pck) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_teacher) REFERENCES Teacher (id\_teacher) ON DELETE CASCADE ON UPDATE CASCADE,

);

INSERT INTO Komplect\_tickets (nom\_kurs, nom\_semester, nom\_protocol, id\_chairman\_pck, id\_teacher)

VALUES (1, 1, 1, 1, 1),

(2, 2, 2, 2, 2),

(3, 3, 3, 3, 3),

(4, 4, 4, 4, 4),

(5, 5, 5, 5, 5);

CREATE TABLE Questions

(

id\_question INT IDENTITY,

id\_discipline int,

question nvarchar(500),

type\_question nvarchar(14),

complexity nvarchar(8),

PRIMARY KEY(id\_question),

FOREIGN KEY (id\_discipline) REFERENCES Disciplines (id\_discipline)

ON DELETE CASCADE ON UPDATE CASCADE

);

INSERT INTO Questions (id\_discipline, question, type\_question, complexity)

VALUES

--ОП.01. Операционные системы практические и теоретические

--сложные вопросы

(1, 'Назначение и функции операционной системы.', 'Теоретический', 'Сложный'),

(1, 'WIMP-интерфейс. SILK-интерфейс. Виды реализации.', 'Теоретический', 'Сложный'),

(1, 'Порядок загрузки операционной системы.

Управление параметрами загрузки операционной системы.', 'Практический', 'Сложный'),

(1, 'Файловая система Linux: структура каталогов файловой системы. Основные каталоги Linux',

'Теоретический', 'Сложный'),

(1, 'Состояние потоков на разных этапах их разработки.Алгоритм планирования процессов

основанный на относительных приоритетах', 'Теоретический', 'Сложный'),

(1, 'Настройка параметров рабочей среды пользователя', 'Практический', 'Сложный'),

(1, 'Типы адресов (символьные, виртуальные, физические). Классификация методов

распределения оперативной памяти', 'Теоретический', 'Сложный'),

(1, 'Понятие оперативной памяти. Распределение памяти перемещаемыми разделами',

'Теоретический', 'Сложный'),

(1, 'Выполнение конфигурирования аппаратных

устройств. Распределение ресурсов', 'Практический', 'Сложный'),

(1, 'Свопинг, как частный случай виртуальной памяти', 'Теоретический', 'Сложный'),

(1, 'Понятие виртуальной памяти. Сегментно - страничное распределение.', 'Теоретический', 'Сложный'),

(1, 'Управление оперативной памятью. Распределение физической памяти', 'Практический', 'Сложный'),

--12

(1, 'Файловая система FAT. Логические области раздела FAT', 'Теоретический', 'Сложный'),

(1, 'Понятие кэш-памяти. Принцип действия кэш-памяти', 'Теоретический', 'Сложный'),

(1, 'Динамическая настройка виртуальной памяти в Linux Ubuntu', 'Практический', 'Сложный'),

(1, 'Составные модули ядра ОС.', 'Теоретический', 'Сложный'),

(1, 'Администрирование учётных записей и групп в ОС Linux.', 'Теоретический', 'Сложный'),

(1, 'Изучение структуры файловой системы. Управление дисками и файловыми системами',

'Практический', 'Сложный'),

(1, 'Управление разделением ресурсов в локальной сети', 'Теоретический', 'Сложный'),

(1, 'Работа с программами архивации файлов', 'Теоретический', 'Сложный'),

(1, 'Использование утилит для работы с файлами', 'Практический', 'Сложный'),

----21

(1, 'Команды обслуживания каталогов. Оболочка NC ', 'Теоретический', 'Сложный'),

(1, 'Нарисуйте цепочку кластеров файла, если его длина 3410 байт, размер кластера – 2 сектора, а

при его записи были свободны только кластеры 5, 7, 10, 13, 14, 18.', 'Теоретический', 'Сложный'),

(1, 'Определите (в %) размер основного раздела диска, если дополнительный (расширенный)

раздел состоит из двух логических разделов размером 30% и 20 %. Запустите программу

DiskDirectSuite. Разбейте диск на указанные разделы.', 'Практический', 'Сложный'),

(1, 'Выведите листинг файлов /etc/passwd и /etc/group.

Дайте расшифровку их содержимого', 'Теоретический', 'Сложный'),

(1, 'Найдите ветвь реестра

HKEY\_CURRENT\_USER\ControlPanel\Desktop\WindowsMetrics и экспортируйте ее в каталог

своей группы.', 'Теоретический', 'Сложный'),

(1, 'Запустите редактор реестра regedit. Найдите ветвь реестра

HKEY\_CURRENT\_USER\ControlPanel\Desktop\WindowsMetrics и экспортируйте ее в каталог

своей группы. Измените параметр IconSpasing на 25. Перезагрузите машину и посмотрите

результат. Импортируйте сохраненный ранее файл в реестр. Для чего необходимо

производить экспорт и импорт ветвей реестра?', 'Практический', 'Сложный'),

(1, 'Установите на виртуальную машину операционную систему Linux.', 'Теоретический', 'Сложный'),

(1, 'Создайте группу news. Создайте учетную запись monthly. Добавьте комментарии к учетке.

Измените имя учетки с monthly на weekly.', 'Теоретический', 'Сложный'),

(1, 'Запустите утилиту KSysGuard. Каковы основные возможности утилиты? Запустите любой

пользовательский процесс. Пошлите процессу сигнал останова? Как это повлияло на работу

процесса? Пошлите процессу сигнал SIGCONT. Что произошло с процессом? Завершите

работу процесса.', 'Практический', 'Сложный'), --10 билетов

--30

----Средние вопросы

(1, 'Что такое операционная система, операционная среда,

операционная оболочка?', 'Теоретический', 'Простой'),

(1, 'Понятие программного интерфейса, его назначение. Виды интерфейсов', 'Теоретический', 'Средний'),

(1, 'Повышение отказоустойчивости операционных систем', 'Практический', 'Средний'),

(1, 'Определение и функции операционных систем. Эволюция ОС', 'Теоретический', 'Средний'),

(1, 'Концепция микроядерной архитектуры', 'Теоретический', 'Средний'),

(1, 'Работа с планировщиком заданий', 'Практический', 'Средний'),

--36

(1, 'Состояние потоков на разных этапах их разработки.Алгоритм планирования процессов

основанный на абсолютных приоритетах', 'Теоретический', 'Средний'),

(1, '. Понятие виртуальной памяти. Страничное распределение.', 'Теоретический', 'Средний'),

(1, 'Исследование соотношения между

представляемым и истинным объёмом занятой дисковой памяти', 'Практический', 'Средний'),

(1, 'Понятие оперативной памяти.Распределение памяти фиксированными разделами',

'Теоретический', 'Средний'),

(1, 'Методы распределения оперативной памяти без использования дискового пространства.',

'Теоретический', 'Средний'),

(1, 'Работа с файлом подкачки', 'Практический', 'Средний'),

----42

(1, 'Цели и задачи файловой системы. Типы файлов. Именование файлов. Атрибуты файлов.',

'Теоретический', 'Средний'),

(1, 'Базовые команды MS DOS.', 'Теоретический', 'Средний'),

(1, 'Работа с видами файловой системы', 'Практический', 'Средний'),

(1, 'Файловая система NTFS. Логические области раздела NTFS.', 'Теоретический', 'Средний'),

(1, 'Аутентификация, пароли, авторизация, аудит', 'Теоретический', 'Средний'),

(1, 'Сравнение распространенных программ средств

диагностики и коррекции ошибок ', 'Практический', 'Средний'),

--48

(1, 'Программное конфигурирование

компьютерной системы', 'Теоретический', 'Средний'),

(1, 'Назначение программы DiskDirectSuite.', 'Теоретический', 'Средний'),

(1, ' Исследование и изучение влияния количества

файлов на время, необходимое для их копирования', 'Практический', 'Средний'),

(1, 'Операционная система MS-DOS ', 'Теоретический', 'Средний'),

(1, 'KSysGuard. Каковы основные возможности утилиты?', 'Теоретический', 'Средний'),

(1, 'Контроль за работой диспетчера задач', 'Практический', 'Средний'),

--54

(1, 'Просмотрите основные сведения о производительности ЦП утилитой KSysInfo. Проанализируйте как

меняется загрузка процессора.', 'Теоретический', 'Средний'),

(1, 'Какие программы обслуживания Windows Вы знаете? Каково их назначение?', 'Теоретический', 'Средний'),

(1, 'Запустите редактор реестра regedit. Найдите ветвь реестра HKEY\_USERS и экспортируйте ее

в каталог своей группы. Импортируйте сохраненный ранее файл в реестр.', 'Практический', 'Средний'),

(1, 'Нарисуйте цепочку кластеров для архивного файла Prog.rar размером 4 кб, если известно,

свободные кластеры начинаются с 15 и кластеры 17, 19 зарезервированы, а 20 кластер

испорченный. При это размер кластера 1024 байт. Приемлем ли выбранный размер кластера

для данного файла? Есть ли потери места на ЖД?', 'Теоретический', 'Средний'),

(1, 'Определите (в %) размер основного раздела диска, если дополнительный (расширенный)

раздел состоит из двух логических разделов размером 20% и 15 %.', 'Теоретический', 'Средний'),

(1, 'Создайте в Windows учётные записи пользователей student и teacher. Установите права

доступа к папкам:

в папку Otvet должны записываться и стираться файлы ответов пользователя student;

в папке Zadanie пользователь student может только читать файлы, а пользователь teacherможет добавлять и изменять файлы.', 'Практический', 'Средний'), --10

----60

----Простые вопросы

(1, 'Командный интерфейс. Виды реализации.', 'Теоретический', 'Простой'),

(1, 'Архитектура ОС: ядро и вспомогательные модули, режимы работы процессора:

привилегированный и пользовательский, работа ядра в привилегированном режиме', 'Теоретический', 'Простой'),

(1, 'Доменная архитектура ОС', 'Практический', 'Простой'),

--63

(1, 'Состояние потоков на разных этапах их разработки.Алгоритм планирования процессов

основанный на квантовании', 'Теоретический', 'Простой'),

(1, 'Назначение и типы прерываний', 'Теоретический', 'Простой'),

(1, 'Создание и редактирование файлов в редакторе Edit', 'Практический', 'Простой'),

(1, 'Понятие оперативной памяти.Распределение памяти динамическими разделами (разделами

переменной величины)', 'Теоретический', 'Простой'),

(1, 'Понятие виртуальной памяти. Сегментное распределение.', 'Теоретический', 'Простой'),

(1, 'Управление учетными записями пользователей', 'Практический', 'Простой'),

(1, 'Методы распределения оперативной памяти с использованием дискового пространства.', 'Теоретический', 'Простой'),

(1, 'Физическая организация магнитного диска: разделы, секторы, кластеры, дорожки, цилиндр,

процесс разбиения диска на разделы.', 'Теоретический', 'Простой'),

(1, 'Динамическая настройка виртуальной памяти в ОС Windows', 'Практический', 'Простой'),

--72

(1, 'Базовые команды ОС Linux', 'Теоретический', 'Простой'),

(1, 'Реестр Windows: основные понятия, ветви, способы редактирования. ', 'Теоретический', 'Простой'),

(1, 'Работа в операционных системах и средах,

наблюдение за распределением ресурсов ПК', 'Практический', 'Простой'),

(1, 'Администрирование в операционной системе Windows', 'Теоретический', 'Простой'),

(1, 'Использование антивирусных программ', 'Теоретический', 'Простой'),

(1, 'Изучение средств управления в ОС Unix

Настройка сетевых параметров в ОС Unix', 'Практический', 'Простой'),

--78

(1, 'Операционная система Free-DOS ', 'Теоретический', 'Простой'),

(1, 'Использование меню в файлах конфигурации', 'Теоретический', 'Простой'),

(1, 'Назначение программы DiskDirectSuite. Размер раздела жесткого диска равен 2048 Кбайт.

Сколько секторов будет содержать раздел? Создайте раздел такого размера с файловой

системой fat.', 'Практический', 'Простой'),

--81

(1, 'Нарисуйте цепочку кластеров файла, если его длина 2,3 Кб, размер кластера – 1 сектор, а при

его записи были свободны только кластеры 2, 4, 8, 10, 14, 23.', 'Теоретический', 'Простой'),

(1, 'Виртуальную машину', 'Теоретический', 'Простой'),

(1, 'Запустите утилиту KSysGuard. Каковы основные возможности утилиты? Просмотрите

системные процессы. Проанализируйте характеристики процесса с идентификатором 10:

состояние, приоритет, размер виртуальной и физической памяти, занимаемой процессом,

родительский процесс, имя пользователя, запустившего процесс. ', 'Практический', 'Простой'),

(1, 'Установлить квоты на использование

дискового пространства и порог выдачи предупреждения для созданного пользователя', 'Теоретический', 'Простой'),

(1, 'Нарисуйте цепочку кластеров файла, если его длина 2000 байт, размер кластера – 1 сектор, а

при его записи были свободны только кластеры 5, 7, 10, 13.', 'Теоретический', 'Простой'),

(1, 'Создайте группу news. В группу добавьте учетные записи tablу и monthly. Назначьте пароли

к учетным записям. Создайте папку WORK. Назначьте общий доступ к этой папке.', 'Практический', 'Простой'),

--87

--9 билетов

----ОП.10. Математическое моделирование теоретические и практические вопросы

----сложные

(2, 'Разделенные разности.

Интерполяционный многочлен в форме Ньютона.

Интерполяция с кратными узлами.

Многочлены Эрмита.

Задачи на построение эрмитовых сплайнов.', 'Теоретический', 'Простой'),

(2, 'Классификация моделей.', 'Теоретический', 'Сложный'),

(2, 'Задача Коши для уравнения теплопроводности', 'Практический', 'Сложный'),

----90

(2, 'Задача линейного программирования и ее общая форма.', 'Теоретический', 'Сложный'),

(2, 'Математическая модель задачи о назначениях.', 'Теоретический', 'Сложный'),

(2, 'Сущность алгоритма Фланда.', 'Практический', 'Сложный'),

(2, 'Возможные множества решений задачи линейного

программирования.', 'Теоретический', 'Сложный'),

(2, 'Критерий оптимальности плана задачи линейного программирования.', 'Теоретический', 'Сложный'),

(2, 'Суть задачи о нахождении кратчайшего пути.', 'Практический', 'Сложный'),

(2, 'Общая характеристика метода потенциалов.', 'Теоретический', 'Сложный'),

(2, 'Модель транспортной задачи в форме таблицы.', 'Теоретический', 'Сложный'),

(2, 'Классификация моделей управления запасами.', 'Практический', 'Сложный'),

(2, 'Предмет, область применения и основные понятия теории графов.', 'Теоретический', 'Сложный'),

(2, 'Параметры событий и работ.', 'Теоретический', 'Сложный'),

(2, 'Достоинства и недостатки моделей с периодическим пополнением

запасов.', 'Практический', 'Сложный'),

----102

(2, 'Постановка задачи о кратчайшем маршруте.', 'Теоретический', 'Сложный'),

(2, 'Теорема Форда – Фалкерсона.', 'Теоретический', 'Сложный'),

(2, 'Суть вероятностной модели управления запасами.', 'Практический', 'Сложный'),

(2, 'Алгоритм приведения матрицы расходов в задаче коммивояжера.', 'Теоретический', 'Сложный'),

(2, 'Процесс Маркова и его свойства.', 'Теоретический', 'Сложный'),

(2, 'Постановка задачи управления запасами с учетом убытков от

неудовлетворенного спроса.', 'Практический', 'Сложный'),

(2, 'Последовательность построения сетевого графика строительства

транспортного сооружения.', 'Теоретический', 'Сложный'),

(2, 'Определения основным элементам сетевой модели: событие, работа,

путь и их характеристики.', 'Теоретический', 'Сложный'),

(2, 'Отличие вероятностных сетевых моделей от детерминированных.', 'Практический', 'Сложный'),

--111

(2, 'Определение термина «надежность транспортного сооружения».', 'Теоретический', 'Сложный'),

(2, 'Роль статистического анализа в проектировании транспортного

сооружения с заданной надежностью.', 'Практический', 'Сложный'),

(2, 'Критерии существования статистического тренда.', 'Практический', 'Сложный'),

(2, 'Суть методики проверки гипотезы о наличии временного тренда на

основе разности средних.', 'Практический', 'Сложный'),

(2, 'Примеры задач в дорожной отрасли, решаемые с применением

моделей линейного программирования.', 'Теоретический', 'Сложный'),

(2, 'Последовательность решения задачи адаптации структур.', 'Практический', 'Сложный'), --10 билетов

--117

----средние

(2, 'Понятие модели, моделирования.

Предметные, аналоговые и математические модели.

Общая схема метода моделирования сложных систем.', 'Теоретический', 'Средний'),

(2, 'Линейная интерполяция.

Практический способ интерполяции.

Прямое построение интерполяционного многочлена Лагранжа

и тригонометрического интерполяционного многочлена.', 'Теоретический', 'Средний'),

(2, 'Построение простейших математических моделей.', 'Практический', 'Средний'),

(2, 'Основные этапы математического моделирования.', 'Теоретический', 'Средний'),

(2, 'Математическая модель задачи о ранце.', 'Теоретический', 'Средний'),

(2, 'Решение задач линейного программирования

симплекс–методом', 'Практический', 'Средний'),

--123

(2, 'Предмет, задача и основные понятия математического

программирования.', 'Теоретический', 'Средний'),

(2, 'Приведение задачи линейного программирования к канонической

форме.', 'Теоретический', 'Средний'),

(2, 'Суть сетевых задач линейного программирования.', 'Практический', 'Средний'),

(2, 'Общая характеристика симплекс – метода.', 'Теоретический', 'Средний'),

(2, 'Метод построения нового плана в рамках симплекс – метода.', 'Теоретический', 'Средний'),

(2, 'Алгоритм задачи определения максимального потока.', 'Практический', 'Средний'),

(2, 'Балансировка транспортной задачи.', 'Теоретический', 'Средний'),

(2, 'Проверка плана транспортной задачи на оптимальность.', 'Теоретический', 'Средний'),

(2, 'Сущность моделей с периодическим пополнением запасов.', 'Практический', 'Средний'),

(2, 'Предмет и область применения системы сетевого планирования и

управления.', 'Теоретический', 'Средний'), --5

(2, 'Методика расчета параметров сетевого графика.', 'Теоретический', 'Средний'),

(2, 'Достоинства и недостатки моделей с непериодическим пополнением

запасов.', 'Практический', 'Средний'),

----135

(2, 'Постановка задачи о максимальном потоке.', 'Теоретический', 'Средний'),

(2, 'Методология метода ветвей и границ.', 'Теоретический', 'Средний'),

(2, 'Порядок обоснования оптимального размера запаса на вероятностной

модели с заданной надежностью при двусторонних интервальных оценках.', 'Практический', 'Средний'),

(2, 'Алгоритм деления множества маршрутов на части.', 'Теоретический', 'Средний'),

(2, 'Основные формы сетевых моделей.', 'Теоретический', 'Средний'),

(2, 'Формула определения оптимального объема запаса с учетом убытков

от неудовлетворенного спроса.', 'Практический', 'Средний'),

----141

(2, 'Определение понятия «критический путь» и «критическая зона» на

сетевой модели.', 'Теоретический', 'Средний'),

(2, 'Зависимость для расчета вероятности завершения работ по сетевому

графику в заданные сроки.', 'Теоретический', 'Средний'),

(2, 'Статистический учет и статистический анализ в работе транспортного

сооружения.', 'Практический', 'Средний'),

(2, 'Суть прогнозирования на основе статистического анализа.', 'Теоретический', 'Средний'),

(2, 'Экстремальный анализ в экономических задачах.', 'Теоретический', 'Средний'),

(2, 'Суть сглаживания статистических данных. Какие методы

статистического сглаживания существуют.', 'Практический', 'Средний'), --10 билетов

--147

----простые

(2, 'Метод математического моделирования.

Классификация моделей.

Перспективы применения многопроцессорных вычислительных систем.', 'Теоретический', 'Простой'),

(2, 'Математическая модель транспортной задачи.', 'Теоретический', 'Простой'),

(2, 'Решение простейших однокритериальных задач', 'Практический', 'Простой'),

--150

(2, 'Случайные процессы и их классификация.', 'Теоретический', 'Простой'),

(2, 'Классификация задач математического программирования.', 'Теоретический', 'Простой'),

(2, 'Нахождение начального решения транспортной

задачи. Решение транспортной задачи методом потенциалов', 'Практический', 'Простой'),

(2, 'Геометрическая интерпретация задачи линейного программирования.', 'Теоретический', 'Простой'),

(2, 'Заполнение начальной симплекс – таблицы.', 'Теоретический', 'Простой'),

(2, 'Суть задачи минимизации сети.', 'Практический', 'Простой'),

(2, 'Вспомогательная задача.', 'Теоретический', 'Простой'),

(2, 'Метод северо-западного угла.', 'Теоретический', 'Простой'),

(2, 'Основные классы задач в области транспортного строительства,

решаемые с применением моделей массового обслуживания.', 'Практический', 'Простой'),

----159

(2, 'Построение нового плана в методе потенциалов.', 'Теоретический', 'Простой'),

(2, 'Сетевой график и его элементы.', 'Теоретический', 'Простой'),

(2, 'Сущность моделей с непериодическим пополнением запасов.', 'Практический', 'Простой'),

(2, 'Критический путь и его содержательный смысл.', 'Теоретический', 'Простой'),

(2, 'Разрез и его пропускная способность.', 'Теоретический', 'Простой'),

(2, 'Методика расчета оптимального периода пополнения запаса на

модели с периодическим спросом.', 'Практический', 'Простой'),

(2, 'Постановка задачи коммивояжера.', 'Теоретический', 'Простой'),

(2, 'Процессы размножения и гибели.', 'Теоретический', 'Простой'),

(2, 'Графическое представление вероятностной модели управления

запасами.', 'Практический', 'Простой'),

--168

(2, 'Определение сетевой модели строительства и ее отличие от

традиционного линейного календарного графика.', 'Теоретический', 'Простой'),

(2, 'Достоинства и недостатки форм сетевых моделей.', 'Теоретический', 'Простой'),

(2, 'Коэффициент напряженности работ, путей и как он рассчитывается.', 'Практический', 'Простой'),

--171

(2, 'Как рассчитать требуемый резерв времени на сетевом графике, если

вероятность своевременного завершения строительства задана, а расчетный срок

окончания работ по графику известен.', 'Теоретический', 'Простой'),

(2, 'Последовательность обработки статистических данных о работе

транспортного сооружения.', 'Теоретический', 'Простой'),

(2, 'Сущность статистического контроля качества.', 'Практический', 'Простой'),

(2, 'Основные методы получения оптимальных моделей с применением

моделей линейного программирования.', 'Теоретичесский', 'Простой'),

(2, 'Эффективность организационной структуры и виды эффективности.', 'Теоретический', 'Простой'),

(2, 'Доказать Теорему Перельмана', 'Практический', 'Простой'), --10 билетов

----177

(3, 'Дать определение избыточных данных.

Рассмотреть на примере аномалии вставки,

удаления, модификации.

Рассмотреть способы устранения аномалий и избыточности данных.', 'Теоретический', 'Сложный'),

(3, 'Применение информационных сетей.', 'Теоретический', 'Сложный'),

(3, 'Монтаж кабельных сред технологий Ethernet', 'Практический', 'Сложный'),

--180

(3, 'Классификация информационных сетей по типу технологии, среды и скорости

передачи.', 'Теоретический', 'Сложный'),

(3, 'Классификация информационных сетей по типу топологии.', 'Теоретический', 'Сложный'),

(3, 'Определение маршрутов следования данных в сети', 'Практический', 'Сложный'),

--183

(3, 'Эталонные модели сети. Эталонная модель TCP/IP.', 'Теоретический', 'Сложный'),

(3, 'Сетевые устройства и сетевые адаптеры. Активные сетевые устройства.', 'Теоретический', 'Сложный'),

(3, 'Протокол DHCP. Аренда IP-адреса', 'Практический', 'Сложный'),

(3, 'Кабельные линии связи. Коаксиальный кабель.', 'Теоретический', 'Сложный'),

(3, 'Базовые сетевые технологии. Метод доступа CSMA/CD и маркерный доступ.', 'Теоретический', 'Сложный'),

(3, 'Протокол ICMP. Недостижимость узла назначения', 'Практический', 'Сложный'),

--189

(3, 'Базовые сетевые технологии. Технология Ethernet. Спецификация Ethernet

10Base-2.', 'Теоретический', 'Сложный'),

(3, 'Базовые сетевые технологии. Технология Ethernet. Спецификации Gigabit

Ethernet..', 'Теоретический', 'Сложный'),

(3, 'Петли в сетях, объединенных с помощью мостов', 'Практический', 'Сложный'),

--192

(3, 'Беспроводные техновлогии. Bluetooth. Архитектура Bluetooth.', 'Теоретический', 'Сложный'),

(3, 'Беспроводные технологии. Bluetooth. Спецификации Bluetooth.', 'Теоретический', 'Сложный'),

(3, 'Протоколы маршрутизации. Протокол RIPv1', 'Практический', 'Сложный'),

(3, 'Беспроводные технологии. Wi-Fi. Метод доступа CSMA/CA и проблема

скрытого узла.', 'Теоретический', 'Сложный'),

(3, 'Адресация в информационных сетях. IP-адрес. Бесклассовая адресация.', 'Теоретический', 'Сложный'),

(3, 'Протоколы маршрутизации. Протокол OSPF', 'Практический', 'Сложный'),

--198

(3, 'Специальные IP-адреса. Групповые адреса.', 'Теоретический', 'Сложный'),

(3, 'Формат IP-пакета. Заголовок пакета IPv4.', 'Теоретический', 'Сложный'),

(3, 'Преобразование сетевых адресов. Преобразование внутренних адресов', 'Практический', 'Сложный'),

----201

(3, 'Адресация в информационных сетях. IP-адрес. Классовая адресация.', 'Теоретический', 'Сложный'),

(3, 'Специальные IP-адреса. Широковещательные адреса.', 'Теоретический', 'Сложный'),

(3, 'Транспортные протоколы TCP/IP. Порты.', 'Практический', 'Сложный'),

(3, 'Специальные IP-адреса. Адреса IPv4, отображенные в IPv6.', 'Теоретический', 'Сложный'),

(3, 'Адресация в информационных сетях. Система доменных имен.', 'Теоретический', 'Сложный'),

(3, 'Транспортные протоколы TCP/IP. Состояния TCP-соединения.', 'Практический', 'Сложный'),

--207

(3, 'Прикладные протоколы TCP/IP. Протокол SMTP.', 'Теоретический', 'Сложный'),

(3, 'Виды протоколов', 'Теоретический', 'Сложный'),

(3, 'Администрирование информационных сетей. Задачи систем управления сетями.', 'Практический', 'Сложный'),

----210

----11 билетов

----средние

(3, 'Сформулировать соотношение операций реляционной алгебры и выражений реляционного исчисления.

Перечислить специальные реляционные операции реляционной алгебры.

Дать точное и полное определение каждой специальной реляционной операции.', 'Теоретический', 'Средний'),

(3, 'Сформулировать определение процесса нормализации.

Дать точное и полное определение нормальной формы.

Перечислить свойства нормальных форм.', 'Теоретический', 'Средний'),

(3, 'Отслеживание прохождения пакетов через сеть', 'Практический', 'Средний'),

(3, 'Сформулировать основные свойства нормальных форм.

Дать определение функциональной зависимости атрибутов от первичного ключа.

Дать определение функционально полной зависимости атрибутов

от первичного ключа.', 'Теоретический', 'Средний'),

(3, 'Классификация информационных сетей по размеру.', 'Теоретический', 'Средний'),

(3, 'Определение оборудования, удовлетворяющего требованиям заказчика', 'Практический', 'Средний'),

(3, 'Эталонные модели сети. Протокол и стек протоколов.', 'Теоретический', 'Средний'),

(3, 'Эталонные модели сети. Гибридная эталонная модель.', 'Теоретический', 'Средний'),

(3, 'Протокол ARP. Определение MAC-адреса для заданного IP-адреса', 'Практический', 'Средний'),

----219

(3, 'Линии и каналы связи. Типы каналов. Первичные и вторичные сети.', 'Теоретический', 'Средний'),

(3, 'Кабельные линии связи. Оптоволоконный кабель.', 'Теоретический', 'Средний'),

(3, 'Протокол ICMP. Определение MAC-адреса для заданного адреса IPv6', 'Практический', 'Средний'),

(3, 'Базовые сетевые технологии. Технология Ethernet. Формат кадра Ethernet.', 'Теоретический', 'Средний'),

(3, 'Базовые сетевые технологии. Технология Ethernet. Спецификации Ethernet

10Base-T и Ethernet 10Base-FL.', 'Теоретический', 'Средний'),

(3, 'Протокол связующего дерева', 'Практический', 'Средний'),

--225

(3, 'Базовые сетевые технологии. Технология Ethernet. Спецификации 10Gigabit

Ethernet.', 'Теоретический', 'Средний'),

(3, 'Беспроводные технологии. Bluetooth. Передача данных в Bluetooth.', 'Теоретический', 'Средний'),

(3, 'Протоколы маршрутизации. Протокол RIPv2', 'Практический', 'Средний'),

(3, 'Беспроводные технологии. Wi-Fi. Архитектура Wi-Fi.', 'Теоретический', 'Средний'),

(3, 'Адресация в информационных сетях. MAC-адрес.', 'Теоретический', 'Средний'),

(3, 'Преобразование сетевых адресов. Перегрузка глобальных адресов', 'Практический', 'Средний'),

--231

(3, 'Специальные IP-адреса. Публичные и частные адреса.', 'Теоретический', 'Средний'),

(3, 'Специальные IP-адреса. Адрес обратной петли.', 'Теоретический', 'Средний'),

(3, 'Транспортные протоколы TCP/IP. UDP-дейтаграмма.', 'Практический', 'Средний'),

--234

(3, 'Адресация в информационных сетях. IP-адрес. Бесклассовая адресация.', 'Теоретический', 'Средний'),

(3, 'Формат IP-пакета. Заголовок пакета IPv6.', 'Теоретический', 'Средний'),

(3, 'Прикладные протоколы TCP/IP. Протокол FTP.', 'Практический', 'Средний'),

--238

(3, 'Специальные IP-адреса. Групповые адреса.', 'Теоретический', 'Средний'),

(3, 'Формат IP-пакета. Заголовок пакета IPv4.', 'Теоретический', 'Средний'),

(3, 'Прикладные протоколы TCP/IP. Протокол POP3.', 'Практический', 'Средний'), --10 билетов

--240

----простые

(3, 'Специальные IP-адреса. Широковещательные адреса.', 'Теоретический', 'Простой'),

(3, 'Адресация в информационных сетях. IP-адрес. Классовая адресация..', 'Теоретический', 'Простой'),

(3, 'Разбиение сети на подсети', 'Практический', 'Простой'), --1

(3, 'Беспроводные технологии. Wi-Fi. Стандарты Wi-Fi.', 'Теоретический', 'Простой'),

(3, 'Специальные IP-адреса. Адрес обратной петли.', 'Теоретический', 'Простой'),

(3, 'Протокол DNS. Заголовок и блоки данных в сообщении DNS-пакета',

'Практический', 'Простой'), --2

(3, 'Специальные IP-адреса. Публичные и частные адреса.', 'Теоретический', 'Простой'),

(3, 'Адресация в информационных сетях. Система доменных имен.', 'Теоретический', 'Простой'),

(3, 'Протокол ICMP. Эхо-сообщения', 'Практический', 'Простой'), --3

----249

(3, 'Специальные IP-адреса. Адреса IPv4, отображенные в IPv6.', 'Теоретический', 'Простой'),

(3, 'Беспроводные технологии. Bluetooth. Профили Bluetooth.', 'Теоретический', 'Простой'),

(3, 'Объединение сетей с помощью мостов. Прозрачное мостовое соединение',

'Практический', 'Простой'), --252 3

(3, 'Базовые сетевые технологии. Технологии Token Ring и FDDI.', 'Теоретический', 'Простой'),

(3, 'Кабельные линии связи. Витая пара.', 'Теоретический', 'Простой'),

(3, 'Объединение сетей с помощью маршрутизаторов. Алгоритмы маршрутизации',

'Практический', 'Простой'),--255 4

(3, 'Беспроводные линии связи. Радиосвязь. Спутниковая связь.', 'Теоретический', 'Простой'),

(3, 'Базовые сетевые технологии. Технология Ethernet. Спецификация Ethernet

10Base-5.', 'Теоретический', 'Простой'),

(3, 'Протоколы маршрутизации. Протокол RIPng.', 'Практический', 'Простой'), --258 5

(3, 'Базовые сетевые технологии. Технология Ethernet. Спецификации Fast Ethernet.',

'Теоретический', 'Простой'),

(3, 'Сетевые устройства и сетевые адаптеры. Пассивные сетевые устройства.', 'Теоретический', 'Простой'),

(3, 'Протоколы маршрутизации. Внешние шлюзовые протоколы', 'Практический', 'Простой'), --6

(3, 'Формат IP-пакета. Заголовок пакета IPv6.', 'Теоретический', 'Простой'),

(3, 'Классификация информационных сетей по типу функционального

взаимодействия.', 'Теоретический', 'Простой'),

(3, 'Написать устройство IP-пакета.', 'Практический', 'Простой'), --7

(3, 'Эталонные модели сети. Эталонная модель OSI.', 'Теоретический', 'Простой'),

(3, 'База данных DNS. Записи A и PTR.', 'Теоретический', 'Простой'),

(3, 'Преобразование сетевых адресов. Преобразование при перекрытии адресов',

'Практический', 'Простой'), --8

--267 9

(3, 'Дать определение 1 нормальной формы.

Дать определение 2 нормальной формы.

Дать определение 3 нормальной формы.', 'Теоретический', 'Простой'),

(3, 'Прикладные протоколы TCP/IP. Протокол HTTP.', 'Теоретический', 'Простой'), --10 билетов

(3, 'Транспортные протоколы TCP/IP. Установление и завершение TCP-соединения.',

'Практический', 'Простой'),

----270

----МДК.02.02. Технология разработки и защиты баз данных теоретические вопросы

--сложные

(4, 'Перечислить основные модели данных.

Проанализировать достоинства и недостатки каждой модели.

Сформулировать преимущества реляционной модели.', 'Теоретический', 'Сложный'),

(4, 'Проектирование реляционной схемы базы

данных в среде СУБД.', 'Теоретический', 'Сложный'),

(4, 'Проектирование реляционной схемы базы

данных в среде СУБД.', 'Практический', 'Сложный'),

(4, 'Создание базы данных в среде

разработки', 'Теоретический', 'Сложный'),

(4, '«Восстановление базы данных из

резервной копии', 'Теоретический','Сложный'),

(4, 'Установка приоритетов', 'Практический', 'Сложный'), --271-276 2 билета

----средние

(4, 'Дать определение СУБД.

Перечислить и кратко охарактеризовать основные функции СУБД.

Перечислить основные элементы типовой организации СУБД.', 'Теоретический', 'Средний'),

(4, 'Сформулировать достоинства и недостатки реляционной модели данных.

Дать определение отношения.

Перечислить и кратко охарактеризовать фундаментальные свойства отношений.', 'Теоретический', 'Средний'),

(4, 'Установить бд Северный ветер', 'Практический', 'Средний'),

----279

(4, 'Методы проектирования базы данных', 'Теоретический', 'Средний'),

(4, 'Импортирование данных в бд', 'Теоретический', 'Средний'),

(4, 'Создать диаграмму в базе данных', 'Практический', 'Средний'), ---277-282 2 билета

----простой

(4, 'Привести примеры современных СУБД.

Перечислить и охарактеризовать основные функции современных СУБД.

Перечислить основные элементы типовой организации СУБД.', 'Теоретический', 'Простой'),

(4, 'Дать определение отношения.

Перечислить и кратко охарактеризовать фундаментальные свойства отношений.

Дать определения элементов отношения:

первичный ключ, кортеж, атрибут, схема отношения, внешний ключ.', 'Теоретический', 'Простой'),

(4, 'Каскадное удаление', 'Теоретический', 'Простой'),

(4, 'Каскадное обновление', 'Теоретический', 'Простой'),

(4, 'Виды связей', 'Теоретический', 'Простой'),

(4, 'Виды связей', 'Практический', 'Простой'), --283-288 2 билета

----ОГСЭ.03. Иностранный язык теоретические вопросы

--сложные

(5, 'Present Simple : образование, употребление.', 'Теоретические', 'Сложный'),

(5, 'Порядок слов в простом предложении.', 'Практический', 'Сложный'),

(5, 'Неопределённый артикль', 'Практический', 'Сложный'),

--289-291

----средний

(5, 'Present Continuous в утвердительной и отрицательной форме', 'Теоретические', 'Средний'),

(5, 'Present Continuous : образование, употребление.', 'Теоретические', 'Средний'),

(5, 'Исправьте ошибки в тексте', 'Практический', 'Средний'), --292-294

----простой

(5, 'Типы вопросов', 'Теоретические', 'Простой'),

(5, 'Примеры модальных глаголов с переводом', 'Теоретические', 'Простой'),

(5, 'Вспомогательные глаголы в английском языке', 'Практический', 'Простой');

--295-297

CREATE TABLE Tickets

(

id\_ticket INT IDENTITY, --IDENTITY,

id\_quest1 INT,

id\_quest2 INT,

id\_quest3 INT,

nom\_komplect INT,

PRIMARY KEY(id\_ticket),

FOREIGN KEY (id\_quest1) REFERENCES Questions (id\_question),

FOREIGN KEY (id\_quest2) REFERENCES Questions (id\_question),

FOREIGN KEY (id\_quest3) REFERENCES Questions (id\_question),

FOREIGN KEY (nom\_komplect) REFERENCES Komplect\_tickets (nom\_komplect) ON DELETE CASCADE ON UPDATE CASCADE

);

INSERT INTO Tickets (id\_quest1, id\_quest2, id\_quest3, nom\_komplect) --доделать

VALUES

--ОП.01. Операционные системы практические и теоретические 1-77

(1, 2, 3, 1 ),

(4, 5, 6, 1),

(7, 8, 9, 1),

(10, 11, 12, 1),

(13, 14, 15, 1),

(16, 17, 18, 1),

(19, 20, 21, 1),

(22, 23, 24, 1),

(25, 26, 27, 1),

(28, 29, 30, 1), --1-30

--средние (+)

(31, 32, 33, 1),

(34, 35, 36, 1),

(37, 38, 39, 1),

(40, 41, 42, 1),

(43, 44, 45, 1),

(46, 47, 48, 1),

(49, 50, 51, 1),

(52, 53, 54, 1),

(55, 56, 57, 1),

(58, 59, 60, 1), --31-60

--простые

(61, 62, 63, 1),

(64, 65, 66, 1),

(67, 68, 69, 1),

(70, 71, 72, 1),

(73, 74, 75, 1),

(76, 77, 78, 1),

(79, 80, 81, 1),

(82, 83, 84, 1),

(85, 86, 87, 1), --61-87

--ОП.10. Математическое моделирование теоретические и практические вопросы -177(100)

--сложные (+)

(88, 89, 90, 2),

(91, 92, 93, 2),

(94, 95, 96, 2),

(97, 98, 99, 2),

(100, 101, 102, 2),

(103, 104, 105, 2),

(106, 107, 108, 2),

(109, 110, 111, 2),

(112, 113, 114, 2),

(115, 116, 117, 2), --88-117

--средние

(118, 119, 120, 2),

(121, 122, 123, 2),

(124, 125, 126, 2),

(127, 128, 129, 2),

(130, 131, 132, 2),

(133, 134, 135, 2),

(136, 137, 138, 2),

(139, 140, 141, 2),

(142, 143, 144, 2),

(145, 146, 147, 2), --118-147

--простые

(148, 149, 150, 2),

(151, 152, 153, 2),

(154, 155, 156, 2),

(157, 158, 159, 2),

(150, 161, 162, 2),

(163, 164, 165, 2),

(166, 167, 168, 2),

(169, 170, 171, 2),

(172, 173, 174, 2),

( 175, 176, 177, 2), --148-177

--МДК.02.01. Инфокоммуникационные системы и сети теоретические вопросы 178-210

--сложные

(178, 179, 180, 3),

(181, 182, 183, 3),

(184, 185, 186, 3),

(187, 188, 189, 3),

(190, 191, 192, 3),

(193, 194, 195, 3),

(196, 197, 198, 3),

(199, 200, 201, 3),

(202, 203, 204, 3),

(205, 206, 207, 3),

(208, 209, 210, 3), --178-210

--средние 211-241

(211, 212, 213, 3),

(214, 215, 216, 3),

(217, 218, 219, 3),

(220, 221, 222, 3),

(223, 224, 225, 3),

(226, 227, 228, 3),

(229, 230, 231, 3),

(232, 233, 234, 3),

(235, 236, 237, 3),

(238, 239, 240, 3), --211-240

----простые 241-270

(241, 242, 243, 3),

(244, 245, 246, 3),

(247, 248, 249, 3),

(250, 251, 252, 3),

(253, 254, 255, 3),

(256, 257, 258, 3),

(259,260, 261, 3),

(262, 263, 264, 3),

(265, 266, 267, 3),

(268,269, 270, 3), --241-270

----МДК.02.02. Технология разработки и защиты баз данных теоретические вопросы

--сложные 271-276

(271, 272, 273, 4),

(274, 275, 276, 4),

--средние 277-282

(277,278, 279, 4),

(280, 281, 282, 4),

--простые 283-288

(283, 284, 285, 4),

(286,287, 288, 4),

----ОГСЭ.03. Иностранный язык теоретические вопросы

--сложные 289-291

(289, 290, 291, 5),

--средние

(292, 293, 294, 5),

--простые

(295, 296, 297, 5);

# **ПРИЛОЖЕНИЕ Б. Код программы**

**Authorization.xaml**

<Page x:Class="WpfApp1.Authorization"

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

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

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

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

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="700" d:DesignWidth="1200"

Title="Authorization">

<Grid Background="#FFD3D3D3">

<Grid.ColumnDefinitions>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"

Height="70" Width="680"

FontSize="40" FontFamily="Verdana" FontWeight="Bold" x:Name="Header">

Авторизация пользователя

</TextBlock>

<TextBlock Grid.Row="1" Grid.Column="0"

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="role">

Роль:

</TextBlock>

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

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="login">

Логин:

</TextBlock>

<TextBlock Grid.Row="3" Grid.Column="0"

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="password">

Пароль:

</TextBlock>

<ComboBox Name="Role"

Grid.Row="1"

Grid.Column="1"

Height="70" Width="400"

HorizontalAlignment="Center"

VerticalAlignment="Top" Margin="0,41,0,0" />

<TextBox Grid.Row="2" Grid.Column="1"

Height="70" Width="400"

MaxLength="50"

TextWrapping="Wrap"

x:Name="textBox\_login" Margin="100,41,100,29">

</TextBox>

<PasswordBox Grid.Row="3" Grid.Column="1"

Height="70" Width="400"

x:Name="textBox\_password" Margin="100,40,100,30"

MaxLength="50"

>

</PasswordBox>

<Button Grid.Row="4" Grid.Column="0" Width="400"

FontSize="30" FontFamily="Verdana" FontWeight="Bold"

x:Name="But2"

Content="Авторизация"

Click="But\_authorization" Margin="0,10,0,16">

<Button.Background>

<SolidColorBrush Color="#FFFA8072" />

</Button.Background>

</Button>

<Button Grid.Row="4" Grid.Column="1" Width="400"

FontSize="30" FontFamily="Verdana" FontWeight="Bold"

x:Name="But1"

Content="Регистрация"

Click="But\_registration" Margin="0,10,0,16">

<Button.Background>

<SolidColorBrush Color="#FF87CEFA" />

</Button.Background>

</Button>

</Grid>

</Page>

**Authorization.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для Authorization.xaml

/// </summary>

public partial class Authorization : Page

{

RandomTicketGenerator db;

public Authorization()

{

InitializeComponent();

bindcombo\_Users();

db = new RandomTicketGenerator();

}

//ComboBox kurs

public List<Users> Users\_list { get; set; }

private void bindcombo\_Users()

{

Role.Items.Add("Admin");

Role.Items.Add("User");

Role.SelectedIndex = 0;

}

/// <summary>

/// сокрытие пароля

/// </summary>

private void log\_in\_load(object sender, EventArgs e)

{

textBox\_password.PasswordChar = '•';

textBox\_password.MaxLength = 100;

}

/// <summary>

/// авторизация пользователя

/// </summary>

private void But\_authorization(object sender, RoutedEventArgs e)

{

string loginUser = textBox\_login.Text;

string roleUser = Role.Text;

string passUser = textBox\_password.Password.ToString();

if ((loginUser != "") && (roleUser != "") && (passUser != ""))

{

if (roleUser == "Admin" || roleUser == "User") //работает Admin

{

//проверка на соответствиее данных

if (db.Users.Any(o => (o.login\_ == loginUser) && (o.password\_ == passUser) && (o.role\_ == roleUser)))

{

MessageBox.Show("Успешная авторизация");

NavigationService.Navigate(new Choice\_admin(roleUser));

}

else

{

MessageBox.Show("Неправильный логин или пароль");

}

}

else

{

MessageBox.Show("Неверно указана роль пользователя");

}

}

else MessageBox.Show("Для продолжения заполните все поля");

}

/// <summary>

/// Переход к регистрации

/// </summary>

private void But\_registration(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new Sign\_up());

}

}

}

**Choice\_admin.xaml**

<Page x:Class="WpfApp1.Choice\_admin"

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

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

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

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

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="700" d:DesignWidth="800"

Title="Choice\_admin">

<Grid Background="#FFD3D3D3">

<Grid.ColumnDefinitions>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0"

Grid.Column="1"

Grid.ColumnSpan="4"

Height="70"

Width="260"

FontSize="20"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="Header"

>

Формирование билета

</TextBlock>

<TextBlock Grid.Row="1"

Height="70"

Width="130"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

HorizontalAlignment="Center"

VerticalAlignment="Center"

x:Name="spec\_header"

>

Специальность:

</TextBlock>

<TextBlock Grid.Row="1"

Grid.Column="3"

Height="70"

Width="120"

TextWrapping="Wrap"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

HorizontalAlignment="Center"

VerticalAlignment="Center"

x:Name="count\_of\_tickets\_header" >

Количество билетов:

</TextBlock>

<TextBlock Grid.Row="1"

Grid.Column="4"

Height="70"

Width="130"

TextWrapping="Wrap"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

HorizontalAlignment="Center"

VerticalAlignment="Center"

x:Name="teacher\_header" >

Преподаватель:

</TextBlock>

<TextBlock Grid.Row="1"

Grid.Column="5"

Height="70"

Width="120"

TextWrapping="Wrap"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

HorizontalAlignment="Center"

VerticalAlignment="Center"

x:Name="ck\_header" >

Председатель цикловой комиссии:

</TextBlock>

<TextBlock Grid.Row="3"

Grid.Column="0"

Height="70"

Width="120"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="kurs\_header"

HorizontalAlignment="Center"

VerticalAlignment="Center">

Курс:

</TextBlock>

<TextBlock Grid.Row="1"

Grid.Column="1"

Height="70"

Width="120"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="Disca\_header"

HorizontalAlignment="Center"

VerticalAlignment="Center">

Дисциплина:

</TextBlock>

<TextBlock Grid.Row="1"

Grid.Column="2"

Height="70"

Width="135"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="Semester\_header"

HorizontalAlignment="Center"

VerticalAlignment="Center">

Семестр:

</TextBlock>

<TextBlock Grid.Row="3"

Grid.Column="1"

Height="70"

Width="120"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="Prot\_header"

HorizontalAlignment="Center"

VerticalAlignment="Center">

Протокол:

</TextBlock>

<TextBlock Grid.Row="3"

Grid.Column="2"

TextWrapping="Wrap"

Height="70"

Width="120"

FontSize="14"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="complexity\_header"

HorizontalAlignment="Center"

VerticalAlignment="Center">

Сложность вопросов:

</TextBlock>

<ComboBox Name="Disca"

Grid.Row="2"

Grid.Column="1"

Width="133"

Height="30"

HorizontalAlignment="Center"

VerticalAlignment="Top"

/>

<TextBox Grid.Row="2"

Grid.Column="3"

Height="40"

Width="133"

TextWrapping="Wrap"

MaxLength="20"

x:Name="count\_of\_tickets"

Background="White"

HorizontalAlignment="Center"

VerticalAlignment="Top"

/>

<ComboBox Name="Teacher"

Grid.Row="2"

Grid.Column="4"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Width="133"

Height="30">

</ComboBox>

<ComboBox Name="Chairman"

Grid.Row="2"

Grid.Column="5"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Width="133"

Height="30">

</ComboBox>

<ComboBox Name="Kurs"

Grid.Row="4"

Grid.Column="0"

Height="30"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Width="133" />

<ComboBox Name="Semester"

Grid.Row="2"

Grid.Column="2"

Height="30"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Width="133" />

<ComboBox Name="Spec"

Grid.Row="2"

Grid.Column="0"

Height="30"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Width="133" />

<ComboBox Name="Protocol"

Grid.Row="4"

Grid.Column="1"

Height="30"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Width="133" />

<ComboBox x:Name="complexity\_of\_question"

Grid.Row="4"

Grid.Column="2"

Height="30"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Width="133"

SelectedIndex="0"

Background="White">

<TextBlock>Сложный</TextBlock>

<TextBlock>Средний</TextBlock>

<TextBlock>Простой</TextBlock>

</ComboBox>

<Button Grid.Row="5" Grid.Column="2" Grid.ColumnSpan="2"

Height="70" Width="250"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_Form"

Content="Формирование билета"

Click="But\_Click\_Form\_Ticket">

<Button.Background>

<SolidColorBrush Color="#FFFA8072" />

</Button.Background>

</Button>

<Button Grid.Row="5" Grid.Column="0" Grid.ColumnSpan="2"

Height="70" Width="250"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_Authorization"

Content="Авторизация"

Click="But\_Auto">

<Button.Background>

<SolidColorBrush Color="#FFFFFACD" />

</Button.Background>

</Button>

<Button Grid.Row="5" Grid.Column="4" Grid.ColumnSpan="2"

Height="70" Width="250"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_viewing"

Content="Просмотр базы данных"

Click="But\_Click\_Viewing\_Table\_Data">

<Button.Background>

<SolidColorBrush Color="#FF87CEFA" />

</Button.Background>

</Button>

<Button Grid.Row="4" Grid.Column="4" Grid.ColumnSpan="2"

Height="70" Width="250"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_Confirmation"

Content="Подтверждение специальности"

Click="But\_Confirmation\_Click">

<Button.Background>

<SolidColorBrush Color="#FFAFEEEE" />

</Button.Background>

</Button>

</Grid>

</Page>

**WordHelper.cs**

using System;

using System.Collections.Generic;

using Word = Microsoft.Office.Interop.Word;

using System.IO;

using DocumentFormat.OpenXml.Packaging;

using DocumentFormat.OpenXml.Wordprocessing;

using DocumentFormat.OpenXml;

namespace WpfApp1

{

internal class WordHelper

{

private FileInfo fileinfo\_;

public WordHelper(string filename)

{

if (File.Exists(filename))

{

fileinfo\_ = new FileInfo(filename);

}

else

{

throw new ArgumentException("File not found");

}

}

/// <summary>

/// формирование word документа

/// </summary>

internal bool Process(Dictionary<string, string> items, int count)

{

using (var doc = WordprocessingDocument.Create("FilePath", WordprocessingDocumentType.Document))

{

MainDocumentPart mainPart = doc.AddMainDocumentPart();

mainPart.Document = new Document();

Body body = mainPart.Document.AppendChild(new Body());

Paragraph para = body.AppendChild(new Paragraph());

Run run = para.AppendChild(new Run());

run.AppendChild(new Text("this new text for test"));

}

Word.Application app = null;

try

{

app = new Word.Application();

Object file = fileinfo\_.FullName;

Object missing = Type.Missing;

app.Documents.Open(file);

foreach (var item in items)

{

Word.Find find = app.Selection.Find;

find.Text = item.Key;

find.Replacement.Text = item.Value;

Object wrap = Word.WdFindWrap.wdFindContinue;

Object replace = Word.WdReplace.wdReplaceAll;

find.Execute(FindText: Type.Missing,

MatchCase: false,

MatchWholeWord: false,

MatchWildcards: false,

MatchSoundsLike: missing,

MatchAllWordForms: false,

Forward: true,

Wrap: wrap,

Format: false,

ReplaceWith: missing, Replace: replace

);

}

Object newFileName = Path.Combine($"C:\\VS Projects\\Commit\\Kursach\\Kursach\\WpfApp1\\Tickets\\",

count.ToString() + "\_Ticket.docx");

app.ActiveDocument.SaveAs2(newFileName);

app.ActiveDocument.Close();

return true;

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

return false;

}

finally

{

if (app != null)

{

app.Quit();

}

}

}

}

}

**ViewingTicketTable.xaml**

<Page x:Class="WpfApp1.ViewingTicketTable"

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

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

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

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

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

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

Title="ViewingTicketTable">

<Grid x:Name="ViewingTicketTable\_" Background="#FFD3D3D3" IsVisibleChanged="ViewingTable\_admin\_IsVisibleChanged">

<Grid.RowDefinitions>

<RowDefinition Height="30"></RowDefinition>

<RowDefinition Height="377\*"></RowDefinition>

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

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock Grid.Row="0"

Grid.Column="1"

Grid.ColumnSpan="2"

Height="30"

Width="300"

FontSize="20"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="Header"

>

Редактирование билетов

</TextBlock>

<DataGrid x:Name="DGridTicket"

AutoGenerateColumns="False"

IsReadOnly="True"

Grid.Column="0"

Grid.ColumnSpan="4"

Grid.Row="1">

<DataGrid.Columns >

<DataGridTextColumn Header="Индентифиатор билета"

Binding="{Binding id\_ticket}"

Width="220">

<DataGridTextColumn.ElementStyle>

<Style>

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

<Setter Property="TextBlock.TextAlignment" Value="Left"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Индетификатор вопроса 1"

Binding="{Binding id\_quest1}"

Width="190">

<DataGridTextColumn.ElementStyle>

<Style>

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

<Setter Property="TextBlock.TextAlignment" Value="Left"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Идентификатор вопроса 2"

Binding="{Binding id\_quest2}"

Width="190">

<DataGridTextColumn.ElementStyle>

<Style>

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

<Setter Property="TextBlock.TextAlignment" Value="Left"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Идентификатор вопроса 3"

Binding="{Binding id\_quest3}"

Width="190">

<DataGridTextColumn.ElementStyle>

<Style>

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

<Setter Property="TextBlock.TextAlignment" Value="Left"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Номер комплекта"

Binding="{Binding nom\_komplect}"

Width="190">

<DataGridTextColumn.ElementStyle>

<Style>

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

<Setter Property="TextBlock.TextAlignment" Value="Left"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" x:Name="BtnEdit" Click="BtnEdit\_Click" Width="177"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Content="Формирование билета"

Grid.Row="2"

Grid.Column="1"

HorizontalAlignment="Left"

Width="150"

Name="Form\_Ticket"

Click="Form\_Ticket\_Click">

<Button.Background>

<SolidColorBrush Color="#FFFA8072" />

</Button.Background>

</Button>

<Button Content="Добавить"

Grid.Row="2"

Grid.Column="0"

HorizontalAlignment="Left"

Width="100"

Name="BtnAdd"

Click="BtnAdd\_Click">

<Button.Background>

<SolidColorBrush Color="#FFFFFACD" />

</Button.Background>

</Button>

<Button Content="Удалить"

Grid.Row="2"

Grid.Column="3"

HorizontalAlignment="Right"

Width="100"

Name="BtnDelete"

Click="BtnDelete\_Click">

<Button.Background>

<SolidColorBrush Color="#FF87CEFA" />

</Button.Background>

</Button>

<Button Content="Таблица вопросов"

Grid.Row="2"

Grid.Column="2"

HorizontalAlignment="Right"

Width="150"

Name="QuestionTable"

Click="NextTable" >

<Button.Background>

<SolidColorBrush Color="#FFAFEEEE" />

</Button.Background>

</Button>

</Grid>

</Page>

**ViewingTicketTable.xaml.cs**

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для ViewingTicketTable.xaml

/// </summary>

public partial class ViewingTicketTable : Page

{

private string roleUser;

public ViewingTicketTable(string roleUser)

{

InitializeComponent();

this.roleUser = roleUser;

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e) //страница редактирования

{

NavigationService.Navigate(new EditingTicket((sender as Button).DataContext as Tickets, roleUser));

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e) //страница добавления

{

NavigationService.Navigate(new AddTicket(roleUser));

}

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

{

var ticketForRemoving = DGridTicket.SelectedItems.Cast<Tickets>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {ticketForRemoving.Count()} элементов?", "Внимание",

MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

RandomTicketGenerator.GetContext().Tickets.RemoveRange(ticketForRemoving);

RandomTicketGenerator.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGridTicket.ItemsSource = RandomTicketGenerator.GetContext().Tickets.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void ViewingTable\_admin\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

RandomTicketGenerator.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridTicket.ItemsSource = RandomTicketGenerator.GetContext().Tickets.ToList();

}

}

private void Form\_Ticket\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new Choice\_admin(roleUser));

}

private void NextTable(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ViewingTableData\_admin(roleUser));

}

}

}

**EditingTicket.xaml**

<Page x:Class="WpfApp1.EditingTicket"

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

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

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

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

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

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

Title="EditingTicket">

<Grid x:Name="EditPage\_admin" Background="#FFD3D3D3">

<Grid.RowDefinitions >

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock Grid.Row="0"

Grid.Column="1"

Grid.ColumnSpan="2"

Height="70"

Width="400"

HorizontalAlignment="Left"

FontSize="25"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="edit\_header"

VerticalAlignment="Center">

Редактирование билетов

</TextBlock>

<TextBlock Grid.Row="1"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="quest\_header"

VerticalAlignment="Center">

Идентификатор вопроса 1:

</TextBlock>

<TextBlock Grid.Row="2"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="type\_question\_header"

VerticalAlignment="Center">

Идентификатор вопроса 2:

</TextBlock>

<ComboBox x:Name="quest1Id"

Grid.Row="1"

Grid.Column="1"

Grid.ColumnSpan="2"

Height="70"

Width="600"

Background="White" />

<ComboBox x:Name="quest2Id"

Grid.Row="2"

Grid.Column="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="70"

Width="600"

SelectedIndex="0"

Background="White"/>

<TextBlock Grid.Row="3"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

TextWrapping="Wrap"

x:Name="complexity\_header"

VerticalAlignment="Center">

Идентификатор вопроса 3:

</TextBlock>

<TextBlock Grid.Row="4"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

TextWrapping="Wrap"

x:Name="nom\_komplect\_header"

VerticalAlignment="Center">

Номер комплекта:

</TextBlock>

<ComboBox x:Name="quest3Id"

Grid.Row="3"

Grid.Column="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="70"

Width="600"

SelectedIndex="0"

Background="White"/>

<ComboBox x:Name="komplectId"

Grid.Row="4"

Grid.Column="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="70"

Width="600"

SelectedIndex="0"

Background="White"/>

<Button Grid.Row="5" Grid.Column="1" Grid.ColumnSpan="2"

Height="55" Width="250"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_Save\_Question"

Content="Сохранение"

Click="But\_Click\_Save\_Question" Margin="120,3,230,2">

<Button.Background>

<SolidColorBrush Color="#FF87CEFA" />

</Button.Background>

</Button>

<Button Grid.Row="5" Grid.Column="0"

Height="55" Width="200"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_viewing"

Content="Просмотр базы данных"

Click="But\_Click\_Viewing\_Table\_Data">

<Button.Background>

<SolidColorBrush Color="#FFFA8072" />

</Button.Background>

</Button>

</Grid>

</Page>

**EditingTicket.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для EditingTicket.xaml

/// </summary>

public partial class EditingTicket : Page

{

private Tickets \_selectedTicket;

private string roleUser;

public EditingTicket(Tickets selectedTick, string roleUser)

{

InitializeComponent();

\_selectedTicket = selectedTick;

DataContext = \_selectedTicket;

InitializationTicket();

this.roleUser = roleUser;

InitializeBoxes();

}

private void InitializationTicket()

{

var TicketList = from i in RandomTicketGenerator.GetContext().Tickets.ToList()

select i;

DataContext = TicketList;

quest1Id.ItemsSource = TicketList;

quest1Id.SelectedValuePath = "";

quest1Id.DisplayMemberPath = "id\_quest1";

quest1Id.SelectedIndex = 0;

quest2Id.ItemsSource = TicketList;

quest2Id.SelectedValuePath = "";

quest2Id.DisplayMemberPath = "id\_quest2";

quest2Id.SelectedIndex = 0;

quest3Id.ItemsSource = TicketList;

quest3Id.SelectedValuePath = "";

quest3Id.DisplayMemberPath = "id\_quest3";

quest3Id.SelectedIndex = 0;

komplectId.ItemsSource = TicketList;

komplectId.SelectedValuePath = "";

komplectId.DisplayMemberPath = "nom\_komplect";

komplectId.SelectedIndex = 0;

}

private void InitializeBoxes()

{

quest1Id.Text = \_selectedTicket.id\_quest1.ToString();

quest2Id.Text = \_selectedTicket.id\_quest2.ToString();

quest3Id.Text = \_selectedTicket.id\_quest3.ToString();

komplectId.Text = \_selectedTicket.nom\_komplect.ToString();

}

private void UpdateTicket()

{

\_selectedTicket.id\_quest1 = ((Tickets)quest1Id.SelectedItem).id\_quest1;

\_selectedTicket.id\_quest2 = ((Tickets)quest2Id.SelectedItem).id\_quest2;

\_selectedTicket.id\_quest3 = ((Tickets)quest3Id.SelectedItem).id\_quest3;

\_selectedTicket.nom\_komplect = ((Tickets)komplectId.SelectedItem).nom\_komplect;

}

private void But\_Click\_Save\_Question(object sender, RoutedEventArgs e)

{

//if (string.IsNullOrWhiteSpace(\_selectedQuestion.question))

//{

// MessageBox.Show("Корректно напишите вопрос");

// return;

//}

UpdateTicket();

try

{

RandomTicketGenerator.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

private void But\_Click\_Viewing\_Table\_Data(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ViewingTicketTable(roleUser));

}

}

}

**AddTicket.xaml**

<Page x:Class="WpfApp1.AddTicket"

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

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

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

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

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

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

Title="AddTicket">

<Grid x:Name="AddTicket\_" Background="#FFD3D3D3">

<Grid.RowDefinitions >

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock Grid.Row="0"

Grid.Column="1"

Grid.ColumnSpan="2"

Height="70"

Width="400"

HorizontalAlignment="Left"

FontSize="25"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="edit\_header"

VerticalAlignment="Center">

Добавление билетов

</TextBlock>

<TextBlock Grid.Row="1"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="quest\_header"

VerticalAlignment="Center">

Идентификатор вопроса 1:

</TextBlock>

<TextBlock Grid.Row="2"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

x:Name="type\_question\_header"

VerticalAlignment="Center">

Идентификатор вопроса 2:

</TextBlock>

<TextBlock Grid.Row="3"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

TextWrapping="Wrap"

x:Name="complexity\_header"

VerticalAlignment="Center">

Идентификатор вопроса 3:

</TextBlock>

<ComboBox x:Name="quest1Id"

Grid.Row="1"

Grid.Column="1"

Grid.ColumnSpan="2"

Height="70"

Width="600"

Background="White" />

<ComboBox x:Name="quest2Id"

Grid.Row="2"

Grid.Column="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="70"

Width="600"

SelectedIndex="0"

Background="White"/>

<ComboBox x:Name="quest3Id"

Grid.Row="3"

Grid.Column="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="70"

Width="600"

SelectedIndex="0"

Background="White"/>

<TextBlock x:Name="komplectTextBox"

Grid.Row="4"

Grid.Column="0"

Height="30"

Width="200"

FontSize="12"

FontFamily="Verdana"

FontWeight="Bold"

TextWrapping="Wrap"

VerticalAlignment="Center">

Номер комплекта билетов:

</TextBlock>

<ComboBox x:Name="komplectId"

Grid.Row="4"

Grid.Column="1"

Grid.ColumnSpan="2"

HorizontalAlignment="Left"

VerticalAlignment="Top"

Height="70"

Width="600"

SelectedIndex="0"

Background="White"/>

<Button Grid.Row="5" Grid.Column="1" Grid.ColumnSpan="2"

Height="70" Width="250"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_Save\_Question"

Content="Сохранение" Click="But\_Click\_Save\_Ticket"

>

<Button.Background>

<SolidColorBrush Color="#FF87CEFA" />

</Button.Background>

</Button>

<Button Grid.Row="5" Grid.Column="0"

Height="70" Width="200"

FontSize="12" FontFamily="Verdana" FontWeight="Bold"

x:Name="But\_viewing"

Content="Просмотр базы данных" Click="But\_viewing\_Click"

>

<Button.Background>

<SolidColorBrush Color="#FFFA8072" />

</Button.Background>

</Button>

</Grid>

</Page>

**AddTicket.xaml.cs**

using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace WpfApp1

{

public partial class AddTicket : Page

{

private string roleUser;

public AddTicket(string roleUser)

{

InitializeComponent();

Initialization();

DataContext = NewTicket();

this.roleUser = roleUser;

}

private void Initialization()

{

var TicketList = from i in RandomTicketGenerator.GetContext().Tickets.ToList()

select i;

DataContext = TicketList;

quest1Id.ItemsSource = TicketList;

quest1Id.SelectedValuePath = "";

quest1Id.DisplayMemberPath = "id\_quest1";

quest1Id.SelectedIndex = 0;

quest2Id.ItemsSource = TicketList;

quest2Id.SelectedValuePath = "";

quest2Id.DisplayMemberPath = "id\_quest2";

quest2Id.SelectedIndex = 0;

quest3Id.ItemsSource = TicketList;

quest3Id.SelectedValuePath = "";

quest3Id.DisplayMemberPath = "id\_quest3";

quest3Id.SelectedIndex = 0;

komplectId.ItemsSource = TicketList;

komplectId.SelectedValuePath = "";

komplectId.DisplayMemberPath = "nom\_komplect";

komplectId.SelectedIndex = 0;

}

private Tickets NewTicket() //новый билет

{

return new Tickets

{

id\_quest1 = ((Tickets)quest1Id.SelectedItem).id\_quest1,

id\_quest2 = ((Tickets)quest2Id.SelectedItem).id\_quest2,

id\_quest3 = ((Tickets)quest3Id.SelectedItem).id\_quest3,

nom\_komplect = ((Tickets)komplectId.SelectedItem).nom\_komplect

};

}

private void But\_Click\_Save\_Ticket(object sender, RoutedEventArgs e)

{

var currentTicket = NewTicket();

try

{

RandomTicketGenerator.GetContext().Tickets.Add(currentTicket);

RandomTicketGenerator.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

private void But\_viewing\_Click(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ViewingTicketTable(roleUser));

}

}

}

**Sign\_up.xaml**

<Page x:Class="WpfApp1.Sign\_up"

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

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

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

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

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="700" d:DesignWidth="1200"

Title="Sign\_up">

<Grid Background="#FFD3D3D3">

<Grid.ColumnDefinitions>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"

Height="70" Width="680"

FontSize="40" FontFamily="Verdana" FontWeight="Bold" x:Name="Header">

Регистрация пользователя

</TextBlock>

<TextBlock Grid.Row="1" Grid.Column="0"

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="role">

Роль:

</TextBlock>

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

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="login">

Логин:

</TextBlock>

<TextBlock Grid.Row="3" Grid.Column="0"

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="password">

Пароль:

</TextBlock>

<ComboBox Name="Role"

Grid.Row="1"

Grid.Column="1"

Height="70" Width="400"

HorizontalAlignment="Center"

VerticalAlignment="Top" Margin="0,41,0,0" />

<TextBox Grid.Row="2" Grid.Column="1"

Height="70" Width="400"

MaxLength="50"

TextWrapping="Wrap"

x:Name="textBox\_login" Margin="100,41,100,29">

</TextBox>

<PasswordBox Grid.Row="3" Grid.Column="1"

Height="70" Width="400"

x:Name="textBox\_password" Margin="100,40,100,30"

MaxLength="50">

</PasswordBox>

<Button Grid.Row="4" Grid.Column="0" Width="400"

FontSize="30" FontFamily="Verdana" FontWeight="Bold"

x:Name="But2"

Content="Регистрация"

Click="But\_registration" Margin="0,10,0,16">

<Button.Background>

<SolidColorBrush Color="#FFFA8072" />

</Button.Background>

</Button>

<Button Grid.Row="4" Grid.Column="1" Width="400"

FontSize="30" FontFamily="Verdana" FontWeight="Bold"

x:Name="But1"

Content="Авторизация"

Click="But\_authorization" Margin="0,10,0,16">

<Button.Background>

<SolidColorBrush Color="#FF87CEFA" />

</Button.Background>

</Button>

</Grid>

</Page>

**Sign\_up.xaml**

<Page x:Class="WpfApp1.Sign\_up"

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

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

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

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

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="700" d:DesignWidth="1200"

Title="Sign\_up">

<Grid Background="#FFD3D3D3">

<Grid.ColumnDefinitions>

<ColumnDefinition>

</ColumnDefinition>

<ColumnDefinition>

</ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

<RowDefinition>

</RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"

Height="70" Width="680"

FontSize="40" FontFamily="Verdana" FontWeight="Bold" x:Name="Header">

Регистрация пользователя

</TextBlock>

<TextBlock Grid.Row="1" Grid.Column="0"

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="role">

Роль:

</TextBlock>

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

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="login">

Логин:

</TextBlock>

<TextBlock Grid.Row="3" Grid.Column="0"

Height="70" Width="350"

FontSize="30" FontFamily="Verdana" FontWeight="Bold" x:Name="password">

Пароль:

</TextBlock>

<ComboBox Name="Role"

Grid.Row="1"

Grid.Column="1"

Height="70" Width="400"

HorizontalAlignment="Center"

VerticalAlignment="Top" Margin="0,41,0,0" />

<TextBox Grid.Row="2" Grid.Column="1"

Height="70" Width="400"

MaxLength="50"

TextWrapping="Wrap"

x:Name="textBox\_login" Margin="100,41,100,29">

</TextBox>

<PasswordBox Grid.Row="3" Grid.Column="1"

Height="70" Width="400"

x:Name="textBox\_password" Margin="100,40,100,30"

MaxLength="50">

</PasswordBox>

<Button Grid.Row="4" Grid.Column="0" Width="400"

FontSize="30" FontFamily="Verdana" FontWeight="Bold"

x:Name="But2"

Content="Регистрация"

Click="But\_registration" Margin="0,10,0,16">

<Button.Background>

<SolidColorBrush Color="#FFFA8072" />

</Button.Background>

</Button>

<Button Grid.Row="4" Grid.Column="1" Width="400"

FontSize="30" FontFamily="Verdana" FontWeight="Bold"

x:Name="But1"

Content="Авторизация"

Click="But\_authorization" Margin="0,10,0,16">

<Button.Background>

<SolidColorBrush Color="#FF87CEFA" />

</Button.Background>

</Button>

</Grid>

</Page>

**Sign\_up.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

using System.Data.SqlClient;

using System.Data;

namespace WpfApp1

{

public partial class Sign\_up : Page

{

RandomTicketGenerator db;

DataBase database = new DataBase();

public Sign\_up()

{

InitializeComponent();

bindcombo\_Users();

db = new RandomTicketGenerator();

}

//ComboBox kurs

public List<Users> Users\_list { get; set; }

private void bindcombo\_Users()

{

Role.Items.Add("Admin");

Role.Items.Add("User");

Role.SelectedIndex = 0;

}

/// <summary>

/// сокрытие пароля

/// </summary>

private void log\_in\_load(object sender, EventArgs e)

{

textBox\_password.PasswordChar = '•';

textBox\_password.MaxLength = 100;

}

/// <summary>

/// проверка на наличие пользователя в бд

/// </summary>

private Boolean check\_user()

{

var login = textBox\_login.Text;

var role = Role.Text;

var password = textBox\_password.Password.ToString();

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querystring = $"select nom\_user, login\_, password\_, role\_ from Users where login\_ = '{login}' and role\_ = '{role}'";

//, password\_ = '{passUser}

SqlCommand command = new SqlCommand(querystring, database.getConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

MessageBox.Show("Пользователь уже существует");

return true;

}

else

{

return false;

}

}

/// <summary>

/// регистрация

/// </summary>

private void But\_registration(object sender, RoutedEventArgs e)

{

var login = textBox\_login.Text;

var role = Role.Text;

var password = textBox\_password.Password.ToString();

if ((login != "") && (role != "") && (password != ""))

{

if ((role == "User") || (role == "Admin"))

{

try

{

if (db.Users.Any(o => o.login\_ == login))

{

throw new Exception("Логин уже занят");

}

Users newUser = new Users();

newUser.login\_ = login;

newUser.password\_ = password;

newUser.role\_ = role;

if (!check\_user())

{

return;

}

db.Users.Add(newUser);

db.SaveChanges();

MessageBox.Show("Регистрация прошла успешно");

NavigationService.Navigate(new Authorization());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else

{

MessageBox.Show("Неверно указана роль пользователя");

}

}

else MessageBox.Show("Для продолжения заполните все поля");

}

/// <summary>

/// Переход к авторизации

/// </summary>

private void But\_authorization(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new Authorization());

}

}

}