|  |  |
| --- | --- |
| Логотип КБК копия | **Автономная некоммерческая организация профессионального образования**  **КАЛИНИНГРАДСКИЙ БИЗНЕС-КОЛЛЕДЖ** |

**Курсовая работа**

На тему: «База данных городской думы»

По предмету: МДК 01.01 «Разработка программных модулей»

Выполнил студент

Группы 23-ИСП-3/1

Одинаев В. М.

Проверил:

Дубинин А. В.

Оценка:

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

Калининград

2024

**Задание**

В базе хранятся имена, адреса, домашние и служебные телефоны всех членов Думы. В Думе работает порядка сорока комиссий, все участники которых являются членами Думы. Каждая комиссия имеет свой профиль, например вопросы образования, проблемы, связанные с жильем и так далее. Данные по каждой из комиссий включают: ее нынешний состав и председатель, прежние председатели и члены этой комиссии, участвовавшие в ее работе за прошедшие 10 лет, даты включения и выхода из состава комиссии, избрания ее председателей. Члены Думы могут заседать в нескольких комиссиях. В базу заносятся время и место проведения каждого заседания комиссии с указанием служащих Думы, которые участвуют в его организации.

Целью курсовой работы является разработка и реализация базы данных городской думы с использованием SQLite и создание приложения на языке C# для управления этой базой данных, включающее проектирование БД (анализ предметной области, инфологическое и физическое моделирование), реализацию заданных и самостоятельно сформулированных запросов на выборку, обновление и удаление данных, а также создание удобного пользовательского интерфейса для взаимодействия с базой данных.

**Содержание**

[Содержание 3](#_Toc211957450)

[Введение 4](#_Toc211957451)

[1. Описание программного продукта 5](#_Toc211957452)

[1.1 Общие сведения о программном продукте 5](#_Toc211957453)

[1.2 Функциональное назначение программного продукта 5](#_Toc211957454)

[1.3 Средства проектирования и разработки 6](#_Toc211957455)

[2. Разработка программного продукта 8](#_Toc211957456)

[2.1 Разработка базы данных 8](#_Toc211957457)

[2.2 Логическая структура программного продукта 10](#_Toc211957458)

[2.3 Основные конструкции и процедуры 11](#_Toc211957459)

[3. Инициализация пользовательского интерфейса 13](#_Toc211957460)

[Заключение 21](#_Toc211957461)

[Список использованных источников 22](#_Toc211957462)

# Введение

В условиях современного развития местного самоуправления и повышения требований к прозрачности деятельности представительных органов власти особую актуальность приобретают вопросы автоматизации процессов управления и документооборота в муниципальных образованиях. Городская дума как ключевой представительный орган местного самоуправления осуществляет значительный объем работы, связанной с организацией деятельности постоянных комиссий, учетом депутатского состава, планированием заседаний и анализом эффективности работы.

Существующие ручные методы учета и отчетности не позволяют оперативно получать актуальную информацию о деятельности думы, затрудняют анализ посещаемости заседаний, усложняют процесс формирования отчетности и принятия управленческих решений. Отсутствие единой информационной системы приводит к дублированию данных, повышению вероятности ошибок и значительным временным затратам на подготовку аналитических материалов.

Разработка специализированного программного обеспечения для автоматизации деятельности городской думы позволяет решить эти проблемы, повысив эффективность работы аппарата думы, обеспечив прозрачность процессов и своевременность получения аналитической информации.

Разработка и внедрение информационной системы для автоматизации учета и управления деятельностью городской думы, обеспечивающей комплексное решение задач по ведению базы данных депутатов, комиссий, заседаний и формированию отчетной документации.

Разрабатываемая система призвана стать эффективным инструментом поддержки принятия решений для руководства городской думы и значительным подспорьем в повседневной работе аппарата думы, способствуя повышению эффективности управления и прозрачности деятельности представительного органа местного самоуправления.

# Описание программного продукта

## 1.1 Общие сведения о программном продукте

Разработанный программный продукт представляет собой специализированную информационную систему для автоматизации деятельности городской думы. Система охватывает ключевые аспекты работы представительного органа местного самоуправления, включая управление депутатским корпусом, организацию работы постоянных комиссий, планирование и учет заседаний, а также комплексный анализ деятельности.

Предметная область программного продукта включает полный цикл документооборота и учета, связанного с организационной структурой думы, кадровым составом депутатов, их участием в работе комиссий и посещаемостью заседаний. Система обеспечивает централизованное хранение и обработку информации обо всех аспектах деятельности думы, начиная от персональных данных депутатов и заканчивая статистикой их рабочей активности.

## 1.2 Функциональное назначение программного продукта

Программный продукт предназначен для применения в аппарате городской думы и предназначен для использования сотрудниками, ответственными за организацию работы представительного органа. Система будет применяться в повседневной работе для ведения базы данных депутатов, формирования состава комиссий, планирования заседаний и подготовки отчетных материалов.

Основное применение системы заключается в замене ручных методов учета и бумажного документооборота на автоматизированное решение, что позволяет значительно сократить временные затраты на подготовку документов и повысить достоверность учетных данных. Продукт будет использоваться при подготовке к заседаниям думы, формировании отчетов о деятельности комиссий, анализе рабочей нагрузки депутатов и планировании организационных мероприятий.

Особенностью программного продукта является его ориентация на специфические потребности представительных органов местного самоуправления, что отличает его от универсальных систем документооборота. Система учитывает особенности организационной структуры думы, правила формирования комиссий и требования к отчетности в соответствии с муниципальными нормативными актами.

К преимуществам системы можно отнести возможность оперативного получения актуальной информации о составе комиссий и их деятельности, автоматическое формирование отчетности, снижение вероятности ошибок при учете данных и повышение прозрачности работы думы. К ограничениям относится необходимость обучения сотрудников работе с системой и зависимость от корректности вводимых данных.

## 1.3 Средства проектирования и разработки

Разработка программного продукта осуществлялась с использованием современной стека технологий Microsoft. В качестве среды разработки применялась Visual Studio 2022, предоставляющая комплексные инструменты для создания desktop-приложений. Основным языком программирования выбран C# с платформой .NET 8, что обеспечивает высокую производительность, типобезопасность и богатую библиотеку компонентов для построения пользовательских интерфейсов.

Для реализации пользовательского интерфейса использовалась технология WPF (Windows Presentation Foundation), позволяющая создавать современные и функциональные приложения с поддержкой сложной графики и анимации. В качестве системы управления базами данных применена SQLite - легковесная реляционная СУБД, не требующая отдельной установки и идеально подходящая для «автономных» приложений.

Для объектно-реляционного отображения использовался Entity Framework Core - современный ORM-фреймворк, обеспечивающий удобную работу с данными через LINQ-запросы и автоматическую генерацию схемы базы данных. Для создания современного пользовательского интерфейса подключена библиотека «Material Design Themes», реализующая принципы «material design» в WPF-приложениях.

Архитектура системы построена по паттерну MVVM (Model-View-ViewModel), что обеспечивает четкое разделение ответственности между компонентами, упрощает тестирование и поддержку кода. Такой подход позволяет легко расширять функциональность системы и модифицировать пользовательский интерфейс без изменения бизнес-логики приложения.

# Разработка программного продукта

## 2.1 Разработка базы данных

На рисунке 2.1 «Схема базы данных» визуально проиллюстрированы все таблицы, поля и связи между ними в разработанной базе данных.

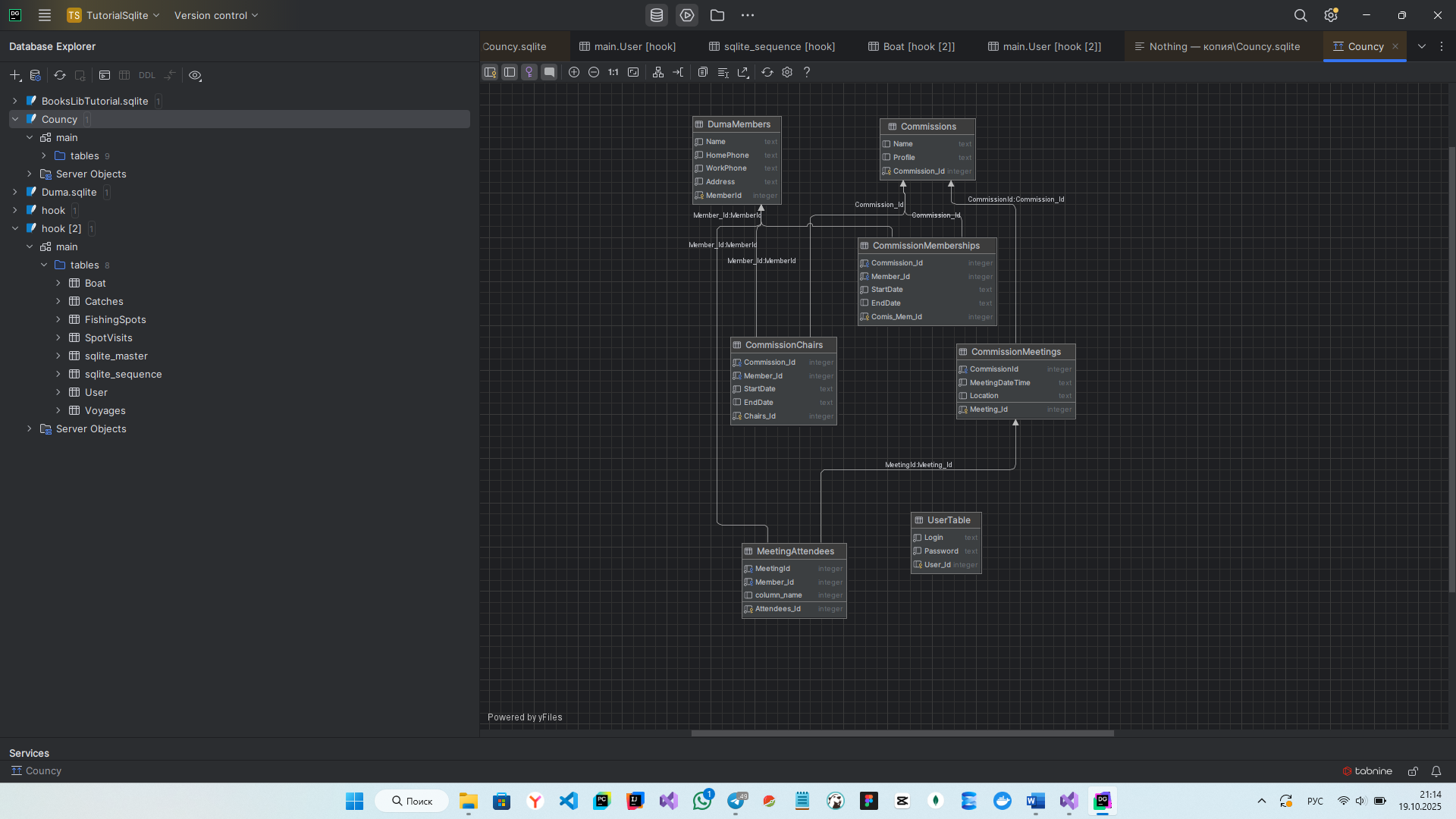


Рис. 2.1. «Схема базы данных»

**DumaMembers –** хранит данные о членах городской думы:

* **MemberId** – первичный ключ, тип данных integer;
* **Name** – полное имя члена, тип данных string;
* **HomePhone** – домашний телефон члена думы, тип данных string;
* **WorkPhone** – рабочий телефон члена думы, тип данных string;
* **Address** – адрес члена думы, тип данных string.

**Commissions** – хранит основную информацию о комиссиях:

* **CommissionId** – уникальный идентификатор, первичный ключ, тип данных integer;
* **Name** – название комиссии, тип данных text;
* **Profile** – хранит информацию о деятельности комиссии, тип данных text.

**CommissionMemberships** – хранит информацию об участниках комиссий:

* **ComisMemId** – уникальный идентификатор, первичный ключ, тип данных ineger;
* **CommissionId** – внешний ключ из таблицы ‘Commissions’, тип данных integer;
* **MemberId** – внешний ключ из таблицы ‘DumaMembers’, тип данных integer;
* **StartDate** – дата начала участия в комиссии, тип данных DateTime;
* **EndDate** – дата окончания участия в комиссии, тип данных DateTime.

**CommissionChairs** – хранит информацию о председателях комиссий:

* **ChairsId** – уникальный идентификатор, первичный ключ, тип данных integer;
* **CommissionId** – уникальный идентификатор, первичный ключ, тип данных integer;
* **MemberId** – внешний ключ из таблицы ‘DumaMembers’, тип данных integer;
* **StartDate** – дата назначения председателем, тип данных DateTime;
* **EndDate** – дата окончания срока назначения председателем, тип данных DateTime.

**CommissionMeetings** – хранит информацию о заседаниях:

* **MeetingId** – уникальный идентификатор, первичный ключ, тип данных integer;
* **CommissionId** – уникальный идентификатор, первичный ключ, тип данных integer;
* **MeetingDateTime** – дата проведения заседания, тип данных DateTime;
* **Location** – место проведения заседания, тип данных text;

**MeetingAttendees –** информация об участниках заседаний:

* **AttendeesId** – уникальный идентификатор, первичный ключ, тип данных integer;
* **MeetingId** - внешний ключ из таблицы ‘Meetings’, тип данных integer;
* **MemberId** - внешний ключ из таблицы ‘DumaMembers’, тип данных integer.

**UserTable** – хранит информацию о пользователях, зарегистрированных в системе управления базы данных городской думы:

* **UserId** – уникальный идентификатор, первичный ключ, тип данных integer;
* **Login** – имя пользователя, тип данных text;
* **Password** – пароль, тип данных text.

## 2.2 Логическая структура программного продукта

Программный продукт построен по трехуровневой архитектуре, где каждый уровень выполняет строго определенные функции. На верхнем уровне находятся пользовательские интерфейсы - WPF-окна, которые обеспечивают визуальное представление данных и взаимодействие с пользователем. На среднем уровне расположены ViewModel-классы, которые содержат логику представления, обрабатывают пользовательские действия и преобразуют данные для отображения. На нижнем уровне находятся модели данных и классы доступа к данным, которые отвечают за хранение и обработку бизнес-сущностей. Схематически взаимосвязь между уровнями можно посмотреть на рисунке 2.1 «Взаимодействие в архитектуре MVVM».

Изображение выглядит как текст, Шрифт, снимок экрана, диаграмма

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 2.1. Взаимодействие в архитектуре MVVM

## 2.3 Основные конструкции и процедуры

Рассмотрим ключевую конструкцию системы - базовый класс BaseClass, который обеспечивает механизм уведомлений об изменении свойств (см. рис. 2.2 «BaseClass»).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 2.2. BaseClass

Данный код реализует интерфейс встроенный ‘INotifyPropertyChanged’, который является фундаментальным для механизма привязки данных в WPF. Когда свойство в модели данных изменяется, вызывается метод ‘OnPropertyChanged’, который генерирует событие ‘PropertyChanged’. Это событие перехватывается системой привязки данных WPF, которая автоматически обновляет все элементы пользовательского интерфейса, привязанные к измененному свойству. Атрибут ‘CallerMemberName’ позволяет автоматически определить имя свойства, из которого был вызван метод, что исключает возможность ошибок при указании имени свойства вручную. Пример реализации данного класса можно увидеть на рисунке 2.3 «Реализация ‘BaseClass’ в классе-наследнике».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 2.3. Реализация ‘BaseClass’в классе-наследнике

# 3. Инициализация пользовательского интерфейса

При запуске программы открывается окно авторизации (см. Рис. 3.1 «Окно авторизации»), где пользователь системы может либо зарегистрироваться, указав имя пользователя и пароль нажав на кнопку ‘Регистрация’, после чего ему покажется окно с уведомлением об успешной регистрации (см. Рис. 3.2 «Сообщение об успешной регистрации»), при этом стоит учесть что пользователь может ввести либо латинские символы верхнего и нижнего регистров (a-z, A-Z) либо арабские цифры от 0 до 9. В случае, если пользователь ранее уже был зарегистрирован, то достаточно ввести валидные данные (условия те же, что при регистрации) и нажать на кнопку ‘Войти’, после чего ему высветится диалоговое окно с приветствием и откроется доступ к системе (см. рис. 3.3 «Диалоговое окно с приветствием»).

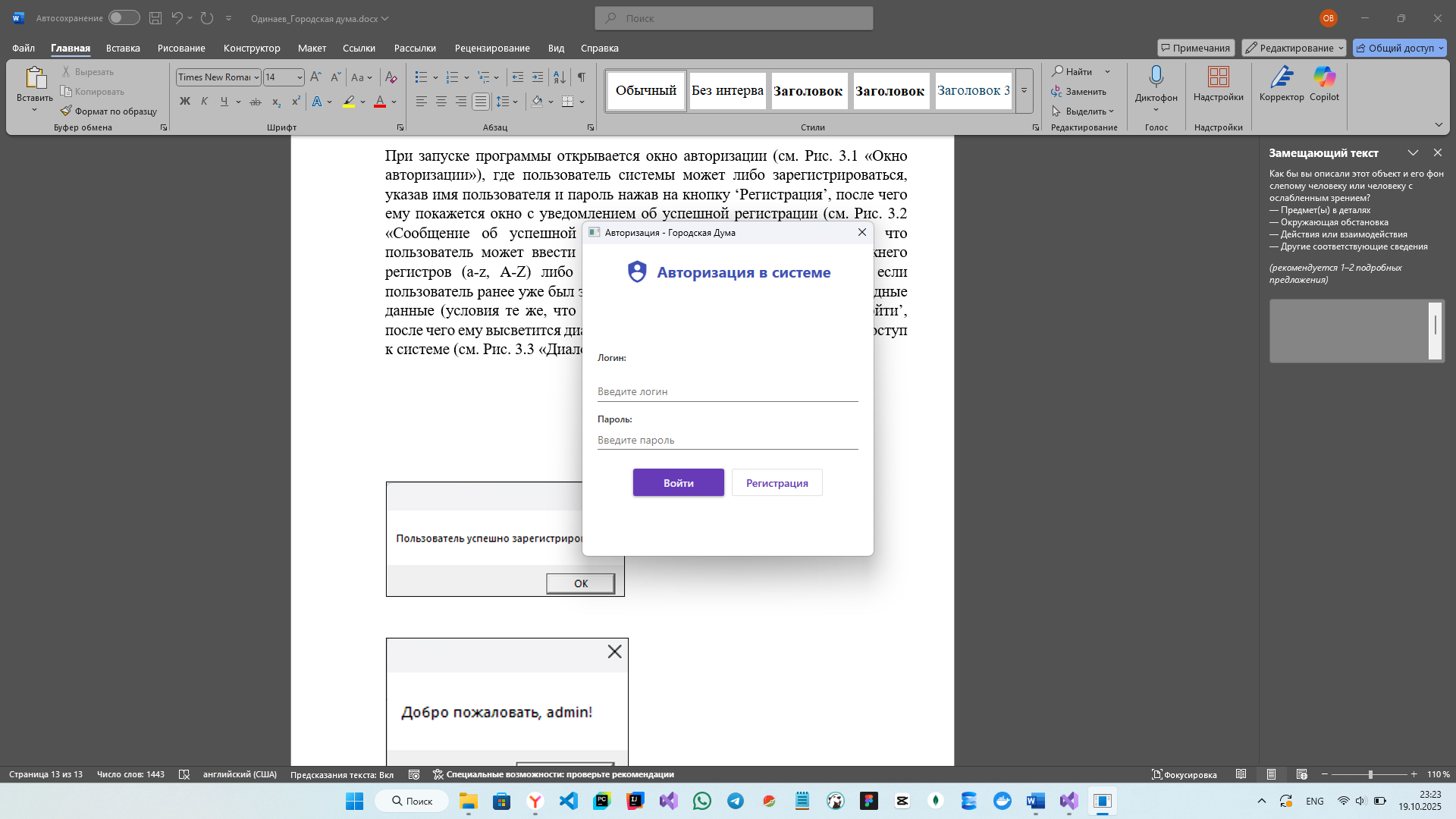


Рис. 3.1. Окно авторизации

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 3.2. Сообщение об успешной регистрации

Изображение выглядит как текст, программное обеспечение, снимок экрана, компьютер

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 3.3. Диалоговое окно с приветствием

После успешной авторизации пользователь попадает в главное меню (см. рис. 3.4 «Главное меню»), где предоставлены все доступные запросы. Каждая кнопка, отображаемая в главном меню, выполняет определённый запрос. О каждом из них подробнее рассказано далее.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 3.4. Главное меню

При нажатии на кнопку ‘Q1: Состав комиссий’ открывается окно отчёта по комиссиям с отображением названия комиссии, его председателя и участников со сроком службы (см. Рис. 3.5 «Отчёт по комиссиям»). В левой части окна находится раздел с комиссиями, где можно выбрать и нажать на любую из них, чтобы посмотреть подробную о ней информацию, которая будет отображаться в правой части окна. Чтобы закрыть окно, можно воспользоваться кнопкой ‘Закрыть’ в правом нижнем углу, после чего пользователь перейдёт на главный экран.

Изображение выглядит как текст, программное обеспечение, снимок экрана, дисплей

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 3.5. Отчёт по комиссиям

Далее идёт кнопка ‘Отчёт по посещаемости’, при нажатии на которую открывается окно с информацией о проведённых заседаниях и их участниках с указанием комиссии и диапазона дат (см. Рис. 3.6 «Отчёт по посещаемости заседаний»). Например, выберем комиссию ‘Миграционная служба’ из выпадающего списка, в полях ‘Дата начала’ и ‘Дата окончания’ введём даты 20.09.2020 и 20.10.2025 соответственно, что задаст временной интервал, в котором будут отображаться проведённые заседания соответствующей комиссии.

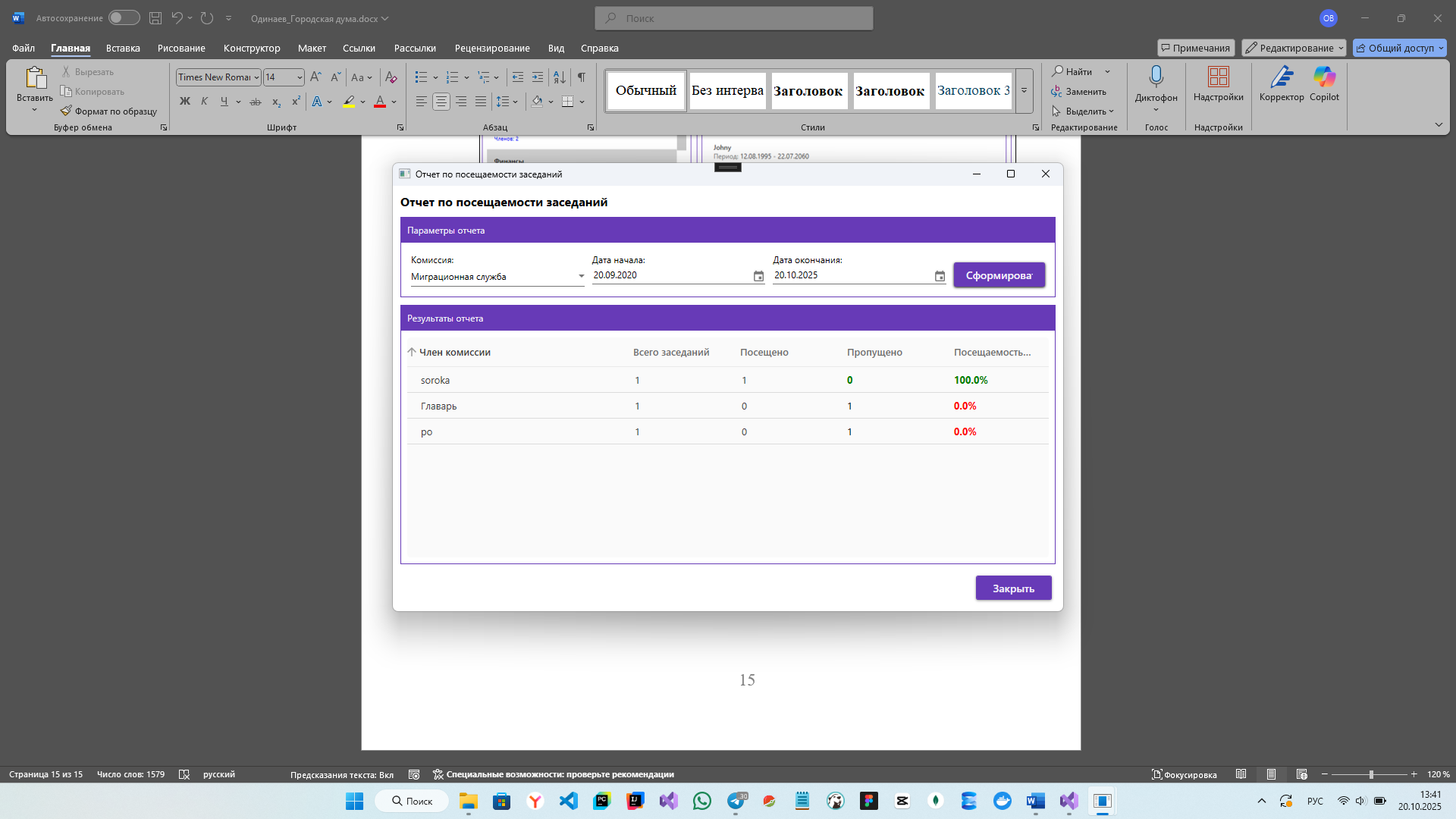


Рис. 3.6. Отчёт по посещаемости заседаний

Ещё один похожий запрос выполнится при нажатии на кнопку ‘Q: Активность комиссий’, где будет отчёт об количестве заседаний в каждой комиссии и её посещаемости в указанный временной интервал (см. рис. 3.6 «Активность комиссий за период».).

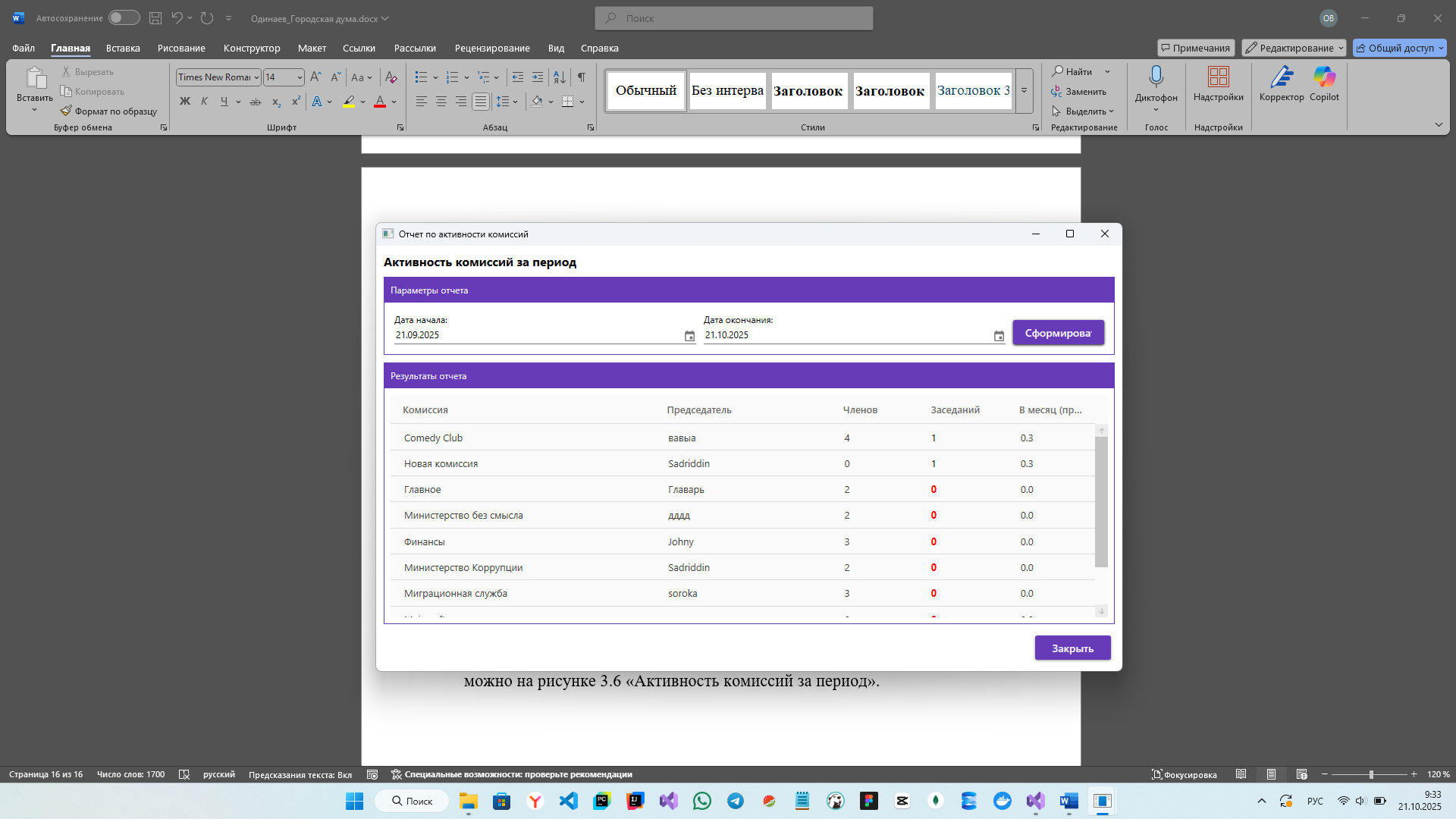


Рис. 3.6. Активность комиссий за период

Далле идут запросы из второй колонки под названием «Запросы», где реализован дополнительный функционал. И нажимая на первую кнопку ‘Члены Думы’ у нас открывается окно, изображённое на рисунке 3.7 «Окно Членов Думы». В нём отображаются все данные обо всех членах думы, их именах, номерах телефонов, адресах, а также комиссиях, в которых они заседают.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 3.7. Окно членов думы

В данном окне также, предоставлен функционал добавления, редактирования и удаления депутатов посредством кнопок ‘Добавить’, ‘Редактировать’ и ’Удалить’ внизу в правой части окна.

Первая кнопка – ‘Добавить’ открывает окно (см. рис. 3.8 «Редактировать участника»). В нём есть поля для ввода имени, домашнего и рабочего телефонов и адреса.

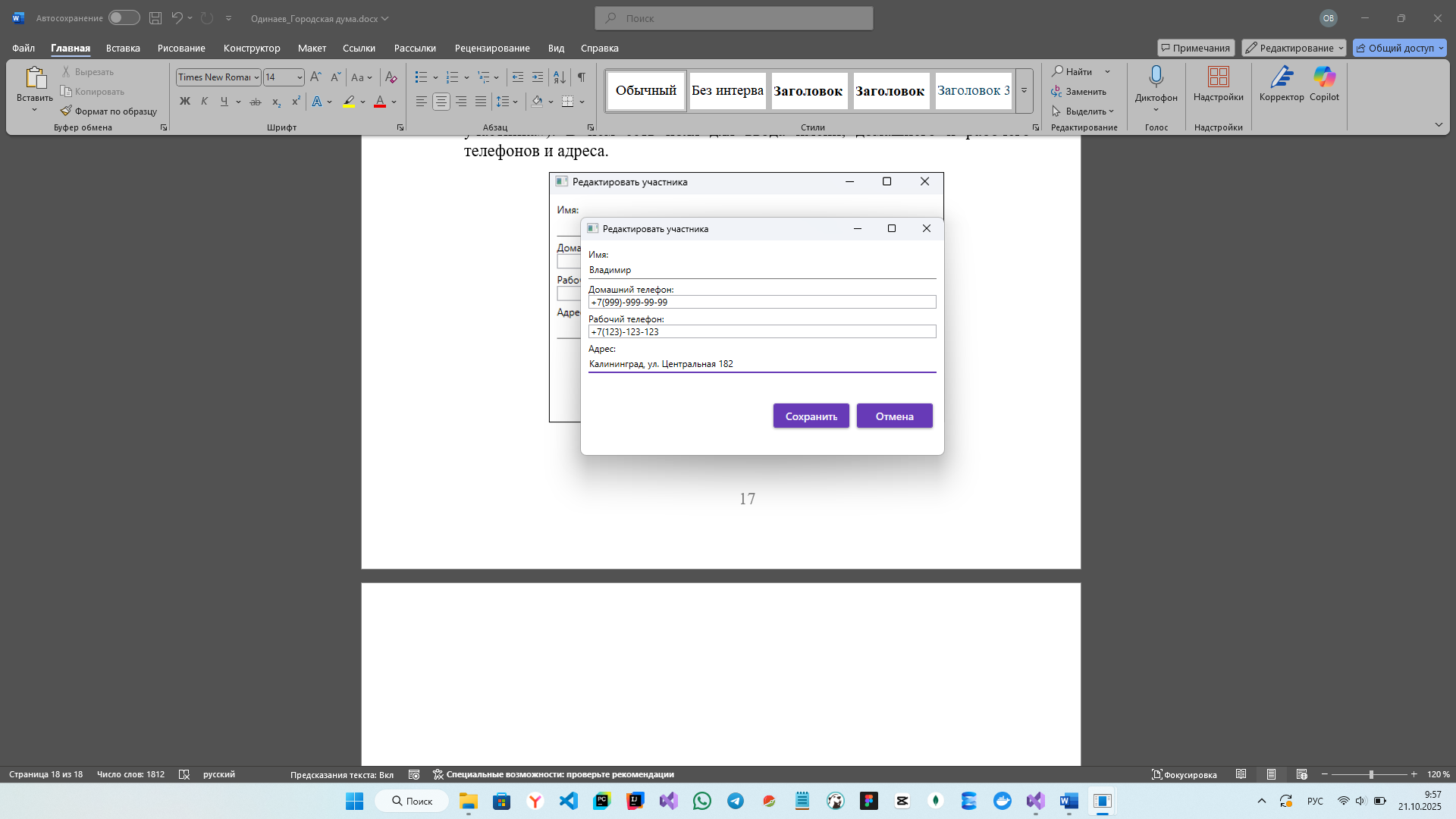


Рис. 3.8. Редактировать участника

Функционал редактирования реализован подобным образом, достаточно выбрать (навести и нажать левой кнопки на существующего участника) и нажать на кнопку ‘Редактировать’ и откроется то же самое окно, которое открывалось при добавлении нового члена Думы (см. рис. 3.8 «Редактировать участника»). Необходимо ввести все данные, что хотели изменить и нажать на кнопку ‘Сохранить’

Для удаления участника достаточно выбрать существующего члена Думы и нажать кнопку ‘Удалить’.

Чтобы посмотреть, добавить и отредактировать существующего члена Думы в комиссии нужно открыть соответствующее окно (см. рис. 3.9 «Членство в комиссиях»).

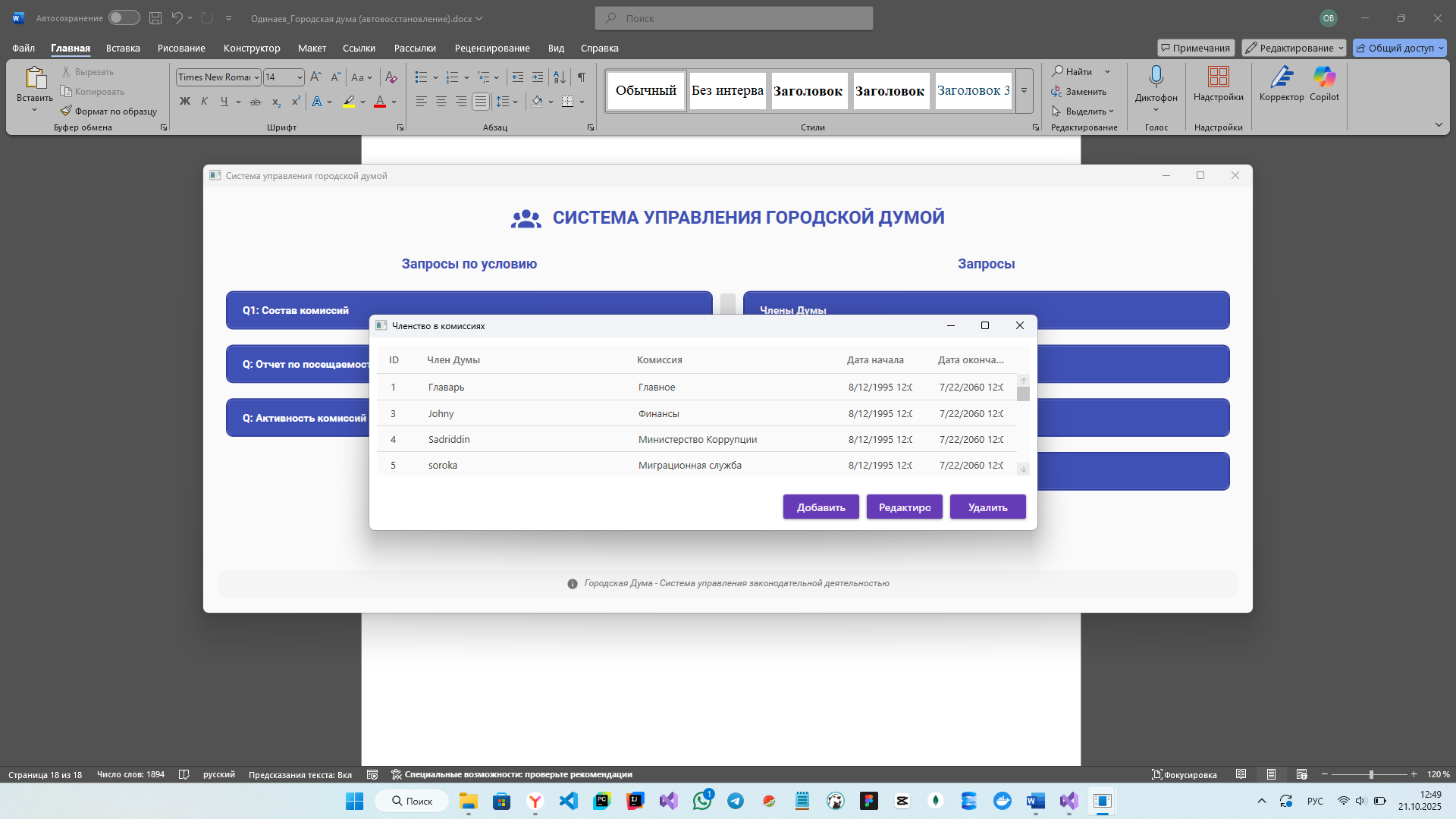


Рис. 3.9. Членство в комиссиях

Нажимая кнопку ‘Добавить’, откроется окно, где можно добавить нового участника комиссии (см. рис. 3.10 «Редактировать членство в комиссии»). Для этого нужно из выпадающих списков выбрать нужную комиссию и члена Думы которого хотим добавить в неё, также указываются даты вступления и ухода из неё.

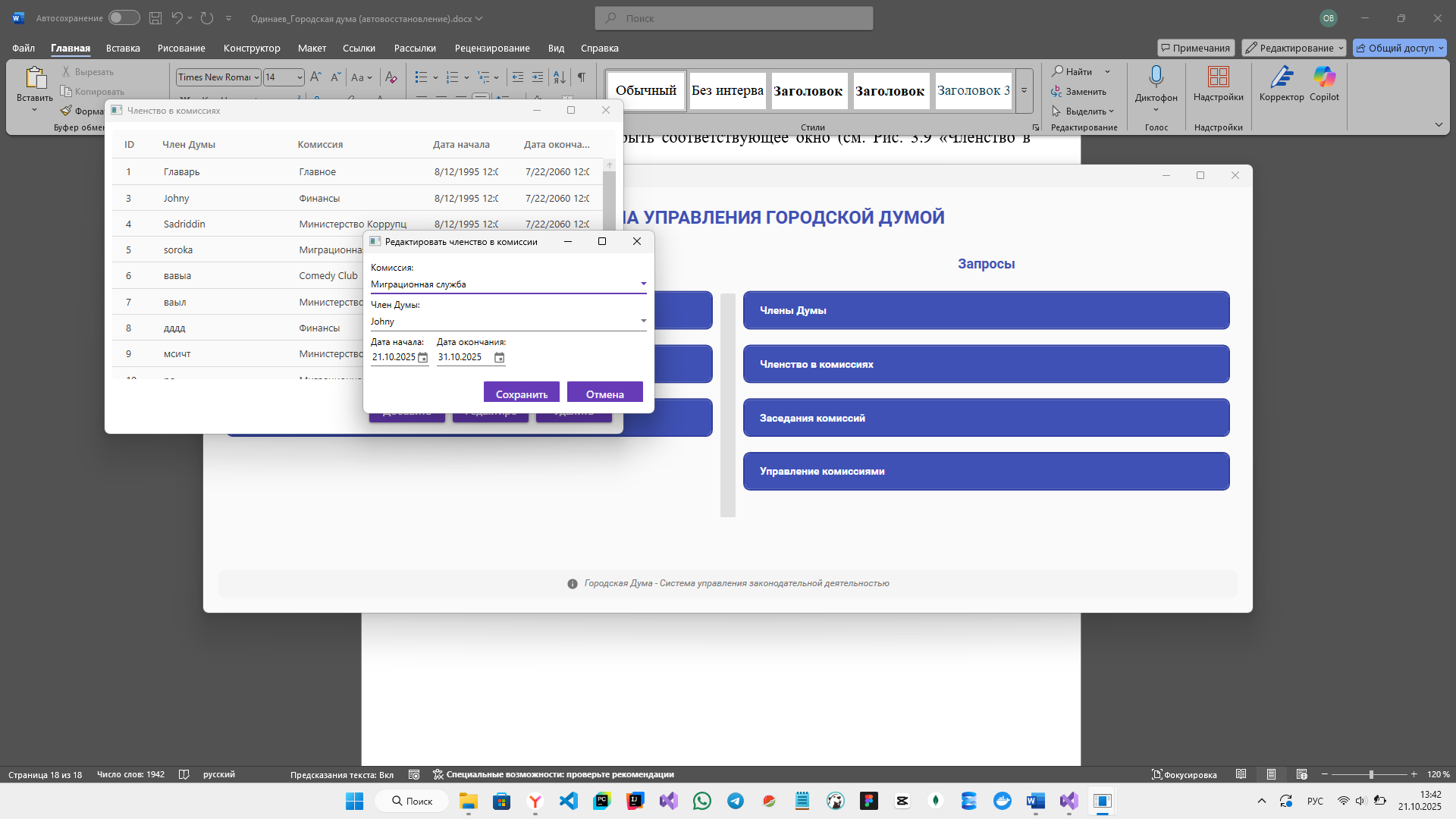


Рис. 3.10. Редактировать членство в комиссии

Для редактирования в отношения членов Думы к комиссиям можно воспользоваться кнопкой ‘Редактировать’, которая откроет то же окно, но выполнит функцию изменения данных в уже имеющемся отношении.

Чтобы удалить отношение, нужно выбрать его из списка, и нажать кнопку ‘Удалить’, что выполнить функцию удаления члена Думы из определённой комиссии.

Следующая кнопка ‘Заседания комиссий’ открывает окно (см. рис. 3.11 «Окно заседаний»). Тут реализован функционал просмотра, редактирования, добавления и удаления заседаний.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 3.11. Окно заседаний

Кнопка ‘Управление комиссиями’ открывает окно, где можно просматривать, создавать, редактировать и удалять комиссии (см. рис. 3.12 «Управление комиссиями»).

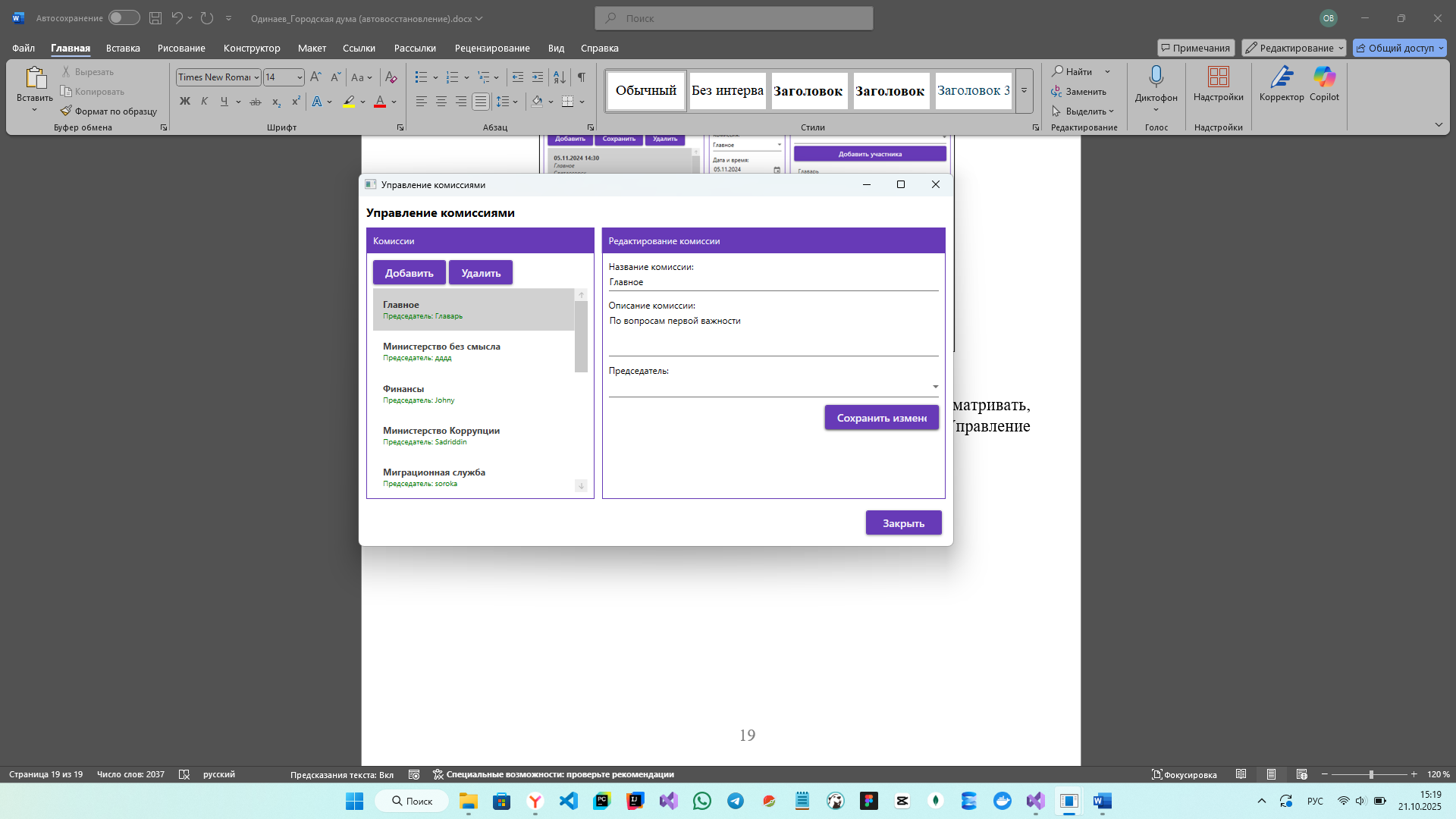


Рис. 3.12. Управление комиссиями

# Заключение

В ходе выполнения курсовой работы было успешно разработано краткое описание проекта, например: система управления базой данных городской Думы, приложение для управлением данной системой. Работа охватила полный цикл создания проекта, от проектирования базы данных до разработки пользовательского интерфейса.

Главное место в архитектуре приложения занял паттерн проектирования Model-View-View Model (MVVM), в сочетании с технологиями WPF (Windows Presentation Foundation) для построения пользовательского интерфейса и C# в качестве языка программирования. Применение MVVM позволило обеспечить четкое разделение модели данных, логики представления и пользовательского интерфейса. Использование привязки данных в WPF упростило синхронизацию данных между моделью и представлением, сделав интерфейс более отзывчивым и интуитивно понятным.

Для работы с данными была выбрана система управления базами данных SQLite, практическое освоение которой позволило получить опыт проектирования и реализации реляционных баз данных, включая определение сущностей, атрибутов, связей и индексов. Взаимодействие с базой данных осуществлялось с помощью Entity Framework, что позволило не затрагивать работу с языком SQL.

В результате выполненной работы был разработан рабочий прототип приложения с интуитивным пользовательским интерфейсом, эффективной архитектурой и надежной базой данных. Опыт, полученный в процессе разработки, позволил углубить понимание принципов объектно-ориентированного программирования, работы с базами данных и построения сложных приложений на основе паттерна MVVM. Полученные навыки позволят эффективно использовать WPF, C#, Entity Framework и SQLite в будущих проектах.

# Список использованных источников

metanit.com: Сайт. URL: <https://metanit.com/>

learn.microsoft: Сайт. URL: [https://learn.microsoft.com/ru-ru/dotnet/](https://learn.microsoft.com/ru-ru/dotnet/csharp)

Режим доступа: для не зарегистрир. gользователей

kotlinlang.org: Сайт. URL: <https://kotlinlang.org/docs/>

Режим доступа: для не зарегистрир. пользователей