



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт
цифровых интеллектуальных систем

Кафедра
компьютерных систем управления

Санталов Михаил Дмитриевич

Выпускная квалификационная работа по направлению подготовки (специальности)
15.03.04 «Автоматизация технологических процессов и производств»,
профиль «Автоматизация технологических процессов и производств»

на тему:

«Разработка автоматизированной системы документооборота при регистрации
объектов интеллектуальной собственности»

Регистрационный номер № _____

Зав. кафедрой,
д.т.н., профессор

Мartiнов Георги Мартинoв

Руководитель,
Старший
преподаватель

(подпись)

Евстафиева Светлана
Владимировна

Обучающийся:
студент гр. АДБ-21-08

(подпись)

Санталов Михаил Дмитриевич

Москва 2025

ГРАФИК
выполнения выпускной квалификационной работы

Мероприятия	Сроки выполнения	Отметка руководителя о выполнении
	до «__» _____ 202__ г.	
	до «__» _____ 202__ г.	
	до «__» _____ 202__ г.	
	до «__» _____ 202__ г.	

График составлен « »202_____ г.

С графиком ознакомлен (а)

(ФИО обучающегося, подпись)

Руководитель Старший преподаватель, Евстафиева Светлана Владимировна

(ФИО, подпись)

СОДЕРЖАНИЕ

Введение.....	4
ГЛАВА 1. Анализ способов и инструментов оформления типовых документов для регистрации объекта интеллектуальной собственности.....	5
1.1. Основные методы подачи государственной заявки на регистрацию.....	5
1.1.1. Собственноручное заполнение документов в word:.....	6
1.1.2. Заполнение документов через сайт ФИПС:.....	6
1.1.3. Заполнение документов путем передачи сторонним организации:.....	7
1.2. Сравнение и выбор базы данных.....	9
1.3. Сравнение и выбор среды разработки.....	12
ГЛАВА 2. Рассмотрение типовых документов при регистрации ОИС. Формирование базы данных и прототипа приложения.....	16
2.1. Документы, заполняемые для подачи заявки.....	16
2.2. Разработка базы данных и прототипа приложения.....	20
2.2.1 Описание и схема базы данных.....	24
2.2.2 Описание диаграммы прецедентов.....	25
2.2.3 Описание прототипа приложения.....	29
ГЛАВА 3. Разработка программного приложения.....	31
3.1. Разработка интерфейса приложения.....	31
3.2. Создание базы данных приложения.....	34
3.3. Разработка внутренней логики приложения.....	35
Список литературы.....	36

Введение

Любое изделие, в том числе программный продукт имеет свои сопроводительные документы, которые заполняются не только для работы с приложением, но также и для официальной его регистрации. Это необходимо в первую очередь тогда, когда это изделие имеет за собой цель не только личного, но и общего (или хотя бы для ограниченного круга лиц) пользования. Регистрация позволяет защитить продукт от посягательств третьих лиц, дает приоритет в судебных разбирательствах, а также позволяет получать официально коммерческую выгоду из зарегистрированного продукта. Помимо деловой стороны вопроса, список зарегистрированных программных продуктов и изобретений может показывать состоятельность человека как разработчика и/или изобретателя программного обеспечения. Однако, регистрация программного обеспечения это достаточно деликатный процесс, требующей высокой точности, так как даже малейшая ошибка в процессе подачи заявки будет стоить не только временных, но и денежных затрат. Данная работа призвана создать приложение, способное упростить и ускорить работу с подачей заявки на регистрацию программного продукта, при этом сократив временные затраты на создание полного комплекта необходимых документов.

ГЛАВА 1. Анализ способов и инструментов оформления типовых документов для регистрации объекта интеллектуальной собственности

1.1. Основные методы подачи государственной заявки на регистрацию

Прежде чем разрабатывать приложение необходимо понять, какие альтернативные методы подачи заявки существуют, в чем их преимущества и недостатки, а также как эти недостатки возможно решить.

Среди основных методов подачи заявления на регистрацию можно выделить три наиболее приоритетных и известных для рядового пользователя. Такими методами являются:

- Собственноручное заполнение документов - самый простой и очевидный способ, в котором заявитель заполняет все бланки самостоятельно в word.
- Онлайн заполнение через официальный сайт ФИПС - способ призванный облегчить заполнение всех нужных форм через официальный сайт федерального института промышленной собственности
- Заполнение через сторонние организации - способ, когда заполнение всех необходимых документов отдается на сторону специальным юридическим компаниям, имеющих специалистов в деле подачи заявок на регистрацию ПО

Рассмотрим преимущества и недостатки каждого из этих вариантов подробнее

1.1.1. Собственноручное заполнение документов в word:

Основной и в большинстве своем всегда доступный способ из представленных. Представляет собой собственноручное заполнение всех необходимых документов в Word для дальнейшей отправки электронной почтой в ФИПС. Крайне простой в своей сути способ, главная сложность которого состоит во времени, затрачиваемом на заполнение каждого бланка. Огромным преимуществом данного метода является независимость от сторонних факторов. Однако при этом этот метод имеет огромный простор где возможен риск человеческой ошибки, вследствие чего будет затрачиваться еще больше времени и сил на перепроверку, либо переписывание бланков.

1.1.2. Заполнение документов через сайт ФИПС:

Сайт федерального института промышленных систем представляет электронную альтернативу заполнения документов. Вписывая данные заявителя в специальные поля, сайт автоматически формирует необходимые документы по заполненным онлайн формам. Преимущество данного метода заключается в автоматизированности процесса, потому что полный комплект необходимых форм выдается в ходе заполнения небольшого числа данных на сайте ФИПС, а не множественного заполнения всех нужных документов в Word. Также этот метод уменьшает шанс человеческой ошибки из-за множественного заполнения одной и той же информации в разные бланки, однако не убирает вероятность ошибки при первичном заполнении. Недостатком же данного метода является, несомненно, строгая необходимость наличия сертификата ключа электронной цифровой подписи (ЭЦП), так как сайт не позволит подать заявление без его наличия. В свою очередь получение тоже является процессом требующим приложения “Госключ”

1.1.3. Заполнение документов путем передачи сторонним организации:

Этот способ предполагает передачу своих личных данных различным юридическим организациям, которые в свою очередь, за определенную плату, возьмут заполнение всех необходимых документов на себя. Главным достоинством этого метода является экономия времени. Передав ответственность за создание документов специальной организации, заявитель не тратит время на подачу документов и может заняться другими делами. Вторым достоинством, несомненно, можно отметить консультацию со специалистом в область регистрационного права, который подскажет какие документы необходимы, а также ответит на все возникающие вопросы. Но с такими крупными достоинствами, метод обладает столь крупными недостатками. Самым главным здесь будет необходимость финансовых трат. Любая такая организация будет требовать денег за свою работу (от 50 тыс. рублей), при этом эти деньги будут заплачены до получения какого-либо результата.

Рассмотрев эти методы необходимо сравнить основные инструменты для создания будущего приложения. В частности, рассмотреть варианты базы данных и среду, в которой будет разрабатываться приложение

Таблица 1.1. - краткое сравнение методов подачи заявления

	Заполнение в word	Сайт ФИПС	Сторонние организации
Временные затраты	Крайне высокие. До 10 мин на каждый из бланков	Средние, затем низкие. 20-30 минут при первичном заполнении. После 15-20 минут на весь пакет	Средние. Консультация длится в основном до 40 минут с учетом наличия всех необходимых документов заранее
Необходимость наличия интернета	Крайне малая, либо отсутствует	Интернет необходим на всех этапах	Интернет может быть нужен для общения с исполнителем
Вероятность человеческой ошибки	Высокая. Каждый из документов заполняется собственноручно	Средняя. Ошибки происходят зачастую при первичном внесении данных	Низкая со стороны заявителя. Ответственность за ошибки лежит на исполнителе
Автоматизированность процесса	Отсутствует	Высокая. Все документы ФИПС формируются автоматически	Средняя, если учитывать невмешательство заявителя
Денежные затраты	Отсутствуют	Отсутствуют	Средне-высокие
Сохранность персональных данных	Высокая. Личные данные находятся внутри узкого круга пользователей	Средняя. Государственные сайты являются основной целью для злоумышленников	Средняя. Личные данные передаются организации, однако они защищены законом в случае неправомерного использования

1.2. Сравнение и выбор базы данных

Сформулировав основные требования, далее необходимо определиться с тем, какие инструменты необходимы для реализации этого приложения.

Первым делом, рассмотрим варианты БД, которые можно использовать, так как именно в базе данных планируется хранение информации о заявителях.

Реляционные базы данных были выбраны в этом приложении в качестве основных. Их ключевыми преимуществами являются поддержка строгой структуризации данных и обеспечение целостности за счет ограничений целостности (например, первичных и внешних ключей) Использование языка SQL позволяет выполнять сложные аналитические запросы и оптимизировать работу с данными. Благодаря своей стабильности, масштабируемости и широкому применению в различных областях такие базы данных остаются основным выбором для систем, требующих надежности и высокой согласованности данных, которой приложение и является. Наиболее популярными и распространенными реляционными базами данных являются:

MySQL – это одна из наиболее популярных систем управления базами данных. Ее спецификация создавалась для повышения производительности и простоты использования. Она ориентирована на работу в рамках веб-приложений, поэтому является одной из самых популярных систем управления данными в интернете. Основное преимущество системы – высокая скорость выполнения операций, особенно чтения данных, поэтому представляется идеальным вариантом для использования в системах, требующих скорости и масштабируемости. Однако, из-за ограниченных возможностей в транзакциях и в поддержки сложных типов данных ее сложно использовать в аналитике и корпоративных приложениях. Эти недостатки преимущественно не будут мешать работе приложения, в то время

как достоинства (в частности масштабируемости) позволят значительно улучшить опыт работы с приложением.

PostgreSQL – мощная и универсальная система с открытым исходным кодом, строго соответствующая стандартам SQL, которая отличается расширенной поддержкой сложных запросов. Система разработана для работы с большими массивами данных и имеет широкие возможности: транзакции с полным поддержанием ACID, сложные типы данных, расширяемость и нативная поддержка JSON-данных. PostgreSQL отдается предпочтение в приложениях, где в первую очередь ценится гибкость и надежность. Для настройки и администрирования такой базы данных может потребоваться значительно большее количество времени и опыта, нежели в случае с MySQL. Достоинства этой базы данных значительны, однако они не будут в полной мере использоваться приложением, а значительно усложненный процесс разработки будет проблематичен в случае расширения приложения

SQLite – это небольшая встроенная база данных с минималистичной внутренней архитектурой, которая не имеет необходимости в установке сервера и хранит все данные в одном файле. Это делает базу данных компактной и идеальной для размещения в мобильных устройствах и встроенных системах. SQLite превосходно подходит для небольших локальных хранилищ данных, так как её функционал ограничен возможностями для множественного пользовательского доступа и больших объемов данных. Компактность является краеугольным камнем всей базы данных SQLite. Однако так как приложение не будет разрабатываться для мобильных устройств, компактность не сыграет большой роли, в то время как сильная ограниченность в запросах - сыграет.

Microsoft SQL Server. Эта мощная коммерческая база данных была разработана корпорацией Microsoft. SQL Server хорошо себя показывает для

корпоративного уровня, так как предоставляет большие ресурсы управления данными, а также аналитики и отчетов. Кроме того, он глубоко интегрирован с продуктами Microsoft, особенно с Excel и Power BI, и особенно популярен именно в сфере бизнес-анализа. Кроме того, SQL Server очень надежен и устойчив к нагрузкам, имеются встроенные инструменты безопасности для работы. В то же время, не смотря на эти плюсы, эта БД дорогостоящая и редко используемая в малых организациях.

Имея краткое описание всех основных систем управления базами данных, можно составить сравнительную характеристику и определиться с тем, какая база данных будет использована в приложении:

Таблица 1.2. - сравнение основных реляционных баз данных

База данных	MySQL	PostgreSQL	SQLite	Microsoft SQL server
Лицензия	Коммерческая	Открытый код	Публичный домен	Проприетарная
Простота настройки	Высокая	Средняя	Крайне высокая	Средняя
Масштабируемость	Высокая	Высокая	Низкая	Крайне высокая
Производительность	Высокая на чтении	Высокая на запросы	Высокая на локальном уровне	Высокая
Сложность администрирования	Низкая	Высокая	Низкая	Высокая

На основе всего вышеописанного хорошим вариантом для разрабатываемого приложения станет MySQL за простоту настройки, большую масштабируемость и быстроедействие с запросами.

1.3. Сравнение и выбор среды разработки

Выбрав базу данных можно сосредоточиться на выборе среды разработки приложения. Здесь важно выбрать такой метод, который будет позволять наглядную работу с бланками как в процессе работы, так и в процессе разработки. Также язык должен иметь относительно простую масштабируемость и быстродействие в работе с бланками и базами данных. Среди таких методов можно выделить:

WinForms — это более старая технология Microsoft для создания настольных приложений, которая была популярна до появления WPF. WinForms представляет собой простой способ разработки пользовательских интерфейсов с использованием элементов управления, похожих на Windows API. WinForms известен своей простотой и быстрой разработкой прототипов, что делает его хорошим выбором для небольших приложений. Однако он менее гибок по сравнению с WPF, особенно в создании сложных интерфейсов и графики.

WPF — это современная по сравнению с WinForms технология разработки приложений для Windows, представленная Microsoft. Она основывается на разметке XAML и использует рендеринг графики через DirectX. WPF предоставляет большую свободу в создании сложных пользовательских интерфейсов, включая 2D и 3D графику, анимации и адаптивные макеты. Эта технология особенно ценится за возможность четкого разделения логики и интерфейса (через паттерн MVVM). Главной проблемой WPF является ограниченность платформой Windows, также как и ее низкая производительность на слабых устройствах или при неправильной оптимизации. В дополнение к этому, для разработки требуется изучение XAML.

UWP — это платформа разработки приложений экосистемы Windows, включая ПК, планшеты, и прочее. UWP также использует XAML для

описания интерфейсов, как и WPF, но при этом предлагает возможности кроссплатформенной разработки внутри Windows-экосистемы. UWP обеспечивает доступ к современным API Windows 10 и более легкую интеграцию с облачными службами Microsoft. Однако эта платформа ограничена Windows 10 и выше, что затрудняет ее использование в средах, где ещё используются более старые версии Windows, в том числе институтах или отдаленных поселениях. Еще одним недостатком является относительная сложность разработки для широкого спектра устройств.

Blazor — это веб-фреймворк от Microsoft для создания интерактивных веб-приложений с использованием C# и .NET. Вместо традиционного JavaScript, Blazor предлагает два подхода: серверный (Blazor Server) и клиентский (Blazor WebAssembly). Blazor позволяет разработчикам, знакомым с .NET, создавать кроссплатформенные приложения, которые работают в браузере. Этот подход особенно удобен для организаций, предпочитающих единую разработку на C#. Однако Blazor ограничен производительностью веб-технологий, а пользовательские интерфейсы часто менее отзывчивы по сравнению с родными настольными решениями.

Таблица 1.3. - сравнение инструментов разработки приложения

Способ разработки	WinForms	WPF	UWP	Blazor
Язык программирования	C#	C# + XAML	C# + XAML	C# + Razor
Платформа	Только Windows	Только Windows	Только Windows 10 и выше	Кроссплатформенная
Графические возможности	Низкие	Высокие	Высокие	Средние
Разделение логики и UI	Низкое	MVVM	MVVM	Возможность использовать MVVM

Производительность	Средняя	Высокая	Высокая	Сильно варьируется
--------------------	---------	---------	---------	--------------------

Исходя из этого, WPF будет являться наилучшим выбором из-за высокой производительности, большого спектра поддерживаемых версий операционных систем (пускай и ограниченных Windows) а также простоты разработки приложения.

Рассмотрев эти аспекты разработки, можно составить основные требования к приложению:

1. Приложение в обязательном порядке должно работать без доступа в интернет и не требовать сертификата ЭЦП. Это позволит избежать проблем возникающих при заполнении через сайт ФИПС.
2. Приложение должно позволять пользователю скачивать бланки которое оно заполняет. Такой критерий позволит сделать приложение более гибким в плане заполнения документов.
3. Приложение должно иметь минимальное требуемое количество полей необходимых для заполнения. Все дополнительные поля, находящиеся в специфичных документах должны заполняться в случае работы с этим документом. Благодаря этому шанс человеческой ошибки при копировании данных можно локализовать в одном месте, быстро доступным для исправления.

ГЛАВА 2. Рассмотрение типовых документов при регистрации ОИС. Формирование базы данных и прототипа приложения.

2.1. Документы, заполняемые для подачи заявки

Создав необходимый базис для разработки приложения теперь необходимо рассмотреть требуемые для работы документы, какие данных в них заполняются, а также какие из этих данных лучше хранить в базе данных на постоянной основе, а какие лучше получать и использовать только при работе с определенным бланком.

Документы которые используются для подачи заявления на регистрацию программного продукта:

- 1) Согласие автора на указание сведений об авторе, указанных в заявлении
- 2) Заявление о государственной регистрации
- 3) Дополнение к заявлению о государственной регистрации и дополнение к графе 7А.
- 4) Запрос на имя ректора
- 5) Уведомление на имя ректора

Рассмотрим поля, которые заполняются для каждого из перечисленных документов:

- Регистрационный номер заявки (при наличии)
- Выбор что регистрируется (Программа для ЭВМ, Базы данных в соответствии с пунктом 4 статьи 1259 Кодекса или в соответствии с пунктом 3 статьи 1334 Кодекса)
- Название представленной на регистрацию программы для ЭВМ или базы данных
- Правообладатель (и) (Заявитель)(и) содержащий:
 - Фамилию(и)
 - Имя(Имена)

- Отчество(а) (при наличии)
- Место жительства физического лица
- Основной государственный регистрационный номер (ОГРН)
- Идентификационный номер налогоплательщика (ИНН)
юридического лица
- Всего правообладателей
- Дата рождения (день, месяц, год)
- Гражданство
- Место жительства включая название страны
- **Краткое описание вклада автора**
- Каким образом автор просит упоминать себя при публикациях
(Упоминать \ Упоминать под псевдонимом \ Не упоминать)
 - В случае упоминания под псевдонимом необходимо его указать
- Подпись автора с расшифровкой (в случае с разрабатываемой программой пункт не будет заполняться автоматически)
- Подпись(и) правообладателя(ей) или его (их) представителя(ей)
 - от имени юридического лица заявление подписывается руководителем организации или иным лицом, уполномоченным на это, с указанием его должности, подпись удостоверяется печатью юридического лица. Подпись любого лица должна быть расшифрована с указанием фамилии и инициалов и даты подписания заявления
- Входящий номер ВК включая дату поступления в формате: день, месяц (числом) и год в формате
- Регистрационный номер программы для ЭВМ или базы данных в реестре включая дату регистрации в формате: день, месяц (числом) и год
- Адрес для переписки
 - Почтовый индекс

- Адрес на территории российской федерации
- Имя или наименование адресата
- Телефон в любом из возможных форматов (+7... или 8...)
- Факс (зачастую не используется)
- Адрес электронной почты
- **Краткое описание творческого вклада автора при создании регистрируемой программы ЭВМ или базы данных**
- Сведения о программе для ЭВМ или базе данных, с привлечением средств бюджетов бюджетной системы Российской Федерации (не заполняется если средства бюджетов Российской Федерации не привлекаются)
 - Регистрируемый объект создан за счет средств (Федерального бюджета \ бюджета субъекта Российской Федерации \ местного бюджета \ государственного внебюджетного фонда Российской Федерации \ территориального государственного внебюджетного фонда)
 - Заказчик работ
- Кем является указанное лицо (государственным заказчиком \ муниципальным заказчиком \ федеральным органом государственной власти, предоставившим субсидии на осуществление капитальных вложений в создание и развитие государственных информационных систем (ГИС))
- Исполнитель работ
 - Указанное лицо является исполнителем по (государственному контракту \ муниципальному контракту \ договору (соглашению) о предоставлении гранта (субсидии) \ государственному заданию \ муниципальному заданию \ иное) Для всех пунктов есть возможность добавить “в том числе на создание ГИС”
 - Дата в формате день, месяц буквами и год, а также номер указанного выше документа

- Основание создания регистрируемой программы для ЭВМ или базы данных
 - Государственный контракт \ муниципальному контракту \ договор (контракт, соглашение) \ договор (соглашение) о предоставлении гранта (субсидии) \ государственное задание \ муниципальное задание \ иное

Для всех пунктов есть возможность добавить “в том числе на создание ГИС”

 - Дата в формате день, месяц буквами и год, а также номер указанного выше документа
- Основания возникновения права не регистрируемую программу ЭВМ или базу данных
 - заявитель является работодателем автора \ передача прав автором или его правопреемником заявителю \ передача прав работодателем заявителю \ в порядке универсального правопреемства (наследования, реорганизации) \ заявитель является изготовителем базы данных \ заявитель является автором
 - Права на регистрируемую программу для ЭВМ или базу данных принадлежат Российской Федерации в соответствии с:
 - с пунктом 12.1 статьи 78.3 Бюджетного кодекса Российской Федерации \ со ст. 5 Федерального закона от 04.08 2023 № 416-ФЗ "О внесении изменений в Бюджетный кодекс Российской Федерации и отдельные законодательные акты Российской Федерации и о признании утратившими силу отдельных положений законодательных актов Российской Федерации" \ с государственным контрактом \ с иным
- Регистрируемый объект
 - Является компонентом государственной информационной \ Не содержит персональные данные \ Содержит персональные

данные Регистрационный номер (номер) в Реестре операторов, осуществляющих обработку персональных данных

- Год создания регистрируемой программы для ЭВМ или базы данных
- Страна и год обнародования (первого опубликования) регистрируемой программы для ЭВМ или базы данных
- Год обновления регистрируемой базы данных, государственная регистрация которой осуществлялась в соответствии с пунктом 3 статьи 1334 Кодекса
- Как зарегистрирована база данных в реестре баз данных
 - Номер и дата регистрации \ нет
- Сведения о содержащихся в заявке документах
- Для одного из авторов, руководителя работы указать ученое звание, ученую степень, WOS Research ID, Scopus Author ID, дату рождения
- Состав потенциальных правообладателей
- Предложения по порядку его использования
- Ключевые слова
- Возможные направления (способы) использования
- Количество опытных образцов, при создании которых использован РИД
- Количество часов, необходимых для создания ОИС (не более 150 часов)
- Планируемая форма правовой охраны

2.2. Разработка базы данных и прототипа приложения

Разобрав основные документы, далее необходимо выделить среди них самые часто встречающиеся пункты, которые будут использоваться в базе данных. Итоговая таблица атрибутов базы данных приведена в таблице 2.1.:

Таблица 2.1. - атрибуты базы данных

Название атрибута	Расшифровка атрибута	Тип данных для атрибута	Комментарий
id	Уникальный номер заявителя	INT	Скрытый атрибут. Нужен в качестве PRIMARY KEY
user_type	Вид правообладателя. Физическое лицо или Юридическое	BINARY	Обязательное поле. Значение 1 будет означать физическое лицо. Значение 0 будет означать юридическое лицо
lastname	Фамилия	TINYTEXT	Обязательное поле. Автоматически будет проверять и заменять по необходимости первую букву на заглавную
name	Имя	TINYTEXT	Обязательное поле. Автоматически будет проверять и заменять по необходимости первую букву на заглавную
middlename	Отчество	VARCHAR	Необязательное поле. Автоматически будет проверять и заменять по необходимости первую букву на заглавную. Значение по умолчанию - NULL
job_position	Должность	VARCHAR	ТОЛЬКО ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ. Обязательное поле.
job_name	Название юридического лица	VARCHAR	ТОЛЬКО ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ. Обязательное поле

pseudo	Псевдоним	VARCHAR	ТОЛЬКО ДЛЯ ФИЗИЧЕСКИХ ЛИЦ. Необязательное поле. Значение по умолчанию - NULL
phonenumb	Номер телефона	INT(10)	Обязательное поле
email	Электронная почта	TEXT	Обязательное поле. Будет проверять наличие одной @ в строке
birthdate	Дата рождения	DATE	Обязательное поле
citizenship	Гражданство	TINYTEXT	Обязательное поле. Свое значение либо РФ
address	Адрес регистрации	TEXT	Обязательное поле. В случае юридического лица заполняется адрес предприятия.
index	Почтовый индекс адреса	INT	Обязательное поле. В случае юридического лица заполняется почтовый индекс предприятия.
passport_type	Вид документа удостоверяющего личность	BINARY	Обязательное поле. Значение 1 будет означать паспорт, значение 0 будет означать вид на жительство
passport_series	Серия паспорта	SMALLINT	Обязательное поле. Пишется без пробела
passport_number	Номер паспорта	MEDIUMINT	Обязательное поле
passport_origin	Кем выдан паспорт	TINYTEXT	Обязательное поле
passport_date	Когда выдан паспорт	DATE	Обязательное поле в формате день \ месяц \

			год
passport_codde	Код подразделения	INT	Обязательное поле. Будет преобразовываться в формат “число-число”
ogrn	ОГРН	INT	Необязательное поле. Значение по умолчанию - NULL
inn	ИНН	INT	Необязательное поле. Значение по умолчанию - NULL
snils	СНИЛС	MEDIUMINT	Необязательное поле. Будет автоматически добавлять “-” и “ “ Значение по умолчанию - NULL
sc_status	Ученое звание	VARCHAR	ТОЛЬКО ДЛЯ ФИЗИЧЕСКИХ ЛИЦ. Необязательное поле. Значение по умолчанию - “Отсутствует”
sc_degree	Ученая степень	VARCHAR	ТОЛЬКО ДЛЯ ФИЗИЧЕСКИХ ЛИЦ. Необязательное поле. Значение по умолчанию - “Отсутствует”
wos	Id в системе web of science (ранее - web of knowledge)	SMALLINT	ТОЛЬКО ДЛЯ ФИЗИЧЕСКИХ ЛИЦ. Необязательно поле. Значение по умолчанию - NULL
scopus	Id в базе данных scopus	SMALLINT	ТОЛЬКО ДЛЯ ФИЗИЧЕСКИХ ЛИЦ. Необязательно поле. Значение по

			умолчанию - NULL
--	--	--	------------------

2.2.1 Описание и схема базы данных

На основе этих атрибутов наилучшим выбором будет разделить базу данных на 5 таблиц. Первая, так называемая “сводная таблица” будет иметь с себе только ключ участника, а также его основные параметры. Удобное разделение получится в случае если поделить всех заявителей по четырех категориям на основе двух критериев: гражданство и вид лица (физическое или юридическое) Получаем таблицу физических лиц граждан РФ, юридических лиц РФ, физических иностранных граждан и юридические иностранные лица. Схема база данных представлена на рисунке 2.1.

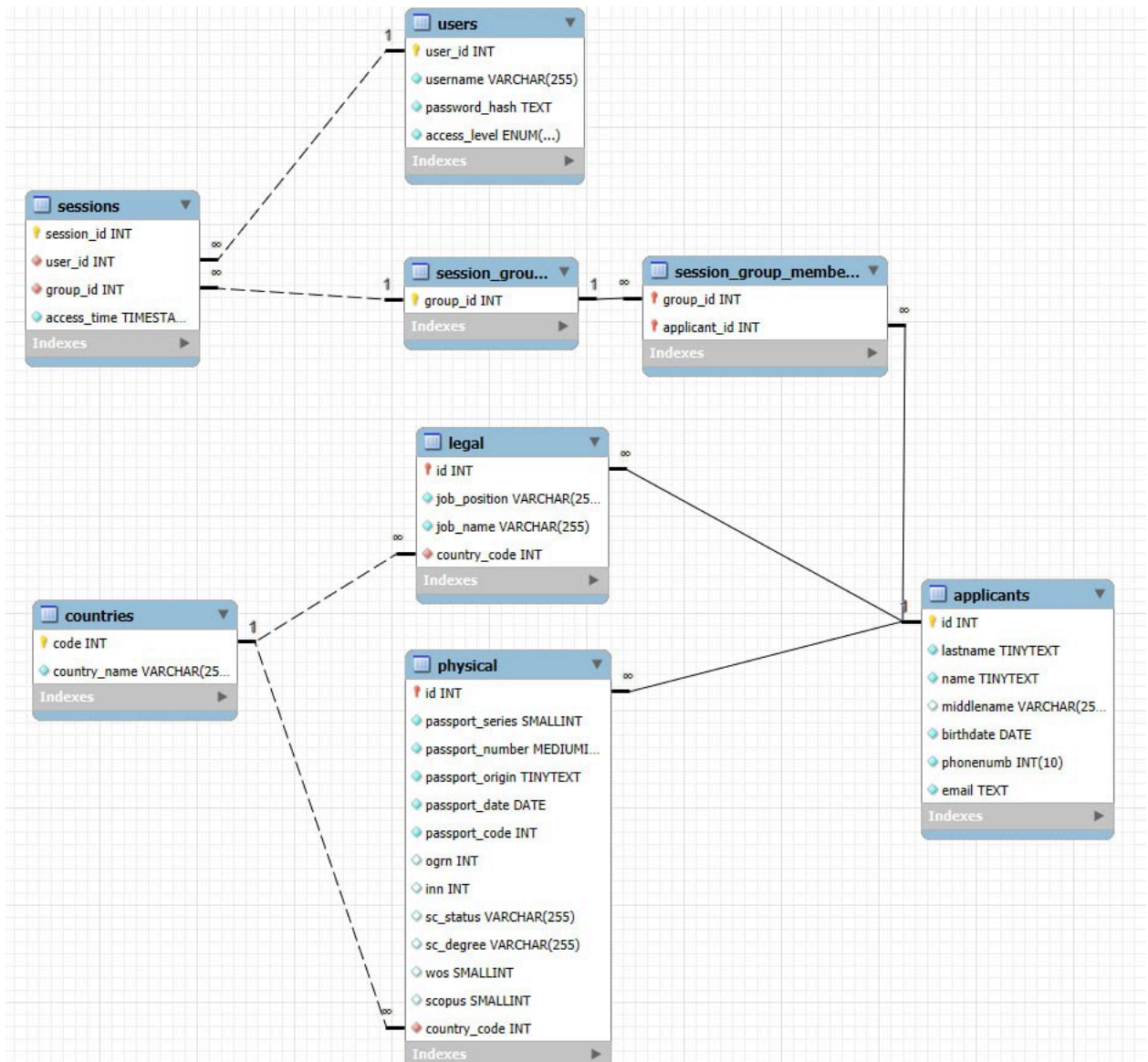


Рис. 2.1. – Схема базы данных

2.2.2 Описание диаграммы прецедентов

Далее, необходимо выделить стандартную схему, по которой будет работать приложение. При открытии программы пользователя будет встречать экран добавления и редактирования пользователей. Заполнив все поля находящиеся там и нажав сохранить пользователь должен будет повторить эту процедуру для каждого из заявителя. После того, как пользователь добавил всех заявителей которых еще нет в базе данных, он переходит на вкладку “Данные о заявлении”. Там, необходимо будет заполнить информацию напрямую относящуюся к заявлению как таковому.

Нажав кнопку “сохранить”, пользователь может переходить к любому из необходимых ему бланков представленных в колонке слева. Каждая вкладка будет иметь в себе дополнительные поля, характерные только для этого документа. Закончив заполнять данные в полях, пользователь должен нажать кнопку “сохранить для печати” чтобы программа автоматически сформировала файл в формате .docx. Диаграмма прецедентов а также схема работы приложения показана на рисунках 2.2 - 2.5.

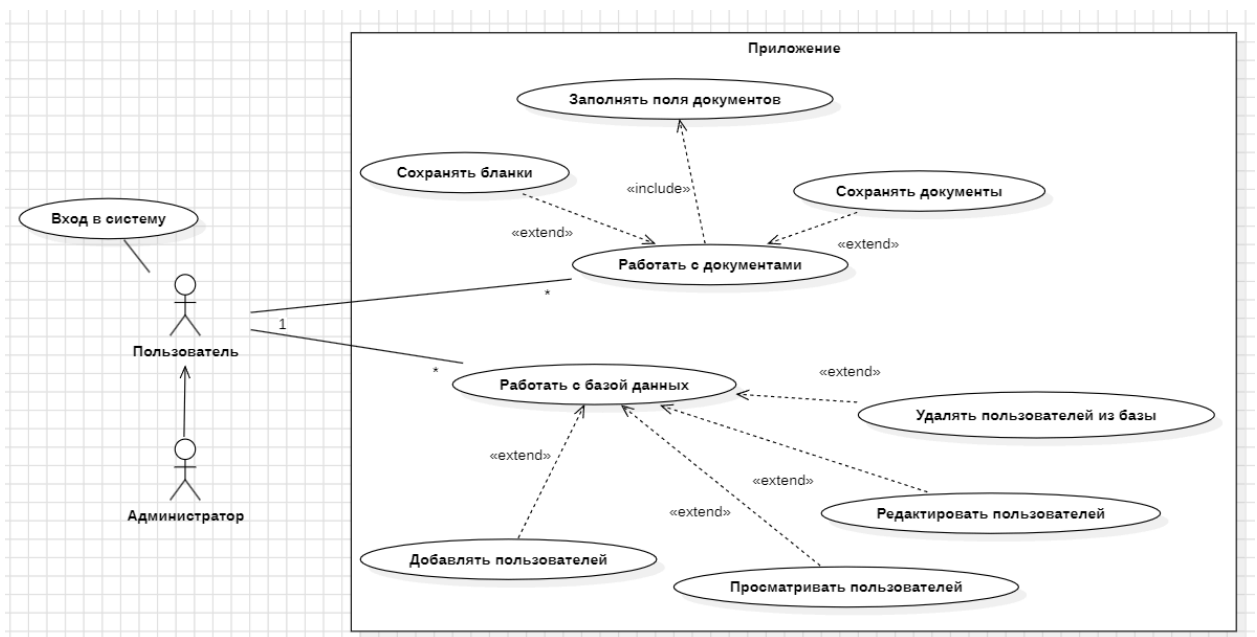


Рис. 2.2. – Диаграмма прецедентов

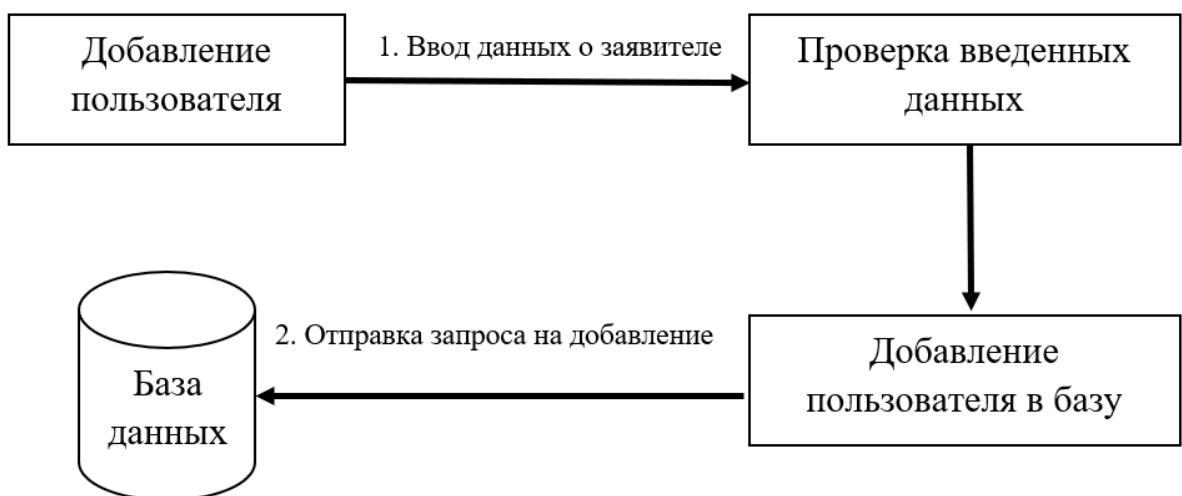


Рис. 2.3. – Схема добавления пользователя в базу данных

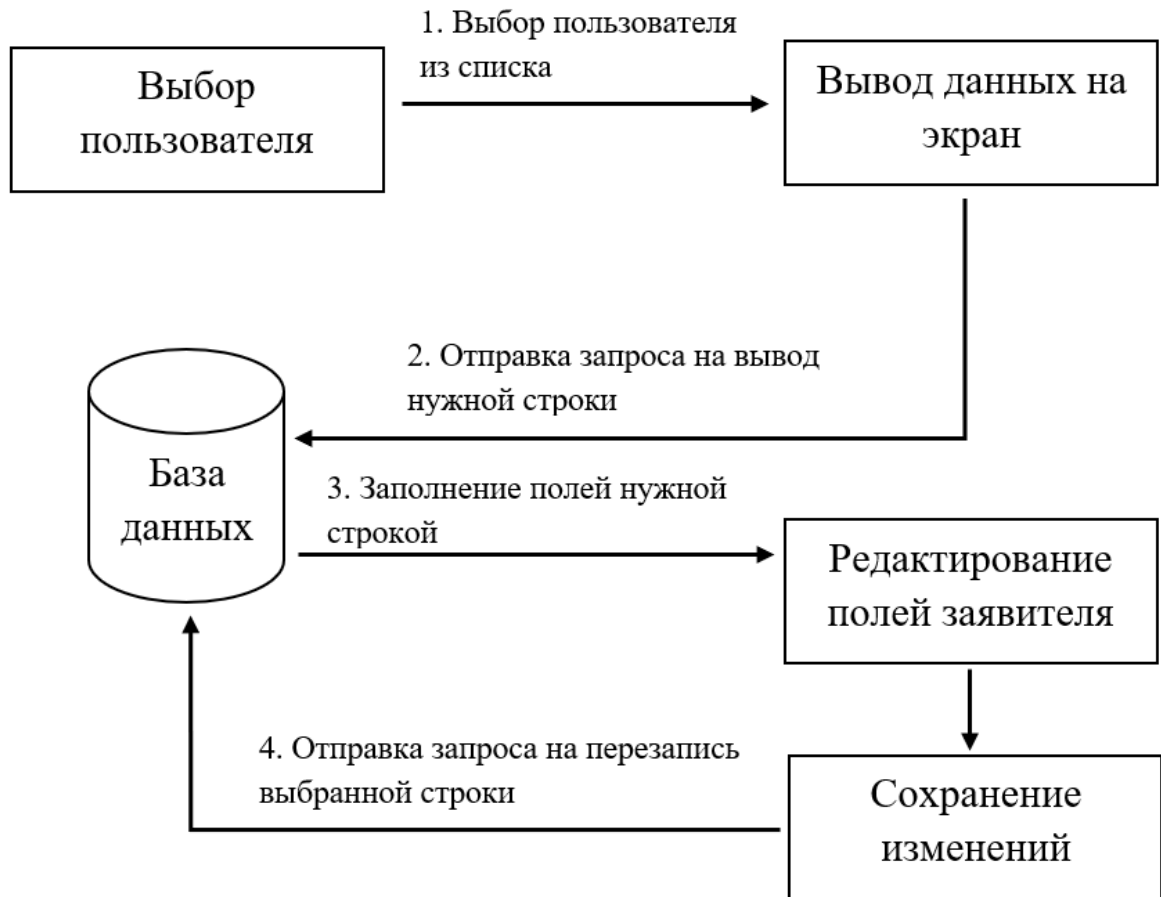


Рис. 2.4. – Схема редактирования пользователя в базе данных



Рис. 2.5. – Схема работы приложения

Рисунок 2.8. показывает каким образом происходит добавление пользователя в базу данных. После введения данных в поля и нажатие на кнопку происходит запрос в базу данных на создание новой строки с данными из заполненных полей в соответствующие поля базы данных.

Рисунок 2.9. описывает принцип редактирования уже существующего заявителя. Сначала выбирается строка из базы данных, которые заполняет данными поля в приложении. Затем, после окончания редактирования, отправляется запрос на изменение уже существующих полей в БД на новые.

Рисунок 2.10. представляет собой общий принцип работы приложения. После входа в систему, пользователь опционально добавляет или изменяет пользователей. Затем, он заполняет все необходимые ему поля в прочих документах. Закончив с заполнением данных пользователь возвращается на главный экран и нажимает кнопку “сформировать документы” после чего

используя заполненные поля а также базу данных приложение записывает их в соответствующие поля специальных бланков и сохраняет их.

2.2.3 Описание прототипа приложения

Получив всю необходимую базу для создания приложения можно начать его создания с интерактивного прототипа, который будет использоваться для схемы работы приложения. Основной экран прототипа показан на рисунке 2.6.:

Основная информация о заявителе			
Основная информация о заявителе	Загрузить из БД		Скачать бланк
Данные по заявлению	Фамилия <input type="text" value="Иванов"/>	Адрес регистрации <input type="text" value="Москва, ул. Пушкина, д. 17, к"/>	ОГРН <input type="text" value="1147847423899"/>
Согласие автора на указание сведений	Имя <input type="text" value="Иван"/>	Индекс регистрации <input type="text" value="126617"/>	ИНН <input type="text" value="12345678912"/>
Заявление на государственную регистрацию	Отчество (при наличии) <input type="text" value="Иванович"/>	Вид удостоверяющего документа <input type="text" value="Паспорт"/>	Ученое звание заявителя <input type="text"/>
Дополнение к заявлению	Вид правообладателя <input type="text" value="Физическое лицо"/>	Серия документа <input type="text" value="1212"/>	Ученая степень заявителя <input type="text"/>
Запрос на имя ректора	Псевдоним (при наличии) <input type="text" value="Котов Сегрей Михайлович"/>	Номер документа <input type="text" value="123456"/>	Web of Science ID <input type="text"/>
Уведомление на имя ректора	Номер телефона <input type="text" value="+78005553535"/>	Кем выдан <input type="text" value="Отделом УФМС России"/>	Scopus ID <input type="text"/>
	Электронная почта <input type="text" value="ivanovii83@mail.ru"/>	Когда выдан <input type="text" value="25.10.2023"/>	
	Дата рождения <input type="text" value="22.10.1983"/>	Код подразделения <input type="text" value="654-245"/>	
	Гражданство <input type="text" value="РФ"/>		
	Сохранить		

Рис. 2.6. – Прототип экрана приложения

Интерфейс приложения будет поделен на 2 основных зоны. В левой части будет находится список из вкладок, каждая из которых будет выводить в рабочую зону справа поля, необходимые для заполнения соответствующего документа. Также помимо вкладок документов, список будет содержать пункты “Основная информация о заявителях” и “Данные по заявлению”

Во вкладке “Основная информация о заявителях” будет содержать все самые часто встречающиеся поля, необходимые для подачи заявки. Тут пользователь сможет заполнять, сохранять и редактировать данные о каждом из заявителей хранящихся в базе данных. Позже эти данные будут использоваться автоматически исходя из того, какие заявители будут выбраны на вкладке “Данные по заявлению”

Вкладка “Данные по заявлению” будет содержать поля, которые нужны только для конкретного заявления, в том числе на этой вкладке будут добавляться основной и прочие правообладатели, сохраненные в базе данных при помощи вкладки “Данные о заявителях”

Все последующие вкладки из списка будут содержать соответствующие названию документы а также возможность сохранять готовый файл или бланк в формате .doc(.docx).

ГЛАВА 3. Разработка программного приложения

3.1. Разработка интерфейса приложения

Основной экран приложения представляет собой два разделенных между собой блок. Блок заголовка в котором отображается текущее положения пользователя а также основной блок, с которым будет работать пользователь. Заголовочный блок представляет собой StackPanel содержащий в себе атрибут Text связанный с текущей выбранной панелью второго блока, в то время как основная рабочая область TabControl, где каждый из необходимых документов представлен свои TabItem. Экран “Основная информация о заявителе” содержит в себе текстовые поля которые созданы на основе StackPanel. Код структуры приложения и пример экрана представлен представлены на 3.1 и 3.2.

Основная информация о заявителе

Основная информация о заявителе Данные по заявлению Согласие автора на сведения Заявление на гос. регистрацию Дополнение к заявлению Запрос на имя ректора Уведомление на имя ректора	<input type="button" value="Загрузить из БД"/> <input type="button" value="Сформировать бланки"/>		
	Имя	Индекс адресе регистрации	ИНН
	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Отчество	Вид удостоверяющего документа	Ученое звание заявителя
	<input type="text"/>	<input type="text" value="v"/>	<input type="text"/>
	Вид правообладателя	Серия документа	Ученая степень заявителя
	<input type="text" value="v"/>	<input type="text"/>	<input type="text"/>
	Псевдоним (при наличии)	Номер документа	Web of Science ID
	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Номер телефона	Кем выдан	Scopus ID
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Электронная почта	Когда выдан	Должность (только для юр. лиц)	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Дата рождения	Код подразделения		
Выбор даты <input type="text" value="15"/>	<input type="text"/>		
Гражданство		<input type="button" value="Сохранить в БД"/>	
<input type="text"/>			

Рис. 3.1. – Основной экран приложения

```

mc:Ignorable="d"
Title="MainWindow" Height="720" Width="1024">
<Grid x:Name="BaseGrid">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="0.15*" />
    <ColumnDefinition Width="0.375*" />
    <ColumnDefinition Width="0.375*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="0.125*" />
    <RowDefinition Height="0.875*" />
  </Grid.RowDefinitions>
  <StackPanel x:Name="Header" Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="3" Background="■" #D9D9D9">
    <TextBlock HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontWeight="Bold" Margin="0,20,0,0" Text="Основная информация о заявителе" />
  </StackPanel>
  <TabControl Grid.Column="0" Grid.Row="1" Grid.ColumnSpan="3" Margin="10" TabStripPlacement="Left">
    <TabItem Header="Основная&#10;информация&#10;о заявителе">
      <Grid>
        <Grid.RowDefinitions>
          <RowDefinition Height="*" />
          <RowDefinition Height="*" />
          <RowDefinition Height="*" />
          <RowDefinition Height="*" />
          <RowDefinition Height="*" />
          <RowDefinition Height="*" />
          <RowDefinition Height="*" />
          <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
          <ColumnDefinition Width="*" />
          <ColumnDefinition Width="*" />
          <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Button Grid.Row="0" Grid.Column="0" Content="Загрузить из БД" Width="auto" Height="35" HorizontalAlignment="Left" Margin="23 0 0 0"/>
        <Button Grid.Row="0" Grid.Column="2" Content="Сформировать бланки" Width="auto" Height="35" HorizontalAlignment="Right" Margin="0 0 23 0"/>
      </Grid>
    </TabItem>
  </TabControl>
</Grid>

```

Рис. 3.2. – Основной экран приложения

Вкладка “Данные по заявлению” содержит в себе поля которые часто повторяются в необходимых документах. Для того чтобы не ограничиваться областью окна приложения, был использован элемент `ScrollView`, добавляющий полосу прокрутки. Реализация а также пример вкладки представлены на 3.3 и 3.4.

Рис. 3.3. – Вкладка “Данные по заявлению” часть 1

[illegible]

Рис. 3.4. – Вкладка “Данные по заявлению” часть 2

Последующие вкладки содержат специфические для каждого документа данные которые в других документах не встречаются. Это необходимо для того чтобы заполнять наименьшее число необходимых полей. Пример необходимых полей представлен на рисунке 3.4.

The screenshot shows a software application window with a sidebar on the left containing several tabs: "Основная информация о заявителе", "Данные по заявлению", "Согласие автора на сведения", "Заявление на гос. регистрацию", "Дополнение к заявлению", "Запрос на имя ректора", and "Уведомление на имя ректора". The main area of the window displays the "Дополнение к заявлению" (Supplement to the application) tab. It contains two checkboxes at the top: "в соответствии с пунктом 4 статьи 1259 Кодекса" and "в соответствии с пунктом 3 статьи 1259 Кодекса". Below these are input fields for "Регистрационный номер" and "Дата регистрации" (with a date picker set to 15). A section titled "Содержащиеся в заявке документы:" (Documents included in the application:) follows, listing various document types with checkboxes: "идентифицирующие программу для ЭВМ материалы в форме распечатки исходных данных", "идентифицирующие программу для ЭВМ материалы в иной форме", "материалы аудиовизуальных отображений, порождаемых программой для ЭВМ", "материалы, идентифицирующие базу данных", "документы, подтверждающие существенные затраты на создание базы данных", "реферат", "иные материалы", "дополнение к заявлению", "доверенность(и)" (which is selected with a blue square), "документ(ы) об уплате государственной пошлины (представляется по инициативе заявителя)", "документ, подтверждающий наличие оснований для освобождения от уплаты пошлины либо для уменьшения ее размера, либо для отсрочки ее уплаты", "ходатайство(а)", and "иные документы". A button labeled "Сформировать бланк" (Generate form) is located at the bottom right of the main area.

Рис. 3.4. – Пример прочих вкладок приложения

При этом стоит отметить, что при необходимости печатать только определенный каждая вкладка содержит кнопку “сформировать документ”, в то время как основная имеет кнопку “сформировать документы”, которая подготавливает весь перечень документов. Пример этих кнопок можно видеть на рисунках 3.1. и 3.4.

3.2. Создание базы данных приложения

С учетом структуры базы данных она была реализована в программе MySQL Workbench. Готовая база данных представлена на **РИСУНКЕ**

Общая таблица заявителей (applicants): В этой таблице хранятся базовые данные для всех заявителей: Фамилия, Имя, Отчество, Дата рождения, Номер телефона и Электронная почта и так далее. Это основной источник информации о каждом заявителе.

Дополнительные таблицы для физических и юридических лиц:

- physical: Здесь хранятся дополнительные данные для физических лиц, такие как информация о паспорте (серия, номер, кем выдан, дата выдачи, код подразделения), а также сведения о финансовых

документах (ОГРН, ИНН) и академические данные (ученое звание, ученая степень, а также ID систем WoS и Scopus). Дополнительно, указывается код страны (для связи с таблицей countries).

- **legal:** Эта таблица содержит данные для юридических лиц, в частности, название юридического лица и должность. Также здесь хранится код страны, с которым связана запись.
- **countries:** Содержит перечень стран с уникальными кодами и названиями. Эти коды используются в таблицах **physical** и **legal** для указания страны, с которой связан заявитель.

Пользователи приложения (**users**): В этой таблице хранятся данные о пользователях приложения – имя пользователя, хеш пароля и уровень доступа (администратор или обычный пользователь). Это обеспечивает контроль доступа к данным.

Сессии и группировка заявителей:

- **session_groups:** Таблица для создания групп заявителей, объединяемых в рамках одной сессии. Каждая группа имеет уникальный идентификатор.
- **session_group_members:** Таблица для связывания группы (из **session_groups**) с конкретными заявителями (из **applicants**). Это позволяет одной сессии пользователя работать сразу с набором заявителей
- **sessions:** Основная таблица для регистрации сессий пользователей. Здесь хранится, какой пользователь инициировал сессию, с каким набором (группой) заявителей работал, а также время доступа. Связь с группой заявителей осуществляется через идентификатор группы из таблицы **session_groups**.

3.3. Разработка внутренней логики приложения

Список литературы

1. Баранов, А. А. Разработка приложений на WPF: Учебное пособие. – Москва: БИНОМ. Лаборатория знаний, 2020. – 240 с.
2. Васильев, И. Н. WPF: создание богатых интерфейсов для Windows. – Санкт-Петербург: Питер, 2019. – 320 с.
3. Гуляев, С. В. Разработка программного обеспечения на C# и WPF. – Новосибирск: Сибирское университетское издательство, 2021. – 280 с.
4. Кузнецов, Е. В. Основы разработки WPF-приложений. – Москва: БИНОМ. Лаборатория знаний, 2019. – 200 с.
5. Левин, А. Б. C# и WPF: создание приложений для Windows. – Москва: Издательство "Сфера", 2018. – 350 с.
6. Мельников, П. Г. Профессиональная разработка WPF-приложений. – Москва: Вильямс, 2021. – 400 с.
7. Никифоров, Т. С. Учебник по WPF: Разработка интерактивных приложений. – Москва: Издательство "Взаимопомощь", 2022. – 350 с.
8. Петров, С. М. WPF для профессионалов. – Москва: IEEE, 2020. – 450 с.
9. Рябов, Х. А. Основы работы с Windows Presentation Foundation. – Екатеринбург: Уральский университет, 2020. – 260 с.
10. Сазонов, А. Н. Интерактивные приложения на WPF: проекты и решения. – Москва: Русский язык, 2020. – 290 с.
11. Сидоров, И. В. WPF и XAML: от основ до продвинутых техник. – Санкт-Петербург: Питер, 2021. – 310 с.
12. Тихонов, В. А. Автоматизация документирования интеллектуальной собственности. – Москва: ТК Велби, 2018. – 280 с.
13. Федоров, Д. П. Основы патентования и регистрация объектов интеллектуальной собственности. – Санкт-Петербург: Вариант, 2021. – 350 с.
14. Шевченко, А. П. Python и WPF для начинающих: разработка графических интерфейсов. – Екатеринбург: Учебное издательство, 2022. – 220 с.
15. Безруков, И. Л. Практика разработки WPF-приложений для бизнеса. – Москва: Аспект Пресс, 2019. – 300 с.
16. Голубев, Е. Н. Дизайн и пользовательский интерфейс в WPF. – Новосибирск: ИнфоЛаб, 2021. – 270 с.

17. Корнильев, С. А. WPF: управление потоками и асинхронное программирование. – Москва: Издательство "Энергетика", 2020. – 320 с.
18. Лукьяненко, Р. П. Разработка корпоративных WPF-приложений. – Москва: Инфраструктура, 2021. – 350 с.
19. Попов, А. Т. Интеллектуальная собственность: задачи и решения. – Санкт-Петербург: Центр правовой информации, 2018. – 230 с.
20. Стародубцева, Н. В. Автоматизация работы с интеллектуальной собственностью. – Москва: Издательство "Юриспруденция", 2021. – 290 с.
21. Федеральный институт промышленной собственности. Официальный сайт ФИПС. [Электронный ресурс]. URL: <https://www.fips.ru/>