

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа Инженерная школа информационных технологий и робототехники  
Направление подготовки 09.03.01 Информатика и ВТ  
Отделение школы (НОЦ) Информационных технологий

**БАКАЛАВРСКАЯ РАБОТА**

Тема работы
Разработка защищенной автоматизированной системы заказа и оформления пропусков на охраняемую территорию закрытого образования (г. Северск)

УДК 681.51:620.11-021.342(1-21)(571.16)

Студент

Группа	ФИО	Подпись	Дата
3-8В3В2	Распопов Андрей Митрофанович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кочегурова Елена Алексеевна	К.Т.Н.		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Баннова Кристина Алексеевна	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Авдеева Ирина Ивановна			

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Погребной Александр Владимирович	К.Т.Н.		

## Планируемые результаты обучения по ООП

Код Результатов	Результат обучения (выпускник должен быть готов)
<b><i>Профессиональные компетенции</i></b>	
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<b><i>Универсальные компетенции</i></b>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

**Министерство образования и науки Российской Федерации**  
 федеральное государственное автономное образовательное учреждение  
 высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки (специальность) 09.03.01 Информатика и ВТ  
 Отделение школы (НОЦ) Информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП

\_\_\_\_\_  
 (Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

Бакалаврской работы
---------------------

Студенту:

Группа	ФИО
3-8В3В2	Распопов Андрей Митрофанович

Тема работы:

Разработка защищенной автоматизированной системы заказа и оформления пропусков на охраняемую территорию закрытого образования (г. Северск)	
Утверждена приказом директора (дата, номер)	19.04.2018 №2752/с

Срок сдачи студентом выполненной работы:	01.06.2018
--	------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	<p><i>Объект исследования: Процесс оформления пропусков для въезда на территорию ЗАТО Северск</i></p> <p><i>Требования к продукту:</i></p> <ul style="list-style-type: none"> <li><i>авторизация через ЕСИА (госуслуги);</i></li> <li><i>подача заявлений гражданами в электронном виде;</i></li> <li><i>формирование списка лиц, подавших заявление на оформление пропусков;</i></li> <li><i>обмен данными с подразделениями, участвующими в процессе оформления пропусков;</i></li> <li><i>возможность сохранения и печати сформированных отчетов;</i></li> <li><i>возможность делать отметку о наличии либо отсутствии оснований для оформления въезда;</i></li> <li><i>возможность делать отметку о разрешении на въезд;</i></li> <li><i>оперативное получение необходимой информации с помощью определенных параметров выборки.</i></li> </ul>
---------------------------------	---

<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	<i>Анализ предметной области Проектирование защищенной автоматизированной системы заказа и оформления пропусков на охраняемую территорию закрытого образования Разработка защищенной автоматизированной системы заказа и оформления пропусков на охраняемую территорию закрытого образования Финансовый менеджмент, ресурсоэффективность и ресурсосбережение Социальная ответственность</i>
<b>Перечень графического материала</b>	<i>Мультимедийная презентация</i>
<b>Консультанты по разделам выпускной квалификационной работы</b>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Баннова Кристина Алексеевна
Социальная ответственность	Авдеева Ирина Ивановна

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	<i>15.03.2018</i>
---	-------------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Начальник отдела информатизации и защиты данных Комитета развития информационного общества Администрации ЗАТО Северск	Олейников Андрей Валерьевич			<i>15.03.2018</i>

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
3-8В3В2	Распопов Андрей Митрофанович		<i>15.03.2018</i>

**Министерство образования и науки Российской Федерации**  
 федеральное государственное автономное образовательное учреждение  
 высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки (специальность) 09.03.01 Информатика и ВТ  
 Уровень образования Бакалавриат  
 Отделение школы (НОЦ) Информационных технологий  
 Период выполнения (осенний / весенний семестр 2017/2018 учебного года) \_\_\_\_\_

Форма представления работы:

Бакалаврская работа
---------------------

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН**  
**выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	01.06.2018
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
25.05.2018	Основная часть	75
18.05.2018	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
18.05.2018	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Кочегурова Елена Алексеевна	К.Т.Н.		

**СОГЛАСОВАНО:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Погребной Александр Владимирович	К.Т.Н.		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
3-8В3В2	Распопов Андрей Митрофанович

<b>Школа</b>	<b>Инженерная школа информационных технологий и робототехники</b>	<b>Отделение</b>	<b>Информационных технологий</b>
<b>Уровень образования</b>	Бакалавриат	<b>Направление/специальность</b>	09.03.01 Информатика и ВТ

### Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

<p>1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i></p>	<p><i>Работа с информацией, представленной в российских научных публикациях, аналитических материалах, статических бюллетенях и изданиях, нормативно-правовых документах.</i></p>
<p>2. <i>Нормы и нормативы расходования ресурсов</i></p>	
<p>3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i></p>	

### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i></p>	<p><i>Проведена оценка коммерческого потенциала:</i>  <i>1. Потенциальные потребители результатов исследования.</i>  <i>2. Анализ конкурентных технических решений</i>  <i>3. SWOT-анализ</i></p>
<p>2. <i>Планирование и формирование бюджета научных исследований</i></p>	<p><i>Произведен расчет бюджета научных исследований.</i></p>
<p>3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i></p>	<p><i>Определена ресурсная, финансовая, бюджетная эффективность исследования посредством расчета интегрального финансового показателя, интегрального показателя ресурсоэффективности и эффективности.</i></p>

### Перечень графического материала:

<ol style="list-style-type: none"> <li>1. <i>Оценка конкурентоспособности технических решений</i></li> <li>2. <i>Матрица SWOT</i></li> <li>3. <i>Альтернативы проведения НИ</i></li> <li>4. <i>График проведения и бюджет НИ</i></li> <li>5. <i>Оценка ресурсной, финансовой и экономической эффективности НИ</i></li> </ol>
--

Дата выдачи задания для раздела по линейному графику	
--	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Баннова Кристина Алексеевна	К.Э.Н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
3-8В3В2	Распопов Андрей Митрофанович		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
3-8В3В2	Распопов Андрей Митрофанович

<b>Школа</b>	<b>Инженерная школа информационных технологий и робототехники</b>	<b>Отделение</b>	<b>Информационных технологий</b>
<b>Уровень образования</b>	Бакалавриат	<b>Направление/специальность</b>	09.03.01 Информатика и ВТ

### Исходные данные к разделу «Социальная ответственность»:

<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<p><i>Работы ведутся в помещении с достаточным освещением (в помещении установлены люминесцентные лампы мощностью 20 Вт). Уровень шума в пределах нормы. Помещение хорошо вентилируется. Рабочее место состоит из стола, стула и персонального компьютера.</i></p> <p><i>Область применения: автоматизация рабочего места.</i></p>
---	--

### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p><b>1. Производственная безопасность</b></p> <p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</p> <ul style="list-style-type: none"> <li>– физико-химическая природа вредности, её связь с разрабатываемой темой;</li> <li>– действие фактора на организм человека;</li> <li>– приведение допустимых норм с необходимой размерностью (со ссылкой на соответствующий нормативно-технический документ);</li> <li>– предлагаемые средства защиты;</li> <li>– (сначала коллективной защиты, затем – индивидуальные защитные средства).</li> </ul> <p>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</p> <ul style="list-style-type: none"> <li>– механические опасности (источники, средства защиты);</li> <li>– термические опасности (источники, средства защиты);</li> <li>– электробезопасность (в т.ч. статическое электричество, молниезащита – источники, средства защиты)</li> </ul>	<p><i>Анализ выявленных вредных факторов:</i></p> <ul style="list-style-type: none"> <li>– недостаточная освещенность рабочей зоны;</li> <li>– отклонение параметров микроклимата в помещении;</li> <li>– повышенный уровень шума.</li> </ul> <p><i>Психофизические факторы:</i></p> <ul style="list-style-type: none"> <li>– длительные статические нагрузки;</li> <li>– монотонность труда.</li> </ul> <p><i>Анализ выявленных опасных производственных факторов рабочей среды, влияющих на организм человека при работе с программным обеспечением в рабочем помещении, а именно:</i></p> <ul style="list-style-type: none"> <li>– опасность поражения электрическим током,</li> <li>– опасность поражения статическим электричеством,</li> <li>– короткое замыкание.</li> </ul> <p><i>В рабочих кабинетах выполняются такие защитные меры электробезопасности как защитное заземление, защитное отключение, электрическая изоляция токоведущих частей.</i></p> <p><i>Помещение отдела оборудовано огнетушителями порошковыми с источником давления ОПУ-05-01, а также помещение оснащено пожарной сигнализацией.</i></p>
<p><b>2. Экологическая безопасность:</b></p> <ul style="list-style-type: none"> <li>– защита селитебной зоны</li> </ul>	<p><i>Предприятие решает проблему отходов, в основной массе это бумага, путем сдачи их в пункты сбора вторичного сырья. Потребляемая</i></p>



<ul style="list-style-type: none"> <li>– анализ воздействия объекта на атмосферу (выбросы);</li> <li>– анализ воздействия объекта на гидросферу (сбросы);</li> <li>– анализ воздействия объекта на литосферу (отходы);</li> <li>– разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.</li> </ul>	<p><i>вода не требует дополнительной очистки перед отводом в канализацию. Вышедшее из строя оборудование складировается, а затем утилизируется. Утилизация люминесцентных ламп производится сторонней компанией.</i></p>
<p><b>3. Безопасность в чрезвычайных ситуациях:</b></p> <ul style="list-style-type: none"> <li>– перечень возможных ЧС при разработке и эксплуатации проектируемого решения;</li> <li>– выбор наиболее типичной ЧС;</li> <li>– разработка превентивных мер по предупреждению ЧС;</li> <li>– разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.</li> </ul>	<p><i>Определены основные причины возникновения пожаров и методы их устранения. В помещении установлены два порошковых огнетушителя с источником давления ОПУ-05-01</i></p>
<p><b>4. Правовые и организационные вопросы обеспечения безопасности:</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<p><i>Основные проводимые правовые и организационные мероприятия по обеспечению безопасности, трудящихся на рабочем месте согласно СанПиН 2.2.2/2.4.1340-03, Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 05.02.2018). Эргономика рабочего места.</i></p>

<b>Дата выдачи задания для раздела по линейному графику</b>	<b>01.03.2018</b>
---	-------------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Авдеева Ирина Ивановна			01.03.2018

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
3-8В3В2	Распопов Андрей Митрофанович		01.03.2018

## Реферат

Выпускная квалификационная работа 90 с., 30 рис., 24 табл., 27 источников литературы.

Ключевые слова: MVC, SQL, база данных, автоматизированная система, пропуск, режим.

Объектом исследования является процесс оформления пропусков на территорию ЗАТО Северск.

Цель работы – Разработка защищенной автоматизированной системы заказа и оформления пропусков на охраняемую территорию закрытого образования (г. Северск).

В процессе исследования проводился анализ функций, процесса оформления пропуска, выделялись функции, которые целесообразно автоматизировать.

В результате исследования разработана защищенная автоматизированная система заказа и оформления пропусков.

Область применения: результаты исследования могут использоваться в городах закрытого типа, где так же, как и в Северске действует пропускной режим для въезда.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ,  
ЕДИНИЦ И ТЕРМИНОВ

СУБД - система управления базой данных;

БД – база данных;

ВКР – выпускная квалификационная работа;

КПП – контрольно-пропускной пункт;

РСО – режимно-секретный отдел;

ЗАТО - закрытое административно-территориальное образование;

ОАО - открытое акционерное общество;

СХК – сибирский химический комбинат;

ГК «Росатом» – Государственная корпорация по атомной энергии «Росатом»;

УФСБ – управление федеральной службы безопасности;

ЕСИА - единая система идентификации и аутентификации;

ФИАС - федеральная информационная адресная система;

АРМ – автоматизированное рабочее место.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	14
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	17
1.1 Описание процесса оформления пропусков.....	17
1.2 Обоснование необходимости разработки защищенной автоматизированной системы.....	24
1.3 Требования к защищенной автоматизированной системе.....	25
2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА .....	28
2.1 Разработка базы данных .....	28
2.1.1 Концептуальная модель данных.....	28
2.1.2 Логическая модель данных .....	31
2.2 Создание базы данных.....	37
2.2.1 Обоснование выбора средств разработки.....	37
2.2.2 Создание базы данных с помощью MySQLWorkbench .....	37
2.2.3 Физическая модель данных.....	40
2.3 Разработка приложения.....	42
2.3.1 Обоснование выбора средств разработки.....	42
2.3.2 Разработка системы.....	45
3 РЕЗУЛЬТАТЫ ПРОВЕДЕНИЯ РАЗРАБОТКИ.....	52
4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	57
4.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	57
4.1.1 Потенциальные потребители результатов исследования .....	57
4.1.2 Анализ конкурентных технических решений .....	57
4.1.3 Технология QuaD .....	57
4.1.4 SWOT-анализ.....	59

4.2	Определение возможных альтернатив проведения научных исследований .....	59
4.3	Планирование научно-исследовательских работ.....	62
4.3.1	Структура работ в рамках научного исследования .....	62
4.3.2	Разработка графика проведения научного исследования.....	62
4.3.3	Бюджет научно-технического исследования .....	65
4.3.3.1	Расчет материальных затрат научно-технического исследования .....	65
4.3.3.2	Основная заработная плата .....	66
4.3.3.3	Дополнительная заработная плата .....	67
4.3.3.4	Отчисления во внебюджетные фонды .....	68
4.3.3.5	Накладные расходы.....	69
4.4	Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования .....	70
5	СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....	75
5.1	Описание рабочего места .....	75
5.2	Вредные факторы .....	77
5.2.1	Освещение.....	77
5.2.2	Шум .....	79
5.2.3	Микроклимат .....	80
5.2.4	Опасные факторы. Опасность поражения электрическим током .....	82
5.3	Охрана окружающей среды .....	83
5.4	Безопасность в чрезвычайных ситуациях.....	84
5.5	Эргономические требования к рабочему месту .....	86
	ЗАКЛЮЧЕНИЕ .....	88
	CONCLUSION .....	89
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	90
	Приложение А	
	Приложение Б	
	Приложение В	
	Приложение Г	

## ВВЕДЕНИЕ

Город Северск имеет статус ЗАТО - закрытое административно-территориальное образование в границах которого расположены промышленные предприятия по разработке, изготовлению, хранению и утилизации оружия массового поражения, переработке радиоактивных и других материалов, военные и иные объекты, для которых устанавливается особый режим безопасного функционирования и охраны государственной тайны, включающий специальные условия проживания граждан. Одним из таких условий является пропускной режим для въезда на территорию ЗАТО Северск. Пропускной режим – совокупность организационных и технических мероприятий, установленных правил, направленных на недопущение бесконтрольного прохода физических лиц, проезда транспортных средств, а также перемещения предметов, материалов на территорию контролируемых зон ЗАТО Северск. Пропускной режим является составной частью мер обеспечения особого режима безопасного функционирования ОАО «СХК» ГК «Росатом», расположенного на территории ЗАТО Северск. Пропускной режим, установленный в контролируемых зонах ЗАТО Северск предназначен для обеспечения порядка пропуска через КПП физических лиц, транспортных средств и исключения возможности несанкционированного прохода на охраняемую территорию посторонних лиц. Доступ физических лиц на территорию контролируемой зоны ЗАТО Северск осуществляется часовыми КПП по пропускам установленного образца. Пропуск является основным документом, дающим право прохода через КПП контролируемой зоны ЗАТО Северск. Установлены следующие виды пропусков: постоянные, временные, разовые и специальные. Порядок оформления въезда граждан в контролируемые зоны ЗАТО Северск в условиях особого режима безопасного функционирования ОАО «СХК» определяет Постановление Правительства РФ от 11 июня 1996 г. № 693 «Об утверждении Положения о порядке обеспечения особого режима в закрытом

административно-территориальном образовании, на территории которого расположены объекты Государственной корпорации по атомной энергии «Росатом» (ред. от 30.12.2012 г.).

Разработана инструкция о пропускном режиме в контролируемые зоны ЗАТО Северск №16-02/373 от 27.08.2013 в соответствии с требованиями законодательных актов РФ:

- Закон РФ от 14.07.1992 № 3297-1 «О закрытом административно-территориальном образовании» (ред. от 22.11.2011 г.);
- Закон РФ от 25.07.2002 № 115 «О правовом положении иностранных граждан в РФ» (ред. от 27.07.2013 г.);
- Постановление Правительства РФ от 11.06.1996 № 693 «Об утверждении Положения о порядке обеспечения особого режима в закрытом административно-территориальном образовании, на территории которого расположены объекты Государственной корпорации по атомной энергии «Росатом» (ред. от 30.12.2012 г.).

Инструкция о пропускном режиме в контролируемые зоны ЗАТО Северск №16-02/373 от 27.08.2013 определяет:

- порядок направления и рассмотрения заявлений о въезде в контролируемые зоны для постоянного проживания или временного пребывания;
- сроки рассмотрения соответствующих заявлений;
- порядок и сроки согласования разрешений о въезде в контролируемые зоны с отделом в г. Северске УФСБ России по Томской области;
- перечень должностных лиц, имеющих право выдачи разрешений на въезд.

Целью данной работы является разработка защищенной автоматизированной системы, позволяющей автоматизировать процесс оформления пропусков.

Для решения поставленной задачи можно выделить следующие этапы проведения работ:

- Произвести детальный анализ функций, процесса оформления пропуска;
- Выделить функции, которые целесообразно автоматизировать;
- Разработать защищенную автоматизированную систему для автоматизации работы;
- Подготовить необходимую документацию;
- Провести экономическое обоснование разработки;
- Провести анализ безопасности представляемого проекта.

Объектом для исследования является процесс оформления пропусков на охраняемую территорию ЗАТО Северск.



# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Описание процесса оформления пропусков

В процессе оформления пропусков на территорию ЗАТО Северск участвуют: граждане, РСО Администрации ЗАТО Северск, ОФЗОиСП ОАО «СХК», отдел УФСБ России по Томской области в г. Северске и бюро пропусков на ЦКПП. Процесс оформления пропусков на территорию ЗАТО Северск представлен в контекстной диаграмме на рисунке 1.

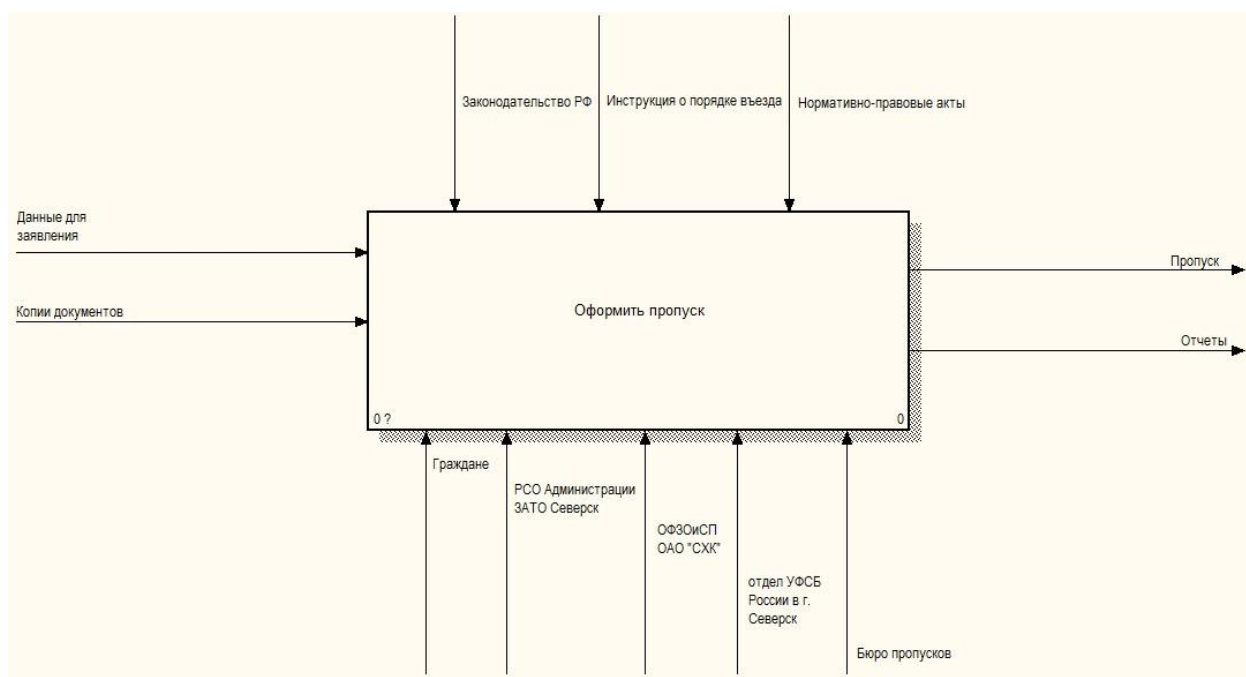


Рисунок 1 - Контекстная диаграмма процесса оформления пропусков на территорию ЗАТО Северск

Декомпозиция модели процесса оформления пропусков представлена на рисунке 2.

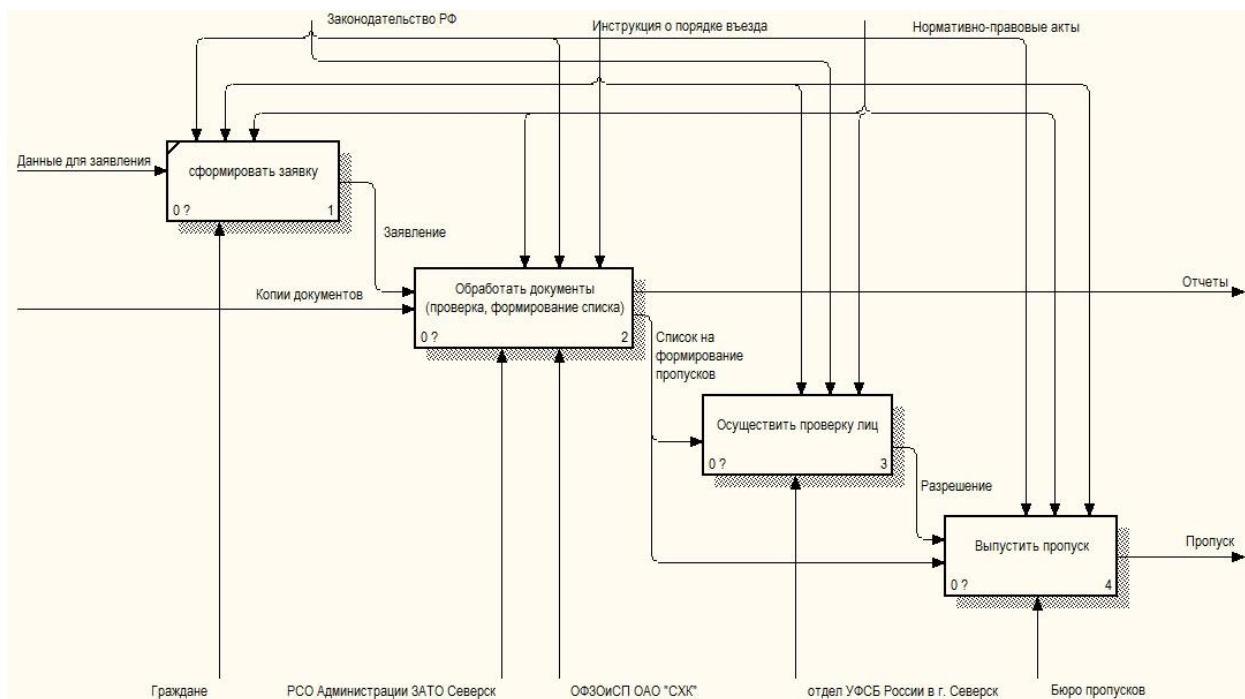


Рисунок 2 - Декомпозиция процесса оформления пропусков на территорию ЗАТО Северск

Проанализируем деятельность каждого участника процесса оформления пропусков поэтапно:

Для получения разрешения на въезд гражданами РФ подается заявление установленного образца (рисунок 3) и прилагаются копии необходимых документов.

В заявлении указываются:

- данные заявителя: фамилия, имя, отчество, дата рождения, место регистрации, место работы, серия и номер паспорта;
- данные на въезжающих граждан: причина въезда для постоянного проживания, степень родства, фамилия, имя, отчество, дата и место рождения, гражданство, серия и номер паспорта, место работы, должность, место регистрации, наличие судимости или нахождение под следствием.

Необходимость въезда в контролируемую зону ЗАТО Северск подтверждается следующими документами (копии):

- приказом о приеме на работу, трудовым договором на неопределенный срок, заключенным с ОАО «СХК», органами государственной власти или Администрацией ЗАТО Северск и подведомственными им организациями;
- контрактом (договором) — для военнослужащих и гражданского персонала воинских частей, должностных лиц, работников и государственных гражданских служащих территориальных подразделений федеральных органов исполнительной власти (сотрудников органов внутренних дел, подразделений противопожарной службы);
- свидетельством о регистрации права собственности жилого помещения, расположенного в контролируемой зоне ЗАТО Северск;
- свидетельством (для граждан РФ, постоянно проживающих в контролируемой зоне ЗАТО Северск и членов их семей) о рождении, о заключении брака, об усыновлении, об опекуновстве.

Главе Администрации ЗАТО Северск  
Диденко Н.В.  
Ф.И.О.

от \_\_\_\_\_  
Ф.И.О.

\_\_\_\_\_

паспорт: \_\_\_\_\_  
число, месяц, год рождения №

\_\_\_\_\_  
кем, когда выдан

проживающей (его) по адресу: \_\_\_\_\_

\_\_\_\_\_  
город, улица, дом, корпус, квартира

телефон: д. \_\_\_\_\_ р. \_\_\_\_\_  
 сот. \_\_\_\_\_

место работы: \_\_\_\_\_

\_\_\_\_\_

должность: \_\_\_\_\_

муж (жена) \_\_\_\_\_  
Ф.И.О.

Место работы: \_\_\_\_\_

#### ЗАЯВЛЕНИЕ

Прошу разрешить временный въезд в город моим родственникам сроком на \_\_\_\_\_ дней

1. \_\_\_\_\_  
степень родства Ф.И.О.

д.р. \_\_\_\_\_  
число, месяц, год рождения гражданин(ин) гражданство место рождения (страна, область, район, город, село)

паспорт: \_\_\_\_\_  
серия номер выдан кем число, месяц, год

Проживает по адресу (регистрация по паспорту): \_\_\_\_\_  
(область, район, город, улица, дом, корпус, квартира)

Место работы \_\_\_\_\_ должность \_\_\_\_\_

Наличие судимости или нахождения под следствием \_\_\_\_\_

Причина въезда: \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г. \_\_\_\_\_  
подпись

В соответствии со ст. 9 ФЗ от 27.06.2006 № 152-ФЗ «О персональных данных» я даю согласие Администрации ЗАТО Северск на обработку моих персональных данных

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г. \_\_\_\_\_  
подпись

**Приложение:** копии документов, подтверждающие родство

Рисунок 3 – Заявления на временный въезд в ЗАТО Северск

Администрация ЗАТО Северск – исполнительно-распорядительный орган местного самоуправления ЗАТО Северск, наделённый полномочиями по реализации вопросов местного значения городского округа ЗАТО Северск, предусмотренных Федеральным законом от 06.10.2003 № 131-ФЗ «Об общих принципах организации местного самоуправления в Российской Федерации». Структура Администрации ЗАТО Северск утверждается Думой ЗАТО Северск по представлению Главы Администрации ЗАТО Северск. В структуру Администрации ЗАТО Северск входят заместители Главы Администрации, отраслевые и территориальные органы – управления, комитеты и отделы. Одним из структурных подразделений является Режимно-секретный отдел Администрации ЗАТО Северск.

Режимно-секретный отдел Администрации ЗАТО Северск выполняет следующие функции:

- Принимает заявления от граждан и документы, подтверждающие необходимость въезда (контракт, договор, трудовой договор и т. п.);
- Проверяет въезжающих по учетам лиц, объявленных в розыск, и проводит проверку изложенных в заявлении данных на основании представленных документов и документов, удостоверяющих личность заявителя;
- Делает отметку на заявлении о наличии либо отсутствии оснований для оформления въезда;
- Выносит решение об отказе, при отсутствии оснований для оформления разрешения;
- Готовит и направляет списки для согласования в отдел в г. Северске УФСБ России по Томской области.

ОАО «Сибирский химический комбинат» — предприятие в Северске, объединяющее четыре завода по обращению с ядерными материалами. Одно из основных направлений работы предприятия — обеспечение потребностей атомных электростанций в уране для ядерного топлива. Входит в состав

топливной компании «ТВЭЛ» госкорпорации «Росатом». Одним из структурных подразделений является отдел физической защиты объектов и спец продукции – ОФЗОиСП ОАО «СХК».

Если въезд связан с нуждами ОАО «СХК», ОФЗОиСП ОАО «СХК» выполняет следующие функции:

- Принимает заявления и документы, подтверждающие необходимость въезда;
- Проверяет въезжающих по учетам лиц, объявленных в розыск, и проводит проверку изложенных в заявлении данных на основании представленных документов и документов, удостоверяющих личность заявителя;
- Делает отметку на заявлении о наличии либо отсутствии оснований для оформления въезда;
- Выносит решение об отказе, при отсутствии оснований для оформления разрешения;
- Готовит и направляет списки для согласования в отдел в г. Северске УФСБ России по Томской области.

Декомпозиция модели процесса деятельности РСО Администрации ЗАТО Северск и ОФЗОиСП ОАО «СХК» при оформлении пропусков представлена на рисунке 4.

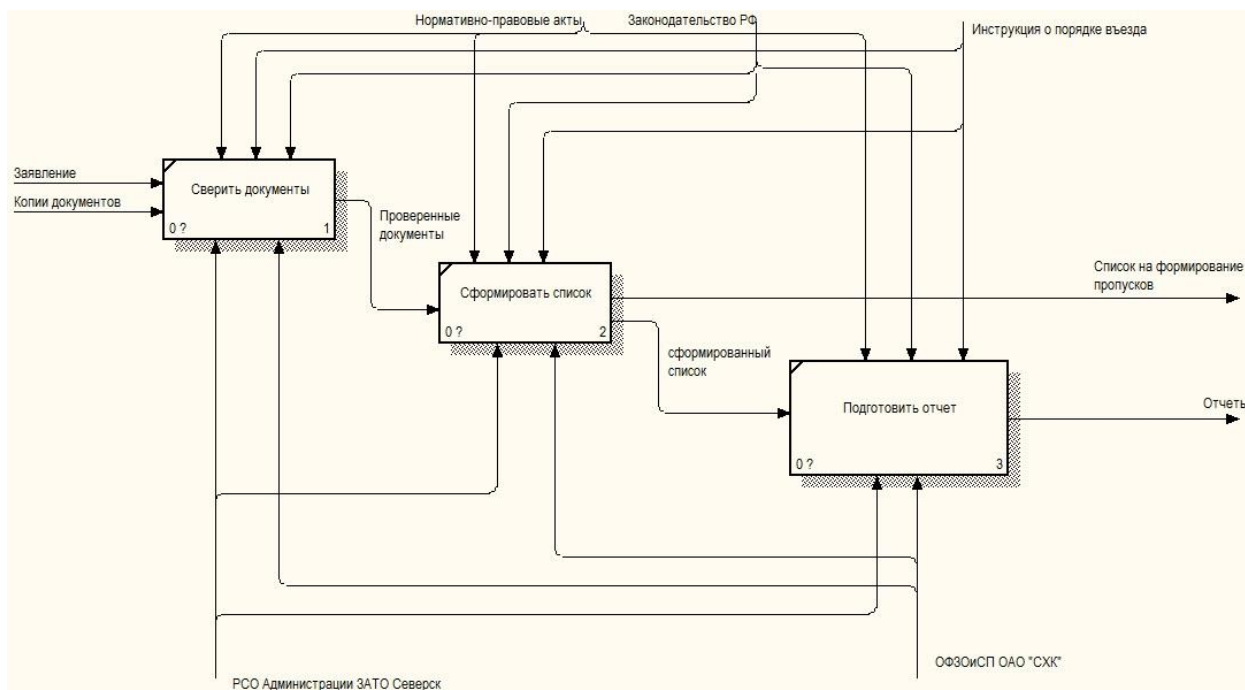


Рисунок 4 - Декомпозиция модели процесса деятельности PCO Администрации ЗАТО Северск и ОФЗОиСП ОАО «СХК» при оформлении пропусков

Управление федеральной службы безопасности Российской Федерации (УФСБ России) - федеральный орган исполнительной власти Российской Федерации, спецслужба, осуществляющая в пределах своих полномочий решение задач по обеспечению безопасности Российской Федерации. Отдел в г. Северске УФСБ России по Томской области исполняет следующие функции:

- Принимает списки на согласование от ОФЗОиСП ОАО «СХК» и PCO Администрации ЗАТО Северск;
- Проводит проверку изложенных в заявлении данных на основании представленных документов и документов, удостоверяющих личность заявителя;
- Отправляет согласованные списки в PCO Администрации ЗАТО Северск либо в ОФЗОиСП ОАО «СХК», если въезд связан с нуждами ОАО «СХК».

Декомпозиция модели процесса деятельности Отдел в г. Северске УФСБ России по Томской области при оформлении пропусков представлена на рисунке 5.

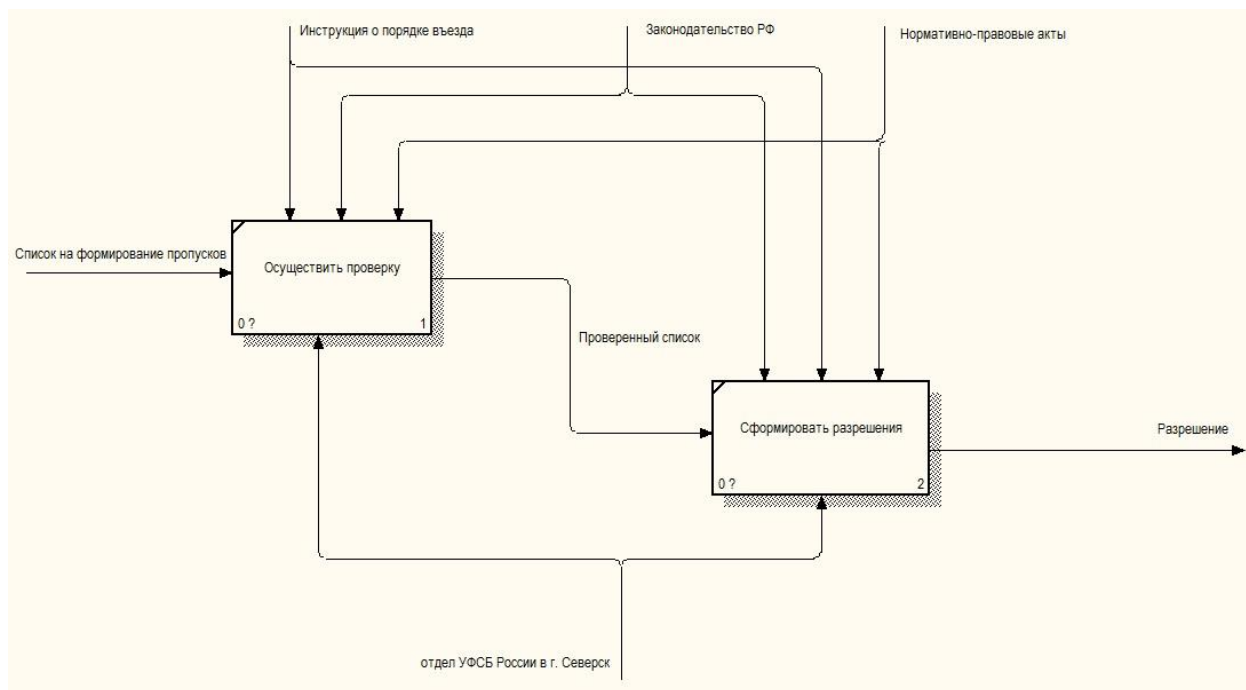


Рисунок 5 – Декомпозиция модели процесса деятельности Отдел в г. Северске УФСБ России по Томской области при оформлении пропусков

При наличии согласования отдела в г. Северске УФСБ России по Томской области должностным лицом ОФЗОиСП ОАО «СХК» или РСО Администрации ЗАТО Северск принимается решение о выдаче разрешения на въезд. После получения разрешения на въезд в ЗАТО Северск гражданам выдаются пропуска в филиале Бюро пропусков, расположенном на Центральном КПП.

Декомпозиция модели процесса деятельности бюро пропусков на ЦКПП при оформлении пропусков представлена на рисунке 6.

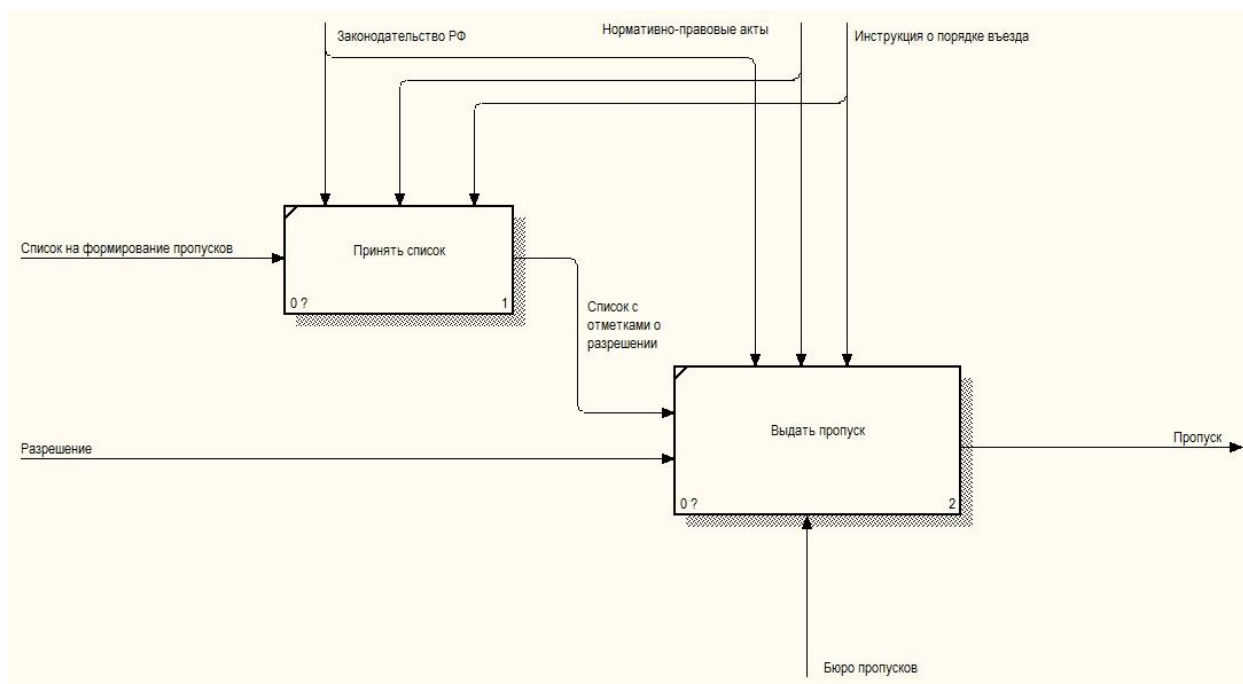


Рисунок 6 - Декомпозиция модели процесса деятельности бюро пропусков на ЦКПП при оформлении пропусков

## 1.2 Обоснование необходимости разработки защищенной автоматизированной системы

В настоящее время для оформления пропуска гражданам необходимо заполнять бланк заявления в печатном виде, делать печатные копии документов и приносить в РСО Администрации ЗАТО Северск или если въезд связан с нуждами ОАО «СХК» в ОФЗОиСП ОАО «СХК», тратить большое количество времени в ожидании приема и проверки корректности заполнения данных. В случае обнаружения ошибок требуется повторное заполнение заявления, при необходимости доносить необходимые документы или копии документов. Сформированные списки хранятся на бумажных носителях, что затрудняет осуществить оперативный поиск требуемой информации. Так как в списках содержатся персональные данные, передача данных при взаимодействии участников процесса требует соблюдения особых условий для работы в соответствии с требованиями регуляторов в области защиты информации.



На данный момент отсутствует полноценная (полнофункциональная) система, позволяющая хранить и обрабатывать данные в электронном виде, а также формировать отчеты по данным и иметь возможность быстрого доступа к информации.

Для оптимизации процесса заказа, оформления и выдачи пропусков необходимо провести автоматизацию следующих функций:

- Формирование и передача комплекта документов ответственному исполнителю с использованием телекоммуникационной сети Интернет;
- Прием и обработку документов, подготовку и передачу списков для оформления пропусков;
- Построение и просмотр отчетов;
- Организацию приема и передачи данных в процессе взаимодействия участников.

Все вышеперечисленные функции являются наиболее трудоемкими и время затратными в процессе оформления пропусков на территорию ЗАТО Северск.

### **1.3 Требования к защищенной автоматизированной системе**

В результате проведенного анализа предметной области были установлены требования к будущему программному продукту.

*Требования к функциональным характеристикам.*

Система должна обеспечить:

- 1) авторизацию через ЕСИА (госуслуги);
- 2) подачу заявлений гражданами в электронном виде;
- 3) формирование списка лиц, подавших заявление на оформление пропусков;
- 4) обмен данными с подразделениями, участвующими в процессе оформления пропусков;

- 5) возможность делать отметку о наличии либо отсутствии оснований для оформления въезда;
- 6) возможность делать отметку о разрешении на въезд;
- 7) возможность сохранения и печати сформированных отчетов;
- 8) оперативное получение необходимой информации с помощью определенных параметров выборки.

*Требования к надежности.*

Система должна обеспечить:

- 1) контроль входной информации;
- 2) целостность данных;
- 3) защиту от несанкционированного доступа к системе;
- 4) разграничение прав доступа к информации.

*Требования к программному обеспечению оператора.*

Система должна функционировать на платформе не ниже ОС Windows 7;

Должен быть установлен:

- Microsoft .NET Framework 4.0 или старше;

Рекомендуемый браузер:

- Microsoft Internet Explorer 11.0;

*Требования к программному обеспечению пользователя.*

Система должна функционировать на платформе Windows, MacOS и Linux;

Поддерживаемые браузеры:

- Google Chrome 28.0 и более поздние;
- Mozilla Firefox 47.0 и более поздние;
- Microsoft Internet Explorer 11.0;
- Microsoft Edge 13.0 и более поздние;
- Opera 36.0 и более поздние.

*Требования к аппаратному обеспечению оператора.*

Защищенная автоматизированная система должна функционировать на персональном компьютере со следующими характеристиками:

- Процессор: 2-х ядерный с частотой не менее 1 GHz или 1-ядерный с частотой не менее 1,8 GHz;
- Оперативная память: не менее 2 ГБ;
- Место на жестком диске: не менее 200 МБ;
- Разрешение экрана: не менее 1024×768.
- Подключение к сети Интернет:
- Скорость соединения с данного АРМ:
  - минимальная: 600 Кбит/с;
  - рекомендуемая: 1 Мбит/с и выше.

*Требования к аппаратному обеспечению пользователя.*

Защищенная автоматизированная система должна функционировать на персональном компьютере со следующими характеристиками:

- Процессор: 2-х ядерный с частотой не менее 1 GHz или 1-ядерный с частотой не менее 1,8 GHz
- Оперативная память: Не менее 256 МБ;
- Место на жестком диске: не менее 100 МБ;
- Разрешение экрана: не менее 1024×768;
- Подключение к сети Интернет:
- Скорость соединения с данного АРМ:
  - минимальная: 600 Кбит/с;
  - рекомендуемая: 1 Мбит/с и выше.

## **2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА**

### **2.1 Разработка базы данных**

Для разработки базы данных выделим этапы, при помощи которых осуществляется переход от предметной области к её конкретной реализации:

- изучение и анализ предметной области;
- разработка концептуальной модели данных;
- разработка логической модели данных;
- разработка физической модели данных;
- разработка информационных объектов базы данных.

Изучение и анализ предметной области были проведены в разделе 1. Разработку концептуальной модели данных осуществим с использованием графической нотации. Концептуальное проектирование базы данных - процесс создания модели используемой на предприятии информации, не зависящей от любых физических аспектов ее представления.

#### **2.1.1 Концептуальная модель данных**

Концептуальная модель базы данных это некая наглядная диаграмма, нарисованная в принятых обозначениях, и подробно показывающая связь между объектами и их характеристиками [4]. Создается концептуальная модель для дальнейшего проектирования базы данных и перевод ее в реляционную базу данных. На концептуальной модели в визуально удобном виде прописываются связи между объектами данных и их характеристиками. Концептуальная модель является некой диаграммой с принятыми обозначениями элементов. Так, все объекты, являющиеся сущностями, обозначаются в виде прямоугольника. Мощность связи обозначаются

стрелками (в направлении, где мощность равна многим - двойная стрелка, а со стороны, где она равна единице - одинарная). Для построения концептуальной модели данных необходимо определить:

- основные типы сущностей, которые требуются для конкретного представления.
- важнейшие типы связей, существующих между сущностями.
- атрибуты и связывание их с типами сущностей и связей

Основными типами сущностей, которые можно выделить на основе диаграмм на данном этапе проектирования, являются: «Заявитель», «Въезжающий», «Заявка», «Список ФСБ», «Список ЦКПП», «Категория въезда», «Цель приезда», «Обработка».

После того как определили основные типы сущностей, необходимо определить их связи:

- Сущность «Заявка» связана с сущностями «Заявитель», «Въезжающий», «Список ФСБ», «Список ЦКПП», «Категория въезда», «Цель приезда», «Обработка»;
- Сущность «Заявитель» связана с сущностью «Заявка»;
- Сущность «Въезжающий» связана с сущностью «Заявка»;
- Сущность «Список ФСБ» связана с сущностями «Заявка» и «Обработка»;
- Сущность «Список ЦКПП» связана с сущностями «Заявка» и «Обработка»;
- Сущность «Категория въезда» связана с сущностью «Заявка»;
- Сущность «Цель приезда» связана с сущностью «Заявка».

Завершающим этапом построения концептуальной модели является определение атрибутов и связывание их с типами сущностей. Для каждой сущности необходимо произвести абстрагирование ее свойств, т.е. выделить

только значимый для нас набор данных. Этот набор и будет являться атрибутами данной сущности.

Определим атрибуты для сущности «Заявитель»: фамилия, имя, отчество, дата рождения, место регистрации, место работы, серия паспорта и номер паспорта.

Для сущности «Въезжающий»: фамилия, имя, отчество, дата рождения, место рождения, гражданство, серия паспорта, номер паспорта, место работы, должность, место регистрации, наличие судимости, скан паспорта и скан родства.

Для сущности «Заявка»: статус въезда, дата заполнения, кто разрешил, примечания.

Для сущности «Список ФСБ»: № списка ФСБ, дата размещения, пояснение на размещение, пояснение на отказ.

Для сущности «Список ЦКПП»: № списка ЦКПП, дата печати ЦКПП.

Для сущности «Обработка»: дата поступления, комментарий.

Для сущности «Категория въезда»: категория въезда

Для сущности «Цель приезда»: цель приезда

Диаграмма концептуальной модели данных с указанием атрибутов сущностей приведена на рисунке 7.

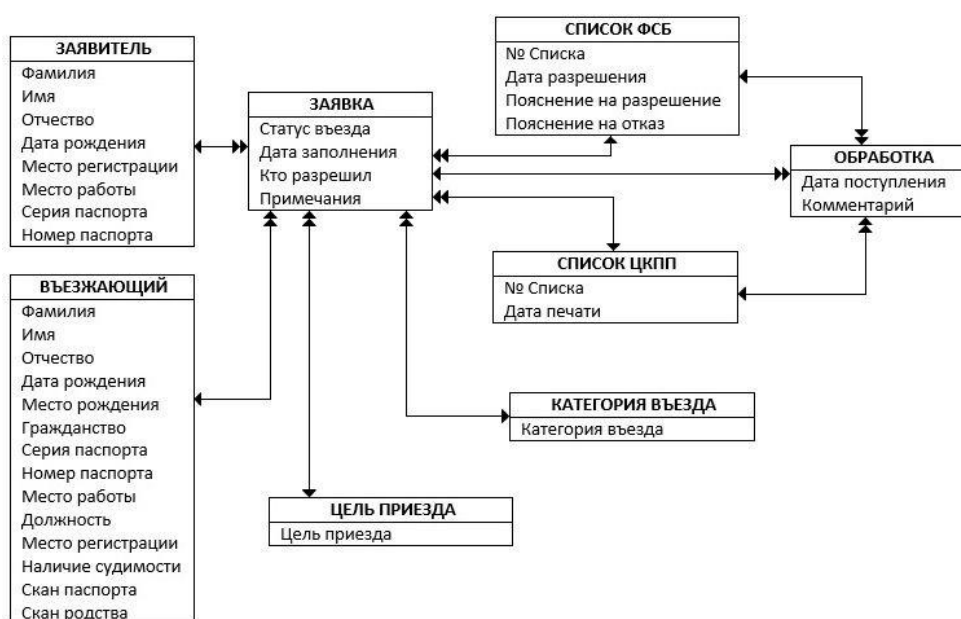


Рисунок 7 - Диаграмма концептуальной модели данных

## 2.1.2 Логическая модель данных

Логическая модель данных создается на основе выбранной модели организации данных целевой СУБД. Выбранная нами модель организации данных является реляционной моделью. Для получения логической модели данных необходимо взять концептуальную модель данных, созданную на предыдущем этапе, уточнить ее и преобразовать в логическую модель данных.

В созданной концептуальной модели данных наблюдаются не полные функциональные зависимости между не ключевыми атрибутами и составными ключами в некоторых сущностях. Для устранения имеющихся недостатков проводится нормализация данных. Опишем полученные сущности:

Сущность «Заявитель», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 1. Ключевым атрибутом будет «ID Заявителя». Данная сущность имеет связи «один ко многим» с сущностью «Заявка».

Таблица 1 – Сущность «Заявитель»

<i>Атрибут</i>	<i>Домен</i>	<i>Примечание</i>
Id_applicant	INT	ID Заявителя
surname	VARCHAR(45)	Фамилия
name	VARCHAR(45)	Имя
patronymic	VARCHAR(45)	Отчество
date_of_birth	DATE	Дата рождения
place_of_registration	VARCHAR(100)	Место регистрации
place_of_work	VARCHAR(100)	Место работы
series	INT	Серия паспорта
number	INT	Номер паспорта

Сущность «Въезжающий», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 2. Ключевым атрибутом будет «ID Въезжающего». Данная сущность имеет связи «один ко многим» с сущностью «Заявка».

Таблица 2 – Сущность «Въезжающий»

<i>Атрибут</i>	<i>Домен</i>	<i>Примечание</i>
Id_entering	INT	ID Въезжающего
surname	VARCHAR(45)	Фамилия
name	VARCHAR(45)	Имя
patronymic	VARCHAR(45)	Отчество
date_of_birth	DATE	Дата рождения
place_of_birth	VARCHAR(45)	Место рождения
citizenship series	VARCHAR(45)	Гражданство
series	INT	Серия паспорта
number	INT	Номер паспорта
place_of_work	VARCHAR(255)	Место работы
position	VARCHAR(100)	Должность
place_of_registration	VARCHAR(45)	Место регистрации
previous_conviction	VARCHAR(45)	Наличие судимости
scan_passport	VARCHAR(255)	Скан паспорта
scan_kinship	VARCHAR(255)	Скан родства

Сущность «Заявка», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 3. Ключевым атрибутом будет «ID Заявки». Данная сущность имеет связи «один ко многим» с сущностью «Обработка» и связи «многие к одному» с сущностями: «Заявитель», «Въезжающий», «Список ФСБ», «Список ЦКПП», «Категория въезда» и «Цель приезда».

Таблица 3 – Сущность «Заявка»

<i>Атрибут</i>	<i>Домен</i>	<i>Примечание</i>
Id_application	INT	ID Заявки
status	VARCHAR(45)	Статус въезда
date_of_completion	DATE	Дата заполнения
allowed	VARCHAR(45)	Кто разрешил
notes	VARCHAR(255)	Примечания

Сущность «Список ФСБ», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 4. Ключевым атрибутом будет «ID Списка». Данная сущность имеет связи «один ко многим» с сущностью «Заявка».



Таблица 4 – Сущность «Список ФСБ»

<i>Атрибут</i>	<i>Домен</i>	<i>Примечание</i>
Id_list	INT	ID Списка
number_list_fsb	INT	№ Списка ФСБ
date_of_authorization	DATE	Дата разрешения
explanation_of_permission	VARCHAR(255)	Пояснение на разрешение
explanation_of_failure	VARCHAR(255)	Пояснение на отказ

Сущность «Список ЦКПП», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 5. Ключевым атрибутом будет «ID Заявки». Данная сущность имеет связи «один ко многим» с сущностью «Заявка».

Таблица 5 – Сущность «Список ЦКПП»

<i>Атрибут</i>	<i>Домен</i>	<i>Примечание</i>
Id_list_KPP	INT	ID Списка
number_list_ckpp	INT	№ Списка ЦКПП
print_date	DATE	Дата печати ЦКПП

Сущность «Категория въезда», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 6. Ключевым атрибутом будет «ID». Данная сущность имеет связи «один ко многим» с сущностью «Заявка».

Таблица 6 – Сущность «Категория въезда»

<i>Атрибут</i>	<i>Домен</i>	<i>Примечание</i>
Id_entry	INT	ID
entry_category	VARCHAR(45)	Категория въезда

Сущность «Цель приезда», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 7. Ключевым атрибутом будет «ID». Данная сущность имеет связи «один ко многим» с сущностью «Заявка».

Таблица 7 – Сущность «Цель приезда»

<i>Атрибут</i>	<i>Домен</i>	<i>Примечание</i>
Id_target	INT	ID
target	VARCHAR(255)	Цель приезда

Сущность «Обработка», включающая в себя атрибуты и домены для атрибутов, представлена в таблице 7. Ключевым атрибутом будет «ID».

Данная сущность имеет связи «многие к одному» с сущностью «Заявка», с сущностью «Список ФСБ» и с сущностью «Список ЦКПП».

Таблица 8 – Сущность «Обработка»

Атрибут	Домен	Примечание
Id_processing	INT	ID
receipt_date	DATE	Дата поступления
comment	VARCHAR(255)	Комментарий

Логическая модель данных с рассмотренными связями представлена на рисунке 8.

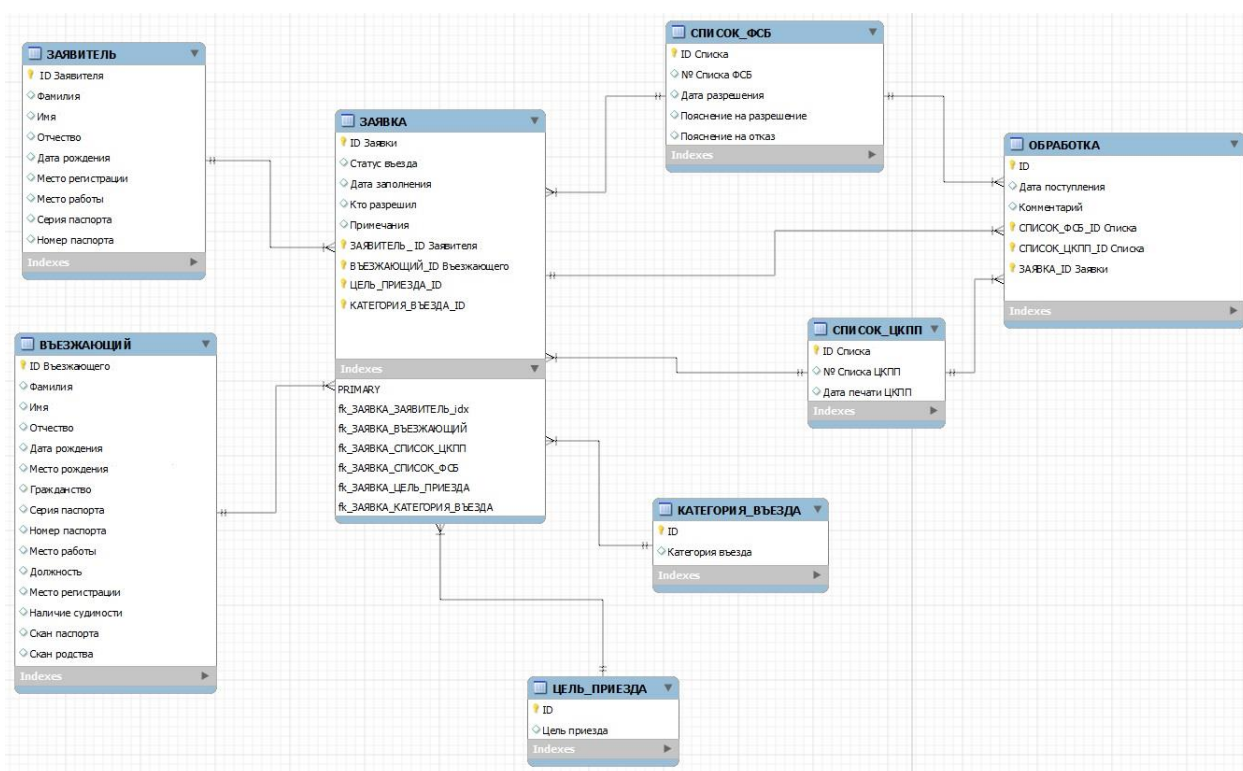


Рисунок 8 – Логическая модель данных

Для полей «Место рождения» и «Место регистрации» реализован ввод адресов по базе ФИАС. Федеральная информационная адресная система (ФИАС) обеспечивает формирование и использование единой адресной базы России. Каждому адресу в ФИАС присваивается уникальный номер. Это помогает исключить ошибки при поиске адреса и дает возможность ведомствам обмениваться данными в электронном формате. Поэтому время

обработки и поиска информации сокращается. ФИАС находится в распоряжении двух ведомств — ФНС России и Минфин России. Налоговая служба обеспечивает функциональные возможности адресной системы. Министерство финансов формирует и регулирует правовую основу ФИАС.

ФИАС регулируется Федеральным законом № 443 «О федеральной информационной адресной системе и о внесении изменений в Федеральный закон «Об общих принципах организации местного самоуправления в Российской Федерации».

### Структура данных ФИАС

Основные таблицы:

- ADDROBJ - Классификатор адресообразующих элементов (край > область > город > район > улица)
- HOUSE - Сведения по номерам домов, улиц, городов и населенных пунктов, номера земельных участков.
- HOUSEINT - Интервалы домов
- LANDMARK - Описание мест расположения имущественных объектов
- NORMDOC - Сведения по нормативному документу, являющемуся основанием присвоения адресному элементу наименования

Справочники:

- ACTSTAT - Статус актуальности ФИАС
- CENTERST - Статус центра
- CURENTST - Статус актуальности КЛАДР 4.0
- ESTSTAT - Признак владения
- HSTSTAT - Статус состояния объектов недвижимости
- INTVSTAT - Статус интервала домов
- NDOCTYPE - Тип нормативного документа (закон, приказ, справка)
- OPERSTAT - Статус действия
- SOCRBASE - Типы адресных объектов (условные сокращения)
- STRSTAT - Признак строения

Полная БД выгружается налоговой в двух форматах - DBF и XML. На основе XSD-схем данных были построены скрипты создания таблиц, полученный код представлен в Приложение А.

MyISAM выбран в качестве движка, что ускоряет импорт. MyISAM — одна из основных (наряду с InnoDB) систем хранения данных в СУБД MySQL. Она основывается на принципах ISAM и обладает в сравнении с ним рядом полезных дополнений. MyISAM имеет возможности по сжатию данных, по созданию полнотекстовых индексов.

Имея готовые структуры, несложно выполнить импорт данных. Имена XML-файлов для удобства были укорочены. Затем построены запросы импорта данных.

```
LOAD XML LOCAL INFILE 'D:\\fias\\ACTSTAT.XML' INTO TABLE `ActualStatus` ROWS
IDENTIFIED BY '<ActualStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\ESTSTAT.XML' INTO TABLE `EstateStatus` ROWS
IDENTIFIED BY '<EstateStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\INTVSTAT.XML' INTO TABLE `IntervalStatus` ROWS
IDENTIFIED BY '<IntervalStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\STRSTAT.XML' INTO TABLE `StructureStatus` ROWS
IDENTIFIED BY '<StructureStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\CENTERST.XML' INTO TABLE `CenterStatus` ROWS
IDENTIFIED BY '<CenterStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\DEL_HOUSEINT.XML' INTO TABLE
`DeletedHouseInterval` ROWS IDENTIFIED BY '<HouseInterval>';
LOAD XML LOCAL INFILE 'D:\\fias\\OPERSTAT.XML' INTO TABLE `OperationStatus`
ROWS IDENTIFIED BY '<OperationStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\NDOCTYPE.XML' INTO TABLE
`NormativeDocumentType` ROWS IDENTIFIED BY '<NormativeDocumentType>';
LOAD XML LOCAL INFILE 'D:\\fias\\HSTSTAT.XML' INTO TABLE `HouseStateStatus`
ROWS IDENTIFIED BY '<HouseStateStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\CURENTST.XML' INTO TABLE `CurrentStatus` ROWS
IDENTIFIED BY '<CurrentStatus>';
LOAD XML LOCAL INFILE 'D:\\fias\\DEL_NORMDOC.XML' INTO TABLE
`DeletedNormativeDocument` ROWS IDENTIFIED BY '<NormativeDocument>';
LOAD XML LOCAL INFILE 'D:\\fias\\SOCRBASE.XML' INTO TABLE `AddressObjectType`
ROWS IDENTIFIED BY '<AddressObjectType>';
LOAD XML LOCAL INFILE 'D:\\fias\\DEL_HOUSE.XML' INTO TABLE `DeletedHouse` ROWS
IDENTIFIED BY '<House>';
LOAD XML LOCAL INFILE 'D:\\fias\\DEL_ADDROBJ.XML' INTO TABLE
`DeletedAddressObject` ROWS IDENTIFIED BY '<Object>';
LOAD XML LOCAL INFILE 'D:\\fias\\LANDMARK.XML' INTO TABLE `Landmark` ROWS
IDENTIFIED BY '<Landmark>';
LOAD XML LOCAL INFILE 'D:\\fias\\HOUSEINT.XML' INTO TABLE `HouseInterval` ROWS
IDENTIFIED BY '<HouseInterval>';
LOAD XML LOCAL INFILE 'D:\\fias\\NORMDOC.XML' INTO TABLE `NormativeDocument`
ROWS IDENTIFIED BY '<NormativeDocument>';
LOAD XML LOCAL INFILE 'D:\\fias\\ADDROBJ.XML' INTO TABLE `AddressObject` ROWS
IDENTIFIED BY '<Object>';
```

Импорт прошёл без проблем. Итоговый размер базы ФИАС составил около 9 Гбайт.

## **2.2 Создание базы данных**

### **2.2.1 Обоснование выбора средств разработки**

Для работы с реляционными моделями данных существует несколько систем управления базами данных (СУБД). Одной из них является СУБД MySQL.

СУБД MySQL обладает очень широкими возможностями. Во-первых, она обладает большинством возможностей, предоставляемых в дорогих и мощных пакетах для работы с базами данных. Во-вторых, MySQL выпускается с открытым кодом и распространяется бесплатно, работает с большим количеством операционных систем и может использоваться со многими языками программирования. Работает очень быстро даже с большими наборами данных. В-третьих, базы данных MySQL используют стандартную форму широко известного языка SQL.

SQL - это специализированный непроцедурный язык, позволяющий описывать данные, осуществлять выборку и обработку информации из реляционных СУБД. Специализированность означает, что sql предназначен лишь для работы с БД.

Существуют специальные приложения и среды разработки, которые предназначены для интерпретации данных SQL и выполнения различного рода операций над ними. Одним из таких специализированных приложений является MySQL Workbench.

### **2.2.2 Создание базы данных с помощью MySQL Workbench**

MySQL Workbench – это средство визуализации базы данных для проектирования и моделирования баз данных для реляционной базы данных MySQL [4]. Это облегчает создание новых моделей физических данных и модификацию существующих баз данных MySQL с помощью функций обратного / прямого проектирования и управления изменениями.

## Возможности MySQL Workbench:

- В MySQL Workbench есть инструменты, которые позволяют разработчикам и администраторам баз данных визуально создавать физические модели проектирования баз данных, которые легко могут быть переведены в базы данных MySQL с использованием форвардной инженерии.
- MySQL Workbench поддерживает создание нескольких моделей в одной среде.
- Он поддерживает все объекты, такие как таблицы, представления, хранимые процедуры, триггеры, которые составляют базу данных.
- MySQL Workbench имеет встроенную утилиту проверки модели, которая сообщает о любых проблемах, которые могут быть обнаружены разработчиком данных.
- MySQL Workbench, имеет встроенный визуальный редактор SQL.
- Редактор Visual SQL позволяет разработчикам создавать, редактировать и запускать запросы к базам данных MySQL. Он имеет утилиты для просмотра данных и их экспорта.
- Его синтаксические цветовые выделения помогают разработчикам легко писать и отлаживать SQL-запросы.
- Можно запускать несколько запросов, и результаты автоматически отображаются на разных вкладках.
- Запросы также сохраняются на панели истории для последующего поиска и запуска.

Последовательность создания базы данных для СУБД MYSQL с помощью инструмента MySQL Workbench следующая:

1. При запуске MySQL Workbench в меню открываем File -> New Model - рисунок 9.

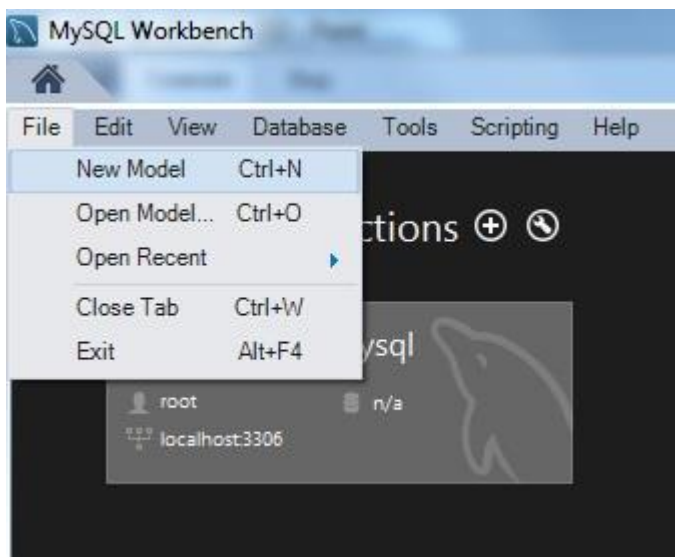


Рисунок 9 – создание новой модели в MySQL Workbench

2. В открывшемся окне необходимо нажать на «Add Diagram» для добавления ER диаграммы – рисунок 10.

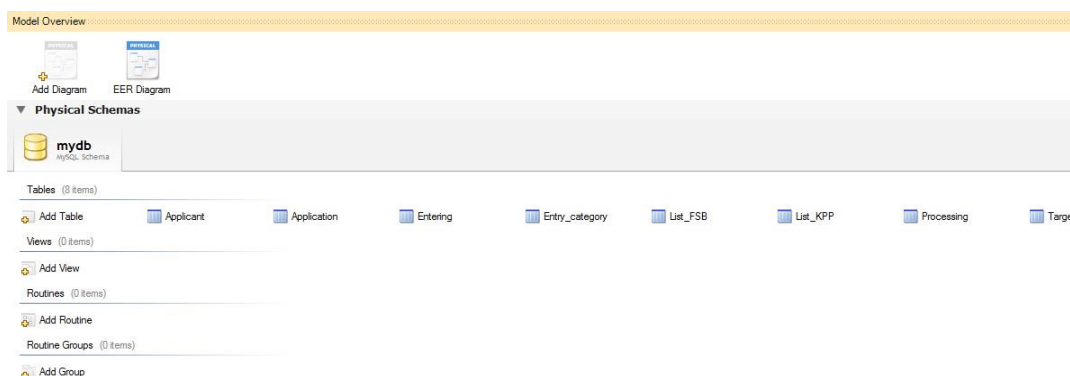


Рисунок 10 – Добавление ER диаграммы

3. Далее добавляем таблицы, поля и тип данных в таблицах и указываем связи – рисунок 11.

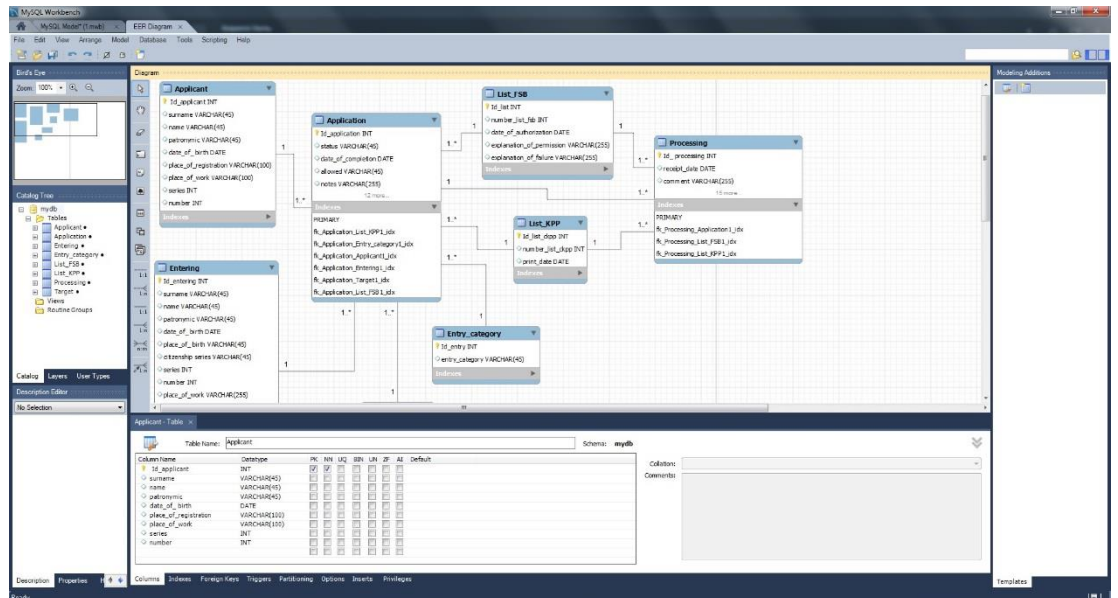


Рисунок 11 – Окно создание ER – модели базы данных

Помимо автоматизированного визуального механизма создания базы данных и таблиц базы, в MySQL Workbench имеется возможность быстрого визуального просмотра и изменения структуры базы данных, а также манипулирования данными.

Необходимый нам программный код на языке SQL получим с помощью средства моделирования баз данных – MySQL Workbench, после преобразования логической модели данных в физическую модель.

### 2.2.3 Физическая модель данных

В физической модели данных проведем преобразования отношений, разработанных на стадии логического моделирования в таблицы, атрибуты в столбцы, домены в типы данных, принятых в СУБД MySQL.

Для преобразования логической модели данных в физическую модель воспользуемся встроенными средствами ПО MySQL Workbench. Физическая модель данных, полученная в результате преобразования в среде MySQL Workbench, представлена на рисунке 12.



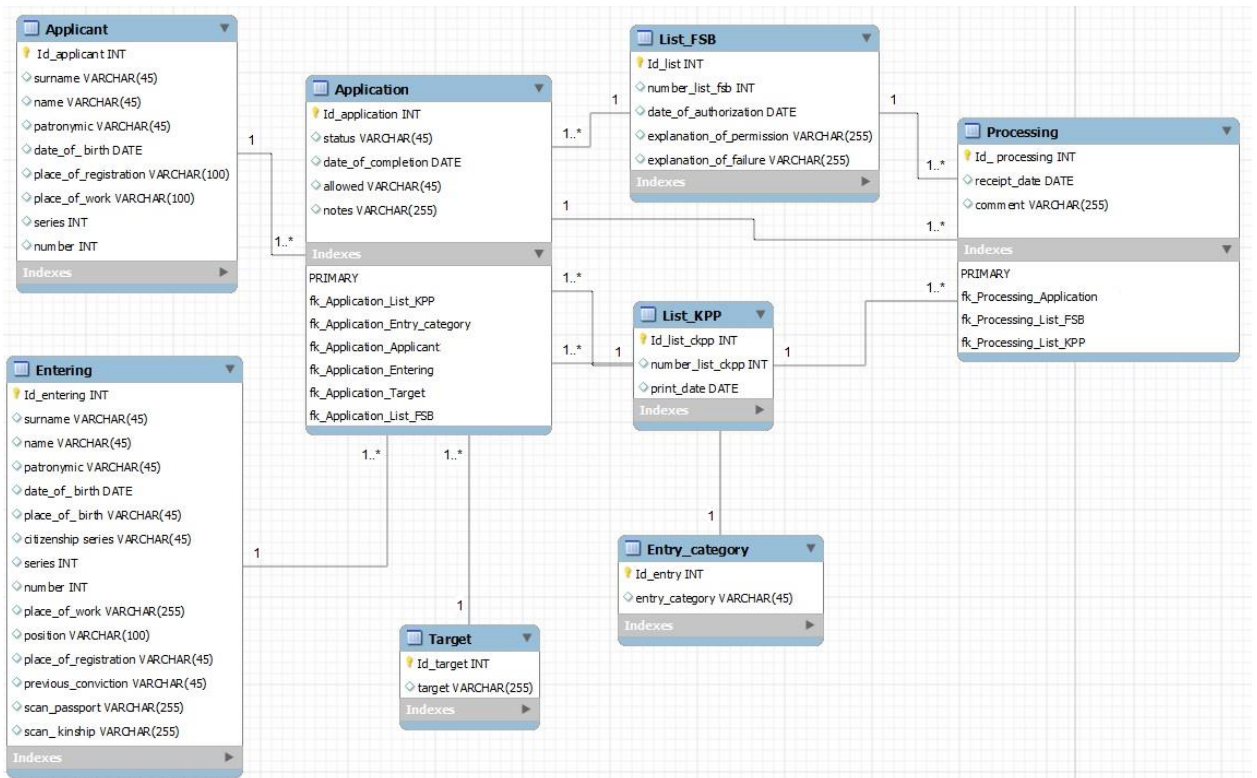


Рисунок 12 – Физическая модель данных в среде MySQL Workbench

После преобразования модели, используем встроенные средства ПО MySQL Workbench для генерации программного кода на языке SQL рисунок 13. Полученный код представлен в Приложении Б. Исполнив данный код в SQL, получим базу данных для нашей информационной системы.

```

1 SET @@OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
2 SET @@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
3 SET @@OLD_SQL_MODE=@@SQL_MODE, SQL_MODE=TRADITIONAL,ALLOW_INVALID_DATES;
4
5 CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
6 USE `mydb`;
7
8 -- Table `mydb`.`Applicant`
9
10
11 CREATE TABLE IF NOT EXISTS `mydb`.`Applicant` (
12   `id_applicant` INT NOT NULL,
13   `surname` VARCHAR(45) NULL,
14   `name` VARCHAR(45) NULL,
15   `patronymic` VARCHAR(45) NULL,
16   `date_of_birth` DATE NULL,
17   `place_of_registration` VARCHAR(100) NULL,
18   `place_of_work` VARCHAR(100) NULL,
19   `series` INT NULL,
20   `number` INT NULL,
21   PRIMARY KEY (`id_applicant`))
22 ENGINE = InnoDB;
23
24
25
26 -- Table `mydb`.`Entering`
27
28
29 CREATE TABLE IF NOT EXISTS `mydb`.`Entering` (
30   `id_entering` INT NOT NULL AUTO_INCREMENT,
31   `surname` VARCHAR(45) NULL,
32   `name` VARCHAR(45) NULL,
33   `patronymic` VARCHAR(45) NULL,
34   `date_of_birth` DATE NULL,
35   `place_of_birth` VARCHAR(45) NULL,
36   `citizenship_series` VARCHAR(45) NULL,
37   `series` INT NULL,
38   `number` INT NULL,
39   `place_of_work` VARCHAR(255) NULL,
40   `position` VARCHAR(100) NULL,
41   `place_of_registration` VARCHAR(45) NULL,
42   `previous_conviction` VARCHAR(45) NULL,
43   `scan_passport` VARCHAR(255) NULL,
44   `scan_kinship` VARCHAR(255) NULL,
45 )

```

Рисунок 13 - Генерация программного кода на языке SQL

## **2.3 Разработка приложения**

После создания базы данных приступим к разработке системы. Данная система должна обеспечить:

- авторизацию через ЕСИА (госуслуги);
- подачу заявлений гражданами в электронном виде;
- формирование списка лиц, подавших заявление на оформление пропусков;
- обмен данными с подразделениями, участвующими в процессе оформления пропусков;
- возможность делать отметку о наличии либо отсутствии оснований для оформления въезда;
- возможность делать отметку о разрешении на въезд;
- возможность сохранения и печати сформированных отчетов;
- оперативное получение необходимой информации с помощью определенных параметров выборки.

Также немало важным аспектом при создании любого программного продукта является наличие эргономичного интерфейса. Таким образом, определив требования к разрабатываемой системе, необходимо выбрать средства, с помощью которых будет реализована разработка нашей защищенной автоматизированной системы.

### **2.3.1 Обоснование выбора средств разработки**

Для реализации системы было принято решение использовать ASP.NET MVC. ASP.NET - это один из самых популярных фреймворков для разработки веб-приложений. Инфраструктура ASP.NET MVC 5 представляет собой последнюю версию веб-платформы ASP.NET от Microsoft. У неё есть

множество преимуществ, которых нет в классической платформе для веб-разработки ASP.NET Web Forms [2].

Встроенные методы представлений генерируют ясный и соответствующий стандартам код разметки (HTML). Таким образом ASP.NET MVC 5 предлагает мощную систему маршрутизации URL, которая позволяет создавать удобочитаемые URL-адреса. Компоненты ASP.NET MVC расширяемы и тестируемы, она обеспечивает расширенную поддержку разработки через тестирование. Огромным преимуществом является также мощная полнофункциональная среда разработки - Microsoft Visual Studio.

Для платформы .NET на сегодняшний день существуют сотни тысяч бесплатных библиотек и компонентов, доступных для скачивания.

ASP.NET MVC вместе с системой VisualStudio.NET является одним из самых совершенных и востребованных средств разработки.

MVC трехсоставная концепция построения веб-сайтов. Аббревиатура MVC происходит от слов Model-View-Controller [3]. MVC распространённый стиль создания сайтов сложной структуры основанная на четком разделении разных составляющих: дизайн, распределение запросов пользователей, программная логика. Каждую из составляющих удобно совершенствовать по отдельности сосредотачиваясь над однородной работой. Схема MVC широко распространена, и применяется в веб-разработках независимо от языка программирования, платформ и операционных систем.

Функции частей MVC:

- Model - программная обработка и обращение к базе данных вызванных запросом пользователя
- View - выводит результат в требуемом виде, т.е. к одной модели можно прикрепить несколько разных представлений
- Controller - распределяет внешние запросы пользователя между моделью и представлением

Концепция MVC используется для многих платформ и языков программирования, не исключение и ASP.NET. В нашем случае мы будем

говорить об MVC на языке C#. C# мощный язык программирования, специально созданный для платформы .NET, являясь языком прикладного уровня, предоставляет огромные возможности по сравнению с другими языками веб-разработок (php, perl, python, ruby). Одно из многих достоинств C# это то, что он играючи справляется с любыми кодировками и символами [5].

В шаблоне MVC технологии ASP.NET в качестве модели выступают «обыкновенные классы» C#, создающие контент и получающие данные из базы данных. Название класса может быть любым, количество классов в модели тоже может быть любым. Модель — это совокупность классов программной логики веб-приложения. Поскольку файлы модели это классы с языком C#, соответственно они имеют расширение \*.cs. Как уже было сказано выше, в модели происходит вся основная логическая работа, через контроллеры в представления передаются уже готовые данные. Некоторую часть программной обработки можно осуществлять и в методах контроллера. Схема взаимодействия MVC приведена на рисунке 14.

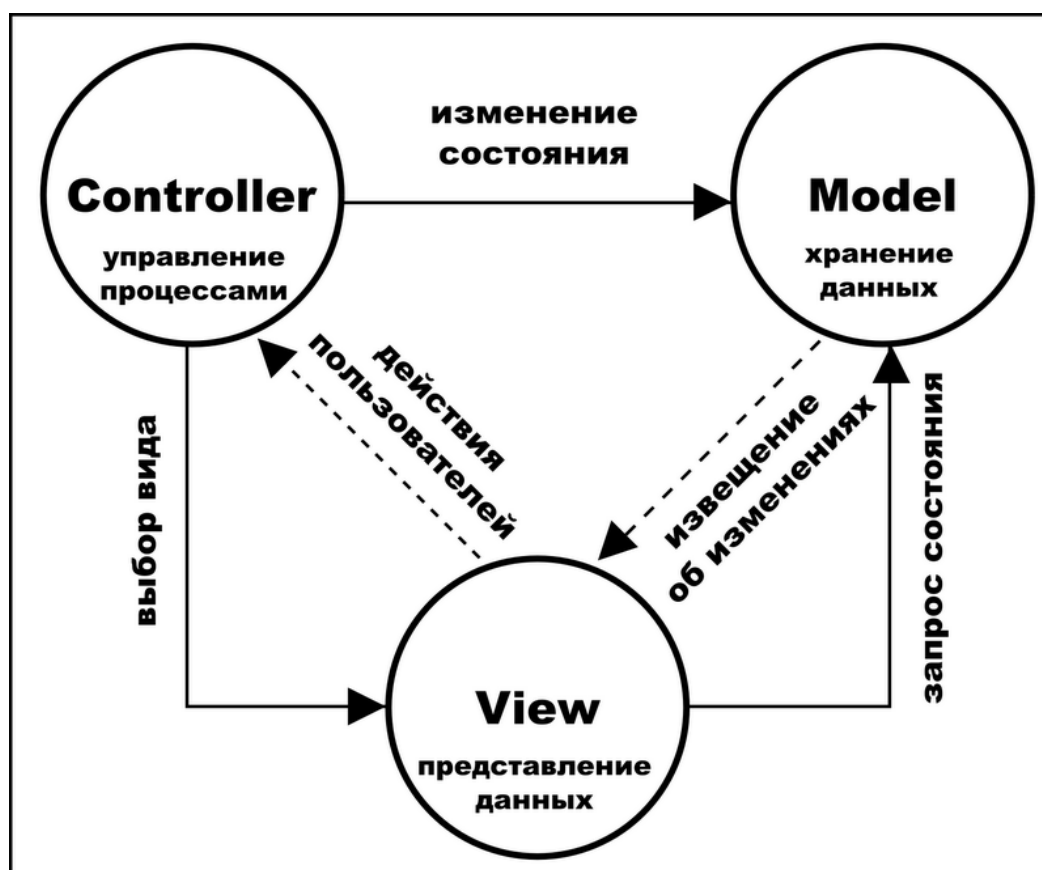


Рисунок 14 – Схема взаимодействия MVC

## 2.3.2 Разработка системы

В Visual Studio 2015 создаем новое ASP.NET приложение – рисунок 15.

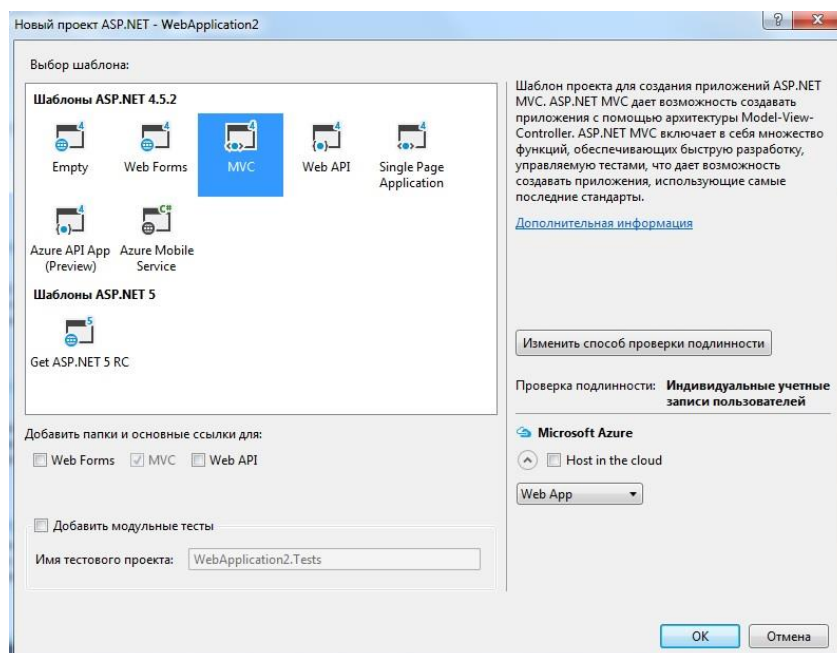


Рисунок 15 - Создание нового ASP.NET приложения

После создания проекта, необходимо подключить нашу базу данных и базу данных «ФИАС». Для начала необходимо соединиться с нашей базой, открываем «Обозреватель серверов» и правой кнопкой мыши на «Подключения данных» выбрать «Добавить подключение...» - рисунок 16

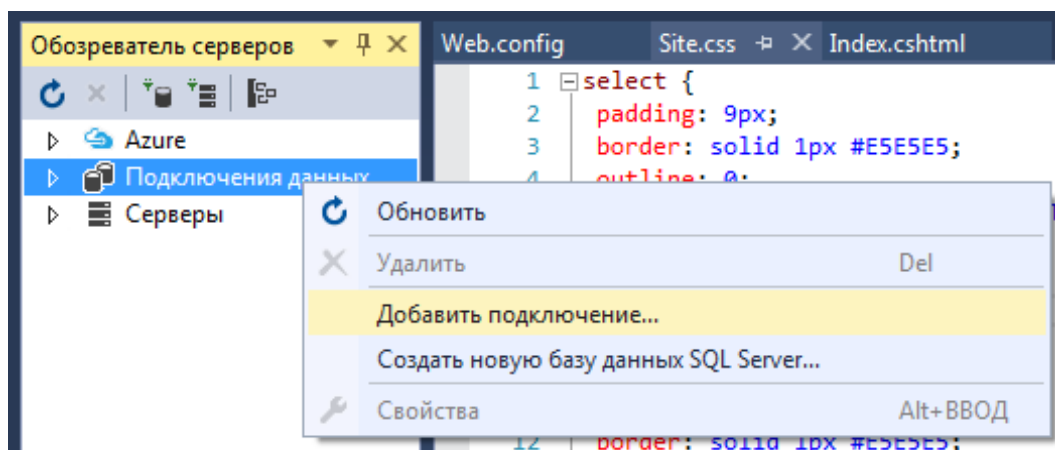


Рисунок 16 – Подключение к БД

Откроется окно подключения к БД рисунок 17.

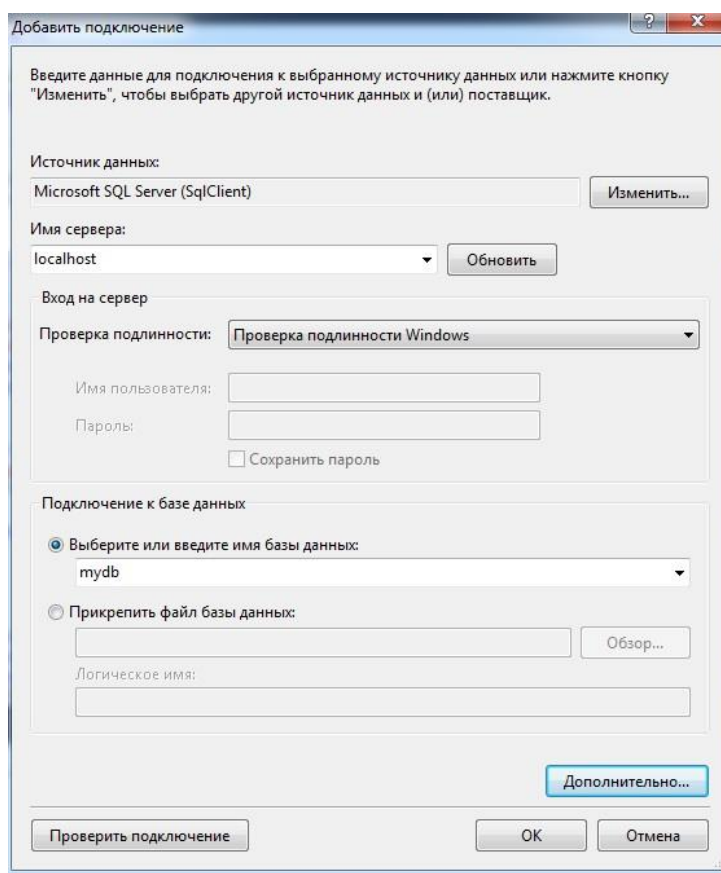


Рисунок 17 – Окно подключения к базе данных

В приложениях ASP.NET конфигурационным файлом является файл Web.config. Для указания строки подключения используется раздел конфигурации connectionString. По умолчанию, строка подключения должна иметь то же имя, что и файл контекста.

```
<connectionStrings>
  <add name="ConnectionString"
    connectionString="Data Source=AdmITSQL;
      Initial Catalog=mydb;Persist Security Info=True;
      User ID=root;Password=12345"
    providerName="System.Data.SqlClient" />
</connectionStrings>
```

где:

*name="ConnectionString"* — имя подключения по которому мы будем обращаться к БД;

*connectionString="Data Source=AdmITSQL* — (AdmIT — имя машины на котором работает SQL, SQL — имя установки SQL сервера);

*Initial Catalog=mydb* — имя нашей БД;

*User ID=root* - пользователь;

*Password=12345* —пароль;

*providerName="System.Data.SqlClient"* — поставщик данных.

По аналогии подключаем базу данных «ФИАС».

Для работы с базой данных удобно пользоваться фреймворком Entity Framework, который позволяет отказаться от написания SQL-запросов. Все сущности принято выделять в отдельные модели. Модели представляют собой классы и располагаются в каталоге Models. Модели описывают логику данных. Модель необязательно состоит только из свойств, кроме того, она может иметь конструктор и вспомогательные методы. Но главное не перегружать класс модели и помнить, что его предназначение - описывать данные. Манипуляции с данными работа контроллера.

Entity Framework поддерживает подход "Code first", который предполагает сохранение или извлечение информации из БД на SQL Server. Entity Framework сам определяет, как и где сохранять объекты этих классов. Для подключения к базе данных через Entity Framework, нужен посредник - контекст данных. Контекст данных представляет собой класс, производный от класса DbContext. Контекст данных содержит одно или несколько свойств типа DbSet<T>, где T представляет тип объекта, хранящегося в базе данных. Создадим контекст данных для работы с моделями:

```
public class DBContext : DbContext
{
    public DbSet<Applicant> applicant { get; set; }
    public DbSet<Entering> entering { get; set; }
    public DbSet<Application> application { get; set; }
    public DbSet<List_FSB> listFSB { get; set; }
    public DbSet<List_KPP> listKPP { get; set; }
    public DbSet<Entry_category> entrycategory { get; set; }
    public DbSet<Target> target { get; set; }
    public DbSet<Processing> processing { get; set; }
}
```

После создания контекста данных, необходимо добавить контроллеры для каждого типах данных в системе. Добавление контроллера изображено на рисунке 18.

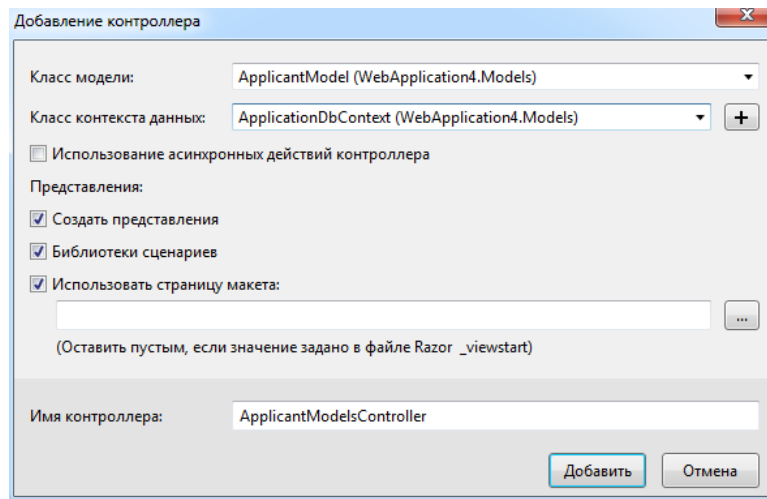


Рисунок 18 – Окно добавление контроллера

Методы добавленных контроллеров представлены в приложении В.

Следующим шагом создаем представления. Именно представления формирует внешний вид нашей системы. В MVC представления - это файлы с расширением cshtml, которые содержат код пользовательского интерфейса в основном на языке html. Хотя представление и содержит код html, оно не является html страницей. При компиляции на основе требуемого представления генерируется класс на языке C#, а затем этот класс компилируется. Для добавления нового представления в проекте выберем папку «Views» в ней нажмем правой кнопкой мыши на подкаталог «Home» и в списке выберем пункт «Добавить представление». Все добавляемые представления, сгруппированы по контроллерам в соответствующие папки в каталоге Views. Представления похоже на обычную веб-страницу html. Однако оно имеет вставки кода на C#, которые обозначаются знаком «@». Данный знак используется движком представлений Razor для перехода к коду на языке C#. При вызове метода View контроллер не генерирует разметку html [6]. Контроллер готовит данные и выбирает представление, которое надо вернуть в качестве объекта ActionResult. Затем объект ActionResult обращается к движку представления для рендеринга представления в выходной результат. Использование Razor позволяет уменьшить синтаксис при вызове кода C#. Razor - это не новый язык, это способ рендеринга представлений, который



имеет свой синтаксис для перехода от разметки html к коду C#. Листинг созданных представлений приведен в приложение В.

Одним из требований к системе стоит необходимость авторизации через ЕСИА. ЕСИА — это Единая Система Идентификации и Аутентификации, которая позволяет получить один пароль для доступа ко всем государственным сайтам. Сейчас ЕСИА Госуслуги позволяет получить доступ к региональным государственным порталам и сайтам электронного правительства. Чтобы получить пароль ЕСИА нужно зарегистрироваться на портале госуслуги. Авторизация через ЕСИА предоставит системе решение по достоверной идентификации пользователей. Достоверность достигается за счет:

- регистрация лица в ЕСИА сопряжена с проверкой значимых для удостоверения личности критериев;
- ЕСИА обеспечивает защиту размещённой в ней информации в соответствии с законодательством Российской Федерации.

Для подключения к ЕСИА необходимо использовать протокол OAuth 2.0 / OpenID Connect.

Сценарий аутентификации пользователя при интеграции по OpenID Connect:

1. Пользователь нажимает на веб-странице системы кнопку «Войти через ЕСИА».
2. Система формирует и отправляет в ЕСИА запрос на аутентификацию и перенаправляет браузер пользователя на страницу предоставления доступа.
3. ЕСИА осуществляет аутентификацию пользователя одним из доступных способов. Если пользователь ещё не зарегистрирован в ЕСИА, то он может перейти к процессу регистрации.

4. Когда пользователь аутентифицирован, ЕСИА сообщает пользователю, что система запрашивает данные о нем в целях проведения идентификации и аутентификации.
5. Если пользователь дает разрешение на проведение аутентификации системой, то ЕСИА выдает системе специальный код.
6. Система формирует в адрес ЕСИА запрос на получение маркера идентификации, вставляя в запрос полученный ранее код.
7. ЕСИА проверяет корректность запроса и кода и передает системе маркер идентификации.
8. Система извлекает идентификатор пользователя из маркера идентификации. Если идентификатор получен, а маркер проверен, то система считает пользователя аутентифицированным. После получения маркера идентификации система использует REST-сервисы ЕСИА для получения дополнительных данных о пользователе.

Схема сценария аутентификации пользователя приведена на рисунке 19.

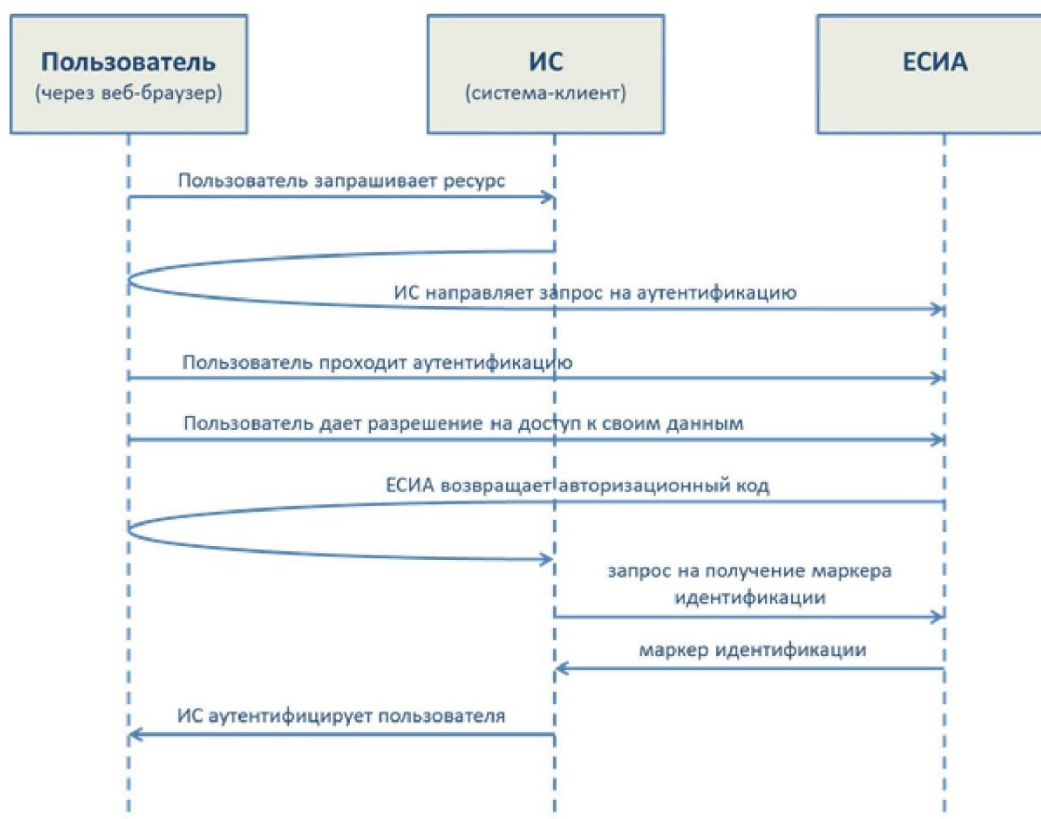


Рисунок 19 – Схема сценария аутентификации пользователя через ЕСИА

Для реализации авторизации через ЕСИА необходимо создать сертификат. Создаем сертификат для генерации подписи с закрытым ключом. Далее устанавливаем сертификат. В ветку «локальное хранилище» импортируем сертификат и копируем в ветку «Доверенные корневые». В ветке «Личные» находим наш сертификат выбираем «Управление закрытыми ключами». Далее добавляем разрешения на группу «Все». После необходимо экспортировать наш сертификат в формате \*.cer и передать в ЕСИА, там его зарегистрируют. Программный код модуля авторизации ЕСИА приведен в приложении Г.

### 3 РЕЗУЛЬТАТЫ ПРОВЕДЕНИЯ РАЗРАБОТКИ

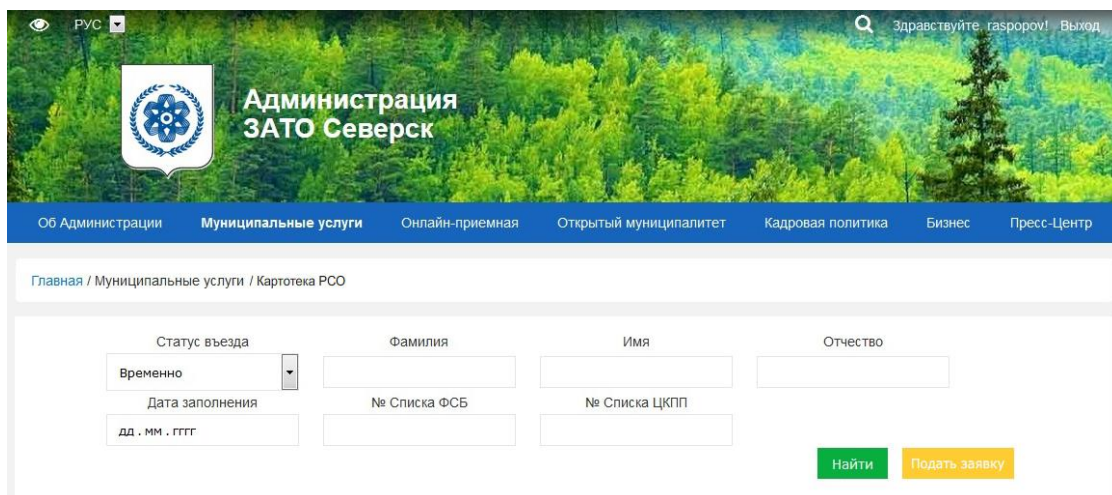
Интерфейс для работы с системой представлен в виде веб-страниц, на которых размещаются функциональные элементы. Для входа в систему необходимо пройти авторизацию через ЕСИА - рисунок 20.

The image shows a web form for logging into the Unified System of Identification and Authentication (ЕСИА). At the top left is the 'госуслуги' logo, and to its right is the text 'Единая система идентификации и аутентификации'. The main heading is 'Вход'. Below this, there are two tabs: 'Телефон или почта' (selected) and 'СНИЛС'. Under the 'Телефон или почта' tab, there is an input field labeled 'Мобильный телефон или почта', another labeled 'Пароль', and a checkbox labeled 'Чужой компьютер'. A blue button labeled 'Войти' is positioned below these fields. Underneath the button is a link that says 'Забыли пароль?'. At the bottom of the form area, there is a link that says 'Зарегистрируйтесь для полного доступа к сервисам' and another link below it that says 'Вход с помощью электронной подписи'.

Рисунок 20 – Окно авторизации через ЕСИА

После успешной авторизации происходит перенаправление на страницу поиска информации о разрешении либо запрете на въезд на территорию ЗАТО Северск по определенным фильтрам. Можно найти информацию по Фамилии, Имени, Отчеству, № Списка ФСБ, № Списка ЦКПП и по дате заполнения.

Страница поиска отображена на рисунке 21.



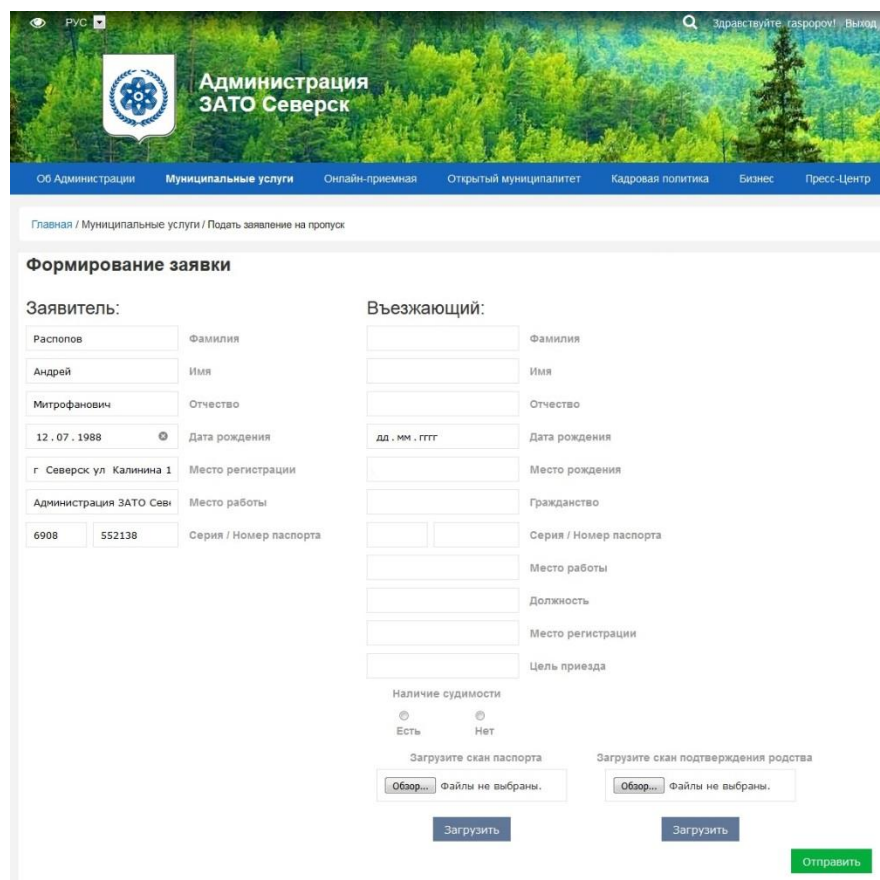
The screenshot shows the search page of the Administration ZATO Severk website. The header features the site logo and navigation menu. The main content area contains a search form with the following fields:

- Статус въезда: Временно
- Дата заполнения: dd . мм . гggg
- Фамилия
- Имя
- Отчество
- № Списка ФСБ
- № Списка ЦКПП

Buttons: Найти, Подать заявку

Рисунок 21 – Страница поиска информации

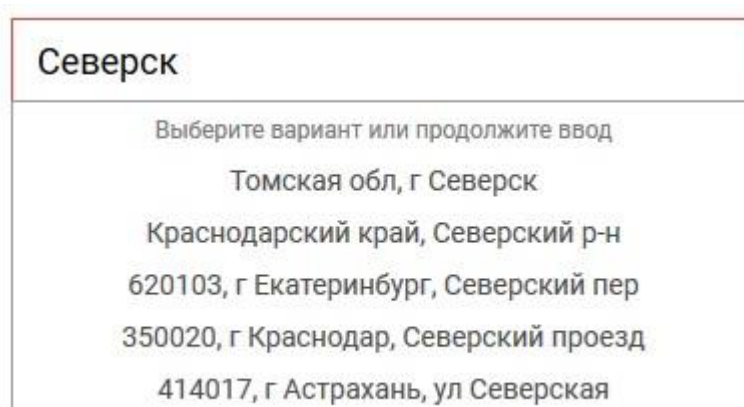
Так же на данной странице располагается кнопка «Подать заявку», при нажатии происходит переход на страницу «Формирования заявки» - рисунок 22.



The screenshot shows the 'Forming application' page. It contains two main sections: 'Заявитель:' and 'Въезжающий:'. The 'Заявитель:' section includes fields for name, date of birth, registration address, and passport details. The 'Въезжающий:' section includes fields for name, date of birth, place of birth, citizenship, passport details, and purpose of visit. There are also radio buttons for 'Наличие судимости' and file upload buttons for 'Загрузите скан паспорта' and 'Загрузите скан подтверждения родства'. A green 'Отправить' button is at the bottom right.

Рисунок 22 – Страница формирования заявки

Необходимо заполнить все поля и нажать кнопку «Отправить», для полей ввода адреса используется авто заполнения адресов из базы данных «ФИАС» рисунок 23.



Северск

Выберите вариант или продолжите ввод

Томская обл, г Северск

Краснодарский край, Северский р-н

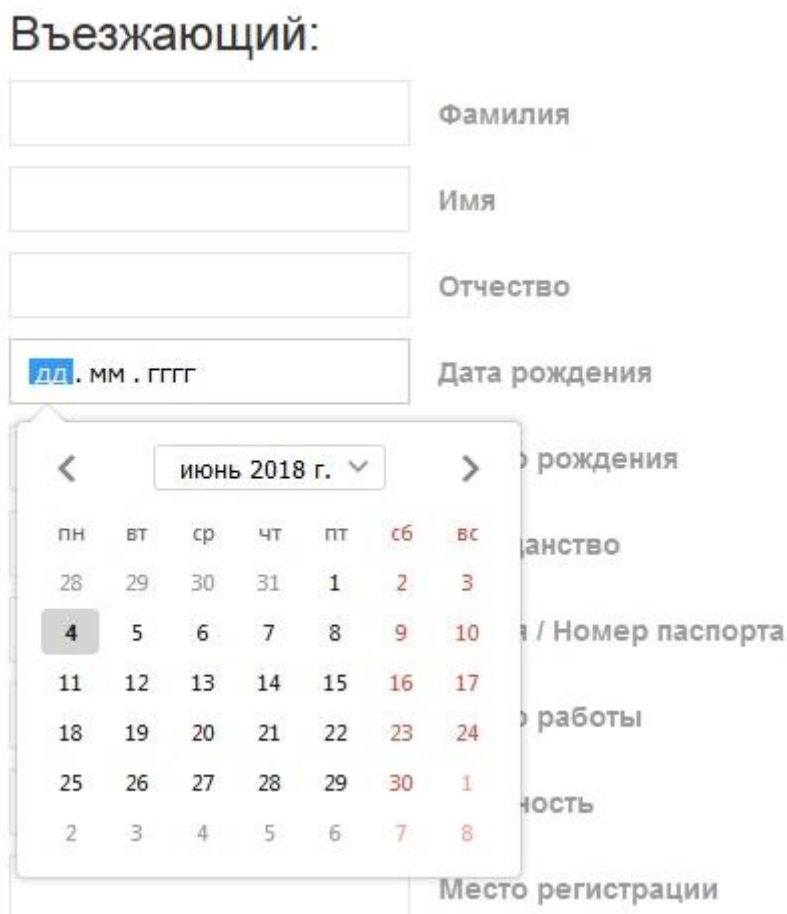
620103, г Екатеринбург, Северский пер

350020, г Краснодар, Северский проезд

414017, г Астрахань, ул Северская

Рисунок 23 – Пример ввода адреса в поле «Место регистрации»

Поля для ввода даты, реализован ввод по календарю, при вводе в поле для даты будет открываться календарь – рисунок 24.



Въезжающий:

Фамилия

Имя

Отчество

Дата рождения

ДД. ММ. ГГГГ

рождения

анство

/ Номер паспорта

работы

ность

Место регистрации

июнь 2018 г.

пн	вт	ср	чт	пт	сб	вс
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Рисунок 24 – Пример ввода даты в поле «Дата рождения»

Для того чтобы загрузить скан паспорта и скан подтверждения родства, необходимо нажать на кнопку «Обзор» и выбрать необходимый скан в формате \*.pdf – рисунок 25.

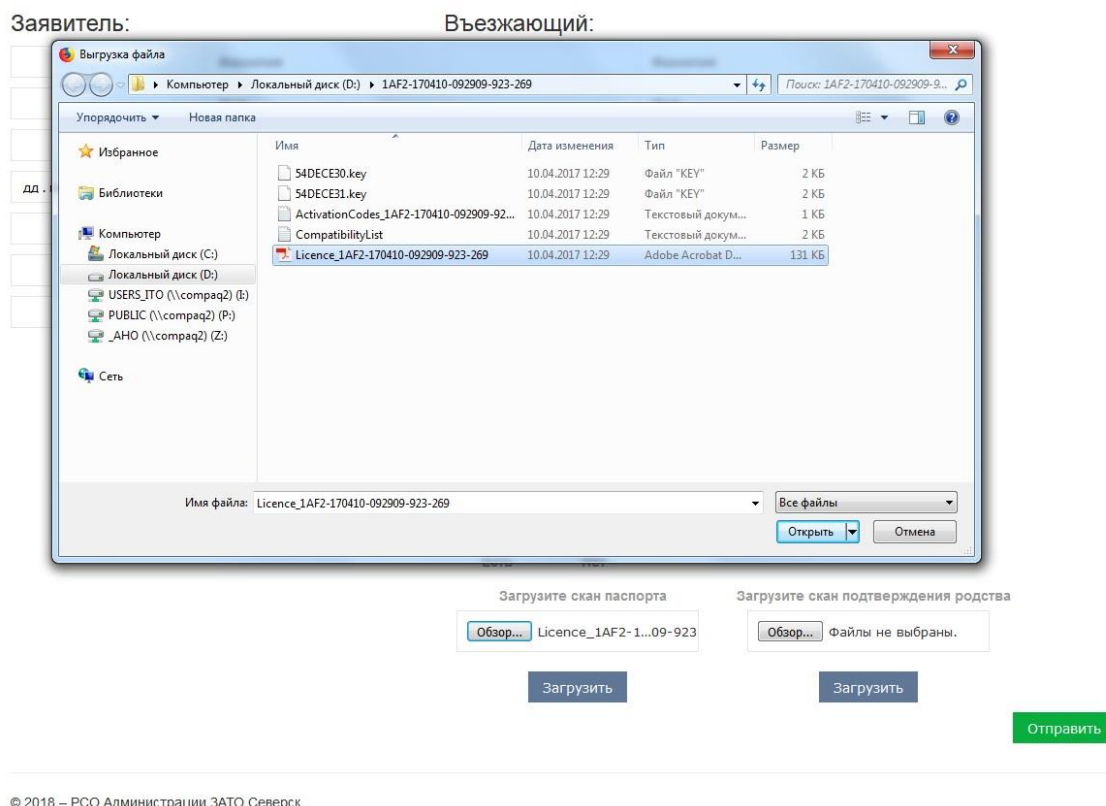


Рисунок 25 – Пример загрузки скана паспорта

В результате поиска выполняется запрос к базе данных с заданным критерием и выводится результат в виде таблицы - рисунок 26, в которой отображаются: Фамилия, Имя, Отчество, Дата рождения, № Списка ФСБ, № Списка ЦКПП, Категория въезда, Статус, Степень родства, Пояснение на разрешение, Пояснение на отказ, Фамилия заявителя, Имя заявителя, Отчество заявителя. Так же имеется возможность выгрузить данный результат в формате Excel (\*.xlsx) нажав на кнопку «Выгрузить».

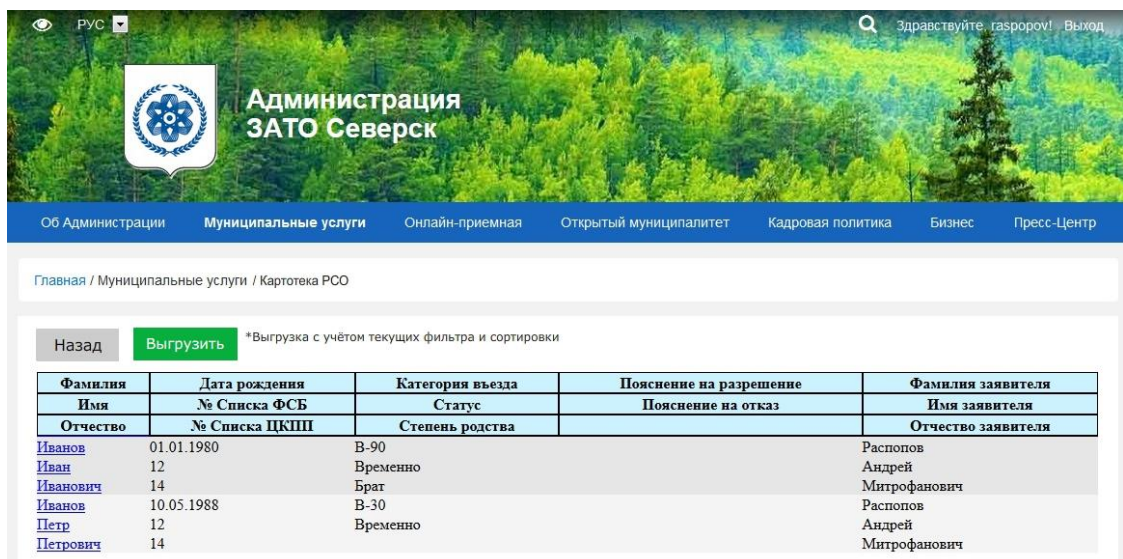


Рисунок 26 – Результат выборки по заданному критерию

В списке сотрудник РСО может выбрать нужного человека, кликнув по фамилии въезжающего и произойдет переход на страницу сформированной заявки с возможностью сверить и в случае необходимости исправить введенные данные о заявителе и въезжающем, а также сделать отметку об разрешении, либо отказе, указать кто разрешил въезд, статус и категорию въезда.

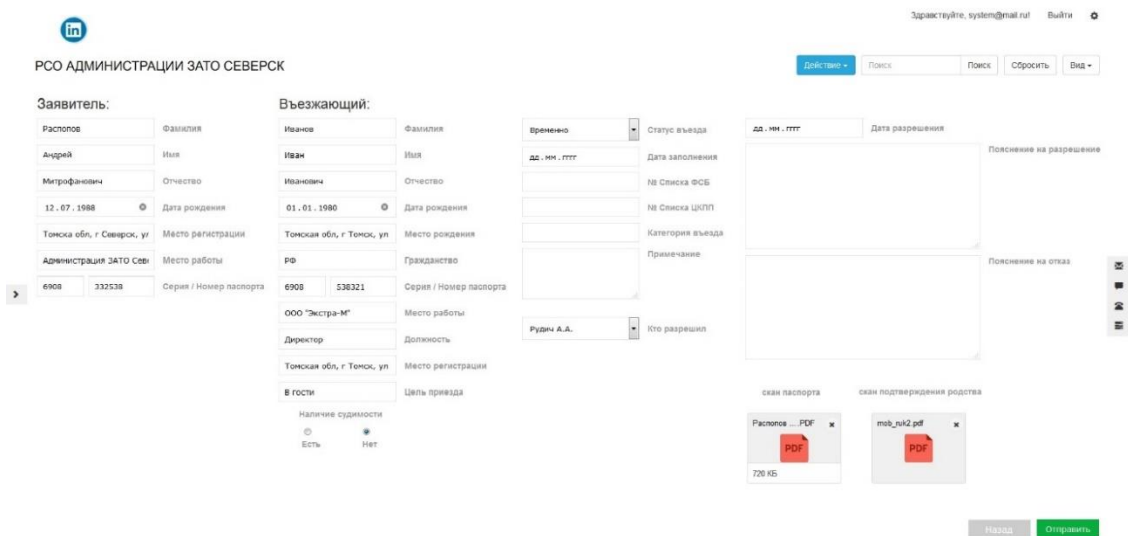


Рисунок 27 – Страница сформированной заявки, доступная только сотрудникам РСО Администрации ЗАТО Северск



## **4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**

### **4.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения**

#### **4.1.1 Потенциальные потребители результатов исследования**

Потенциальным потребителем защищенной автоматизированной системы заказа и оформления пропусков на охраняемую территорию закрытого образования (г. Северск) являются жители города Северска, у которых есть необходимость оформления пропусков для временного въезда родственников на территорию ЗАТО Северск.

#### **4.1.2 Анализ конкурентных технических решений**

Анализ конкурентов проверить невозможно так как аналогов разработки нет либо они являются закрытыми разработками в рамках предприятий.

#### **4.1.3 Технология QuaD**

В соответствии с технологией *QuaD* каждый показатель оценивается по сто балльной шкале, где 1 – наиболее слабая позиция, а 100 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Результат оценки по технологии *QuaD* представлены в таблице 9.

Таблица 9 – Оценочная карта технологии *QuaD*

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	6
<b>Показатели оценки качества разработки</b>					
1. Удобство в эксплуатации (соответствует требованиям потребителей)	0.1	90	100	0.9	0.09
2. Надежность (отказоустойчивость)	0.05	95	100	0.95	0.0475
3. Потребность в ресурсах памяти	0.05	75	100	0.75	0.0375
4. Простота эксплуатации	0.15	95	100	0.95	0.1425
5. Качество интеллектуального интерфейса	0.05	85	100	0.85	0.0425
<b>Показатели оценки коммерческого потенциала разработки</b>					
1. Конкурентоспособность продукта	0.1	70	100	0.7	0.07
2. Уровень проникновения на рынок	0.05	10	100	0.1	0.005
3. Цена	0.2	80	100	0.8	0.16
4. Послепродажное обслуживание	0.1	80	100	0.8	0.08
5. Финансирование научной разработки	0.1	75	100	0.75	0.075
6. Срок выхода на рынок	0.05	35	100	0.35	0.0175
Итого	1				0,7675

Средневзвешенное значение показателя качества и перспективности научной разработки равно 0,7675. Это означает, что разработка является перспективной.

#### 4.1.4 SWOT-анализ

Для исследования внешней и внутренней среды проекта был проведен SWOT-анализ.

Таблица 10 – SWOT-анализ

	<b>Сильные стороны научно-исследовательского проекта:</b> С1. Современные технологии разработки. С2. Уникальность функционала приложения. С3. Близость разработки к завершенности. С4. Интерес к разработанному приложению.	<b>Слабые стороны научно-исследовательского проекта:</b> Сл1. Уровень проникновения на рынок. Сл2. Необходимость в финансировании. Сл3. Трудоемкость разработки.
<b>Возможности:</b> В1. Обеспечение поддержки автоматизированной системы после внедрения. В2. Расширение функциональных возможностей автоматизированной системы. В3. Финансирование проекта коммерческими организациями.	Внедрение дополнительных возможностей повысит удобство эксплуатации. Привлечение финансирования коммерческих организаций за счет уникальности продукта. Расширение автоматизированной системы обеспечит дополнительный функциональные возможности. Повышение эффективности использования.	Уровень проникновения на рынок может увеличить за счет расширения системы, обеспечение поддержки и повышение стоимости конкурентов. Постепенный переход к новым технологиям разработки. Повышение квалификации кадров у потребителя. Сложность в расширении информационной составляющей ресурса за счет необходимости финансирования.
<b>Угрозы:</b> У1. Отсутствие спроса на расширение разработки. У2. Отказ от технической поддержки системы после внедрения. У3. Непопулярность продукта на рынке.	Отсутствие спроса на расширение может снизить цену автоматизированной системы. Отказ от технической поддержки. Появление конкурентов может снизить спрос пользователей на продукт.	Отсутствие спроса понижает уровень проникновения на рынок. Вирусные и хакерские атаки снаружи. Высокий темп развития современных информационных технологий, требующий непрерывного обновления.

Таблица 11 - Интерактивная матрица

Направления развития		C1	C2	C3	C4
	B1	+	-	+	+
	B2	-	-	+	+
	B3	+	+	-	-
Угрозы проекта	У1	-	-	+	+
	У2	-	+	-	-
	У3	-	-	+	+
Сдерживающие факторы		Сл1	Сл2	Сл3	
	B1	-	-	+	
	B2	-	+	-	
	B3	-	-	-	
Уязвимости	У1	-	+	+	
	У2	+	-	-	
	У3	-	+	-	

B1B2C3C4 – развитие функционала интерфейса приложения, а также возможность расширения информационной составляющей ресурса.

B3C1C2 – привлечение финансирования коммерческих организаций за счет уникальности продукта.

B1C1 – оптимизация работы с ресурсом за счет использования современных технологий

B1Сл3 – сложность в оптимизации продукта, за счет трудоемкости разработки.

B2Сл2 – сложность в расширении информационной составляющей ресурса за счет необходимости финансирования.

У1С3С4 – появление конкурентов может снизить спрос пользователей на продукт.

У2С2 – закрытие информационных ресурсов скажется на уникальность продукта на рынке.

У3С3С4 – Непопулярность продукта на рынке приведет к потерям интереса широкой аудитории пользователей.

У2Сл2 – Вирусные и хакерские атаки снаружи.

У3Сл2 – непопулярность продукта на рынке значительно может ограничить финансирование.

Положительные и слабые стороны, которые были обозначены в ходе проведенного анализа, дают возможность спланировать необходимые изменения, слабые стороны необходимо по возможности минимизировать, базируясь прежде всего на имеющихся сильных сторонах.

#### **4.2 Определение возможных альтернатив проведения научных исследований**

Раскрытие возможных вариантов по каждой характеристике. В рамках этого этапа составляется морфологическая матрица. Морфологическая матрица приведена в таблице 12.

Таблица 12 - Морфологическая матрица

	<b>1</b>	<b>2</b>	<b>3</b>
А. Среда разработки	MS Visual Studio 2015	Eclipse	NetBeans
Б. Язык программирования	C#	C#	PHP
В. Тип БД	SQL	MySQL	SQL
Г. Парадигма программирования	Объектно-ориентированное	Объектно-ориентированное	Объектно-ориентированное

Для данной матрицы выберем сочетания *A1B2B1Г3*.

### 4.3 Планирование научно-исследовательских работ

#### 4.3.1 Структура работ в рамках научного исследования

Перечень этапов и работ, распределение исполнителей по данным видам работ приведен в таблице 13.

Таблица 13 - Перечень этапов, работ и распределение исполнителей

<b>Основные этапы</b>	<b>№ раб</b>	<b>Содержание работ</b>	<b>Должность исполнителя</b>
Разработка технического задания	1	Составление и утверждение технического задания.	Руководитель
Выбор направления исследований	2	Подбор материалов по теме	Руководитель
	3	Изучение материалов по теме	Бакалавр
	4	Выбор направления	Руководитель, бакалавр
	5	Календарное планирование работ по теме	Руководитель, бакалавр
Разработка программы	6	Кодирование	Бакалавр
	7	Отладка	Бакалавр
Тестирование программы	8	Разработка, согласование и утверждение и методики испытаний	Руководитель, бакалавр
	9	Проведение испытаний	Бакалавр
Оформление отчета	10	Составление пояснительной записки	Бакалавр

#### 4.3.2 Разработка графика проведения научного исследования

Временные показатели проведения научного исследования представлены в таблице 14.

Таблица 14 – Трудоемкость выполнения работ

Наименование работ	Исполнители (Р - руководитель, Б – бакалавр)	Продолжительность работ, дни									Длительность работ, чел/дн.					
		tmin			tmax			toж			T <sub>р</sub>			T <sub>к</sub>		
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Составление и утверждение технического задания	Р	4	4	4	6	6	6	4,8	4,8	4,8	4,8	4,8	4,8	7	7	7
Подбор материалов по теме	Р	8	8	8	12	12	12	9,6	9,6	9,6	9,6	9,6	9,6	14	14	14
Изучение материалов по теме	Б	14	14	14	20	20	20	16,4	16,4	16,4	16,4	16,4	16,4	24	24	24
Выбор направления исследований	Р	7	7	7	12	12	12	9	9	9	4,5	4,5	4,5	7	7	7
	Б	7	7	7	12	12	12	9	9	9	4,5	4,5	4,5	7	7	7
Календарное планирование работ по теме	Р	4	4	4	10	10	10	6,4	6,4	6,4	3,2	3,2	3,2	5	5	5
	Б	4	4	4	10	10	10	6,4	6,4	6,4	3,2	3,2	3,2	5	5	5
Кодирование	Б	22	20	25	24	23	32	22,8	21,2	27,8	22,8	21,2	27,8	34	31	41
Отладка	Б	20	17	15	23	20	21	21,2	18,2	17,4	21,2	18,2	17,4	31	27	26
Разработка, согласование и утверждение методики испытаний	Р	3	3	3	5	5	5	4	4	4	2	2	2	3	3	3
	Б	3	3	3	5	5	5	4	4	4	2	2	2	3	3	3
Проведение испытаний	Б	4	4	4	7	7	7	5,2	5,2	5,2	5,2	5,2	5,2	8	8	8
Составление пояснительной записки	Б	13	13	13	16	16	16	14,2	14,2	14,2	14,2	14,2	14,2	21	21	21
Итого:	Всего:										113,6	109	114,6	169	162	171
	Руководитель:										24,1	24,1	24,1	36	36	36
	Бакалавр:										89,5	84,9	90,7	133	126	135

На основе таблицы 14 строится календарный план-график. График строится для максимального по длительности исполнения работ в рамках научно-исследовательского. План-график приведен в таблице 15.

Таблица 15 - Календарный план-график

№ работ	Вид работ	Исполнители	$T_{ki}$ , кал. дн.	Продолжительность выполнения работ															
				январь		февраль			март			апрель			май				
				2	3	1	2	3	1	2	3	1	2	3	1	2	3		
1	Составление и утверждение технического задания.	Руководитель	7	■															
2	Подбор материалов по теме.	Руководитель	14		■														
3	Изучение материалов по теме.	Бакалавр	24			▨													
4	Выбор направления.	Руководитель, бакалавр	7					■	▨										
5	Календарное планирование работ по теме	Руководитель, бакалавр	5					■	▨										
6	Кодирование	Бакалавр	41							▨									
7	Отладка	Бакалавр	26										▨						
8	Разработка, согласование и утверждение и методики испытаний	Руководитель, бакалавр	3																■
9	Проведение испытаний	Бакалавр	8																▨
10	Составление пояснительной записки	Бакалавр	21																▨

▨ -бакалавр

■ -руководитель



### 4.3.3 Бюджет научно-технического исследования

#### 4.3.3.1 Расчет материальных затрат научно-технического исследования

Произведем расчет материальных затрат, включающий в себя следующие пункты:

- Стоимость оборудования;
- Стоимость программного обеспечения;
- Стоимость офисных принадлежностей.

Расчет материальных затрат осуществляется по следующей формуле

$$Z_m = (1 + k_T) \sum_{i=1}^m C_i \cdot N_{расxi} \quad (2)$$

где  $m$  – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расxi}$  – количество материальных ресурсов  $i$ -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м<sup>2</sup> и т.д.);

$C_i$  – цена приобретения единицы  $i$ -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м<sup>2</sup> и т.д.);

$k_T$  – коэффициент, учитывающий транспортно-заготовительные расходы.

Расчеты приведены для 3-х вариантов исполнения:

- Исполнение 1 – с использованием Visual Studio;
- Исполнение 2 – с использованием Eclipse;
- Исполнение 3 – с использованием NetBeans.

Материальные затраты представлены в таблице 16.

Таблица 16 - Материальные затраты

Наименование	Ед. изм.	Количество			Цена за ед., руб.			Затраты на материалы, (Зм), руб.		
		И.1	И.2	И.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Ноутбук	шт	1	1	1	27500	27500	27500	27500	27500	27500
Среда Разработки	шт	1	1	1	2900	3100	3150	2900	3100	3150
Картридж для принтера, бумага, папка для бумаги	шт.	1	1	1	650	650	650	650	650	650
Итого:								31050	31250	31300

Из расчета материальных затрат видно, что Исполнение 1 имеет наименьшую величину затрат 31050 рублей, Исполнение 2 - 31250 рублей, Исполнение 3 – 31300 рублей.

#### 4.3.3.2 Основная заработная плата

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Количество рабочих дней в месяце примем равным 20-ти. Оклад руководителя составляет 33664 руб/мес. Оклад бакалавра составляет 10745 руб/мес. Для расчета заработной платы умножаем оклад на районный коэффициент (1,3 для Томска).

Месячный должностной оклад работника рассчитывается по формуле:

$$З_m = З_{тс} * k_p$$

где  $k_p$  - районный коэффициент, равный 1,3 (для Томска).

$$З_m = 33664 * 1,3 = 43763,2$$

$$З_m = 10745 * 1,3 = 13968,5$$

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = З_m * M / F_d$$

где M – количество месяцев работы без отпуска в течение года,

$F_d$  - действительный годовой фонд рабочего времени научно- технического персонала.

$$Z_{\text{дн}} = 43763,2 * 12 / 365 = 1438,8$$

$$Z_{\text{дн}} = 13968,5 * 12 / 365 = 459,3$$

Основная заработная плата рассчитывается по формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} * T_p$$

где  $T_p$  - продолжительность работ, выполняемых научно-техническим работником, раб. дн. (таблица. б);

$$Z_{\text{осн}} = 1438,8 * 24,1 = 34675,08$$

$$Z_{\text{осн}} = 459,3 * 89,5 = 41107,35$$

$$Z_{\text{осн}} = 459,3 * 84,9 = 38994,57$$

$$Z_{\text{осн}} = 459,3 * 90,7 = 41658,51$$

Расчет основной заработной платы сводится в таблице 17.

Таблица 17 – Затраты на основную заработную плату

Исполнитель	Оклад с районным коэф., руб.	Средняя заработная плата, руб./дн.	Трудоемкость, раб. дн.			Основная заработная плата, руб.		
			Исп.1	Исп.2	Исп.3	Исп. 1	Исп. 2	Исп. 3
Руководитель	43763,2	1438,8	24,1	24,1	24,1	34675,08	34675,08	34675,08
Бакалавр	13968,5	459,3	89,5	84,9	90,7	41107,35	38994,57	41658,51
Итого:						75782,43	73669,65	76333,59

В результате проведенных расчетов, основная заработная плата в исполнение 1 равна 75782,43 рублей, в исполнение 2 равна 73669,65 рублей, в исполнение 3 равна 76333,59 рублей.

#### 4.3.3.3 Дополнительная заработная плата

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} * Z_{\text{осн}}$$

где  $k_{\text{доп}}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).

Расчет дополнительной заработной платы руководителя:

$$Z_{\text{доп}} (\text{Исп.1}) (\text{Исп.2}) (\text{Исп.3}) = 34675,08 * 0.15 = 5201,262$$

Расчет дополнительной заработной платы бакалавра:

$$Z_{\text{доп}} (\text{Исп.1}) = 41107,35 * 0.15 = 6166,1025$$

$$Z_{\text{доп}} (\text{Исп.2}) = 38994,57 * 0.15 = 5849,1855$$

$$Z_{\text{доп}} (\text{Исп.3}) = 41658,51 * 0.15 = 6248,7765$$

Расчет дополнительной заработной платы сводится в таблице 18.

Таблица 18 – Расчет дополнительной заработной платы

Исполнитель	Основная заработная плата, руб.			Коэффициент дополнительной заработной платы	Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3		Исп.1	Исп.2	Исп.3
Руководитель	34675,08	34675,08	34675,08	0,15	5201,262	5201,262	5201,262
Бакалавр	41107,35	38994,57	41658,51		6166,1025	5849,1855	6248,7765
Итого:					11367,4	11050,4	11450,1

В результате проведенных расчетов, дополнительная заработная плата в исполнение 1 равна 11367,4 рублей, в исполнение 2 равна 11050,4 рублей, в исполнение 3 равна 11450,1 рублей.

#### 4.3.3.4 Отчисления во внебюджетные фонды

Отчисления во внебюджетные фонды (страховые отчисления) отражают обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}})$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

На 2014 г. в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%.

Величина отчислений во внебюджетные фонды у руководителя:

$$Z_{\text{внеб}} = 0,302 * (34675,08 + 5201,262) = 12042,66$$

Величина отчислений во внебюджетные фонды у бакалавра:

$$Z_{\text{внеб}} = 0,302 * (41107,35 + 6166,1025) = 14276,58$$

$$Z_{\text{внеб}} = 0,302 * (38994,57 + 5849,1855) = 13542,82$$

$$Z_{\text{внеб}} = 0,302 * (41658,51 + 6248,7765) = 14468,05$$

Отчисления во внебюджетные фонды представлены в таблице 19.

Таблица 19 - Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	34675,08	34675,08	34675,08	5201,262	5201,262	5201,262
Бакалавр	41107,35	38994,57	41658,51	6166,1025	5849,1855	6248,7765
Коэф. отчислений во внебюджетные фонды	0,302					
<b>Итого</b>						
Исполнение 1	26319,24					
Исполнение 2	25585,48					
Исполнение 3	26510,71					

В результате вычислений получили отчисления во внебюджетные фонды в исполнение 1 - 26319,24 рублей, в исполнение 2 – 25585,48 рублей, в исполнение 3 – 26510,71 рублей.

#### 4.3.3.5 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергия, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 4) \cdot k_{\text{нр}},$$

где  $k_{\text{нр}}$  – коэффициент, учитывающий накладные расходы (16%).

$$Z_{\text{накл1}} = 26319,24 * 0,16 = 4211,07$$

$$Z_{\text{накл2}} = 25585,48 * 0,16 = 4093,68$$

$$Z_{\text{накл3}} = 26510,71 * 0,16 = 4241,72$$

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 20.

Таблица 20 - Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.			Примечание
	Исп.1	Исп.2	Исп.3	
1 Материальные затраты НИИ	31050	31250	31300	Пункт 4.3.3.1
2 Затраты по основной заработной плате исполнителей темы	75782,43	73669,65	76333,59	Пункт 4.3.3.2
3 Затраты по дополнительной заработной плате исполнителей темы	11367,4	11050,4	11450,1	Пункт 4.3.3.3
4 Отчисления во внебюджетные фонды	26319,24	25585,48	26510,71	Пункт 4.3.3.4
5 Накладные расходы	4211,07	4093,68	4241,72	16 % от суммы ст. 1-4
6 Бюджет затрат НИИ	148730,14	145649,21	149836,12	Сумма ст. 1-6

В ходе определения бюджета затрат на научно-исследовательский проект в Исполнение 1 бюджет затрат составил 148730,14 рублей, Исполнение 2 – 145649,21 рублей, Исполнение 3 – 149836,12 рублей.

#### 4.4 Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}},$$

где  $I_{\text{финр}}^{\text{исп.}i}$  – интегральный финансовый показатель разработки;

$\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта.

Исполнение 1:  $I_{\text{финр}} = 148730,14 / 149836,12 = 0,9$

Исполнение 2:  $I_{\text{финр}} = 145649,21 / 149836,12 = 0,9$

Исполнение 3:  $I_{\text{финр}} = 149836,12 / 149836,12 = 1$

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в разгах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в разгах (значение меньше единицы, но больше нуля).

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i,$$

где  $I_{pi}$  – интегральный показатель ресурсоэффективности для  $i$ -го варианта исполнения разработки;

$a_i$  – весовой коэффициент  $i$ -го варианта исполнения разработки;

$b_i^a, b_i^p$  – балльная оценка  $i$ -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

$n$  – число параметров сравнения.

Расчет интегрального показателя приведен в таблице 21.

Таблица 21 - Сравнительная оценка характеристик вариантов исполнения проекта

Критерии \ Объект исследования	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
Способствует росту производительности труда пользователя	0.1	5	3	4
Удобство в эксплуатации (соответствует требованиям потребителей)	0.15	4	2	3
Помехоустойчивость	0.15	5	3	3
Энергосбережение	0.20	4	3	3
Надежность	0.25	4	4	4
Материалоемкость	0.15	4	4	4
ИТОГО:		1		

$$I_{p-исп1} = 5*0.1 + 4*0.15 + 5*0.15 + 4*0.20 + 4*0.25 + 4*0.15 =$$

$$= 0.5 + 0.6 + 0.75 + 0.8 + 1 + 0.6 = 4.25;$$

$$I_{p-исп2} = 3*0.1 + 2*0.15 + 3*0.15 + 3*0.20 + 4*0.25 + 4*0.15 =$$

$$= 0.3 + 0.3 + 0.45 + 0.6 + 1 + 0.6 = 3.25;$$

$$I_{p-исп3} = 4*0.1 + 3*0.15 + 3*0.15 + 3*0.20 + 4*0.25 + 4*0.15 =$$

$$= 0.4 + 0.45 + 0.45 + 0.6 + 1 + 0.6 = 3.5.$$

Интегральный показатель эффективности вариантов исполнения разработки ( $I_{исп.i}$ ) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.1} = \frac{I_{p-исп1}}{I_{финр}}$$

где  $I_{p-исп1}$  - интегрального показателя ресурсоэффективности.

$$I_{исп1} = 4,25 / 0,9 = 4,7$$

$$I_{исп2} = 3,25 / 0,9 = 3,6$$

$$I_{исп3} = 3,5 / 1 = 3,5$$

Сравнительная эффективность проекта:



$$\mathcal{E}_{cp} = \frac{I_{ucn.i}}{I_{ucn.min}}$$

где  $I_{ucn1}$  - интегральный показатель эффективности вариантов исполнения разработки 1;

$I_{ucn2}$  - интегральный показатель эффективности вариантов исполнения разработки 2.

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта. Сравнительная эффективность разработки представлена в таблице 22.

Таблица 22 - Сравнительная эффективность разработки

№ п/п	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,9	0,9	1
2	Интегральный показатель ресурсоэффективности разработки	4,7	3,25	3,5
3	Интегральный показатель эффективности	4,7	3,6	3,8
4	Сравнительная эффективность вариантов исполнения	1,3	1	1,05

Вывод:

В ходе оценки перспективности и альтернатив проведения научного исследования с позиции ресурсоэффективности и ресурсосбережения коммерческого потенциала для выпускной квалификационной работы были определены потенциальные потребители.

Был сделан анализ конкурентных решений, выявлены сильные и слабые стороны разработанного решения, его возможности и угрозы, положительные и слабые стороны, которые были обозначены в ходе проведенного анализа, дали возможность спланировать необходимые изменения, слабые стороны были минимизированы, базируясь прежде всего на имеющихся сильных сторонах, определенных в ходе SWOT-анализа.

В ходе составления структуры работ в рамках научного исследования был составлен перечень этапов и работ, и проведено распределение исполнителей по видам работ.

Из расчета материальных затрат видно, что выбранный вариант исполнения имеет наименьшую величину затрат - 31050 рублей. Отчисления во внебюджетные фонды - 26319,24 рублей

В ходе определение бюджета затрат на научно-исследовательский проект бюджет затрат составил 148730,14 рублей.

Исходя из полученных данных и проведенного анализа эффективности, можно сделать вывод, что выбранный вариант исполнения является наиболее эффективным за счет своих возможностей, что делает его разработку целесообразной, несмотря на то, что выбранное решение достаточно дорогое.

## **5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

Имеющийся в настоящее время комплекс разработанных организационных мероприятий и технических средств защиты показывает, что имеется возможность добиться значительно больших успехов в деле устранения воздействия на работников опасных и вредных производственных факторов. Операторы ЭВМ и другие работники сталкиваются с воздействием таких вредных производственных факторов, как повышенный уровень шума, повышенная температура внешней среды, отсутствие или недостаточная освещенность рабочей зоны, электрический ток, статическое электричество и другие.

Оборудованное рабочее место, состоящее из персонального компьютера и МФУ, потребляющих напряжение 220 вольт переменного тока и используемый монитор типа LED с низким уровнем излучения, находится в отделе информатизации и защиты данных в здании Администрации ЗАТО Северск.

### **5.1 Описание рабочего места**

Рабочее место – это зона, где сотрудник организации выполняет возложенные на него трудовые функции. Рабочее место должно быть оснащено всеми необходимыми предметами для возможности ведения рабочего процесса в рамках должностных обязанностей.

Правила и порядок безопасной работы с ЭВМ описаны санитарными правилами и нормами СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» [10].

План рабочего кабинета и рабочего места представлены на рисунке 28.

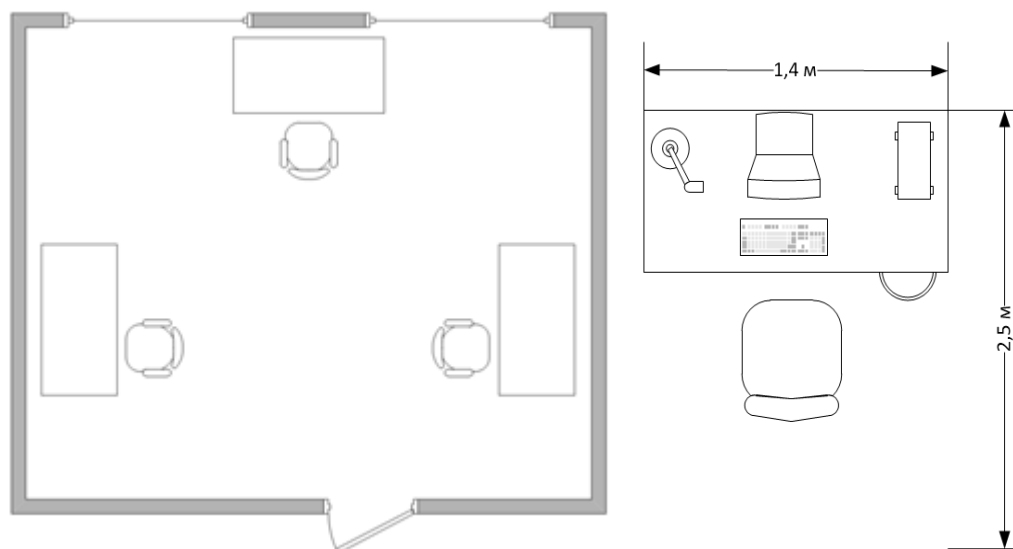


Рисунок 28 – План рабочего кабинета и рабочего места

Рабочий кабинет, располагается на третьем этаже здания. Кабинет представляет собой комнату длиной – 6 м, шириной – 5 м и высотой – 3 м. Естественное освещение кабинета осуществляется посредством двух окон размерами 2 м x 1 м каждое. Дверь – деревянная. Высота двери – 2 м, ширина - 1 м. Стены комнаты окрашены водоэмульсионной краской. Цвет стен – бежевый. Потолок подвесной открытого кассетного типа со встроенными светильниками. Пол покрывает линолеум бежевого цвета. Площадь кабинета составляет 30 м<sup>2</sup>, объем – 90 м<sup>3</sup>.

Помещение оборудовано на три рабочих места. Пункт 3.4 СанПиН 2.2.2/2.4.1340-03. определяет требования к минимальной площади и объему на одно рабочее место. Согласно СанПиН 2.2.2/2.4.1340-03 при периметральном расположении рабочих мест площадь одного рабочего места должна составлять не менее 4,0 кв.м. Для данного помещения эти требования выполнены. Согласно п.9.1 СанПиН 2.2.2/2.4.1340-03, расстояние между столами с видеомониторами должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов должно быть не менее 1,2 м. Расстояние между столами 3 м, между боковыми поверхностями видеомониторов 1,5 м. Требования пункта 9.1 СанПиН 2.2.2/2.4.1340-03 выполняются.

## 5.2 Вредные факторы

### 5.2.1 Освещение

Освещение рабочего места – является важнейшим фактором создания нормальных условий труда. В производственных помещениях используется три вида освещения: естественное (источником его является солнце), искусственное (когда используются только искусственные источники света), совмещенное (характеризуется одновременным сочетанием естественного и искусственного освещения).

Естественное освещение должно осуществляться через боковые светопроемы ориентированные преимущественно на север и северо-восток. Величина коэффициента естественной освещенности (КЕО) должна соответствовать нормативным уровням по СНиП 23-05-95 "Естественное и искусственное освещение" и создавать КЕО не ниже 1,2% в зонах с устойчивым снежным покровом и не ниже 1,5% на остальной территории.

Искусственное освещение следует осуществлять в виде системы комбинированного освещения. В качестве источников света применяются люминесцентные лампы типа ЛБ мощностью 20 Вт.

Схема расположения светильников представлена на рисунке 29.

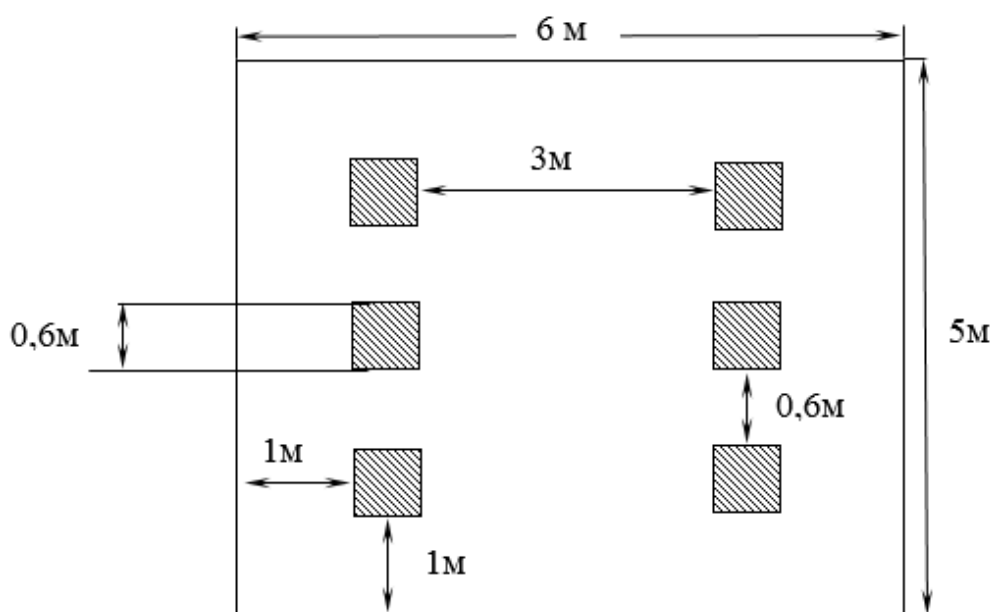


Рисунок 29 – Схема освещения

В помещении установлены светильники типа ARS418, укомплектованные четырьмя люминесцентными лампами мощностью 20 Вт. Геометрические размеры светильников 595x595x36 мм,  $\lambda = 1,4$ . Учитывая, что в каждом светильнике установлено по 4 лампы, количество ламп составит:

$$N = 24$$

Согласно СНиП 23-05-95 [12] норма освещённости рабочих поверхностей в помещениях для работы с дисплеями и видеотерминалами принимается:

$$E_n = 200 \text{лк.}$$

Фактическая освещённость определяется по формуле:

$$E_{\phi} = (N \cdot n \cdot \Phi_{ст} \cdot \eta) / (S \cdot K_z \cdot Z), \quad (1)$$

где  $S$  – площадь освещаемого помещения, м<sup>2</sup>;

$K_z$  – коэффициент запаса, учитывающий загрязнение светильника;

$\Phi_{ст}$  – световой поток люминесцентных ламп, лм;

$Z$  – коэффициент неравномерности освещения, отношение  $E_{ср.}/E_{min}$ .

$N$  – число светильников;

$n$  – число ламп в светильнике;

$\eta$  – коэффициент использования светового потока, %.

Для определения коэффициента использования светового потока необходимо найти индекс помещения.

Индекс помещения определяется по формуле:

$$i = S/h \cdot (A+B), \quad (2)$$

где  $h$  – высота помещения;

$S$  – площадь помещения;

$A$  – ширина помещения;

$B$  – длина помещения.

Длина помещения  $B$  составляет 6 метров, ширина  $A$  – 5 м, высота  $h$  – 3 м.

Находим индекс помещения по формуле 2:

$$i = 30/3 \cdot (6+5) = 0,91$$

Зная индекс помещения  $i$ , коэффициент отражения светлых стен  $\rho_c = 50\%$  и светлого потолка  $\rho_n = 70\%$ , определим коэффициент использования светового потока из таблицы взятой из СНиП 23-05-95,  $\eta = 0,49$ .

Коэффициент неравномерности освещения  $Z$  принимается в пределах 1.1-1.2, в данной работе примем  $Z = 1.1$ . Коэффициент запаса определяется по таблице из СНиП 23-05-95. в зависимости от характеристик помещения, в нашем случае  $K = 1.5$ . Световой поток люминесцентных ламп типа ЛБ с мощностью 20 Вт согласно таблицы из СНиП 23-05-95 составляет 1060 лк.

Фактическая освещенность:

$$E\phi = \frac{6*4*1060*0,49}{30*1,5*1,1} = 251,83 \text{ лк.}$$

Отклонение от нормы  $\Delta E$  рассчитываются по формуле:

$$\Delta E = ((E\phi - E_n) / E_n) * 100 \quad (3)$$

$$\Delta E = ((251,83 - 200) / 200) * 100 = 25.9\%$$

Фактическое значение освещенности выше нормативного. Установка местного освещения не требуется. Так как значение выше нормативного, для комфортного освещения, реализована возможность включения не всех светильников одновременно.

### 5.2.2 Шум

Шум - звук, мешающий разговорной речи и негативно влияющий на здоровье человека.

В помещениях при работе на ПЭВМ источниками шума, повышенного уровня являются принтер, системный блок, устройства кондиционирования и вентиляции, находящийся в помещении персонал. Уровень шума в помещении на рабочих местах пользователей ПЭВМ не должен превышать значений, установленных санитарных нормы 2.2.4/2.1.8.562-96 и ГОСТ 12.1.003-83 [9]. Допустимые уровни звукового давления и уровни звука на рабочих местах

связанных с творческой или научной деятельностью, программированием и преподаванием, согласно СН 2.2.4/2.1.8.562-96, приведены в таблице 23.

Таблица 23 - Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни
31,5	63	125	250	500	1000	2000	4000	8000	звукa (в дБА)
86	71	61	54	49	45	42	40	38	50

При выполнении основной работы на ПЭВМ уровень шума на рабочем месте не превышает 50 дБА.

### 5.2.3 Микроклимат

Микроклимат производственных помещений – климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также интенсивности теплового излучения от нагретых поверхностей.

С целью создания нормальных условий труда для персонала существуют требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе помещений эксплуатации ПЭВМ определены в документах ГОСТ 12.1.005-88 ССБТ. «Общие санитарно-гигиенические требования к воздуху рабочей зоны» и СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» [15].

Микроклимат определяют следующие параметры:

- температура воздуха;
- температура поверхностей;
- относительная влажность воздуха;
- скорость движения воздуха.



Так как разработка программных продуктов относится к работам, производимым сидя и сопровождающиеся незначительным физическим напряжением, то данный вид деятельности относят к категории работ *Ia*. Параметры микроклимата холодного и теплого периода года для категории *Ia* представлены в таблице 24.

Таблица 24 - Параметры микроклимата для помещений.

Период года	Категория работ	Температура, °С				Относительная влажность		Скорость движения, м/с		
		оптимальная	допустимая				оптимальная	допустимая на рабочих местах	оптимальная, не более	допустимая на рабочих местах постоянных и непостоянных
			верхняя граница		нижняя граница					
			на рабочих местах							
постоянных	непостоянных	постоянных	непостоянных							
Холодный	Легкая – <i>I a</i>	22- 24	25	26	21	18	40-60	75	0,1	Не более 0,1
Теплый	Легкая – <i>I a</i>	23- 25	28	30	22	20	40-60	55 (при 28 °С)	0,1	0,1-0,2

Микроклимат оказывает огромное влияние на функциональную деятельность человека, его здоровье и на надёжность работы средств вычислительной техники.

При обеспечении оптимальных показателей микроклимата, температура внутренних поверхностей конструкций, ограждающих рабочую зону (стен, пола, потолка), или устройств, а также температура наружных поверхностей технологического оборудования или ограждающих его устройств не должны выходить более чем на 2 °С за пределы оптимальных величин температуры воздуха, установленных в таблице 2. При температуре поверхностей ограждающих конструкций ниже или выше оптимальных величин температуры воздух на рабочие места должны быть удалены от них на расстояние менее 1м. Температура воздуха в рабочей зоне, измеренная на разной высоте и в различных участках помещений, не должна выходить в течение смены за пределы оптимальных величин, указанных в таблице 24.

Поддержание параметров микроклимата в заданных пределах осуществляется естественной вентиляцией в летнее время года и при помощи систем отопления в зимний период года. Для повышения влажности воздуха в помещениях с ПЭВМ следует применять увлажнители воздуха, заправляемые ежедневно дистиллированной водой. Помещения с ПЭВМ перед началом работы и после окончания, должны быть проветрены, что обеспечивает улучшение качественного состава воздуха.

Для поддержания требуемых параметров микроклимата в кабинете установлен кондиционер. Кондиционер – это автоматизированная вентиляционная установка, поддерживающая в помещении заданные параметры микроклимата. Кондиционированием воздуха называется создание и автоматическое поддержание в помещениях независимо от внешних метеорологических условий постоянных или изменяющихся по определённой программе температуры, влажности, чистоты и скорости движения воздуха, сочетания которых создаёт комфортные условия труда или требуется для нормального протекания технологического процесса.

Для поддержания заданной температуры воздуха в Администрации ЗАТО Северск в холодное время года используют водяную систему отопления. В системах водяного отопления в качестве теплоносителя используется вода. Такие системы отопления наиболее эффективны в санитарно-гигиеническом отношении.

#### **5.2.4 Опасные факторы. Опасность поражения электрическим током**

Электрические установки, к которым относятся оборудования ЭВМ, представляет потенциальную опасность для человека.

В рабочих кабинетах выполняются такие защитные меры как защитное заземление, защитное отключение, электрическая изоляция токоведущих частей, малое напряжение. Защитные меры должны обеспечивать напряжение

прикосновения не выше 42 В – в помещении с повышенной опасностью. Рабочее место относится к помещениям без повышенной опасности, согласно ПУЭ.

Питание оборудования осуществляется от сети напряжением 220 В при частоте 50 Гц. Сопротивление изоляции должно быть не менее 0.5 мОм.

Токи статического электричества возникают при прикосновении персонала к любому из элементов ЭВМ. Статическое электричество существует вблизи экрана дисплея, но электроопасности не несет [23].

Требования электробезопасности при эксплуатации ЭВМ с ВДТ и ЧП:

- В помещении, где одновременно эксплуатируются более пяти ЭВМ с ВДТ и ЧП, на доступном месте устанавливается аварийный резервный выключатель, который может полностью отключить электропитание помещения;
- ЭВМ с ВДТ и ЧП должны подключаться к электросети только с помощью исправных соединений и электророзеток;
- Не допускается подключать ЭВМ с ВДТ и ЧП к обычной двухпроводной электросети, в том числе - с использованием переходных устройств;
- Электросети штепсельных соединений и электророзеток для питания ЭВМ с ВДТ и ЧП нужно выполнять по магистральной схеме, по 3-6 соединений или электророзеток в одном круге;
- Токи короткого замыкания и электрические перегрузки проводов, вызывающие их недопустимый перегрев.

### **5.3 Охрана окружающей среды**

Общие требования к хозяйственной и иной деятельности, оказывающей вредное воздействие на атмосферный воздух регламентированы Федеральным законом "Об охране атмосферного воздуха" статья 15.

Требования охраны атмосферного воздуха при проектировании, размещении, строительстве, реконструкции и эксплуатации объектов

хозяйственной и иной деятельности указаны в статье 16 СанПиН 2.2.1/2.1.1.1200-03 [26].

В Администрации ЗАТО Северск в качестве искусственного освещения используются ртутьсодержащие лампы. Ртуть относится к группе веществ 1 класса опасности. Поэтому утилизация ртутьсодержащих ламп – обязательная мера по обеспечению безопасности. При нарушении целостности корпуса осветительного элемента или по окончании срока его службы: лампочки утилизируются сразу или подлежат временному хранению в специальном помещении на предприятии с дальнейшей транспортировкой и переработкой/утилизацией, при этом используются закрытые контейнеры [25].

В качестве промышленных отходов Администрации ЗАТО Северск выступают: бумага, картон. Предприятие решает проблему отходов, в основной массе это бумага, путем сдачи их в пункты сбора вторичного сырья для дальнейшей переработки.

#### **5.4 Безопасность в чрезвычайных ситуациях**

Чрезвычайная ситуация (ЧС) – обстановка на определенной территории, сложившаяся в результате аварии, опасного природного явления, катастрофы, стихийного или иного бедствия, которые могут повлечь или повлекли за собой человеческие жертвы, ущерб здоровью людей или окружающей природной среде, значительные материальные потери и нарушение условий жизнедеятельности людей.

Рассмотрим пожар, как наиболее вероятную чрезвычайную ситуацию техногенного характера на производстве.

Рабочее помещение по взрывопожарной и пожарной опасности относится к категории "В" "Горючие и трудногорючие жидкости, твердые горючие и трудногорючие вещества и материалы (в том числе пыли и волокна), вещества и материалы, способные при взаимодействии с водой, кислородом воздуха или

друг с другом только гореть, при условии, что помещения, в которых они находятся, не относятся к категории А или Б".

Причины возникновения пожара в помещениях с ПЭВМ и меры по их устранению:

- Неисправности электропроводки, розеток и выключателей которые могут привести к короткому замыканию. Необходимо вовремя выявлять и устранять неисправности электропроводки, розеток и выключателей, проводить плановый осмотр и своевременно устранять все неисправности. В целях уменьшения вероятности возникновения пожара вследствие короткого замыкания необходимо, чтобы электропроводка была скрытой;
- Использование неисправных электроприборов. Необходимо для исключения пожара своевременно ремонтировать электроприборы, качественно исправлять поломки, не использовать неисправные электроприборы;
- Использование в помещении электронагревательных приборов с открытыми нагревательными элементами. В помещении технического отдела находятся бумажные документы и справочная литература в виде книг, пособий, а бумага – легковоспламеняющийся предмет, поэтому использование открытых обогревательных приборов в помещении технического отдела не допускается;
- Возгорание здания вследствие внешних воздействий. В летний период во время грозы возможно попадание молнии вследствие чего возможен пожар. Во избежание этого на крыше здания установлен молниеотвод;
- Неаккуратное обращение с огнем и несоблюдение мер пожарной безопасности. Для устранения возгорания в результате курения, в помещении категорически запрещено курение, оно разрешено только в строго отведенном для этого месте. В целях предотвращения пожара с сотрудниками технического отдела проводится противопожарный

инструктаж. Помещение отдела оборудовано порошковыми огнетушителями с источником давления ОПУ-05-01.

Помещение оснащено пожарной сигнализацией и знаками пожарной безопасности, согласно ГОСТ Р 12.4.026-2001 [22]. Общие требования к знакам пожарной безопасности регламентированы нормами пожарной безопасности.

### 5.5 Эргономические требования к рабочему месту

Для обеспечения оптимальной работоспособности и сохранения здоровья пользователя в течение рабочего дня необходимо проводить перерывы. Время перерывов для различных категорий работы с ПЭВМ в течение рабочего дня, согласно СанПиН 9-131 РБ, представлены на рисунке 30.

Категория работы с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное время регламентированных перерывов, мин	
	Группа А, количество знаков	Группа Б, количество знаков	Группа В, ч	при 8-часовой смене	при 12-часовой смене
I	До 20 000	До 15 000	До 2	50	80
II	До 40 000	До 30 000	До 4	70	110
III	До 60 000	До 40 000	До 6	90	140

Рисунок 30 – Суммарное время регламентированных перерывов в зависимости от продолжительности работы, вида и категории трудовой деятельности с ПЭВМ

В промышленных помещениях необходимо соблюдать нормы полезной площади для сотрудников. Площадь на одно рабочее место с ПЭВМ для пользователей, согласно СанПиН 2.2.2/2.4.1340-03 [10], должна составлять не менее 4,0 кв.м. Данные требования в рабочем помещении соблюдены.

При организации рабочего места необходимо произвести требования эргономики, т.е. учитывать факторы, которые влияют на эффективность действий человека при обеспечении безопасных условий работы. Оптимальная планировка обеспечивает экономию сил и времени человека, удобство при выполнении работы.

Для профилактики производственных заболеваний следует соблюдать следующие правила:

- при работе с клавиатурой, угол сгиба руки в локте должен быть прямым (90 градусов);
- при работе с мышкой кисть должна быть прямой, и лежать на столе как можно дальше от края.

Вывод:

На основании произведённого анализа условий труда в рабочем помещении, можно сделать вывод, что уровень условий труда признан допустимым, и данные, полученные из расчетов полностью удовлетворяют требованиям стандартов безопасности жизнедеятельности.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был проведен анализ функций, процесса оформления пропуска для возможности автоматизации. Были выделены функции, которые целесообразно автоматизировать. Для выполнения поставленных задач с использованием современных технологий была разработана защищенная автоматизированная система. Все элементы созданной автоматизированной системы соответствует установленным требованиям.

Разработанная защищенная автоматизированная система будет интегрирована в портал Администрации ЗАТО Северска. В дальнейшем планируется расширение функциональности системы, а также поддержка и сопровождение данной автоматизированной системы.

Так же в ходе выпускной квалификационной работы выполнено технико-экономическое обоснование проекта, был проведен анализ возможных решений поставленной задачи, выявлены сильные и слабые стороны решения, посчитана основная и дополнительная зарплата исполнителей проекта, проведена оценка трудоемкости и материальных затрат, а также рассмотрены вопросы безопасности труда и производственной санитарии, рассмотрены факторы, оказывающие вредное воздействие на окружающую среду и меры их предотвращения.

Задачи, поставленные перед выполнением выпускной квалификационной работы выполнены, цель достигнута.



## CONCLUSION

During the performance of the final qualifying work, an analysis of the functions, the process of issuing the pass for the possibility of automation was carried out. Functions have been singled out that are expedient to automate. To accomplish these tasks using modern technologies, a secure automated system was developed. All elements of the automated system created meet the requirements.

The developed secure automated system will be integrated into the portal of the Seversk Administration. In the future, it is planned to expand the functionality of the system, as well as support and maintenance of this automated system.

Also in the course of the final qualifying work the feasibility study of the project was carried out, the analysis of possible solutions of the task was carried out, the strengths and weaknesses of the solution were identified, the main and additional salary of the project executors was calculated, labor and material costs were evaluated, and industrial sanitation, factors that have a harmful effect on the environment and measures to prevent them are considered.

The tasks set for the performance of the final qualification work have been fulfilled, the goal has been achieved.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Банокин П.И. Методы и средства проектирования информационных систем и технологий: учебное пособие / П.И. Банокин; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2015. – 80 с.
2. ASP.NET Web API | Microsoft Docs // [Электронный ресурс]: MSDN. URL: <https://docs.microsoft.com/aspnet/web-api/>, Дата обращения 28.05.2018 г.;
3. Изучение ASP.NET MVC [Электронный ресурс]. - Электрон.дан..-ProfessorWeb. URL: [http://professorweb.ru/my/ASP\\_NET/mvc/level/](http://professorweb.ru/my/ASP_NET/mvc/level/), свободный. - Яз. рус. Дата обращения 28.05.2018 г.;
4. MySQLWorkbench [Электронный ресурс] – Электрон.дан. – MySQL. URL: <https://dev.mysql.com/doc/relnotes/workbench/wb-news-6-3-3.html>, свободный. – Яз.англ. Дата обращения: 20.05.2018 г.
5. Обобщенный MVC [Электронный ресурс] – Электрон.дан. – RSDN. URL: <http://rdsn.ru/article/patterns/generic-mvc>, свободный – Яз.рус. Дата обращения: 03.05.2018 г.
6. Модель-Представление-Контроллер [Электронный ресурс] – Электрон.дан. – Полное руководство. URL: <http://yiiframework.ru/doc/guide/basics.mvc>, свободный. – Яз.рус. Дата обращения: 28.04.2018г.
7. Чрезвычайные ситуации [Электронный ресурс] — энциклопедия. – Электрон. дан. URL: <http://ru.wikipedia.org/>, свободный. – Загл. с экрана. – Яз. рус., англ.
8. Нормы пожарной безопасности (Утверждены 18.06.2003);
9. ГОСТ 12.0.003-74 ССБТ Опасные и вредные производственные факторы. Классификация. – М.: Информационно-издательский центр Минздрава России, 1974.

10. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы: с изменениями от 25 апреля 2007 г. – М.: Информационно-издательский центр Минздрава России, 2003.
11. Алексеев С.В., Усенко В.Р. Гигиена труда. М.: Медицина, 1988. — 576 с.
12. СНиП 23-05-95 «Естественное и искусственное освещение» (Утверждены 02.08.1995г.);
13. Д.А. Кривошеин. Экология и безопасность жизнедеятельности. М.: ЮНИТИ-ДАНА, 2000. - 447 с.
14. СНиП 2.2.4/2.1.8.562-96 Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. – М.: Информационно-издательский центр Минздрава России, 1996.
15. СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»
16. Санитарные правила и нормы СанПиН 9-131 РБ "Гигиенические требования к видео дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы" (Утверждены 14.07.1996г.);
17. Правила устройства электроустановок. 6-е изд. с изм. и допол. – СПб, 1999г. – 123с.;
18. НПАОП 0.00-1.28-10 ПРАВИЛА охраны труда при эксплуатации электронно-вычислительных машин (ЭВМ, компьютеров) Охрана труда <http://www.ohranatruda.in.ua/pages/184/>
19. Чрезвычайная ситуация [Электронный ресурс] / Википедия — свободная энциклопедия. — Электрон. дан. URL:<http://ru.wikipedia.org/wiki/>, свободный. – Загл. с экрана. – Яз. рус. англ.

20. ГОСТ Р 22.0.07-95 Безопасность в чрезвычайных ситуациях. Источники техногенных чрезвычайных ситуаций. Классификация и номенклатура поражающих факторов и их параметров. – М.: Госстандарт России, 1995.
21. Нормы пожарной безопасности (Утверждены 18.06.2003);
22. ГОСТ Р 12.4.026-2001 Цвета сигнальные, знаки безопасности и разметка сигнальная»;
23. Нормы пожарной безопасности 160-97 (Утверждены 24.07.1997г.);
24. Ахатов А. Г. Экология. Энциклопедический словарь. - Казань, ТКИ, Экополис, 1995. — с. 168
25. СанПиН 2.2.1/2.1.1.1200-03 Санитарно-защитные зоны и санитарная классификация предприятий, сооружений и иных объектов. – М.: Госкомсанэпиднадзор России, 2003.
26. ГОСТ 30775-2001. Классификация, идентификация и кодирование отходов
27. ГОСТ 6825-91 Лампы люминесцентные трубчатые для общего освещения

## Приложение А

### Программный код на языке SQL для создания БД «ФИАС»

```
CREATE TABLE IF NOT EXISTS `ActualStatus` (
  `ACTSTATID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Идентификатор
статуса (ключ)',
  `NAME` VARCHAR(100) /* NOT NULL */ COMMENT 'Наименование'
) ENGINE=MyISAM COMMENT 'Статус актуальности ФИАС';

CREATE TABLE IF NOT EXISTS `AddressObject` (
  `AOGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Глобальный уникальный
идентификатор адресного объекта ',
  `FORMALNAME` VARCHAR(120) /* NOT NULL */ COMMENT 'Формализованное
наименование',
  `REGIONCODE` VARCHAR(2) /* NOT NULL */ COMMENT 'Код региона',
  `AUTOCODE` VARCHAR(1) /* NOT NULL */ COMMENT 'Код автономии',
  `AREACODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код района',
  `CITYCODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код города',
  `STARCODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код внутригородского района',
  `PLACECODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код населенного пункта',
  `STREETCODE` VARCHAR(4) COMMENT 'Код улицы',
  `EXTRCODE` VARCHAR(4) /* NOT NULL */ COMMENT 'Код дополнительного
адресообразующего элемента',
  `SEXTCODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код подчиненного
дополнительного адресообразующего элемента',
  `OFFNAME` VARCHAR(120) COMMENT 'Официальное наименование',
  `POSTALCODE` VARCHAR(6) COMMENT 'Почтовый индекс',
  `IFNSFL` VARCHAR(4) COMMENT 'Код ИФНС ФЛ',
  `TERRIFNSFL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ФЛ',
  `IFNSUL` VARCHAR(4) COMMENT 'Код ИФНС ЮЛ',
  `TERRIFNSUL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ЮЛ',
  `OKATO` VARCHAR(11) COMMENT 'ОКАТО',
  `OKTMO` VARCHAR(11) COMMENT 'ОКТМО',
  `UPDATEDATE` DATE /* NOT NULL */ COMMENT 'Дата внесения (обновления)
записи',
  `SHORTNAME` VARCHAR(10) /* NOT NULL */ COMMENT 'Краткое наименование типа
объекта',
  `AOLEVEL` INTEGER /* NOT NULL */ COMMENT 'Уровень адресного объекта ',
  `PARENTGUID` VARCHAR(36) COMMENT 'Идентификатор объекта родительского
объекта',
  `AOID` VARCHAR(36) /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Уникальный
идентификатор записи. Ключевое поле',
  `PREVID` VARCHAR(36) COMMENT 'Идентификатор записи связывания с предыдущей
исторической записью',
  `NEXTID` VARCHAR(36) COMMENT 'Идентификатор записи связывания с последующей
исторической записью',
  `CODE` VARCHAR(17) COMMENT 'Код адресного объекта одной строкой с признаком
актуальности из КЛАДР 4.0. ',
  `PLAINCODE` VARCHAR(15) COMMENT 'Код адресного объекта из КЛАДР 4.0 одной
строкой без признака актуальности (последних двух цифр)',
  `ACTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус актуальности адресного
объекта ФИАС. Актуальный адрес на текущую дату. Обычно последняя запись об
адресном объекте.',
  `CENTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус центра',
  `OPERSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус действия над записью -
причина появления записи (см. описание таблицы OperationStatus)',
  `CURRSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус актуальности КЛАДР 4
(последние две цифры в коде)',
  `STARTDATE` DATE /* NOT NULL */ COMMENT 'Начало действия записи',
  `ENDDATE` DATE /* NOT NULL */ COMMENT 'Окончание действия записи',
  `NORMDOC` VARCHAR(36) COMMENT 'Внешний ключ на нормативный документ',
```

```

`LIVESTATUS` ENUM('0','1') /* NOT NULL */ COMMENT 'Признак действующего
адресного объекта'
) ENGINE=MyISAM COMMENT 'Классификатор адресообразующих элементов';

CREATE TABLE IF NOT EXISTS `CenterStatus` (
`CENTERSTID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Идентификатор
статуса (ключ)',
`NAME` VARCHAR(100) /* NOT NULL */ COMMENT 'Наименование'
) ENGINE=MyISAM COMMENT 'Статус центра';

CREATE TABLE IF NOT EXISTS `CurrentStatus` (
`CURENTSTID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Идентификатор
статуса (ключ)',
`NAME` VARCHAR(100) /* NOT NULL */ COMMENT 'Наименование'
) ENGINE=MyISAM COMMENT 'Статус актуальности КЛАДР 4.0';

CREATE TABLE IF NOT EXISTS `EstateStatus` (
`ESTSTATID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Признак
владения (ключ)',
`NAME` VARCHAR(20) /* NOT NULL */ COMMENT 'Наименование',
`SHORTNAME` VARCHAR(20) COMMENT 'Краткое наименование'
) ENGINE=MyISAM COMMENT 'Признак владения';

CREATE TABLE IF NOT EXISTS `House` (
`POSTALCODE` VARCHAR(6) COMMENT 'Почтовый индекс',
`IFNSFL` VARCHAR(4) COMMENT 'Код ИФНС ФЛ',
`TERRIFNSFL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ФЛ',
`IFNSUL` VARCHAR(4) COMMENT 'Код ИФНС ЮЛ',
`TERRIFNSUL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ЮЛ',
`OKATO` VARCHAR(11) COMMENT 'ОКАТО',
`OKTMO` VARCHAR(11) COMMENT 'ОКТМО',
`UPDATEDATE` DATE /* NOT NULL */ COMMENT 'Дата время внесения (обновления)
записи',
`HOUSENUM` VARCHAR(20) COMMENT 'Номер дома',
`ESTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Признак владения',
`BUILDNUM` VARCHAR(10) COMMENT 'Номер корпуса',
`STRUCNUM` VARCHAR(10) COMMENT 'Номер строения',
`STRSTATUS` INTEGER COMMENT 'Признак строения',
`HOUSEID` VARCHAR(36) /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Уникальный
идентификатор записи дома',
`HOUSEGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Глобальный уникальный
идентификатор дома',
`AOGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Guid записи родительского
объекта (улицы, города, населенного пункта и т.п.)',
`STARTDATE` DATE /* NOT NULL */ COMMENT 'Начало действия записи',
`ENDDATE` DATE /* NOT NULL */ COMMENT 'Окончание действия записи',
`STATSTATUS` INTEGER /* NOT NULL */ COMMENT 'Состояние дома',
`NORMDOC` VARCHAR(36) COMMENT 'Внешний ключ на нормативный документ',
`COUNTER` INTEGER /* NOT NULL */ COMMENT 'Счетчик записей домов для КЛАДР 4'
) ENGINE=MyISAM COMMENT 'Сведения по номерам домов улиц городов и населенных
пунктов, номера земельных участков и т.п.';

CREATE TABLE IF NOT EXISTS `HouseInterval` (
`POSTALCODE` VARCHAR(6) COMMENT 'Почтовый индекс',
`IFNSFL` VARCHAR(4) COMMENT 'Код ИФНС ФЛ',
`TERRIFNSFL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ФЛ',
`IFNSUL` VARCHAR(4) COMMENT 'Код ИФНС ЮЛ',
`TERRIFNSUL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ЮЛ',
`OKATO` VARCHAR(11) COMMENT 'ОКАТО',
`OKTMO` VARCHAR(11) COMMENT 'ОКТМО',
`UPDATEDATE` DATE /* NOT NULL */ COMMENT 'Дата время внесения (обновления)
записи',
`INTSTART` INTEGER /* NOT NULL */ COMMENT 'Значение начала интервала',

```

```

`INTEND` INTEGER /* NOT NULL */ COMMENT 'Значение окончания интервала',
`HOUSEINTID` VARCHAR(36) /* NOT NULL */ /* PRIMARY KEY */ COMMENT
'Идентификатор записи интервала домов',
`INTGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Глобальный уникальный
идентификатор интервала домов',
`AOGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Идентификатор объекта
родительского объекта (улицы, города, населенного пункта и т.п.)',
`STARTDATE` DATE /* NOT NULL */ COMMENT 'Начало действия записи',
`ENDDATE` DATE /* NOT NULL */ COMMENT 'Окончание действия записи',
`INTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус интервала (обычный,
четный, нечетный)',
`NORMDOC` VARCHAR(36) COMMENT 'Внешний ключ на нормативный документ',
`COUNTER` INTEGER /* NOT NULL */ COMMENT 'Счетчик записей домов для КЛАДР 4'
) ENGINE=MyISAM COMMENT 'Интервалы домов';

CREATE TABLE IF NOT EXISTS `HouseStateStatus` (
`HOUSESTID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Идентификатор
статуса',
`NAME` VARCHAR(60) /* NOT NULL */ COMMENT 'Наименование'
) ENGINE=MyISAM COMMENT 'Статус состояния домов';

CREATE TABLE IF NOT EXISTS `IntervalStatus` (
`INTVSTATID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Идентификатор
статуса (обычный, четный, нечетный)',
`NAME` VARCHAR(60) /* NOT NULL */ COMMENT 'Наименование'
) ENGINE=MyISAM COMMENT 'Статус интервала домов';

CREATE TABLE IF NOT EXISTS `Landmark` (
`LOCATION` VARCHAR(500) /* NOT NULL */ COMMENT 'Месторасположение ориентира',
`POSTALCODE` VARCHAR(6) COMMENT 'Почтовый индекс',
`IFNSFL` VARCHAR(4) COMMENT 'Код ИФНС ФЛ',
`TERRIFNSFL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ФЛ',
`INFSUL` VARCHAR(4) COMMENT 'Код ИФНС ЮЛ',
`TERRIFNSUL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ЮЛ',
`OKATO` VARCHAR(11) COMMENT 'ОКАТО',
`OKTMO` VARCHAR(11) COMMENT 'ОКТМО',
`UPDATEDATE` DATE /* NOT NULL */ COMMENT 'Дата внесения (обновления) записи',
`LANDID` VARCHAR(36) /* NOT NULL */ COMMENT 'Уникальный идентификатор записи
ориентира',
`LANDGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Глобальный уникальный
идентификатор ориентира',
`AOGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Уникальный идентификатор
родительского объекта (улицы, города, населенного пункта и т.п.)',
`STARTDATE` DATE /* NOT NULL */ COMMENT 'Начало действия записи',
`ENDDATE` DATE /* NOT NULL */ COMMENT 'Окончание действия записи',
`NORMDOC` VARCHAR(36) COMMENT 'Внешний ключ на нормативный документ'
) ENGINE=MyISAM COMMENT 'Описание мест расположения имущественных объектов';

CREATE TABLE IF NOT EXISTS `NormativeDocumentType` (
`NDTYPEID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Идентификатор
записи (ключ)',
`NAME` VARCHAR(250) /* NOT NULL */ COMMENT 'Наименование типа нормативного
документа'
) ENGINE=MyISAM COMMENT 'Тип нормативного документа';

CREATE TABLE IF NOT EXISTS `NormativeDocument` (
`NORMDOCID` VARCHAR(36) /* NOT NULL */ COMMENT 'Идентификатор нормативного
документа',
`DOCNAME` VARCHAR(21783) COMMENT 'Наименование документа',
`DOCDATE` DATE COMMENT 'Дата документа',
`DOCNUM` VARCHAR(20) COMMENT 'Номер документа',
`DOCTYPE` INTEGER /* NOT NULL */ COMMENT 'Тип документа',
`DOCIMGID` INTEGER COMMENT 'Идентификатор образа (внешний ключ)'
)

```

```

) ENGINE=MyISAM COMMENT 'Сведения по нормативному документу, являющемуся
основанием присвоения адресному элементу наименования';

CREATE TABLE IF NOT EXISTS `OperationStatus` (
`OPERSTATID` INTEGER /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Идентификатор
статуса (ключ)',
`NAME` VARCHAR(100) /* NOT NULL */ COMMENT 'Наименование'
) ENGINE=MyISAM COMMENT 'Статус действия';

CREATE TABLE IF NOT EXISTS `AddressObjectType` (
`LEVEL` INTEGER /* NOT NULL */ COMMENT 'Тип адресного объекта',
`SCNAME` VARCHAR(10) COMMENT 'Краткое наименование типа объекта',
`SOCRNAME` VARCHAR(50) /* NOT NULL */ COMMENT 'Полное наименование типа
объекта',
`KOD_T_ST` VARCHAR(4) /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Ключевое
поле'
) ENGINE=MyISAM COMMENT 'Тип адресного объекта';

CREATE TABLE IF NOT EXISTS `StructureStatus` (
`STRSTATID` INTEGER /* NOT NULL */ COMMENT 'Признак строения',
`NAME` VARCHAR(20) /* NOT NULL */ COMMENT 'Наименование',
`SHORTNAME` VARCHAR(20) COMMENT 'Краткое наименование'
) ENGINE=MyISAM COMMENT 'Признак строения';

CREATE TABLE IF NOT EXISTS `DeletedAddressObject` (
`AOGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Глобальный уникальный
идентификатор адресного объекта ',
`FORMALNAME` VARCHAR(120) /* NOT NULL */ COMMENT 'Формализованное
наименование',
`REGIONCODE` VARCHAR(2) /* NOT NULL */ COMMENT 'Код региона',
`AUTOCODE` VARCHAR(1) /* NOT NULL */ COMMENT 'Код автономии',
`AREACODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код района',
`CITYCODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код города',
`STARCODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код внутригородского района',
`PLACECODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код населенного пункта',
`STREETCODE` VARCHAR(4) COMMENT 'Код улицы',
`EXTRCODE` VARCHAR(4) /* NOT NULL */ COMMENT 'Код дополнительного
адресообразующего элемента',
`SEXTCODE` VARCHAR(3) /* NOT NULL */ COMMENT 'Код подчиненного
дополнительного адресообразующего элемента',
`OFFNAME` VARCHAR(120) COMMENT 'Официальное наименование',
`POSTALCODE` VARCHAR(6) COMMENT 'Почтовый индекс',
`IFNSFL` VARCHAR(4) COMMENT 'Код ИФНС ФЛ',
`TERRIFNSFL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ФЛ',
`IFNSUL` VARCHAR(4) COMMENT 'Код ИФНС ЮЛ',
`TERRIFNSUL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ЮЛ',
`OKATO` VARCHAR(11) COMMENT 'ОКАТО',
`OKTMO` VARCHAR(11) COMMENT 'ОКТМО',
`UPDATEDATE` DATE /* NOT NULL */ COMMENT 'Дата внесения (обновления)
записи',
`SHORTNAME` VARCHAR(10) /* NOT NULL */ COMMENT 'Краткое наименование типа
объекта',
`AOLEVEL` INTEGER /* NOT NULL */ COMMENT 'Уровень адресного объекта ',
`PARENTGUID` VARCHAR(36) COMMENT 'Идентификатор объекта родительского
объекта',
`AOID` VARCHAR(36) /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Уникальный
идентификатор записи. Ключевое поле',
`PREVID` VARCHAR(36) COMMENT 'Идентификатор записи связывания с предыдущей
исторической записью',
`NEXTID` VARCHAR(36) COMMENT 'Идентификатор записи связывания с последующей
исторической записью',
`CODE` VARCHAR(17) COMMENT 'Код адресного объекта одной строкой с признаком
актуальности из КЛАДР 4.0. ',

```



```

`PLAINCODE` VARCHAR(15) COMMENT 'Код адресного объекта из КЛАДР 4.0 одной
строкой без признака актуальности (последних двух цифр)',
`ACTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус актуальности адресного
объекта ФИАС. Актуальный адрес на текущую дату. Обычно последняя запись об
адресном объекте.',
`CENTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус центра',
`OPERSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус действия над записью -
причина появления записи (см. описание таблицы OperationStatus)',
`CURRSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус актуальности КЛАДР 4
(последние две цифры в коде)',
`STARTDATE` DATE /* NOT NULL */ COMMENT 'Начало действия записи',
`ENDDATE` DATE /* NOT NULL */ COMMENT 'Окончание действия записи',
`NORMDOC` VARCHAR(36) COMMENT 'Внешний ключ на нормативный документ',
`LIVESTATUS` ENUM('0','1') /* NOT NULL */ COMMENT 'Признак действующего
адресного объекта'
) ENGINE=MyISAM COMMENT 'Классификатор адресообразующих элементов';

```

```

CREATE TABLE IF NOT EXISTS `DeletedHouse` (
`POSTALCODE` VARCHAR(6) COMMENT 'Почтовый индекс',
`IFNSFL` VARCHAR(4) COMMENT 'Код ИФНС ФЛ',
`TERRIFNSFL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ФЛ',
`IFNSUL` VARCHAR(4) COMMENT 'Код ИФНС ЮЛ',
`TERRIFNSUL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ЮЛ',
`OKATO` VARCHAR(11) COMMENT 'ОКАТО',
`OKTMO` VARCHAR(11) COMMENT 'ОКТМО',
`UPDATEDATE` DATE /* NOT NULL */ COMMENT 'Дата время внесения (обновления)
записи',
`HOUSENUM` VARCHAR(20) COMMENT 'Номер дома',
`ESTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Признак владения',
`BUILDNUM` VARCHAR(10) COMMENT 'Номер корпуса',
`STRUCNUM` VARCHAR(10) COMMENT 'Номер строения',
`STRSTATUS` INTEGER COMMENT 'Признак строения',
`HOUSEID` VARCHAR(36) /* NOT NULL */ /* PRIMARY KEY */ COMMENT 'Уникальный
идентификатор записи дома',
`HOUSEGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Глобальный уникальный
идентификатор дома',
`AOGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Guid записи родительского
объекта (улицы, города, населенного пункта и т.п.)',
`STARTDATE` DATE /* NOT NULL */ COMMENT 'Начало действия записи',
`ENDDATE` DATE /* NOT NULL */ COMMENT 'Окончание действия записи',
`STATUS` INTEGER /* NOT NULL */ COMMENT 'Состояние дома',
`NORMDOC` VARCHAR(36) COMMENT 'Внешний ключ на нормативный документ',
`COUNTER` INTEGER /* NOT NULL */ COMMENT 'Счетчик записей домов для КЛАДР 4'
) ENGINE=MyISAM COMMENT 'Сведения по номерам домов улиц городов и населенных
пунктов, номера земельных участков и т.п.';

```

```

CREATE TABLE IF NOT EXISTS `DeletedHouseInterval` (
`POSTALCODE` VARCHAR(6) COMMENT 'Почтовый индекс',
`IFNSFL` VARCHAR(4) COMMENT 'Код ИФНС ФЛ',
`TERRIFNSFL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ФЛ',
`IFNSUL` VARCHAR(4) COMMENT 'Код ИФНС ЮЛ',
`TERRIFNSUL` VARCHAR(4) COMMENT 'Код территориального участка ИФНС ЮЛ',
`OKATO` VARCHAR(11) COMMENT 'ОКАТО',
`OKTMO` VARCHAR(11) COMMENT 'ОКТМО',
`UPDATEDATE` DATE /* NOT NULL */ COMMENT 'Дата время внесения (обновления)
записи',
`INTSTART` INTEGER /* NOT NULL */ COMMENT 'Значение начала интервала',
`INTEND` INTEGER /* NOT NULL */ COMMENT 'Значение окончания интервала',
`HOUSEINTID` VARCHAR(36) /* NOT NULL */ /* PRIMARY KEY */ COMMENT
'Идентификатор записи интервала домов',
`INTGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Глобальный уникальный
идентификатор интервала домов',

```

```

`AOGUID` VARCHAR(36) /* NOT NULL */ COMMENT 'Идентификатор объекта
родительского объекта (улицы, города, населенного пункта и т.п.)',
`STARTDATE` DATE /* NOT NULL */ COMMENT 'Начало действия записи',
`ENDDATE` DATE /* NOT NULL */ COMMENT 'Окончание действия записи',
`INTSTATUS` INTEGER /* NOT NULL */ COMMENT 'Статус интервала (обычный,
четный, нечетный)',
`NORMDOC` VARCHAR(36) COMMENT 'Внешний ключ на нормативный документ',
`COUNTER` INTEGER /* NOT NULL */ COMMENT 'Счетчик записей домов для КЛАДР 4'
) ENGINE=MyISAM COMMENT 'Интервалы домов';

CREATE TABLE IF NOT EXISTS `DeletedNormativeDocument` (
`NORMDOCID` VARCHAR(36) /* NOT NULL */ COMMENT 'Идентификатор нормативного
документа',
`DOCNAME` VARCHAR(21783) COMMENT 'Наименование документа',
`DOCDATE` DATE COMMENT 'Дата документа',
`DOCNUM` VARCHAR(20) COMMENT 'Номер документа',
`DOCTYPE` INTEGER /* NOT NULL */ COMMENT 'Тип документа',
`DOCIMGID` INTEGER COMMENT 'Идентификатор образа (внешний ключ)'
) ENGINE=MyISAM COMMENT 'Сведения по нормативному документу, являющемуся
основанием присвоения адресному элементу наименования';

```

## Приложение Б

### Программный код на языке SQL для создания БД

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,
ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci;
USE `mydb`;

-----
-- Table `mydb`.`Applicant`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Applicant` (
  `Id_applicant` INT NOT NULL,
  `surname` VARCHAR(45) NULL,
  `name` VARCHAR(45) NULL,
  `patronymic` VARCHAR(45) NULL,
  `date_of_birth` DATE NULL,
  `place_of_registration` VARCHAR(100) NULL,
  `place_of_work` VARCHAR(100) NULL,
  `series` INT NULL,
  `number` INT NULL,
  PRIMARY KEY (`Id_applicant`))
ENGINE = InnoDB;

-----
-- Table `mydb`.`Entering`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Entering` (
  `Id_entering` INT NOT NULL AUTO_INCREMENT,
  `surname` VARCHAR(45) NULL,
  `name` VARCHAR(45) NULL,
  `patronymic` VARCHAR(45) NULL,
  `date_of_birth` DATE NULL,
  `place_of_birth` VARCHAR(45) NULL,
  `citizenship_series` VARCHAR(45) NULL,
  `series` INT NULL,
  `number` INT NULL,
  `place_of_work` VARCHAR(255) NULL,
  `position` VARCHAR(100) NULL,
  `place_of_registration` VARCHAR(45) NULL,
  `previous_conviction` VARCHAR(45) NULL,
  `scan_passport` VARCHAR(255) NULL,
  `scan_kinship` VARCHAR(255) NULL,
  PRIMARY KEY (`Id_entering`))
ENGINE = InnoDB;

-----
-- Table `mydb`.`List_KPP`
-----
```

```

-----
CREATE TABLE IF NOT EXISTS `mydb`.`List_KPP` (
  `Id_list_ckpp` INT NOT NULL,
  `number_list_ckpp` INT NULL,
  `print_date` DATE NULL,
  PRIMARY KEY (`Id_list_ckpp`))
ENGINE = InnoDB;
-----
-- Table `mydb`.`Entry_category`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Entry_category` (
  `Id_entry` INT NOT NULL,
  `entry_category` VARCHAR(45) NULL,
  PRIMARY KEY (`Id_entry`))
ENGINE = InnoDB;
-----
-- Table `mydb`.`Target`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Target` (
  `Id_target` INT NOT NULL,
  `target` VARCHAR(255) NULL,
  PRIMARY KEY (`Id_target`))
ENGINE = InnoDB;
-----
-- Table `mydb`.`List_FSB`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`List_FSB` (
  `Id_list` INT NOT NULL,
  `number_list_fsb` INT NULL,
  `date_of_authorization` DATE NULL,
  `explanation_of_permission` VARCHAR(255) NULL,
  `explanation_of_failure` VARCHAR(255) NULL,
  PRIMARY KEY (`Id_list`))
ENGINE = InnoDB;
-----
-- Table `mydb`.`Application`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`Application` (
  `Id_application` INT NOT NULL,
  `status` VARCHAR(45) NULL,
  `date_of_completion` DATE NULL,
  `allowed` VARCHAR(45) NULL,
  `notes` VARCHAR(255) NULL,
  `List_KPP_Id_list_ckpp` INT NOT NULL,
  `Entry_category_Id_entry` INT NOT NULL,
  `Applicant_Id_applicant` INT NOT NULL,
  `Entering_Id_entering` INT NOT NULL,
  `Target_Id_target` INT NOT NULL,
  `List_FSB_Id_list` INT NOT NULL,
  PRIMARY KEY (`Id_application`, `List_KPP_Id_list_ckpp`,
`Entry_category_Id_entry`, `Applicant_Id_applicant`,
`Entering_Id_entering`, `Target_Id_target`, `List_FSB_Id_list`),

```

```

INDEX `fk_Application_List_KPP1_idx` (`List_KPP_Id_list_ckpp`
ASC),
INDEX `fk_Application_Entry_category1_idx`
(`Entry_category_Id_entry` ASC),
INDEX `fk_Application_Applicant1_idx` (`Applicant_
Id_applicant` ASC),
INDEX `fk_Application_Entering1_idx` (`Entering_Id_entering`
ASC),
INDEX `fk_Application_Target1_idx` (`Target_Id_target` ASC),
INDEX `fk_Application_List_FSB1_idx` (`List_FSB_Id_list` ASC),
CONSTRAINT `fk_Application_List_KPP1`
FOREIGN KEY (`List_KPP_Id_list_ckpp`)
REFERENCES `mydb`.`List_KPP` (`Id_list_ckpp`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Application_Entry_category1`
FOREIGN KEY (`Entry_category_Id_entry`)
REFERENCES `mydb`.`Entry_category` (`Id_entry`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Application_Applicant1`
FOREIGN KEY (`Applicant_Id_applicant`)
REFERENCES `mydb`.`Applicant` (`Id_applicant`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Application_Entering1`
FOREIGN KEY (`Entering_Id_entering`)
REFERENCES `mydb`.`Entering` (`Id_entering`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Application_Target1`
FOREIGN KEY (`Target_Id_target`)
REFERENCES `mydb`.`Target` (`Id_target`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Application_List_FSB1`
FOREIGN KEY (`List_FSB_Id_list`)
REFERENCES `mydb`.`List_FSB` (`Id_list`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```
ENGINE = InnoDB;
```

```
-----
-- Table `mydb`.`Processing`
-----
```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Processing` (
  `Id_processing` INT NOT NULL,
  `receipt_date` DATE NULL,
  `comment` VARCHAR(255) NULL,
  `Application_Id_application` INT NOT NULL,
  `Application_List_KPP_Id_list_ckpp` INT NOT NULL,
  `Application_Entry_category_Id_entry` INT NOT NULL,
  `Application_Applicant_Id_applicant` INT NOT NULL,
  `Application_Entering_Id_entering` INT NOT NULL,

```

```

`Application_Target_Id_target` INT NOT NULL,
`Application_List_FSB_Id_list` INT NOT NULL,
`List_FSB_Id_list` INT NOT NULL,
`List_KPP_Id_list_ckpp` INT NOT NULL,
PRIMARY KEY (`Id_processing`, `Application_Id_application`,
`Application_List_KPP_Id_list_ckpp`,
`Application_Entry_category_Id_entry`, `Application_Applicant_
Id_applicant`, `Application_Entering_Id_entering`,
`Application_Target_Id_target`, `Application_List_FSB_Id_list`,
`List_FSB_Id_list`, `List_KPP_Id_list_ckpp`),
INDEX `fk_Processing_Application1_idx`
(`Application_Id_application` ASC,
`Application_List_KPP_Id_list_ckpp` ASC,
`Application_Entry_category_Id_entry` ASC,
`Application_Applicant_Id_applicant` ASC,
`Application_Entering_Id_entering` ASC,
`Application_Target_Id_target` ASC,
`Application_List_FSB_Id_list` ASC),
INDEX `fk_Processing_List_FSB1_idx` (`List_FSB_Id_list` ASC),
INDEX `fk_Processing_List_KPP1_idx` (`List_KPP_Id_list_ckpp`
ASC),
CONSTRAINT `fk_Processing_Application1`
FOREIGN KEY (`Application_Id_application`,
`Application_List_KPP_Id_list_ckpp`,
`Application_Entry_category_Id_entry`, `Application_Applicant_
Id_applicant`, `Application_Entering_Id_entering`,
`Application_Target_Id_target`, `Application_List_FSB_Id_list`)
REFERENCES `mydb`.`Application` (`Id_application`,
`List_KPP_Id_list_ckpp`, `Entry_category_Id_entry`,
`Applicant_Id_applicant`, `Entering_Id_entering`,
`Target_Id_target`, `List_FSB_Id_list`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Processing_List_FSB1`
FOREIGN KEY (`List_FSB_Id_list`)
REFERENCES `mydb`.`List_FSB` (`Id_list`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Processing_List_KPP1`
FOREIGN KEY (`List_KPP_Id_list_ckpp`)
REFERENCES `mydb`.`List_KPP` (`Id_list_ckpp`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

## Приложение В

### Листинг созданных представлений

```
<script>
function Delete(controller, id) {
    if (confirm("Вы действительно хотите это удалить?")) {
        $('#' + controller + 'Div' + id).load('/' + controller +
'/Delete/' + id);
        $('.' + controller + 'Div' + id).load('/' + controller +
'/Delete/' + id);
    }
}

function Cancel(controller, id) {
    if (confirm("Продолжаем?")) {
        $("#" + controller + "Tab").find('#' + controller + 'Div' +
id).load('/' + controller + '/Cancel/' + id);
        $("#" + controller + "Tab").find('.' + controller + 'Div' +
id).load('/' + controller + '/Cancel/' + id);
    }
}

function CancelEdit(controller, id) {
    $.get('/' + controller + '/Details/' + id, function (data) {
        res = $.parseHTML('<div>' + data + '</div>');
        if ($(res).find('.' + controller + 'Div' + id).html() != "") {
            var content = $(res).find('.' + controller + 'Div' +
id).html();
        }
        else {
            var content = $(res).find('#' + controller + 'Div' +
id).html();
        }
        $('#' + controller + 'Div' + id).html(content);
        $('.' + controller + 'Div' + id).html(content);
    });
}

function LoadEditForm(controller, id) {
    if ($("#" + controller + "Tab").hasClass("active"))
    {
        $("#" + controller + "Tab").find('#' + controller + 'Div' +
id).load('/' + controller + '/Edit/' + id);
        $("#" + controller + "Tab").find('.' + controller + 'Div' +
id).load('/' + controller + '/Edit/' + id);
    }
    else
    {
        $("#dateTab").find('#' + controller + 'Div' + id).load('/' +
controller + '/Edit/' + id);
        $("#dateTab").find('.' + controller + 'Div' + id).load('/' +
controller + '/Edit/' + id);
    }
}

function PostEditForm(controller, id, mce) {
    if (mce == true) tinyMCE.triggerSave();
    $.ajax({
        type: "POST",
        url: '/' + controller + '/Edit/' + id,
        data: $('.' + controller + 'Edit' + id).serialize() + $('#' +
controller + 'Edit' + id).serialize(),
        success: function (data) {
            res = $.parseHTML('<div>' + data + '</div>');
        }
    });
}
```

```

        if ($(res).find('.' + controller + 'Div' + id).html() != "")
        {
            var content = $(res).find('.' + controller + 'Div' +
id).html();
        }
        else {
            var content = $(res).find('#' + controller + 'Div' +
id).html();
        }
        $('#' + controller + 'Div' + id).html(content);
        $('.' + controller + 'Div' + id).html(content);
    }
});

}

function PostCreateForm(controller, num, mce) {
    if (mce == true) tinyMCE.triggerSave();
    $.ajax({
        type: "POST",
        url: '/' + controller + '/Create/',
        data: $('#' + controller + 'Create').serialize(),
        success: function (data) {
            res = $.parseHTML('<div><div>' + data + '</div></div>');
            $('#documentData' +
num).prepend($(res).find('div').first().html());
            $('#' + controller +
'Tab').find('.tabContent').prepend($(res).find('div').first().html());
            $('#' + controller + 'Create').trigger('reset');
        }
    });
}
}

```

</script>

```

<form class="form">
    <p><select name="status" id="status">
        <option selected="selected">Временно</option>
        <option>Постоянно</option>
        <option>На работу</option>
    </select>
    <label>Статус въезда</label></p>
    <p><input type="date" id="date_of_completion" name="date_of_completion "
/>
        <label for="date_of_completion ">Дата заполнения</label></p>
    <p><input type="text" name=" number_list_fsb " id=" number_list_fsb " />
        <label for="number_list_fsb">№ Списка ФСБ</label></p>
    <p><input type="text" name=" number_list_ckpp" id=" number_list_ckpp " />
        <label for=" number_list_ckpp ">№ Списка ЦКПП</label></p>
    <p><input type="text" name=" entry_category " id="entry_category" />
        <label for=" entry_category ">Категория въезда</label></p>
    <p><textarea name="text"></textarea>
        <label>Примечание</label></p><br />
    <p><select name=" allowed " id="allowed">
        <option selected="selected">Рудич А.А.</option>
        <option>Мятлев П.А.</option>
        <option>Веденин И.И.</option>
    </select>
    <label>Кто разрешил</label></p>
</form>

```



```

<div class="tab-content">
  <div role="tabpanel" class="tab-pane active fade in" id="dateTab">
    @if (Model.visits.Count == 0)
    {
      <div class="bs-callout bs-callout-success">

        </div>
      }
      @if (Model.visits.Count != 0 &&
Model.visits.First().date.Equals(DateTime.Today))
      {
        @Html.Partial("~/Views/documentList.cshtml", new documentList
{ num = 1, add = true, visit = Model.visits.First() })
      }
      @{
        int num = 9;
      }
      @foreach (var visit in Model.visits)
      {
        if (visit.date.Equals(DateTime.Today))
        {
          continue;
        }
        @Html.Partial("~/Views/documentList.cshtml", new documentList
{ num = num, add = false, visit = visit })
        num = num + 8;
      }
    }
  </div>
<table style="width: 100%" cellpadding="0" cellspacing="0" align="center">
  <tr>
    <td colspan="2"><h3><b>Формирование заявки</b></h3></td>
  </tr>
  <tr>
    <td valign="top">
      <h3>Заявитель:</h3>
    </td>
    <td valign="top">
      <h3>Въезжающий:</h3>
    </td>
  </tr>
  <tr>
    <td valign="top">
      <form class="form">
        <p><input type="text" name="surname" id="surname" />
          <label for="surname">Фамилия</label></p>
        <p><input type="text" name="name" id=" name " />
          <label for=" name " >Имя</label></p>
        <p><input type="text" name="patronymic" id="patronymic" />
          <label for="patronymic">Отчество</label></p>
        <p><input type="date" id="date_of_birth"
name="date_of_birth" />
          <label for="date_of_birth">Дата рождения</label></p>
        <p><input type="text" name="place_of_registration"
id="place_of_registration" />
          <label for="place_of_registration">Место регистрации</label></p>
        <p><input type="text" name="place_of_work" id="place_of_work" />
          <label for="place_of_work">Место работы</label></p>
        <input class="series" type="text" name="series" id="series"
/>&nbsp;&nbsp;&nbsp;<input class="number" type="text" name="number" id="number" />
          <label for="series">Серия / Номер паспорта</label></p>
      </form>
    </td>
    <td valign="top">

```

```

<form class="form">
<p><input type="text" name="surname" id="surname" />
  <label for="surname">Фамилия</label></p>
<p><input type="text" name="name" id="name" />
  <label for="name">Имя</label></p>
<p><input type="text" name="patronymic" id="patronymic" />
  <label for="patronymic">Отчество</label></p>
<p><input type="date" id="date_of_birth" name="date_of_birth" />
  <label for="date_of_birth">Дата рождения</label></p>
<p><input type="text" name="place_of_birth" id="place_of_birth" />
  <label for="place_of_birth">Место рождения</label></p>
<p><input type="text" name="sitizenship" id="sitizenship" />
  <label for="sitizenship">Гражданство</label></p>
<p><input class="series" type="text" name="series" id="series" />
  &nbsp;&nbsp;<input class="number" type="text" name="number" id="number" />
  <label for="series">Серия / Номер паспорта</label></p>
<p><input type="text" name="place_of_work" id="place_of_work" /> <label
for="place_of_work">Место работы</label></p>
<p><input type="text" name="position" id="position" />
  <label for="position">Должность</label></p>
<p><input type="text" name="place_of_registration" id="place_of_registration"
/>
  <label for="place_of_registration">Место регистрации</label></p>
<p><input type="text" name="target" id="target" />
  <label for="target">Цель приезда</label></p>
<table style="width: 200px" cellspacing="0" cellpadding="0">
  <tr>
    <td colspan="2" align="center"><label
for="previous_conviction">Наличие судимости</label></td>
  </tr>
  <tr>
    <td><p align="center"><input class="radio"
name="yes" type="radio" value="yes"><label>Есть</label></p></td>
    <td><p align="center"><input class="radio"
name="no" type="radio" value="no"><label>Нет</label></p></td>
  </tr>
</table>
<table style="width: 600px" cellspacing="0" cellpadding="0">
  <tr>
    <td align="center">
      <label for="scan_passport">Загрузите скан паспорта</label>
<p><input class="load" type="file" name="pdf" multiple
accept="image/*,pdf/pdf"><br />
<input class="button_load" type="submit" value="Загрузить"></p>
      <td align="center">
        <label for="scan_kinship">Загрузите скан подтверждения родства</label><p>
          <input class="load" type="file" name="pdf"
multiple accept="image/*,pdf/pdf"><br />
          <input class="button_load" type="submit" value="Загрузить">
        </p>
      </td>
    </tr>
</table>
</form>
</td>
<tr>
  <td>
    <tr>
      <td colspan="2" align="right">
        <p class="submit"><input type="submit" value="Отправить" /></p>
      </td>
    </tr>
  </tr>
</table>

```

## Приложение Г

### Программный код модуля авторизации ЕСИА

```
public class EsiaModule : NancyModule
{
    static string client_id = "1200334877"; //Мнемоника системы, его
можно узнать у ЕСИА
    static string state = Guid.NewGuid().ToString("D"); //гуйд для всяких
проверок
    static string server_url = "https://esia.gosuslugi.ru/aas/oauth2/ac";
//адрес по которому есиа вернет авторизационный код
    static string server_url_2 =
"https://esia.gosuslugi.ru/aas/oauth2/te"; //адрес по которому получим маркер
    static string server_url_prns = "https://esia.gosuslugi.ru/rs/prns/";
//тут мы узнаем ФИО пользователя, который к нам хочет залогиниться

//тестовые настройки
    /* static string server_url = "https://esia-
portall.test.gosuslugi.ru/aas/oauth2/ac";
    static string server_url_2 = "https://esia-
portall.test.gosuslugi.ru/aas/oauth2/te";
    static string server_url_prns = "https://esia-
portall.test.gosuslugi.ru/rs/prns/";*/
    public EsiaModule()
    {
        Get["/ESIA"] = _ =>
        {
            //string scope = "openid";
            string scope = "fullname";
            string timestamp = DateTime.UtcNow.ToString("yyyy.MM.dd
HH:mm:ss +0000");
            string access_type = "online";
            string response_type = "code";
            string redirect_uri = Request.Url.Scheme + "://" +
Request.Url.HostName + (Request.Url.Port == 80 ? "" : (":" +
Request.Url.Port.ToString())) + "/ESIA-OK"; //адрес редиректа, после того
как пользователь ввел данные в есиа
            string client_secret = "";
            //Генерим подпись с помощью нашего сертификата
            string msg = scope + timestamp + client_id + state;
            byte[] msgBytes = Encoding.UTF8.GetBytes(msg);
            var signerCert = DetachedSignature.GetSignerCert();
            byte[] encodedSignature = DetachedSignature.SignMsg(msgBytes,
signerCert);
            client_secret =
HttpServerUtility.UrlTokenEncode(encodedSignature);
            //генерим строку с параметрами
            RequestBuilder builder = new RequestBuilder();
            builder.AddParam("client_id", client_id);
            builder.AddParam("client_secret", client_secret);
            builder.AddParam("redirect_uri", redirect_uri);
            builder.AddParam("scope", scope);
            builder.AddParam("response_type", response_type);
            builder.AddParam("state", state);
            builder.AddParam("timestamp", timestamp);
            builder.AddParam("access_type", access_type);

            string red_url = server_url + "?" +
builder.ToString().Replace("+", "%2b");
            return Response.AsRedirect(red_url);
        }
    }
}
```

```

};
Get["/ESIA-OK"] = _ =>
{
    //сюда нас редиректит есиа, мы проверяем state который был
    послан и который пришел, потом используем code
    string state_r = Request.Query["state"];
    string code = Request.Query["code"];

    if (state == state_r)
    {
        //string scope = "openid";
        string scope = "fullname";
        string timestamp = DateTime.UtcNow.ToString("yyyy.MM.dd
HH:mm:ss +0000");
        string redirect_uri = Request.Url.Scheme + "://" +
Request.Url.HostName + (Request.Url.Port == 80 ? "" : (":" +
Request.Url.Port.ToString())) + "/ESIA-OK";
        string client_secret = "";
        string msg = scope + timestamp + client_id + state;
        byte[] msgBytes = Encoding.UTF8.GetBytes(msg);
        var signerCert = DetachedSignature.GetSignerCert();
        byte[] encodedSignature =
DetachedSignature.SignMsg(msgBytes, signerCert);
        client_secret =
HttpServerUtility.UrlTokenEncode(encodedSignature);

        string result;
        //генерим post запрос для получения маркера
        {
            RequestBuilder builder = new RequestBuilder();
            builder.AddParam("client_id", client_id);
            builder.AddParam("code", code);
            builder.AddParam("grant_type", "authorization_code");
            builder.AddParam("client_secret", client_secret);
            builder.AddParam("state", state);
            builder.AddParam("redirect_uri", redirect_uri);
            builder.AddParam("scope", scope);
            builder.AddParam("timestamp", timestamp);
            builder.AddParam("token_type", "Bearer");

            var httpWebRequest =
(HttpWebRequest)WebRequest.Create(server_url_2);
            httpWebRequest.ContentType = "application/x-www-form-
urlencoded";

            httpWebRequest.Method = "POST";
            httpWebRequest.Timeout = int.MaxValue;

            httpWebRequest.Proxy = p;
            using (var streamWriter = new
StreamWriter(httpWebRequest.GetRequestStream()))
            {
                streamWriter.Write(builder.ToString().Replace("+", "%2b"));
                streamWriter.Flush();
                streamWriter.Close();
            }

            var httpResponse =
(HttpWebResponse)httpWebRequest.GetResponse();
            using (var streamReader = new
StreamReader(httpResponse.GetResponseStream()))
            {
                result = streamReader.ReadToEnd();
            }
        }
    }
}

```

```

    }
}
ESIA_Marker_Answer marker =
JsonConvert.DeserializeObject<ESIA_Marker_Answer>(result);//наш маркер

string[] marker_parts = marker.access_token.Split('.');

string header =
Encoding.UTF8.GetString(base64urldecode(marker_parts[0]));
string payload =
Encoding.UTF8.GetString(base64urldecode(marker_parts[1]));
string oid =
(JsonConvert.DeserializeObject<dynamic>(payload))["urn:esia:sbj_id"];

//генерим запрос для получения информации о пользователе
string user_info = "";
{
    var httpWebRequest =
(HttpWebRequest)WebRequest.Create(server_url_prns + oid);
    httpWebRequest.ContentType = "application/x-www-form-
urlencoded";

    httpWebRequest.Method = "GET";
    httpWebRequest.Headers["Authorization"] = "Bearer " +
marker.access_token;

    httpWebRequest.Timeout = int.MaxValue;

    var httpResponse =
(HttpWebResponse)httpWebRequest.GetResponse();
    using (var streamReader = new
StreamReader(httpResponse.GetResponseStream()))
    {
        user_info = streamReader.ReadToEnd();
    }
}

string firstName =
JsonConvert.DeserializeObject<dynamic>(user_info)["firstName"];
string lastName = "";
try
{
    lastName =
JsonConvert.DeserializeObject<dynamic>(user_info)["lastName"];
}
catch { }
string middleName = "";
try
{
    middleName =
JsonConvert.DeserializeObject<dynamic>(user_info)["middleName"];
}
catch { }

// Записываю в бд данные если он не зареган и проверяю
если зареган

MyEntities db = new MyEntities();
Helper h = new Helper();
users u;
if (db.users.Any(a => a.esia_oid.Trim() == oid.Trim()))
{
    u = db.users.FirstOrDefault(a => a.esia_oid.Trim() ==
oid.Trim());
}

```

```

    }
    else
    {
        u = new users
        {
            mail = oid,
            name = firstName,
            pass = h.HashWithSalt(oid),
            patronymic = middleName ?? "",
            surname = lastName ?? "",
            token = Guid.NewGuid().ToString("N"),
            role_id = 2,
            esia_oid = oid
        };
        db.users.Add(u);
        db.SaveChanges();
    }

    //добавляю куки и отправляю на главную
    DateTime expires = DateTime.UtcNow.AddYears(20);
    List<NancyCookie> cs = new List<NancyCookie> {new
NancyCookie("id", u.id.ToString(), false) { Expires = expires }, new
NancyCookie("token", u.token, false) { Expires = expires } };
    var resp = Response.AsRedirect("/");
    resp.AddCookie(cs[0]);
    resp.AddCookie(cs[1]);

    return resp;
}
else
{
    return Response.AsJson(new { err = 1 });
}
};
}
static byte[] base64urldecode(string arg)
{
    string s = arg;
    s = s.Replace('-', '+'); // 62nd char of encoding
    s = s.Replace('_', '/'); // 63rd char of encoding
    switch (s.Length % 4) // Pad with trailing '='s
    {
        case 0: break; // No pad chars in this case
        case 2: s += "=="; break; // Two pad chars
        case 3: s += "="; break; // One pad char
        default: throw new System.Exception(
            "Illegal base64url string!");
    }
    return Convert.FromBase64String(s); // Standard base64 decoder
}
}

public class RequestBuilder
{
    List<RequesItemClass> items = new List<RequesItemClass>();
    public void AddParam(string name, string value)
    {
        items.Add(new RequesItemClass { name = name, value = value });
    }
    override public string ToString()
    {
        return string.Join("&", items.Select(a => a.name + "=" +
a.value));
    }
}

```

```

    }
}

class DetachedSignature
{
    static public X509Certificate2 GetSignerCert()
    {
        X509Store storeMy = new X509Store(StoreName.My,
StoreLocation.LocalMachine);
        storeMy.Open(OpenFlags.ReadOnly);
        X509Certificate2Collection certColl =
storeMy.Certificates.Find(X509FindType.FindBySubjectKeyIdentifier, "12 34 56
78 90 00 00 00 11 11 11 11 11 11 11 11 22 33 44", false);
        storeMy.Close();
        return certColl[0];
    }
    static public byte[] SignMsg(Byte[] msg, X509Certificate2 signerCert)
    {
        ContentInfo contentInfo = new ContentInfo(msg);
        SignedCms signedCms = new SignedCms(contentInfo, true);
        CmsSigner cmsSigner = new CmsSigner(signerCert);
        signedCms.ComputeSignature(cmsSigner);
        return signedCms.Encode();
    }

    static public bool VerifyMsg(Byte[] msg, byte[] encodedSignature)
    {
        ContentInfo contentInfo = new ContentInfo(msg);
        SignedCms signedCms = new SignedCms(contentInfo, true);
        signedCms.Decode(encodedSignature);

        try
        {
            signedCms.CheckSignature(true);
        }
        catch (System.Security.Cryptography.CryptographicException e)
        {
            return false;
        }

        return true;
    }
}

public class ESIA_Marker_Answer
{
    public string state { get; set; }
    public string token_type { get; set; }
    public int expires_in { get; set; }
    public string refresh_token { get; set; }
    public string id_token { get; set; }
    public string access_token { get; set; }
}

public class RequesItemClass
{
    public string name;
    public string value;
}

```