

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
федерального государственного бюджетного образовательного учреждения высшего образования  
«Владимирский государственный университет  
Имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(МИВлГУ)

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИИ \_\_\_\_\_

# КУРСОВАЯ РАБОТА

по \_\_\_\_\_ Разработка корпоративных приложений \_\_\_\_\_

Тема \_\_\_\_\_ Распределенная ИС Страхового агентства \_\_\_\_\_

\_\_\_\_\_

(оценка)

Руководитель

Кульков Я.Ю. \_\_\_\_\_

(фамилия, инициалы)

\_\_\_\_\_ (подпись) \_\_\_\_\_ (дата)

Члены комиссии

Студент ПИИ-119 \_\_\_\_\_

(группа)

\_\_\_\_\_ (подпись) \_\_\_\_\_ (Ф.И.О.)

Лямина И.А. \_\_\_\_\_

(фамилия, инициалы)

\_\_\_\_\_ (подпись) \_\_\_\_\_ (Ф.И.О.)

\_\_\_\_\_ (подпись) \_\_\_\_\_ (дата)

Муром 2022

Задание

Задание

В данной курсовой работе была разработана БД и распределенная ИС Страхового агентства. В ходе выполнения курсовой работы выявлены требования к программе, разработаны модели данных и диаграммы. На основе разработанных моделей реализован набор классов и разработано приложение на языке программирования Java в среде разработки IntelliJ IDEA с использованием фреймворка Spring. На заключительном этапе работы произведено тестирование разработанного продукта.

In this course work, a database and a distributed information system of an Insurance agency were developed. During the course work, the requirements for the program were identified, data models and diagrams were developed. Based on the developed models, a set of classes is implemented and an application in the Java programming language is developed in the IntelliJ IDEA development environment using the Spring framework. At the final stage of the work, the developed product was tested.

## Содержание

Введение.....	6
1. Анализ технического задания .....	8
1.1 Действующие документы.....	8
1.2 Обоснование выбора средств реализации .....	9
1.3 Функциональные возможности .....	10
2. Разработка моделей данных.....	11
2.1 Концептуальная модель данных.....	12
2.2 Логическая модель данных .....	14
2.3 Соответствие сущностей логической и физической моделей.....	15
2.4 Физическая модель данных.....	20
3.Проектирование работы системы.....	21
3.1 Диаграмма вариантов использования .....	21
3.2 Функциональная диаграмма .....	21
3.3 Диаграммы последовательностей .....	22
4. Разработка и реализация системы .....	25
5. Тестирование системы.....	34
Заключение .....	38
Список используемой литературы .....	39
Приложение 1. Модели данных и диаграммы .....	40
Приложение 2. Код программы .....	46
Приложение 3. Снимки окон программы (скриншоты программы) .....	47

					МИВУ 09.03.04-8.000 ПЗ		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Лямина И.А.			Распределенная ИС Страхового агентства	Лит.	Лист
Провер.		Кульков Я.Ю.					Листов
Реценз.							5
Н. Контр.							53
Утверд.						МИ ВлГУ ПИН-119	

## Введение

Распределённая система — система, для которой отношения местоположений элементов (или групп элементов) играют существенную роль с точки зрения функционирования системы, а, следовательно, и с точки зрения анализа и синтеза системы.

Для распределённых систем характерно распределение функций, ресурсов между множеством элементов (узлов) и отсутствие единого управляющего центра, поэтому выход из строя одного из узлов не приводит к полной остановке всей системы.

Эффективное управление предприятием в современных условиях невозможно без использования компьютерных технологий. Разработка распределённой информационной системы страхового агентства важна, так как будет разработано приложение для автоматизации процесса страхования, которое поможет страховым агентам сократить время на работу с документацией.

Внедрение информационных технологий в процесс планирования и управления деятельностью страховых компаний предусматривает не только обработку больших и взаимосвязанных массивов данных, но может использоваться также для их анализа и обоснований вариантов управленческих решений.

Объемы информации, высокие требования к точности и достоверности, необходимость эффективного анализа финансового состояния клиентуры и страховой фирмы — вот основные причины, предопределяющие автоматизацию страхового бизнеса.

Функции, которые будут автоматизированы:

– если приходил страхователь, то сотруднику приходится искать вручную его бумажные договора, возможно спрашивая самого клиента. В ИС достаточно написать номер телефона или паспорт страхователя и сотрудник

					МИВУ 09.03.04-8.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

получит всю информацию как о самом клиенте, так и о всех заключенных договорах;

– чтобы продлить договор приходилось переписывать и перепроверять все данные. В ИС можно просто нажать на кнопку продлить и выбрать срок, на который нужно продлить;

– автоматически составляются отчёты. Сотруднику достаточно выбрать вид страхования и период времени, и он получит следующие данные: количество и сумма заключенных договоров, сумма страховых выплат;

– автоматически происходит рассылка финансового отчёта на почту администратора, а пользователям приходят уведомления о истечение срока действия, оформленного на них полиса.

Целью курсовой работы является проектирование и разработка распределённой информационной системы страхового агентства.

Для достижения поставленной цели были поставлены следующие задачи:

- проанализировать предметную область;
- разработать модели данных;
- реализовать базу данных;
- разработать приложение ASP.Net;
- протестировать программный продукт.

					МИВУ 09.03.04-8.000 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		



## 1. Анализ технического задания

### 1.1 Действующие документы

Для наиболее точного анализа предоставленного технического задания были найдены действующие полисы ОСАГО и КАСКО. Примеры данных документов приведены на рисунках 1 и 2.

**1** — **ЭЛЕКТРОННЫЙ СТРАХОВОЙ ПОЛИС** серия XXX № 000000000 — **2**

ОБЯЗАТЕЛЬНОГО СТРАХОВАНИЯ ГРАЖДАНСКОЙ ОТВЕТСТВЕННОСТИ  
ВЛАДЕЛЬЦЕВ ТРАНСПОРТНЫХ СРЕДСТВ

Росгосстрах  
ООО «Росгосстрах»  
Т: 0530 (моб.) или 8-800-200-0-900  
www.RGS.ru

Срок страхования с **17** ч. **30** мин. **01** **05** . 20 **16** г. — **3**  
по 24 ч. 00 мин. **30** **04** . 20 **17** г.

Страхование распространяется на страховые случаи, произошедшие в период использования транспортного средства в течение срока страхования  
**4** — с **01** **05** . 20 **16** г. по **31** **08** . 20 **16** г., с **01** **10** . 20 **16** г. по **30** **04** . 20 **17** г.,  
с . 20 г. по . 20 г.

**5** — 1. Страхователь (полное наименование юридического лица или фамилия, имя, отчество\* гражданина)  
**Николаев Николай Николаевич**

**6** — Собственник транспортного средства (полное наименование юридического лица или фамилия, имя, отчество\* гражданина)  
**Николаева Мария Сергеевна**

2. Транспортное средство используется с прицепом: да, нет.

Марка, модель транспортного средства **Toyota Corolla** Идентификационный номер транспортного средства **JTHBA00E490144371** Государственный регистрационный знак транспортного средства **T110 EA 63**

**7** — Паспорт транспортного средства, свидетельство о регистрации транспортного средства, паспорт самоходной машины (либо аналогичный документ)  
Вид документа **паспорт транспортного средства** серия **63 УТ** номер **348543**

Цель использования транспортного средства (отметить нужное): ☒ личная, учебная езда, такси, перевозка опасных и легковоспламеняющихся грузов, прокат/краткосрочная аренда, регулярные пассажирские перевозки/перевозки пассажиров по заказам, дорожные и специальные транспортные средства, экстренные и коммунальные службы, прочие.

3. Договор заключен в отношении: неограниченного количества лиц, допущенных к управлению транспортным средством, лиц, допущенных к управлению транспортным средством

№ п/п	Лица, допущенные к управлению транспортным средством (фамилия, имя, отчество*)	Водительское удостоверение (серия, номер)
<b>8</b> — <b>1</b>	<b>Николаев Николай Николаевич</b>	<b>63 01 112113</b>
<b>2</b>	<b>Николаев Степан Николаевич</b>	<b>63 02 673749</b>
—	—	—
—	—	—

4. Страховая сумма, в пределах которой страховщик при наступлении каждого страхового случая (независимо от количества страховых случаев в течение срока страхования по договору обязательного страхования) обязуется возместить потерпевшим причиненный вред, установлена Федеральным законом от 25 апреля 2002 года №40-ФЗ «Об обязательном страховании гражданской ответственности владельцев транспортных средств» в редакции, действующей на дату заключения (изменения\*) настоящего договора.

5. Страховой случай - наступление гражданской ответственности владельца транспортного средства за причинение вреда жизни, здоровью или имуществу потерпевших при использовании транспортного средства, влечущее за собой в соответствии с договором обязательного страхования обязанность страховщика осуществить страховую выплату.

6. Страховой полис действует на территории Российской Федерации.

**9** — 7. Страховая премия **8 345 руб. 00 копеек** рублей.

**10** — 8. Особые отметки  
Данный полис был приобретен на сайте ООО «Росгосстрах» по адресу: [www.RGS.ru](http://www.RGS.ru) (заказ № 1241538)

**11** — Дата заключения договора **27** апреля 20 **16** г.

Страхователю выданы перечень представителей страховщика в субъектах Российской Федерации согласно приложению и два бланка извещения о дорожно-транспортном происшествии.


**12** — Страхователь: **27** апреля 20 **16** г. (дата выдачи полиса)

Страховщик / представитель страховщика  
**Маргаров Д.Э.** (подпись) (фамилия, имя, отчество)

**13** —

Рисунок 1 - пример страхового полиса ОСАГО

Изм.	Лист	№ докум.	Подпись	Дата

 <b>ВСК СТРАХОВОЙ ДОМ</b> ВОЕННО-СТРАХОВАЯ КОМПАНИЯ MILITARY INSURANCE COMPANY			
<b>СТРАХОВОЙ ПОЛИС</b>		№ <b>10868C5000</b>	<b>INSURANCE POLICY</b>
<p>Настоящим Полисом подтверждается заключение Договора страхования транспортного средства (ТС) и дополнительного оборудования, установленного на нем, на условиях Полиса добровольного страхования автотранспорта, гражданской ответственности владельца транспортного средства и водителя и пассажиров от несчастного случая ОАО "ВСК" от 28.03.2008 г., являющегося неотъемлемой частью Договора.</p>			
<b>ВИД ПОЛИСА / Л-Како</b>			
<b>ДОГОВОР ЗАКЛЮЧАЕТСЯ В:</b> <input checked="" type="checkbox"/> Рубли РФ <input type="checkbox"/> в валютном эквиваленте: <input type="checkbox"/> злв. Долларов США <input type="checkbox"/> злв. Евро			
<b>СТРАХОВАТЕЛЬ</b>			
Фамилия Имя Отчество _____			
Адрес места жительства, контактный телефон _____			
Паспортные данные _____			
Выгодоприобретатель _____			
<b>ТРАНСПОРТНОЕ СРЕДСТВО (ТС)</b>			
Марка и модель _____	<b>MAZDA 6</b>	Паспорт ТС _____	<b>78 УЕ</b>
Идентификационный № (VIN) _____	_____	Экз. выдана ТС _____	<b>2008</b>
Кузов (высота) № _____	_____	Моделность, п.с. _____	<b>120</b>
Номер двигателя _____	<b>20260576</b>	Регистрационный знак _____	_____
На момент заключения заявления ТС: <input checked="" type="checkbox"/> Оборудовано: <input type="checkbox"/> Подлежит оборудованию в течение 10 дней с даты начала действия Полиса			
<input checked="" type="checkbox"/> механическим противотуманным устройством (наименование) _____ <input checked="" type="checkbox"/> электронными противотуманными устройствами (наименование) _____ Шипы/протекторы _____ <input checked="" type="checkbox"/> спутниковой охранно-поисковой системой (наименование) _____			
Лица, допущенные к управлению ТС: <input checked="" type="checkbox"/> Водителем согласно перечню: _____ / <input type="checkbox"/> Любое лицо, допущенное к управлению ТС на законных основаниях			
Фамилия Имя Отчество _____ Дата рождения _____ Срок validity _____			
1. _____	_____	_____	_____
2. _____	_____	_____	_____
3. _____	_____	_____	_____
4. _____	_____	_____	_____
5. _____	_____	_____	_____
<b>ДОПОЛНИТЕЛЬНОЕ ОБОРУДОВАНИЕ (ДО)</b>			
Наименование объекта ДО _____	Страховая сумма по риску "Автострахование" _____	Наименование объекта ДО _____	Страховая сумма по риску "Автострахование" _____
_____	_____	_____	_____
_____	_____	_____	_____
<b>Страхование транспортного средства и дополнительного оборудования осуществляется по рискам:</b>			
Риск	Общая страховая сумма	Безусловная франшиза	Страховая премия
Автострахование	ТС <b>650 000,00</b>	не установлена	<b>50 245,00</b>
Внешнее воздействие	ЛО <b>32 500,00</b>	не установлена	
Возмещение ущерба ТС (ДО) в результате любого внешнего механического воздействия производится без предоставления подтверждающих документов комплектных органов в пределах страховой суммы, установленной по риску "Внешние воздействия". При этом лимит ответственности Страховщика по каждому объекту ДО устанавливается в размере страховой суммы по риску "Автострахование", установленной для данного объекта, не в любом случае не выше страховой суммы по риску "Внешние воздействия".			
<b>СТРАХОВАНИЕ ГРАЖДАНСКОЙ ОТВЕТСТВЕННОСТИ ВЛАДЕЛЬЦА ТРАНСПОРТНОГО СРЕДСТВА</b>			
Страховая сумма _____	Страховая премия _____	В соответствии с п. 8.3.7. Правил, выплата страхового возмещения производится с учетом произведенного страхового возмещения по договору обязательного страхования гражданской ответственности владельца ТС	
<b>СТРАХОВАНИЕ ВОДИТЕЛЯ ИЛИ ПАССАЖИРОВ ОТ НЕЩАСНОГО СЛУЧАЯ</b>			
Лимит ответственности на одно застрахованное место:		Временная утрата трудоспособности: <input type="checkbox"/> Застрахована <input type="checkbox"/> Не застрахована	Страховая премия _____
Установлен в размере: <input type="checkbox"/> Не установлен (паушальная система)			
весь салон _____	_____		
на месте водителя _____	_____		
на место(а) рядом с водителем (кол-во) _____	_____		
на места пассажиров _____	(кол-во) _____		

## 1.2 Обоснование выбора средств реализации

Чтобы разработать БД Страхового агентства выбрана СУБД MySQL. MySQL является надежной базой данных для любых целей, может продолжаться расширяться по мере наполнения информацией, без заметного уменьшения быстродействия операций с записями в многопользовательском режиме. Обеспечивается максимальная безопасность. Техническое обслуживание MySQL очень простое и не требует больших знаний.

В качестве среды для разработки прикладной программы для работы с созданной в MySQL базой данных, была выбрана среда программирования IntelliJ IDEA, язык программирования Java. IntelliJ IDEA предоставляет массу возможностей для быстрой и эффективной разработки: умное автодополнение, анализ кода в реальном времени и надежные рефакторинги.

Для более быстрого и легкого создания приложения был использован фреймворк Spring. Spring — это набор взаимосвязанных мини-фреймворков, созданных для работы над разными частями приложения. Подключать их можно по отдельности, в зависимости от задач.

### 1.3 Функциональные возможности

Пользоваться системой страхового агентства будут сотрудники, которые будут иметь возможность просмотра, добавления, изменения и удаления данных, а также пользователи, у которых есть возможность посмотреть все оформленные полисы.

Программа должна содержать следующие функциональные возможности:

- добавление данных о страхователях, сотрудниках, автомобилях, лицах, допущенных к управлению;
- добавление полисов;
- возможность работы со страховыми случаями;
- возможность изменения информации;
- возможность удаления выбранной информации;
- возможность анализировать количество и суммы заключенных договоров по каждому из видов, а также оценивать риски, подсчитывая суммы страховых выплат по каждому виду договоров, а также составлять финансовый отчет деятельности компании за заданный период времени.

## 2. Разработка моделей данных

Модели базы данных разрабатываются, чтобы определить логическую структуру базы данных и каким образом данные могут храниться, организовываться и обрабатываться. Разработка моделей данных является очень важным этапом в разработке АИС, на котором выделяются сущности, атрибуты сущностей и связи между ними. Необходимо разработать концептуальную, логическую и физическую модели данных.

Но прежде, чем разрабатывать модели данных нужно выявить ограничения предметной области. В данной предметной области существуют следующие ограничения:

- страхователь может оформлять несколько полисов;
- полис оформляется одним страхователем;
- полис оформляется на один автомобиль;
- на каждый автомобиль можно оформить несколько полисов;
- одновременно на автомобиль могут быть оформлены полисы ОСАГО и КАСКО. Пока не пройдет срок действия ОСАГО нельзя оформлять ещё один полис ОСАГО. Аналогично с КАСКО;
- у автомобиля должна быть хотя бы одна фотография, но есть возможность закреплять за каждым автомобилем сразу несколько фотографий;
- каждая фотография может закрепляться только за одним автомобилем;
- полис не перестаёт действовать после наступления страхового случая, поэтому за каждым полисом могут быть закреплены несколько страховых случаев;
- страховой случай закрепляется за одним конкретным полисом;
- при оформлении полиса должен указываться сотрудник, который его оформил;
- сотрудник может оформить несколько полисов;

					МИВУ 09.03.04-8.000 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

– в полисе должно указываться хотя бы одно лицо, допущенное к управлению, но есть возможность закреплять за каждым полисом сразу несколько лиц, допущенных к управлению;

– лицо, допущенное к управлению, может быть закреплено за несколькими полисами;

– если страхователь оформляет полис, то он не является лицом, допущенным к управлению по умолчанию, поэтому при необходимости его тоже нужно добавлять в данный список.

## 2.1 Концептуальная модель данных

Концептуальная модель — это отражение предметной области, для которой разрабатывается база данных. Так, все сущности обозначаются в виде прямоугольника. Атрибуты, характеризующие объект - в виде овала, а связи между объектами - ромбами. Мощность связи обозначаются стрелками (в направлении, где мощность равна многим - двойная стрелка, а со стороны, где она равна единице - одинарная).

Концептуальная модель, разработанная для БД Страхового агентства представлена в приложении 1 на рисунке 3. Объектами на разработанной модели являются Страхователи, Автомобили, Фотографии, Полисы, Лица, допущенные к управлению, Страховые случаи, Сотрудники. Страхователь может оформить полис, который включает Лица, допущенные к управлению, Страховые случаи и Автомобили, у которых есть изображения. Полис регистрирует Сотрудник.

Страхователи

Dom (ФИО) = {строка символов длиной не более 64}

Dom (Дата рождения) = {строка символов длиной 10, символами являются быть цифры или «-»}

					МИВУ 09.03.04-8.000 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

Dom (Телефон) = {строка символов длиной не более 15, символами являются цифры}

Dom (Email) = {строка символов длиной не более 30}

Dom (Паспортные данные) = {строка символов длиной 10, символами являются цифры}

#### Автомобили

Dom (Марка, модель) = {строка символов длиной не более 50}

Dom (Идентификационный номер) = {строка символов длиной 17, символами являются цифры и буквы}

Dom (Регистрационный знак) = {строка символов длиной не более 25, символами являются цифры и буквы}

Dom (Паспорт ТС) = {строка символов длиной 10, символами являются цифры и буквы}

#### Изображения

Dom (Путь) = {строка символов длиной не более 100}

Dom (Дата загрузки) = {строка символов длиной 10, символами являются быть цифры или «-»}

#### Полисы

Dom (Вид страхования) = {строка символов длиной 5, символами являются буквы}

Dom (Страховая сумма) = {строка символов длиной не более 10, символами являются цифры}

Dom (Страховая премия) = {строка символов длиной не более 10, символами являются цифры}

Dom (Дата заключения) = {строка символов длиной 10, символами могут быть цифры или «-»}

					МИВУ 09.03.04-8.000 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Dom (Дата окончания действия) = {строка символов длиной 10, символами могут быть цифры или «-»}

Лица, допущенные к управлению

Dom (ФИО) = {строка символов длиной не более 64}

Dom (Водительское удостоверение) = {строка символов длиной 10, символами являются цифры}

Страховые случаи

Dom (Дата) = {строка символов длиной 10, символами являются цифры или «.»}

Dom (Страховая выплата) = {строка символов длиной не более 50, символами являются цифры}

Dom (Описание) = {строка символов длиной 1000}

Сотрудники

Dom (ФИО) = {строка символов длиной не более 64}

Dom (Дата рождения) = {строка символов длиной 10, символами являются цифры или «.»}

Dom (Телефон) = {строка символов длиной 15, символами являются цифры}

Dom (Email) = {строка символов длиной не более 30}

Dom (Паспортные данные) = {строка символов длиной 10, символами являются цифры}

## 2.2 Логическая модель данных

Целью построения логической модели является получение графического представления логической структуры исследуемой предметной области. Логическая модель предметной области иллюстрирует сущности, а также их взаимоотношения между собой.

					МИВУ 09.03.04-8.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

Логическая модель должна читаться по схеме: <Сущность 1> — <отношение / влияние> — <Сущность 2>. Чтение логической модели, представленной в приложении 1 на рисунке 4: Страхователь оформляет полис. Полис включает страховые случаи и автомобиль. Автомобиль включает фотографии. Полис регистрирует сотрудник. Полис включает лица, допущенные к управлению (соединение через дополнительную таблицу, так как связь “многие ко многим”).

### 2.3 Соответствие сущностей логической и физической моделей

Построение физической модели БД производится на основе логической модели. В таблицах 1-8 описано соответствие сущностей логической модели и таблиц физической.

Таблица 1 - Таблица Страхователи(Policyholders)

Сущность	Таблица
ID	ID
ФИО	FullName
Дата рождения	Birthday
Телефон	Telephone
Паспортные данные	Passport

Таблица 2 - Таблица Автомобили(Cars)

Сущность	Таблица
ID	ID
Марка, модель	Model
Идентификационный номер	VIN
Регистрационный знак	RegistrationPlate
Паспорт ТС	VehiclePassport



Таблица 3 - Таблица Фотографии(Photos)

Сущность	Таблица
ID	ID
Путь	Path
Дата загрузки	UploadDate
ID автомобиля	Car_ID

Таблица 4 - Таблица Полисы(Policies)

Сущность	Таблица
ID	ID
Вид страхования	InsuranceType
Страховая премия	InsurancePremium
Страховая сумма	InsuranceAmount
Дата заключения	DateOfConclusion
Дата окончания действия	ExpirationDate
ID страхователя	Policyholder_ID
ID автомобиля	Car_ID
ID сотрудника	Employee_ID

Таблица 5 - Таблица Соединения(Connections)

Сущность	Таблица
ID полиса	Policy_ID
ID лица, допущенного к управлению	PersonAllowedToDrive_ID

Таблица 6 - Таблица Лица, допущенные к управлению  
(PersonsAllowedToDrive)

Сущность	Таблица
ID	ID
ФИО	FullName
Водительское удостоверение	DrivingLicence

Таблица 7 - Таблица Страховые случаи (InsuranceEvents)

Сущность	Таблица
ID	ID
Дата	Date
Страховая выплата	InsurancePayment
Описание	Discription
ID полиса	Policy_ID

Таблица 8 - Таблица Сотрудники(Employees)

Сущность	Таблица
ID	ID
ФИО	FullName
Дата рождения	Birthday
Телефон	Telephone
Паспортные данные	Passport
Логин	Login
Пароль	Password

Подробное описание каждой сущности приведено в таблицах 9-16.

Также в данных таблица приведён расчет памяти, необходимой для хранения одной записи.

Таблица 9 - Таблица Policyholders

Имя столбца	Тип	Размер(байт)
ID	INT	4
FullName	VARCHAR(64)	64
Birthday	DATE	3
Telephone	VARCHAR(15)	15
Email	VARCHAR(30)	30
Passport	VARCHAR(10)	10
Итого		126

Таблица 10 - Таблица Connections

Имя столбца	Тип	Размер(байт)
Policy_ID	INT	4
PersonAllowedToDrive_ID	INT	4
Итого		8

Таблица 11 - Таблица Cars

Имя столбца	Тип	Размер(байт)
ID	INT	4
Model	VARCHAR(50)	50
VIN	VARCHAR(17)	17
RegistrationPlate	VARCHAR(25)	25
VehiclePassport	VARCHAR(10)	10
Итого		106

Таблица 12 - Таблица Photos

Имя столбца	Тип	Размер(байт)
ID	INT	4
Path	VARCHAR(20)	20
uploadDate	DATE	3
Car_ID	INT	4
Итого		31

Таблица 13 - Таблица PersonsAllowedToDrive

Имя столбца	Тип	Размер(байт)
ID	INT	4
FullName	VARCHAR(64)	64
DrivingLicence	VARCHAR(10)	10
Итого		78

Таблица 14 - Таблица Policies

Имя столбца	Тип	Размер(байт)
ID	INT	4
InsuranceType	VARCHAR(5)	5
InsurancePremium	INT	4
InsuranceAmount	INT	4
DateOfConclusion	DATE	3
ExpirationDate	DATE	3
Policyholder_ID	INT	4
Car_ID	INT	4
Employee_ID	INT	4
Итого		35

Таблица 15 - Таблица InsuranceEvents

Имя столбца	Тип	Размер(байт)
ID	INT	4
Date	DATE	3
InsurancePayment	INT	4
Description	VARCHAR(1000)	1000
Policy_ID	INT	4
Итого		1015

Таблица 16 - Таблица Employees

Имя столбца	Тип	Размер(байт)
ID	INT	4
FullName	VARCHAR(64)	64
Birthday	DATE	3
Telephone	VARCHAR(15)	15
Email	VARCHAR(30)	30
Passport	VARCHAR(10)	10
Итого		126

Определим объем внешней памяти, необходимой для размещения данных за год использования. Для того чтобы оценить объем, занимаемый таблицами базы данных, необходимо оценить объем каждой таблицы. Примерный расчет необходимо объема памяти приведен в таблице 17.

Таким образом, при максимальном заполнении БД объем таблиц составит:  $V_{\text{данных}} = 482,441 \text{ Кбайт/год}$ .

Таблица 17 - Расчет объема ПЗУ для хранения данных

Таблица	Размер записи, байт	Максимальное (оценочное) количество записей	Всего, Кбайт
Policyholders	126	700	86,133
Cars	106	700	72,461
Photos	31	2000	60,546
Connections	8	1700	13,281
Policies	35	1000	34,180
PersonsAllowedToDrive	78	1500	114,258
InsuranceEvents	1015	100	99,121
Employees	126	20	2,461
Итого			482,441

## 2.4 Физическая модель данных

Физическая модель данных, зависит от конкретной СУБД, фактически являясь отображением системного каталога. Физическая модель БД определяет способ размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне.

Физическая модель, разработанная для БД Страхового агентства представлена в приложении 1 на рисунке 5.

### 3. Проектирование работы системы

#### 3.1 Диаграмма вариантов использования

Создав диаграмму вариантов использования, которая представлена в приложении 1 на рисунке 6, были сформулированы общие требования к функциональному поведению проектируемой системы и определены общие границы и контекст моделируемой предметной области.

Актерами в диаграмме являются Пользователь, Оператор и Администратор. В самой диаграмме отображен план работы с ИС Страхового агентства. Пользователь, Оператор и Администратор – внешняя часть программы. Основной частью программы является:

- общее: авторизация и просмотр профиля;
- для пользователя: просмотр всех своих полисов;
- для оператора и администратора: работа с полисом, работа с автомобилем, работа с лицом, допущенным к управлению, работа со страхователем, работа с отчётами;
- для администратора: работа с сотрудником.

#### 3.2 Функциональная диаграмма

Создадим функциональную диаграмму. Представим всю систему в виде простейшей компоненты – одного блока и дуг, изображающих интерфейсы с функциями вне системы. Созданная функциональная диаграмма показана на рисунке 7 в приложении 1.

Управляющая информация (нормативы, формы отчётов) входит в блок сверху, в то время как информация, которая подвергается обработке (данные пользователя, информация об автомобиле, информация о страхователе, информация о лице, допущенном к управлению, информация о сотруднике), показана с левой стороны блока, а результаты выхода (скаченный файл) показаны с правой стороны. Механизмы (пользователь, оператор,

администратор), которые осуществляют операцию, представляются дугой, входящей в блок снизу.

Для получения подробной информации о функционировании ИС необходимо создать подробную функциональную диаграмму. Она показана на рисунке 8 в приложении 1.

Пользователь передает данные для авторизации в системе. После авторизации при правильном вводе логина и пароля пользователь получает право доступа в приложение и ему назначается определённая роль: “Пользователь”, “Оператор” или “Администратор”. Пользователь может работать с автомобилями, страхователями, лицами, допущенными к управлению, сотрудниками, полисами или отчётами. При работе с отчётами, пользователь получает загруженный отчёт.

Функциональные диаграммы “Работа со страхователем” и “Работа с полисом”, которые показаны на рисунках 9 - 10 в приложении 1, раскрывают функционирование этих блоков. Также “Работа с автомобилем”, “Работа с лицами, допущенными к управлению” и “Работа с сотрудниками” похожи на диаграмму “Работа со страхователями”.

### 3.3 Диаграммы последовательностей

Диаграммы, представленные на рисунках 11 - 13 в приложении 1 изображают диаграммы последовательностей и отображают временные особенности передачи и приема сообщений объектами.

На рисунке 11 в приложении 1 представлена диаграмма последовательности при авторизации пользователя. Пользователь передаёт логин и пароль для ИС, а ИС передает эту информацию БД, в которой происходит поиск этого пользователя. После поиска БД передает результаты ИС, а ИС предоставляет права доступа пользователю.

На рисунке 12 в приложении 1 представлена диаграмма последовательности при работе со страхователем. Также работа с

автомобилем, работа с лицами, допущенными к управлению, и работа с сотрудниками похожи на диаграмму последовательности при работе со страхователями.

При добавлении страхователя в БД пользователь передаёт данные о страхователе, а ИС формирует запрос на добавление и отправляет его в БД, в которой происходит добавление этого страхователя. БД передаёт результат добавления, а ИС формирует сообщение о результате добавления.

При поиске страхователя в БД пользователь передаёт данные о страхователе, а ИС формирует запрос поиска и отправляет его в БД, в которой происходит поиск этого страхователя. БД передаёт результат поиска, а ИС возвращает найденного страхователя или сообщение об ошибке.

При изменении страхователя в БД пользователь передаёт выбранного страхователя и измененные данные, а ИС формирует запрос на изменение и отправляет его в БД, в которой происходит изменение этого страхователя. БД передаёт результат изменения, а ИС формирует сообщение о результате изменения.

При удалении страхователя из БД пользователь передаёт выбранного страхователя, а ИС формирует запрос на удаление и отправляет его в БД, в которой происходит удаление этого страхователя. БД передаёт результат удаления, а ИС формирует сообщение о результате удаления.

На рисунке 13 в приложении 1 представлена диаграмма последовательности при работе с полисом. Она похожа на диаграмму последовательности при работе со страхователями, но есть несколько отличающихся моментов.

После добавления полиса в БД происходит добавление лиц, допущенных к управлению. Пользователь передаёт список лиц, допущенных к управлению, а ИС формирует запрос на добавление лиц, допущенных к управлению, к полису и отправляет его в БД, в которой происходит добавление. БД передаёт результат добавления, а ИС формирует сообщение о результате добавления.



После выбора полиса в БД можно добавить ему страховой случай. Пользователь передаёт выбранный полис и данные о страховом случае, а ИС формирует запрос на добавление страхового случая к полису и отправляет его в БД, в которой происходит добавление. БД передаёт результат добавления, а ИС формирует сообщение о результате добавления.

После выбора полиса в БД можно изменить список лиц, допущенных к управлению, прикрепленных к полису. Пользователь передаёт выбранный полис и изменённый список лиц, допущенных к управлению, а ИС формирует запрос на изменение лиц, допущенных к управлению, прикрепленных к полису и отправляет его в БД, в которой происходит изменение. БД передаёт результат изменения, а ИС формирует сообщение о результате изменения.

					МИВУ 09.03.04-8.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

## 4. Разработка и реализация системы

### 4.1 Добавление

Рассмотрим механизм добавления на примере страхователя. Сначала GET метод выводит пользователю представление Create, которое показано на рисунке 28 в приложении 3.

```
@GetMapping("/create")
public String createGet(Model model) {
    model.addAttribute("policyholder", new Policyholder());
    return "policyholder/create";
}
```

После заполнения данной формы в методе POST происходит проверка правильности введённых данных. При успешной проверке в БД добавляется страхователь, иначе происходит вывод ошибок. Также после успешного добавления пользователь переходит на форму с подробной информацией о страхователе.

```
@PostMapping("/create")
public String createPost(@ModelAttribute("policyholder") @Valid Policyholder
policyholder, BindingResult bindingResult) {
    checkForUniqueness(policyholder, bindingResult);
    if (bindingResult.hasErrors()) {
        return "policyholder/create";
    }

    policyholder = policyholderService.create(policyholder);
    return "redirect:/policyholder/details/" + policyholder.getId();
}

private void checkForUniqueness(Policyholder policyholder, BindingResult
bindingResult) {
    if (checkTelephone(policyholder)) {
        bindingResult.addError(new FieldError(
            "policyholder", "telephone",
            policyholder.getTelephone(),
            false, null, null,
            "Данный Телефон уже используется"
        ));
    }
    if (checkEmail(policyholder)) {
        bindingResult.addError(new FieldError(
            "policyholder", "email",
            policyholder.getEmail(),
            false, null, null,
            "Данный Email уже используется"
        ));
    }
}
```

					МИВУ 09.03.04-8.000 ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

```

        );
    }
    if (checkPassport(policyholder)) {
        bindingResult.addError(new FieldError(
            "policyholder", "passport",
            policyholder.getPassport(),
            false, null, null,
            "Данный Паспорт уже используется"
        ));
    }
}

private boolean checkTelephone(Policyholder policyholder) {
    if (policyholder.getId() == null) {
        return
        policyholderService.checkTelephone(policyholder.getTelephone());
    } else {
        return
        policyholderService.checkTelephoneExceptId(policyholder.getId(),
        policyholder.getTelephone());
    }
}

private boolean checkEmail(Policyholder policyholder) {
    if (policyholder.getId() == null) {
        return policyholderService.checkEmail(policyholder.getEmail());
    } else {
        return policyholderService.checkEmailExceptId(policyholder.getId(),
        policyholder.getEmail());
    }
}

private boolean checkPassport(Policyholder policyholder) {
    if (policyholder.getId() == null) {
        return policyholderService.checkPassport(policyholder.getPassport());
    } else {
        return
        policyholderService.checkPassportExceptId(policyholder.getId(),
        policyholder.getPassport());
    }
}

```

## 4.2 Изменение

Рассмотрим механизм изменения на примере страхователя. Сначала GET метод выводит пользователю представление Edit, в которое передается модель Policyholder - страхователь, которого нужно изменить.

```

@GetMapping("/edit/{id}")
public String editGet(Model model, @PathVariable("id") Long id) {
    model.addAttribute("policyholder", policyholderService.getById(id));
    return "policyholder/edit";
}

```

После заполнения данной формы в методе POST происходит проверка правильности введенных данных, как и при добавлении. При успешной проверке в БД изменяется страхователь, иначе происходит вывод ошибок.

```
@PostMapping("/edit")
public String editPost(@ModelAttribute("policyholder") @Valid Policyholder
policyholder, BindingResult bindingResult) {
    checkForUniqueness(policyholder, bindingResult);
    if (bindingResult.hasErrors()) {
        return "policyholder/edit";
    }

    policyholderService.edit(policyholder);
    return "redirect:/policyholder/details/" + policyholder.getId();
}
```

### 4.3 Удаление

Рассмотрим механизм добавления на примере автомобиля. Сначала GET метод выводит пользователю представление Delete, в которое передается модель Car - автомобиль, который нужно удалить. Эта форма является формой подтверждения. Данная страница показана на рисунке 26 в приложении 3.

```
@GetMapping("/delete/{id}")
public String deleteGet(Model model, @PathVariable("id") Long id) {
    Car car = carService.getById(id);
    model.addAttribute("car", car);
    model.addAttribute("hasPolicies", car.getPolicies().size() != 0);
    return "car/delete";
}
```

С учётом того, что на автомобилях могут быть оформлены полисы, то их нельзя удалять, иначе удалится сам полис. Запрет на удаление также представлен на рисунке 26 в приложении 3.

Удаление автомобиля:

```
@PostMapping("/delete")
public String deletePost(Model model, @ModelAttribute("car") Car car) {
    Car carDelete = carService.getById(car.getId());
    if (carDelete.getPolicies().size() != 0) {
        model.addAttribute("car", carDelete);
        model.addAttribute("hasPolicies", true);
        return "car/delete";
    }
}
```

					МИВУ 09.03.04-8.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

```

    }

    for (var photo: carDelete.getPhotos()) {
        photoService.delete(photo.getId());
    }
    carService.delete(carDelete.getId());
    return "redirect:/car/list";
}

```

#### 4.4 Поэтапное добавление полиса

На первом этапе происходит выбор страхователя из списка. Это продемонстрировано на рисунке 16 в приложении 3. После выбора страхователя его id передаётся на следующий этап.

```

@GetMapping()
public String createGet(Model model) {
    model.addAttribute("policyholders", policyholderService.list());
    return "policy/create/choosePolicyholder";
}

@GetMapping("/choosePolicyholder/{policyholderId}")
public String createChoosePolicyholder(Model model,
    @PathVariable("policyholderId") Long policyholderId) {
    try {
        Policyholder policyholder =
            policyholderService.getById(policyholderId);
    } catch (Exception exp) {
        model.addAttribute("policyholders", policyholderService.list());
        return "policy/create/choosePolicyholder";
    }

    model.addAttribute("policyholderId", policyholderId);
    model.addAttribute("cars", carService.list());
    return "policy/create/chooseCar";
}

```

На втором этапе происходит выбор автомобиля из списка. Это продемонстрировано на рисунке 17 в приложении 3. После выбора автомобиля его id и id страхователя передаются на следующий этап. Также тут получается id сотрудника, который авторизован в системе.

```

@GetMapping("/chooseCar/{policyholderId}/{carId}")
public String createChooseCar(Model model, @PathVariable("policyholderId")
    Long policyholderId, @PathVariable("carId") Long carId) {
    Policyholder policyholder;
    Car car;
    try {
        policyholder = policyholderService.getById(policyholderId);
    }
}

```

					МИВУ 09.03.04-8.000 ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

```

    } catch (Exception exp) {
        model.addAttribute("policyholders", policyholderService.list());
        return "policy/create/choosePolicyholder";
    }
    try {
        car = carService.getById(carId);
    } catch (Exception exp) {
        model.addAttribute("policyholderId", policyholderId);
        model.addAttribute("cars", carService.list());
        return "policy/create/chooseCar";
    }

    Policy policy = new Policy();

    policy.setExpirationDate(LocalDate.now());
    policy.setPolicyholder(policyholder);
    policy.setCar(car);

    Authentication authentication =
    SecurityContextHolder.getContext().getAuthentication();
    String email = authentication.getName();
    Employee employee = employeeService.getByEmail(email);
    policy.setEmployee(employee);

    model.addAttribute("policy", policy);
    return "policy/create/chooseInfo";
}

```

На третьем этапе происходит заполнение информации о полисе. Это продемонстрировано на рисунке 18 в приложении 3. Если вся информация заполнена верно, то происходит переход на следующий этап с передачей полиса.

```

@PostMapping("/chooseInfo")
public String createChooseInfo(Model model, @ModelAttribute("policy") @Valid
Policy policy, BindingResult bindingResult) {
    if (policy.getPolicyholder() == null) {
        model.addAttribute("policyholders", policyholderService.list());
        return "policy/create/choosePolicyholder";
    }
    if (policy.getCar() == null) {
        model.addAttribute("policyholderId",
policy.getPolicyholder().getId());
        model.addAttribute("cars", carService.list());
        return "policy/create/chooseCar";
    }
    if (policy.getEmployee() == null) {
        return "redirect:/policy/list";
    }

    if (bindingResult.hasErrors()) {
        return "policy/create/chooseInfo";
    }

    if (policy.getExpirationDate().isEqual(LocalDate.of(2006, 6, 6))) {
        policy.setExpirationDate(policy.getDateOfConclusion().plusMonths(6));
    } else if (policy.getExpirationDate().isEqual(LocalDate.of(2012, 12,

```

					МИВУ 09.03.04-8.000 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

```

12))) {
    policy.setExpirationDate(policy.getDateOfConclusion().plusYears(1));
} else {
    bindingResult.addError(new FieldError(
        "policy", "expirationDate",
        policy.getExpirationDate(),
        false, null, null,
        "Некорректная дата заключения"
    ));
}
checkForErrors(policy, bindingResult);
if (bindingResult.hasErrors()) {
    return "policy/create/chooseInfo";
}

model.addAttribute("policy", policy);
model.addAttribute("listPersonsAllowedToDrive",
    personAllowedToDriveService.list());
return "policy/create/choosePersonsAllowedToDrive";
}

```

На четвертом этапе происходит выбор лиц, допущенных к управлению. Это продемонстрировано на рисунке 19 в приложении 3. При условии, что выбрано хотя бы одно лицо, допущенное к управлению, происходит добавление полиса. После добавления происходит переход на форму с подробной информацией о добавленном полисе.

```

@PostMapping("/choosePersonsAllowedToDrive")
public String editPersonsAllowedToDrivePost(Model model,
    @ModelAttribute("policy") @Valid Policy policy, BindingResult bindingResult)
{
    if (policy.getPolicyholder() == null) {
        model.addAttribute("policyholders", policyholderService.list());
        return "policy/create/choosePolicyholder";
    }
    if (policy.getCar() == null) {
        model.addAttribute("policyholderId",
            policy.getPolicyholder().getId());
        model.addAttribute("cars", carService.list());
        return "policy/create/chooseCar";
    }
    if (policy.getEmployee() == null) {
        return "redirect:/policy/list";
    }

    if (policy.getPersonsAllowedToDrive().size() == 0) {
        bindingResult.addError(new FieldError(
            "policy", "personsAllowedToDrive",
            policy.getPersonsAllowedToDrive(),
            false, null, null,
            "Выберите хотя бы одно лицо, допущенное к управлению"
        ));
    }
    if (bindingResult.hasErrors()) {
        model.addAttribute("policy", policy);
    }
}

```

```

        model.addAttribute("listPersonsAllowedToDrive",
personAllowedToDriveService.list());
        return "policy/create/choosePersonsAllowedToDrive";
    }

    policy = policyService.create(policy);
    return "redirect:/policy/details/" + policy.getId();
}

```

## 4.5 Изображения

Пользователь может загрузить фотографию для автомобиля (продемонстрировано на рисунке 25 в приложении 3). В базе данных хранятся только пути к фотографии, поэтому при добавлении фотографии автомобилю, она сохраняется по специальному жёстко заданному пути.

```

public static String uploadImage(MultipartFile file, Long carId) throws
IOException {
    Path fileNameAndPath = Paths.get(UPLOAD_DIRECTORY + "\\\" + carId,
file.getOriginalFilename());
    Files.write(fileNameAndPath, file.getBytes());
    return file.getOriginalFilename();
}

```

Получение фотографии происходит по пути, который был получен из БД.

```

public static List<PhotoLayout> getPhotos(Car car) throws IOException {
    List<PhotoLayout> photos = new ArrayList<>();
    for (var photoDB : car.getPhotos()) {
        String pathFile = UPLOAD_DIRECTORY + "\\\" + car.getId() + "\\\" +
photoDB.getPath();
        PhotoLayout photoLayout = new PhotoLayout(photoDB.getId(),
toBase64(pathFile), photoDB.getUploadDate());
        photos.add(photoLayout);
    }
    return photos;
}

private static String toBase64(String path) throws IOException {
    File file = new File(path);
    byte[] fileContent = Files.readAllBytes(file.toPath());
    Base64 base64 = new Base64();
    byte[] encodeBase64 = base64.encode(fileContent);
    return new String(encodeBase64, StandardCharsets.UTF_8);
}

```



## 4.6 Отчёты

Чтобы сформировать финансовый отчёт нужно, чтобы пользователь выбрал вид страхования: ОСАГО, КАСКО или ОСАГО и КАСКО, а также дату начала и дату окончания - период, за который будет сформирован отчёт.

```
@PostMapping()
public String reportPost(Model model, @ModelAttribute("report") @Valid Report
report, BindingResult bindingResult) {
    if (bindingResult.hasErrors()) {
        return "report";
    }
    if (report.getStartDate().isAfter(report.getEndDate())) {
        bindingResult.addError(new FieldError(
            "report", "startDate",
            report.getStartDate(),
            false, null, null,
            "Дата начала не может быть больше даты окончания"
        ));
        return "report";
    }

    if (report.getInsuranceType().equals("all")) {
        List<Policy> listPolicies = policyService.list();
        List<InsuranceEvent> listInsuranceEvents =
insuranceEventService.list();

        listPolicies.removeIf(value ->
value.getDateOfConclusion().isBefore(report.getStartDate()) ||
value.getDateOfConclusion().isAfter(report.getEndDate()));
        int countPolicies = listPolicies.size();
        int sumInsurancePremium = 0;
        for (var item : listPolicies) {
            sumInsurancePremium += item.getInsurancePremium();
        }

        listInsuranceEvents.removeIf(value ->
value.getIncidentDate().isBefore(report.getStartDate()) ||
value.getIncidentDate().isAfter(report.getEndDate()));
        int sumInsurancePayment = 0;
        for (var item : listInsuranceEvents) {
            sumInsurancePayment += item.getInsurancePayment();
        }

        model.addAttribute("report", report);
        model.addAttribute("countPolicies", countPolicies);
        model.addAttribute("sumInsurancePremium", sumInsurancePremium);
        model.addAttribute("sumInsurancePayment", sumInsurancePayment);
        return "report";
    } else {
        List<Policy> listPolicies = policyService.list();
        List<InsuranceEvent> listInsuranceEvents =
insuranceEventService.list();

        listPolicies.removeIf(value ->
!value.getInsuranceType().equals(report.getInsuranceType()) ||
```

					МИВУ 09.03.04-8.000 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

```

value.getDateOfConclusion().isBefore(report.getStartDate()) ||
value.getDateOfConclusion().isAfter(report.getEndDate()));
    int countPolicies = listPolicies.size();
    int sumInsurancePremium = 0;
    for (var item : listPolicies) {
        sumInsurancePremium += item.getInsurancePremium();
    }

    listInsuranceEvents.removeIf(value ->
!value.getPolicy().getInsuranceType().equals(report.getInsuranceType()) ||
value.getIncidentDate().isBefore(report.getStartDate()) ||
value.getIncidentDate().isAfter(report.getEndDate()));
    int sumInsurancePayment = 0;
    for (var item : listInsuranceEvents) {
        sumInsurancePayment += item.getInsurancePayment();
    }

    model.addAttribute("report", report);
    model.addAttribute("countPolicies", countPolicies);
    model.addAttribute("sumInsurancePremium", sumInsurancePremium);
    model.addAttribute("sumInsurancePayment", sumInsurancePayment);
    return "report";
}
}

```

					МИВУ 09.03.04-8.000 ПЗ	Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

## 5. Тестирование системы

Одним из важнейших этапов создания приложения является его тестирование и отладка. Целью тестирования является проверка работоспособности программы, правильности выполнения всех функций, а также правильности обработки всех исключений, возникающих в ходе работы программы.

Таблица 18 - Чек-лист «Тестирование приложения»

Группа тестов без названия			
№	Приоритет	Проверка	Результат
1	Средний	Запуск программы	Корректное отображение главной страницы
2	Средний	Тестирование главного меню приложения	Пользователю и сотруднику видны разные типы меню
Права пользователей			
№	Приоритет	Проверка	Результат
3	Средний	Пользователь	Пользователю доступен только просмотр своих полисов
4	Средний	Оператор	Оператор может переходить на все страницы, кроме сотрудников. Также ему не доступны удаления полисов и страховых случаев
5	Средний	Администратор	Имеет доступ ко всем страницам
Списки			
№	Приоритет	Проверка	Результат
6	Средний	Отображение списков и переход по нажатию на кнопки	Корректно отображается информация и работают кнопки для подробной информации, изменения и удаления Также над списком корректно работает кнопка для создания нового элемента
Добавление и изменение сотрудника и страхователя			
№	Приоритет	Проверка	Результат
7	Средний	Корректно заполнить все поля	Добавление/изменение сотрудника/страхователя и переход на страницу с подробной информацией о нём

**Продолжение таблицы 18 – Чек-лист «Тестирование приложения»**

			Появляются соответствующие ошибки о незаполненных поля
8	Средний	Не заполнить поле	Страхователю можно не заполнять поле email
9	Средний	Ввести ФИО превышающее 64 символа	Ошибка "ФИО не должно превышать 64 символа"
10	Средний	Ввести в поле номер телефона символы помимо цифр	Ошибка "Номер телефона может включать только цифры"
11	Средний	Ввести номер телефона превышающий 15 символов	Ошибка "Номер телефона не должен превышать 15 цифр"
12	Средний	Ввести Email превышающее 32 символа	Ошибка "Email не должен превышать 32 символа"
13	Средний	Ввести некорректный Email	Ошибка "Некорректный ввод для Email"
14	Средний	Ввести в поле паспорт символы помимо цифр	Ошибка "Паспорт может включать только цифры"
15	Средний	Ввести паспорт не длиной в 10 цифр	Ошибка "Паспорт должен содержать 10 цифр"
16	Средний	Уникальность	Номер телефона, email и паспорт должны быть уникальными, иначе появляются соответствующие ошибки

**Добавление и изменение автомобиля**

№	Приоритет	Проверка	Результат
			Добавление автомобиля и переход на страницу с выбором фотографий для автомобиля (можно не добавлять)
17	Средний	Корректно заполнить все поля	Изменение автомобиля и переход на страницу с подробной информацией о нём
18	Средний	Не заполнить поле	Появляются соответствующие ошибки о незаполненных поля
19	Средний	Ввести модель, превышающую 50 символов	Ошибка "Модель не должна превышать 50 символов"
20	Средний	Ввести VIN не длиной в 17 символов	Ошибка "VIN номер должен содержать 17 символов"
21	Средний	Ввести регистрационный знак превышающую 25 символов	Ошибка "Регистрационный знак не должен превышать 25 символов"
22	Средний	Ввести паспорт ТС не длиной в 10 символов	Ошибка "Паспорт ТС должен содержать 10 символов"
23	Средний	Уникальность	VIN должен быть уникальными, иначе появляется соответствующая ошибка

**Продолжение таблицы 18 – Чек-лист «Тестирование приложения»**

Добавление и изменение лиц, допущенных к управлению			
№	Приоритет	Проверка	Результат
24	Средний	Корректно заполнить все поля	Добавление/изменение лица, допущенного к управлению, и переход на страницу с подробной информацией о нём
25	Средний	Не заполнить поле	Появляются соответствующие ошибки о незаполненных полях
26	Средний	Ввести ФИО, превышающее 64 символа	Ошибка "ФИО не должно превышать 64 символа"
27	Средний	Ввести в поле водительское удостоверение символы помимо цифр	Ошибка "Водительское удостоверение может включать только цифры"
28	Средний	Ввести водительское удостоверение не длиной в 10 цифр	Ошибка "Водительское удостоверение должно содержать 10 цифр"
29	Средний	Уникальность	Водительское удостоверение должен быть уникальным, иначе появляется соответствующая ошибка
Добавление полиса			
№	Приоритет	Проверка	Результат
30	Средний	Отображение этапов добавления и переходы между ними	Корректное отображение этапов добавления и корректные переходы между ними 1. Выбор страхователя 2. Выбор автомобиля 3. Ввод информации 4. Выбор лиц, допущенных к управлению
31	Средний	Корректно выбрать и ввести все данные	Добавление полиса и переход на страницу с подробной информацией о нём
32	Средний	Не заполнить поле	Появляются соответствующие ошибки о незаполненных полях
33	Средний	Не выбрать ни одного лица, допущенного к управлению	Ошибка "Выберите хотя бы одно лицо, допущенное к управлению"
34	Средний	Некорректный ввод страховой премии и суммы	Ошибка "Страховая премия не может быть меньше или равна 0" Ошибка "Страховая сумма не может быть меньше или равна 0" Ошибка "Страховая премия не может быть больше или равна Страховой сумме"

**Продолжение таблицы 18 – Чек-лист «Тестирование приложения»**

Добавление страхового случая			
№	Приоритет	Проверка	Результат
35	Средний	Корректно ввести все данные	Добавление страхового случая и переход на страницу с подробной информацией о нём
36	Средний	Не заполнить поле	Появляются соответствующие ошибки о незаполненных полях
37	Средний	Ввести описание, превышающее 1000 символов	Ошибка "Описание не должно превышать 1000 символов"
38	Средний	Некорректная дата	Ошибка "Дата не может быть меньше даты заключения полиса" Ошибка "Дата не может быть больше даты окончания действия полиса"
39	Средний	Ввести страховую выплату больше страховой суммы	У полиса КАСКО ошибка "Страховая выплата должна быть меньше Страховой суммы" У полиса ОСАГО ошибок нет
Удаление			
№	Приоритет	Проверка	Результат
40	Средний	Страницы удаления	Корректно отображается информация о выбранном элементе
41	Средний	Удаление автомобилей, страхователей, сотрудников и лиц, допущенных к управлению	Доступно только если не закреплены полисы
42	Средний	Удаление страховых случаев	Доступно администратору
43	Средний	Удаление полисов	Доступно администратору при условии, что за полисом не закреплены страховые случаи

## Заключение

В данной курсовой работе в соответствии с заданием была разработана ИС Страхового агентства.

В ходе выполнения курсовой работы были выполнены следующие задачи:

- выявлены требования к программе;
- разработаны модели данных;
- создана база данных;
- разработано приложение ASP.Net;
- осуществлено ее тестирование.

Разработанная программа обеспечивает осуществление следующих функций:

- добавление данных о страхователях, сотрудниках, автомобилях, лицах, допущенных к управлению;
- добавление полисов;
- возможность работы со страховыми случаями;
- возможность изменения информации;
- возможность удаления выбранной информации;
- возможность анализировать количество и суммы заключенных договоров по каждому из видов, а также оценивать риски, подсчитывая суммы страховых выплат по каждому виду договоров, а также составлять финансовый отчет деятельности компании за заданный период времени.

## Список используемой литературы

1. Аналог Полисы ОСАГО 1.0.9: [Электронный ресурс] // URL: <https://araxgroup.ru/index.php/products/53-other-programs/271-polisi-osago> (Дата обращения – 1.10.2022)
2. Аналог Учёт автострахования ОСАГО: [Электронный ресурс] // URL: [http://bestsoft.moy.su/load/ofis\\_programmy\\_dlja\\_kompjutera\\_skachat\\_besplatno/ofis\\_programmy\\_dlja\\_kompjutera\\_skachat\\_besplatno/skachat\\_besplatno\\_programmu\\_uchjot\\_i\\_zapolnenie\\_polisov\\_avtostrakhovanija\\_osago\\_5\\_6\\_bez\\_registracii\\_i\\_sms\\_licenzionnyj\\_kljuch\\_aktivacii/11-1-0-108](http://bestsoft.moy.su/load/ofis_programmy_dlja_kompjutera_skachat_besplatno/ofis_programmy_dlja_kompjutera_skachat_besplatno/skachat_besplatno_programmu_uchjot_i_zapolnenie_polisov_avtostrakhovanija_osago_5_6_bez_registracii_i_sms_licenzionnyj_kljuch_aktivacii/11-1-0-108) (Дата обращения – 1.10.2022)
3. Руководство по языку программирования Java: [Электронный ресурс] // URL: <https://metanit.com/java/tutorial/> (Дата обращения – 11.11.2022)
4. Руководство по Spring Boot: [Электронный ресурс] // URL: <https://www.baeldung.com/spring-boot> (Дата обращения – 10.12.2022)
5. Учебник: Использование Thymeleaf: [Электронный ресурс] // URL: <https://habr.com/ru/post/350862/> (Дата обращения – 8.12.2022)



## Приложение 1. Модели данных и диаграммы

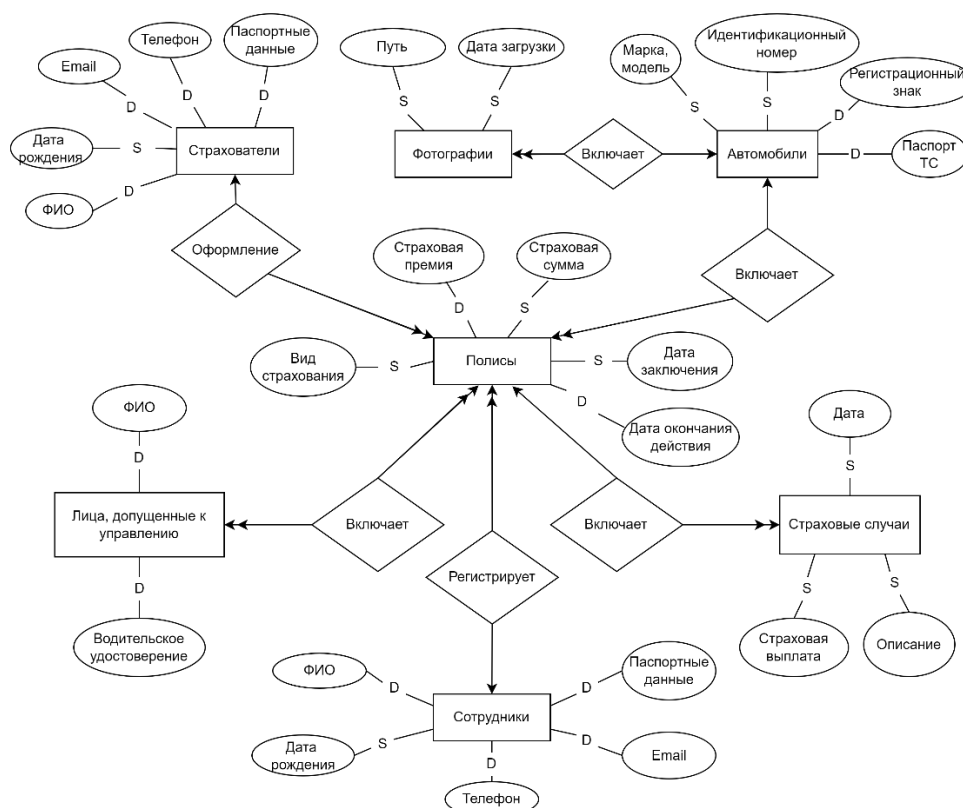


Рисунок 3 - Концептуальная модель данных

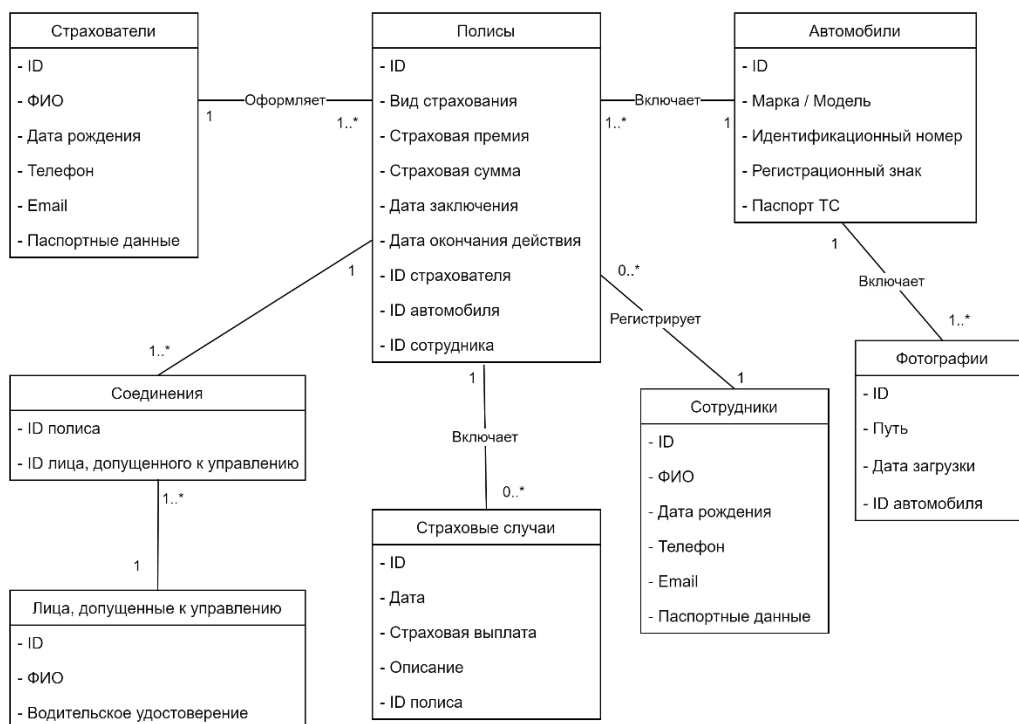


Рисунок 4 - Логическая модель данных

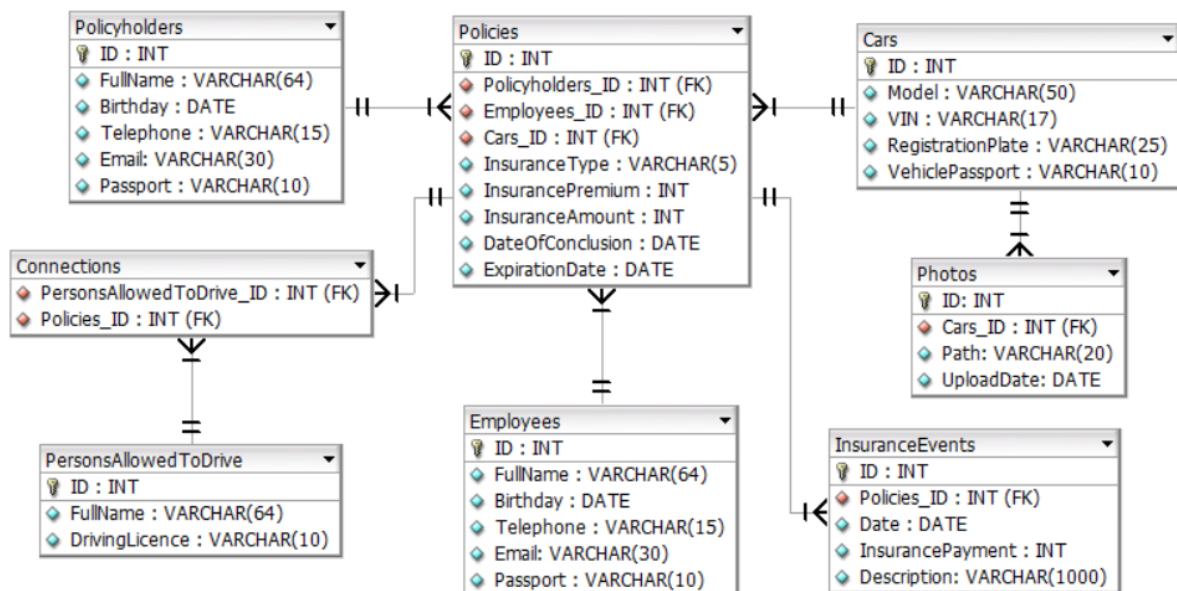


Рисунок 5 - Физическая модель данных

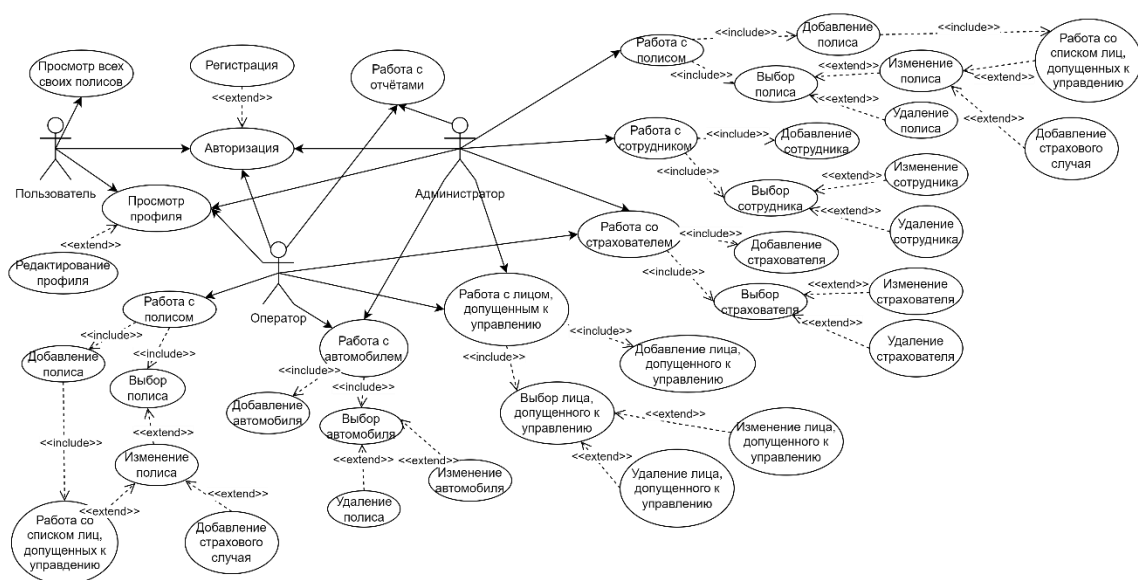


Рисунок 6 - Диаграмма вариантов использования

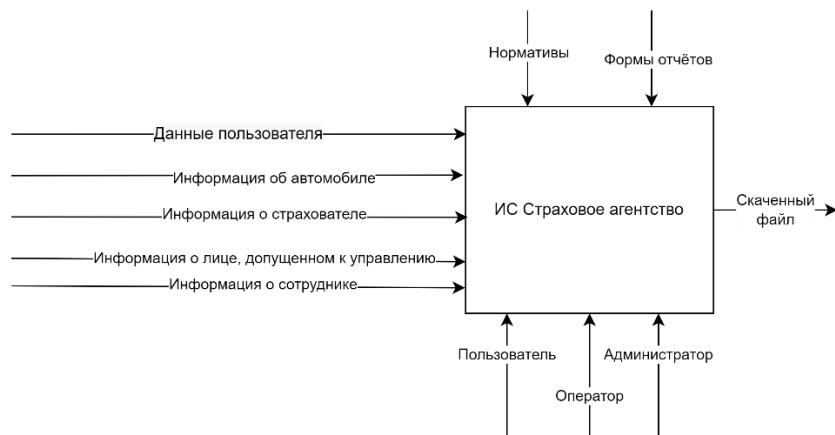


Рисунок 7 - Диаграмма IDEF0

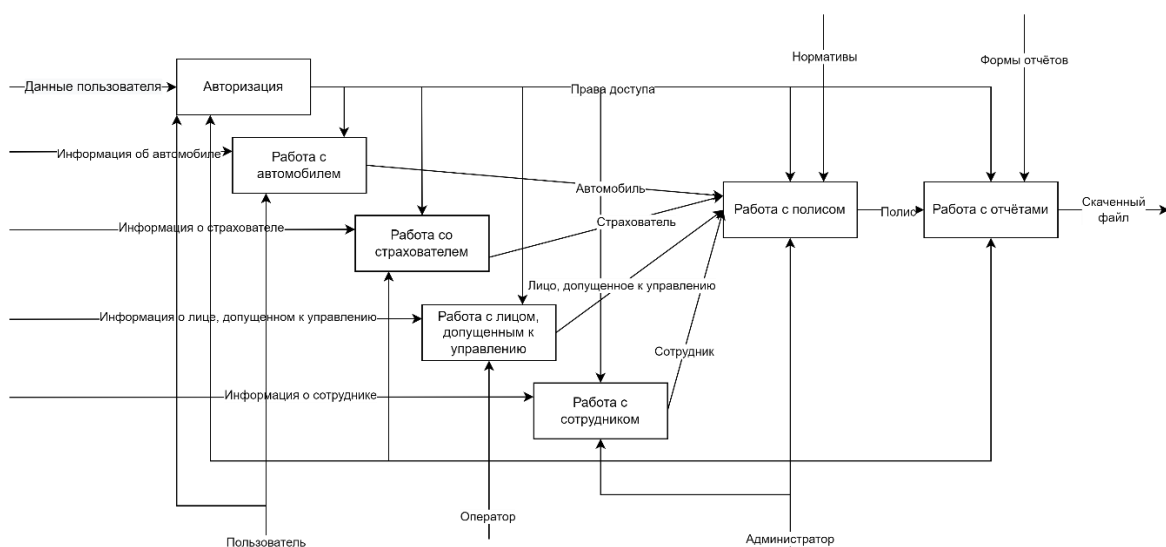


Рисунок 8 - Подробная диаграмма IDEF0

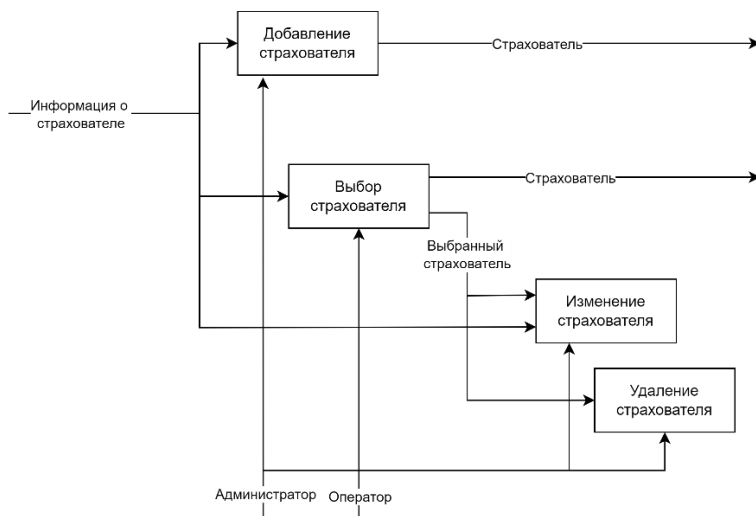


Рисунок 9 - Подробная диаграмма IDEF0 (Работа со страхователем)

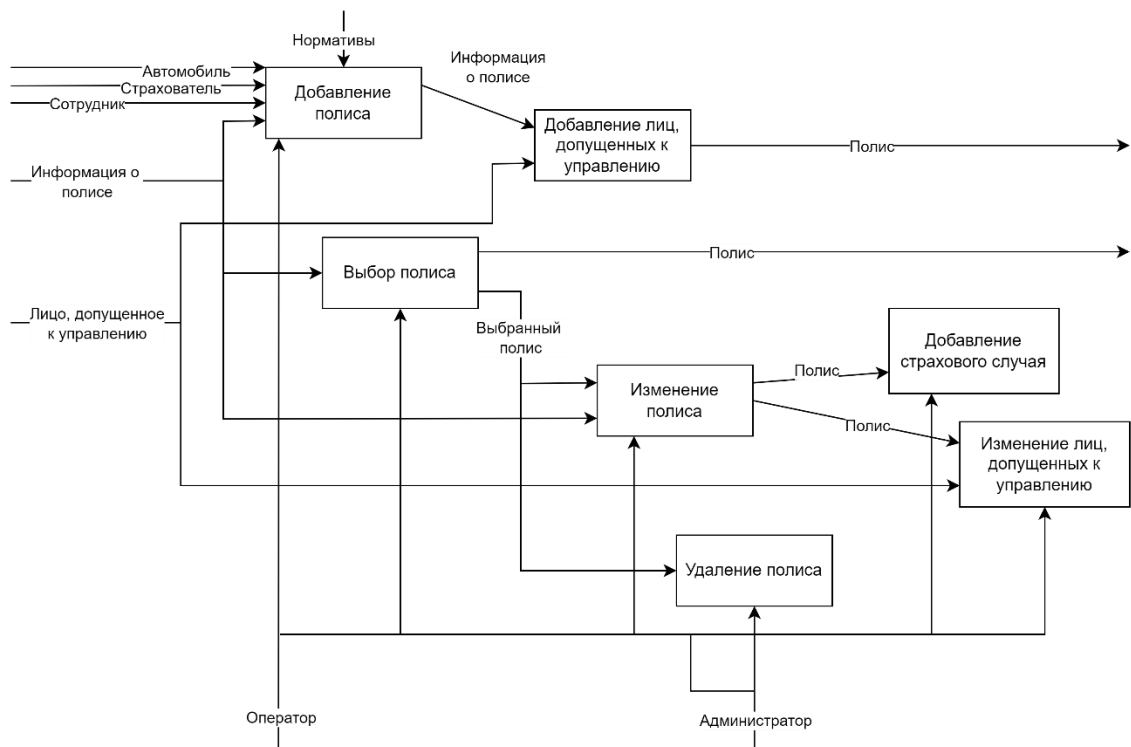


Рисунок 10 - Подробная диаграмма IDEF0 (Работа с полисом)

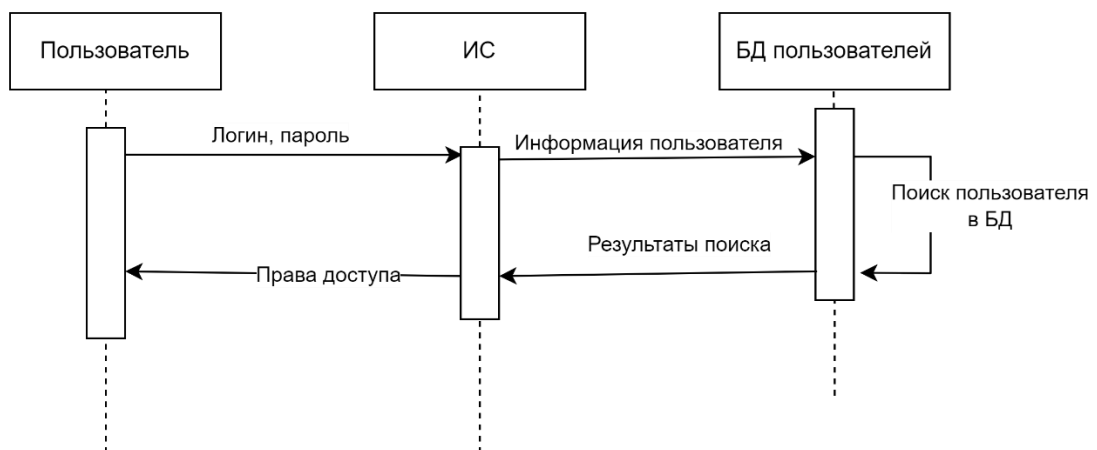


Рисунок 11 - Диаграмма последовательности (Авторизация)

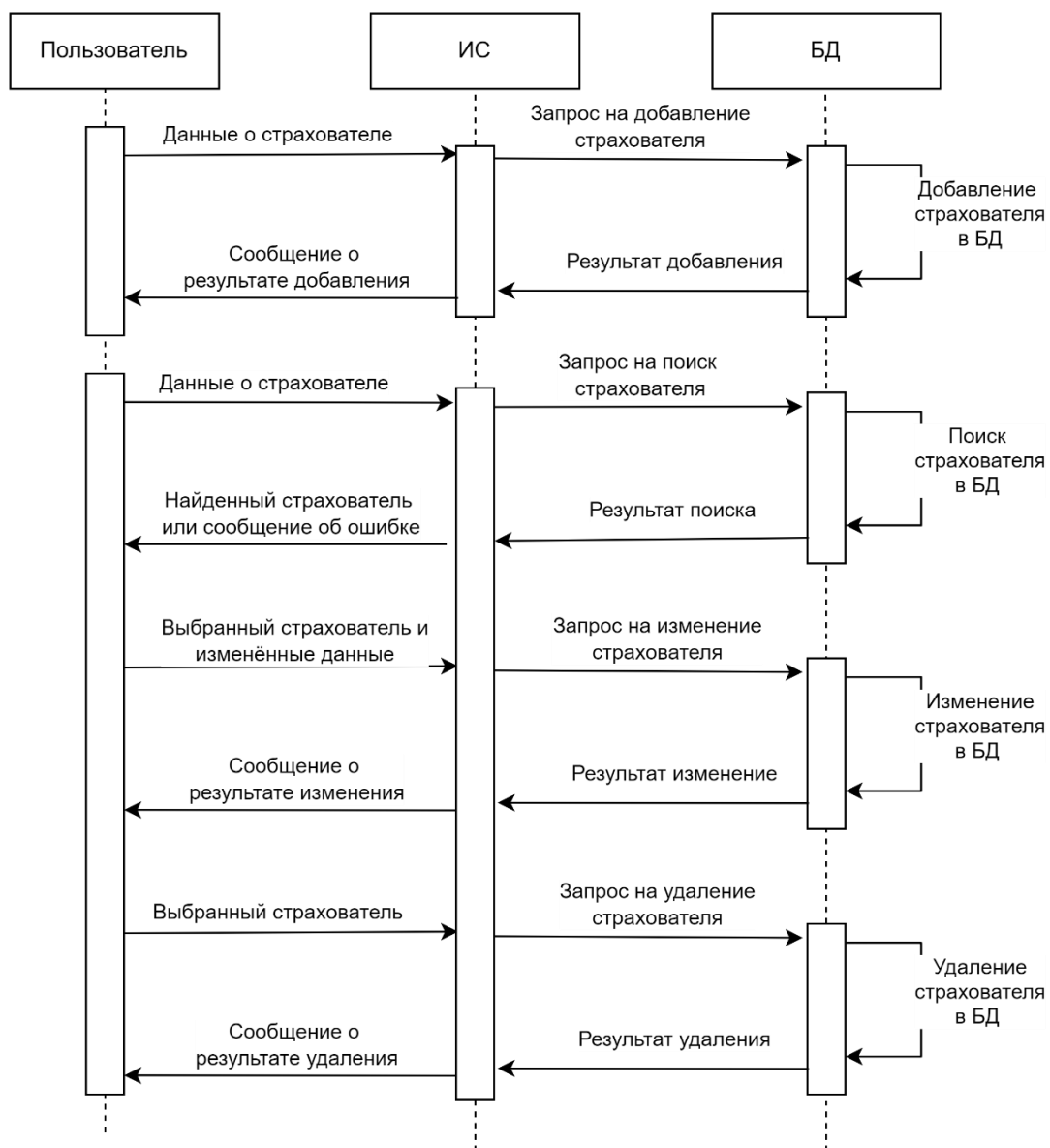


Рисунок 12 - Диаграмма последовательности (Работа со страхователем)

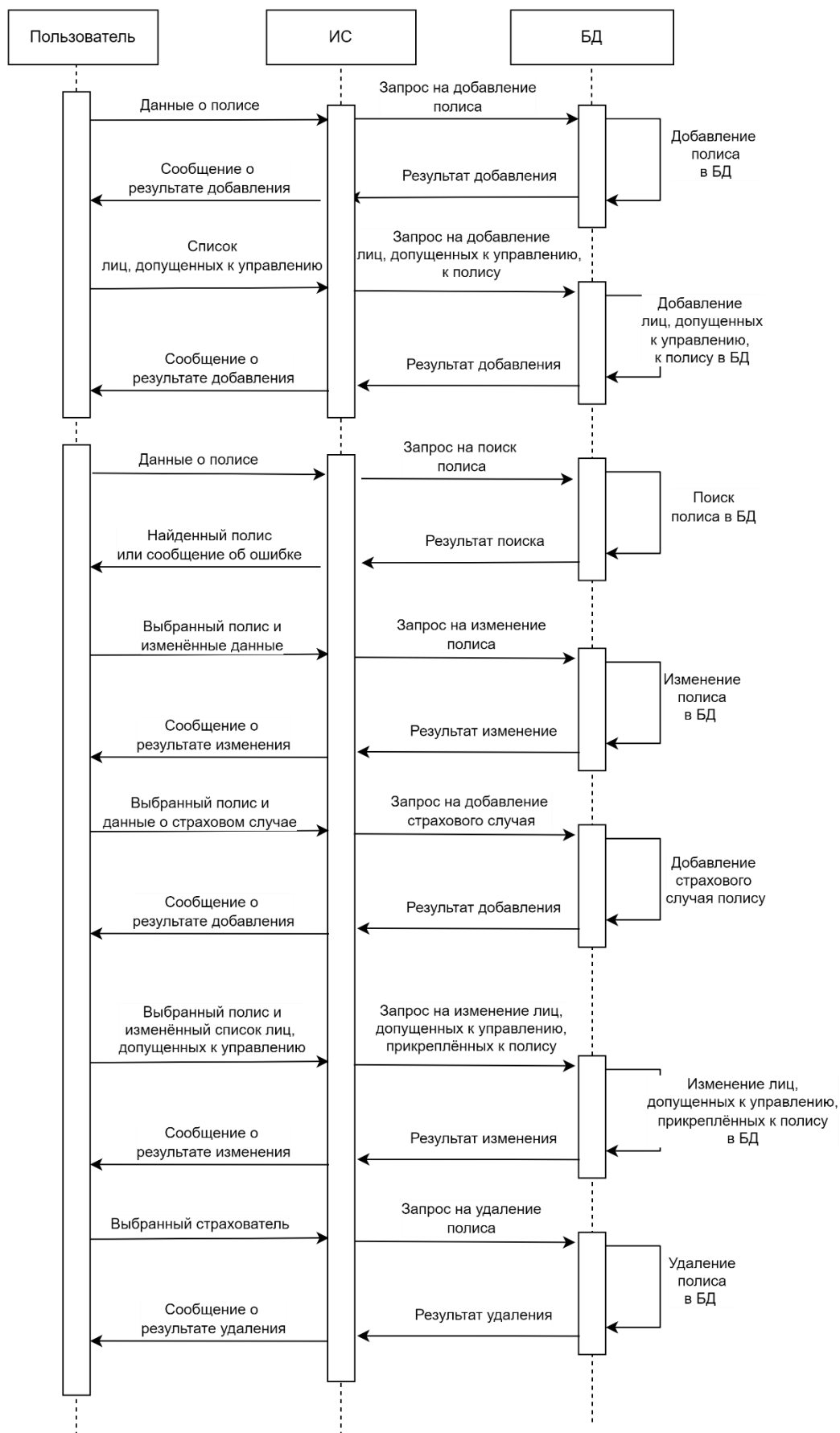


Рисунок 13 - Диаграмма последовательности (Работа с полисом)

## Приложение 2. Код программы

Полный код приложения представлен по ссылке:

<https://github.com/vivir-para-volar/SpringBootCoursework>

					МИВУ 09.03.04-8.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		46

### Приложение 3. Снимки окон программы (скриншоты программы)

Вид страхования	Страховая сумма	Страховая премия	Дата заключения	Дата окончания действия	Страхователь	Автомобиль
ОСАГО	8000	500000	07.10.2021	07.10.2022	Шестаков Даниил Данилович	Hyundai Santa Fe (3G)
КАСКО	10000	600000	07.10.2021	07.10.2022	Шестаков Даниил Данилович	Hyundai Santa Fe (3G)

Рисунок 14 - список полисов для пользователя

Вид страхования	Страховая сумма	Страховая премия	Дата заключения	Дата окончания действия	Страхователь	Автомобиль	Подробнее	Изменить	Удалить
ОСАГО	8000	500000	07.10.2021	07.10.2022	Шестаков Даниил Данилович	Hyundai Santa Fe (3G)	Подробнее	Изменить	Удалить
КАСКО	10000	600000	07.10.2021	07.10.2022	Шестаков Даниил Данилович	Hyundai Santa Fe (3G)	Подробнее	Изменить	Удалить
ОСАГО	4500	500000	01.10.2021	01.04.2022	Бессонов Иван Савельевич	KIA Ceed (2G)	Подробнее	Изменить	Удалить
ОСАГО	9800	500000	15.06.2021	15.06.2022	Морозова Софья Яновна	Renault Fluence	Подробнее	Изменить	Удалить

Рисунок 15 - список полисов

**Добавление полиса**

Выберите страхователя

ФИО	Дата рождения	Номер телефона	Email	Паспорт	Выбрать
Шестаков Даниил Данилович	01.01.1990	89969798752	shestakov@gmail.com	4395200670	Выбрать
Бессонов Иван Савельевич	16.01.1971	89180741287		7330729340	Выбрать
Морозова Софья Яновна	15.08.1979	89347650895	aspmorozovasofi@gmail.com	2218864204	Выбрать
Гаврилов Борис Дмитриевич	15.03.1999	89017654595	gavrilov@gmail.com	7818864232	Выбрать
Колесова Эмилия Тимофеевна	08.11.1991	89217652333		1134711573	Выбрать
Корнилов Ярослав Максимович	12.01.1976	89200755299		1220729563	Выбрать

[Вернуться к списку](#)

Рисунок 16 - добавление полиса (1 этап) - выбор страхователя



Добавление

localhost:8080/policy/create/choosePolicyholder/4

Главная Полисы Компоненты Отчёт Выйти

### Добавление полиса

Выберите автомобиль

Модель	VIN	Регистрационный знак	Паспорт ТС	
Hyundai Santa Fe (3G)	3GNDA23D08S544387	ОР897Ш09	67Тб907867	Выбрать
KIA Ceed (2G)	JN1CZ24A8LX094080	АЛ090РВ12	25Т4082337	Выбрать
Renault Fluence	3N1CC1AP8AL392058	E222XC22	89ТЦ348280	Выбрать
Rover 45	1N6BF0KL5CN138975	A343PP90	34ТВ734342	Выбрать
Лада 2101	JN8AF5MV4DT293416	ВО378Р79	38ТЕ489348	Выбрать
Opel Astra H	1GCE58FE2C8192977	ЛО839Р23	45ТФ893489	Выбрать
Mazda 626 IV	12343HJKL4564320	239GHj67	1237767833	Выбрать

Рисунок 17 - добавление полиса (2 этап) - выбор автомобиля

Добавление

localhost:8080/policy/create/chooseCar/4/3

Главная Полисы Компоненты Отчёт Выйти

### Добавление полиса

Введите оставшуюся информацию

Вид страхования: ОСАГО

Страховая сумма: 8000

Страховая премия: 500000

Дата заключения: 28.12.2022

Дата окончания действия: 12 месяцев

Страхователь: Гаврилов Борис Дмитриевич

Автомобиль: Renault Fluence

Сотрудник: AgencyAdmin

Далее

Рисунок 18 - добавление полиса (3 этап) - ввод оставшейся информации

Изменение

localhost:8080/policy/create/chooseInfo

Главная Полисы Компоненты Отчёт Выйти

### Добавление полиса

Выберите лица, допущенные к управлению

ФИО	Водительское удостоверение	
Бессонов Иван Савельевич	7345901132	<input type="checkbox"/>
Васильева Мария Ильинична	2929222999	<input checked="" type="checkbox"/>
Попов Иван Александрович	1731634821	<input checked="" type="checkbox"/>
Маркова Таисия Серафимовна	3489344688	<input type="checkbox"/>
Сидоров Максим Максимович	7292742382	<input checked="" type="checkbox"/>
Корнилов Ярослав Максимович	1907540089	<input type="checkbox"/>
Морозова Софья Яновна	5278982365	<input type="checkbox"/>
Гаврилов Борис Дмитриевич	7342367288	<input type="checkbox"/>
Шестаков Даниил Данилович	4397562113	<input checked="" type="checkbox"/>
Колесова Эмилия Тимофеевна	8734284883	<input type="checkbox"/>

Сохранить

Рисунок 19 - добавление полиса (4 этап) - выбор лиц, допущенных к управлению

Изменение

localhost:8080/policy/edit/7

Главная Полисы Компоненты Отчёт Выйти

### Изменение информации о полисе

Вид страхования: ОСАГО

Страховая сумма: 8000

Страховая премия: 500000

Дата заключения: 28.12.2022

Дата окончания действия: 28.12.2023

Страхователь: Гаврилов Борис Дмитриевич

Автомобиль: Renault Fluence

Сотрудник: AgencyAdmin

Сохранить

Добавить страховой случай | Лица, допущенные к управлению | Вернуться к списку

Рисунок 20 - редактирование полиса

Добавление

localhost:8080/insuranceEvent/create/7

Главная

### Изменение информации о полисе

#### Добавление страхового случая

Дата  
30.12.2022

Страховая выплата  
5000

Описание  
Авария

Добавить

Рисунок 21 - добавление страхового случая

Подробнее

localhost:8080/policy/details/7

Главная

### Подробнее о полисе

Вид страхования: ОСАГО  
Страховая сумма: 8000  
Страховая премия: 500000  
Дата заключения: 28.12.2022  
Дата окончания действия: 28.12.2023

#### Страхователь

ФИО: Гаврилов Борис Дмитриевич  
Дата рождения: 15.03.1999  
Номер телефона: 89017654595  
Email: gavrilov@gmail.com  
Паспорт: 7818864232

#### Автомобиль

Модель: Renault Fluence  
VIN: 3N1CC1AP8AL392058  
Регистрационный: E222XC22

Сотрудник

ФИО: AgencyAdmin  
Номер телефона: 89209380662  
Email: agencyaadm@gmail.com

#### Лица, допущенные к управлению

ФИО	Водительское удостоверение
Васильева Мария Ильинична	2929222999
Попов Иван Александрович	1731634821
Морозова Софья Яновна	5278982365

#### Страховые случаи

Дата	Страховая выплата	Описание
30.12.2022	5000	Авария

[Изменить](#) | [Вернуться к списку](#)

Рисунок 22 - подробная информация о полисе

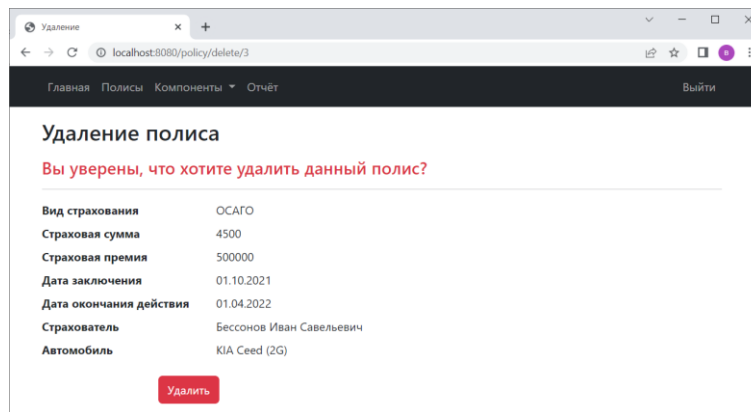


Рисунок 23 - удаление полиса

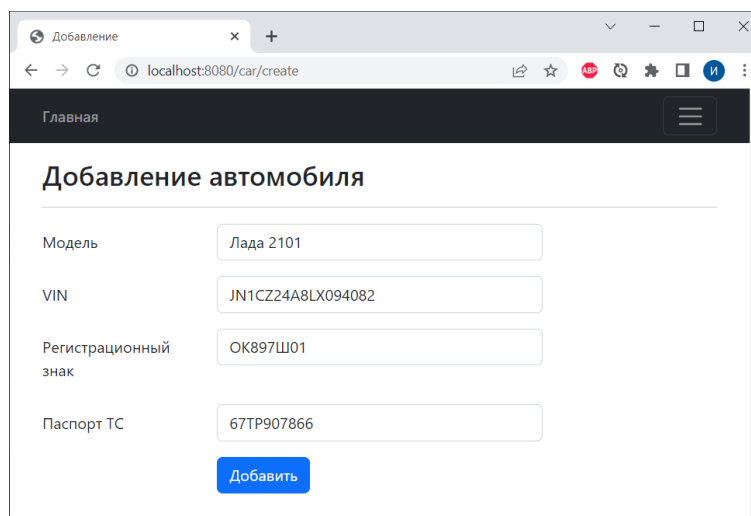


Рисунок 24 - добавление автомобиля (1 этап) - добавление информации об автомобиле

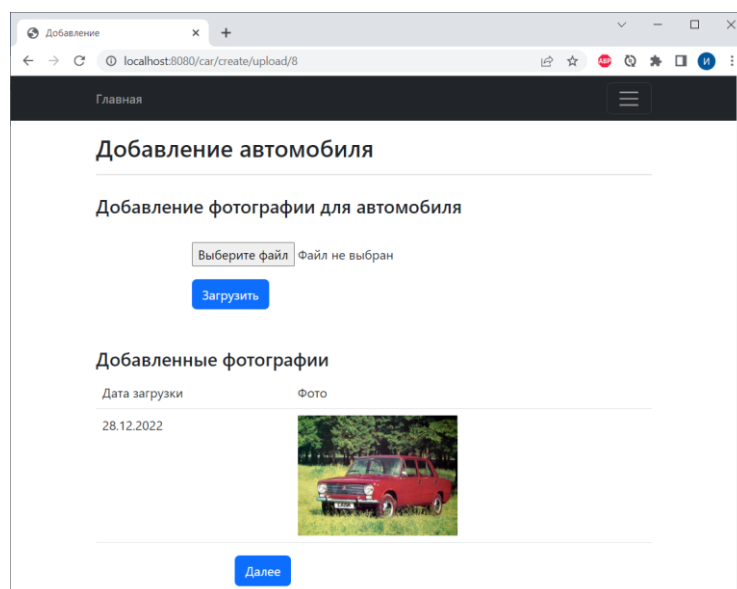


Рисунок 25 - добавление автомобиля (2 этап) - добавление фотографий

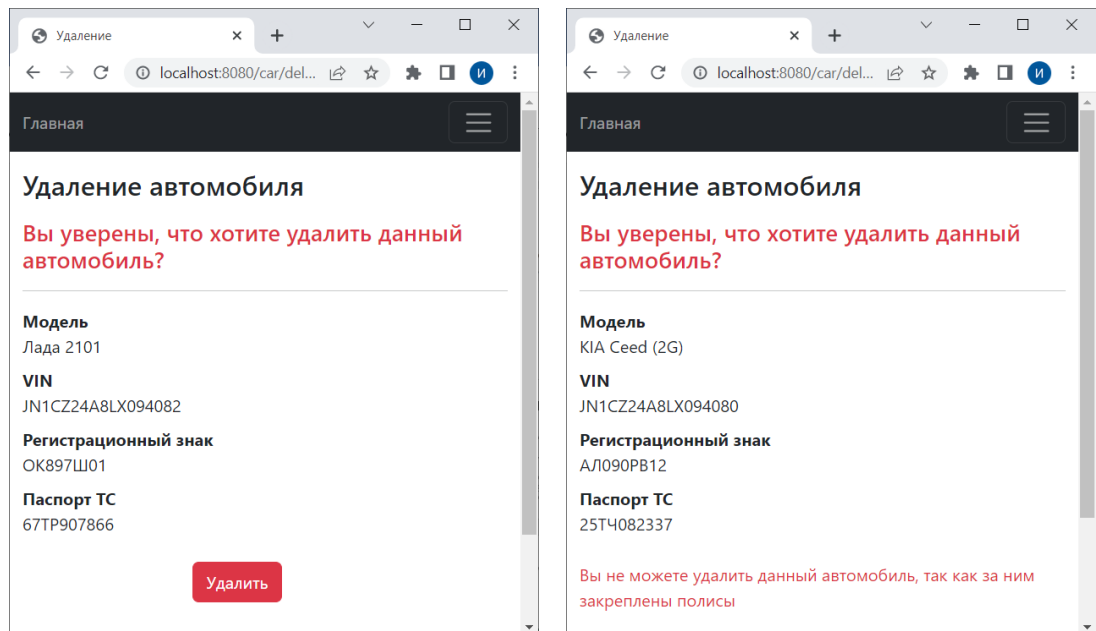


Рисунок 26 - удаление автомобиля

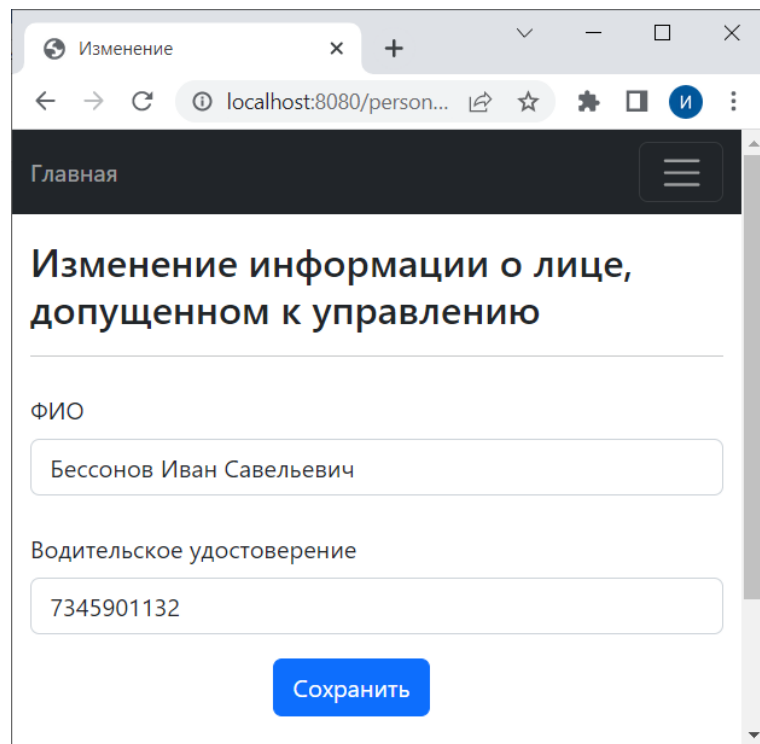


Рисунок 27 - изменение лица, допущенного к управлению

Добавление
localhost:8080/policyh...

Главная

### Добавление страхователя

ФИО

Лямина Ирина Алексеевна

Дата рождения

05.07.2001

Номер телефона

89209000655

Email (не обязательно)

Паспорт

8989888800

Добавить

Добавление
localhost:8080/emplo...

Главная

### Добавление сотрудника

ФИО

Афониная Ксения Николаевна

Дата рождения

13.10.1977

Номер телефона

89349380656

Email

Введите Email

Паспорт

8989866888

Добавить

Рисунок 28 - добавление страхователя и сотрудника

Добавление
localhost:8080/report

Главная Полисы Компоненты Отчёт Выйти

### Получить финансовый отчёт деятельности компании

Вид страхования

ОСАГО и КАСКО

Дата начала

01.01.2021

Дата окончания

01.01.2022

Получить

Количество заключённых договоров

5

Сумма заключённых договоров

39300

Сумма страховых выплат

1000

Рисунок 29 - форма с отчётами