

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»

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

# КУРСОВАЯ РАБОТА

По \_\_\_\_\_ Разработка корпоративных приложений \_\_\_\_\_

Тема \_\_\_\_\_ Веб-приложение «Детейлинг центр» \_\_\_\_\_

\_\_\_\_\_  
(оценка)

Руководитель  
\_\_\_\_\_  
Кульков Я.Ю.  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) (дата)

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

\_\_\_\_\_  
(подпись) (Ф.И.О.)

Студент ПИН-121  
(группа)  
\_\_\_\_\_  
Кокурин Я.Д.  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) (дата)



В данной курсовой работе было необходимо спроектировать веб-приложение для детейлинг центра. В качестве средств разработки приложения была использована среда IntelliJ IDEA Community Edition 2024.2.3. Язык разработки: Java. Технология Spring Framework. Управление версиями осуществлялось через систему контроля версий GitHub.

In this course project, it was necessary to develop a web application for a car detailing center. The IntelliJ IDEA Community Edition 2024.2.3 environment was used as the application development tools. Development language: Java. Spring Framework technology. Version control was carried out through the GitHub version control system

## Содержание

Введение.....	6
1 Анализ технического задания.....	8
2 Разработка моделей данных.....	12
2.1 Концептуальная модель данных .....	12
2.2 Логическая модель данных.....	12
2.3 Физическая модель данных .....	13
2.4 Разработка диаграммы прецедентов.....	14
3 Проектирование работы системы.....	15
4 Разработка и реализация системы.....	17
4.1 Руководство программиста.....	18
5 Тестирование системы.....	23
Заключение .....	25
Список литературы .....	26
Приложение 1 .....	27
Приложение 2 .....	29
Приложение 3 .....	30

					МИВлГУ 09.03.04 - 0.012				
Изм.	Лист	№ докум.	Подпись	Дата					
Разраб.		Кокурин Я.Д.			Распределённая ИС «Детейлинг центр»	Лит.	Лист	Листов	
Провер.		Кульков Я.Ю.					5	37	
Реценз.						МИ ВлГУ ПИН-121			
Н. Контр.									
Утверд.									

## Введение

В настоящее время для достижения успехов любому предприятию необходимо точно понимать свои затраты, прибыли, ресурсы, бизнес-процессы и многое другое. Наглядная информация о происходящем поможет глубже проанализировать процесс и сделать правильные выводы, что в конечном итоге приведет к повышению эффективности.

Лидерами становятся наиболее эффективные предприятия, имеющие минимальные издержки, высочайший уровень производительности и полностью контролируемые и четко отлаженные процессы. Ни что так не способствует контролю и анализу деятельности на предприятии как внедрение информационной системы – программно-аппаратный комплекс для авторизации деятельности организации, который обеспечивает хранение, передачу и обработку информации.

Внедрив информационную систему, руководство предприятия сможет принимать правильные решения по повышению эффективности отдельных процессов, снизит затраты, улучшит коммуникацию что несомненно поспособствует достижению успехов в своем виде деятельности.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется преимущественно на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами.

В данной курсовой работе рассматривается создание веб-приложения для автоматизации предметной области детейлинг центра на основе технологии Spring Framework.

Детейлинг центр – предприятие, занимающееся оказанием специализированных процедур по уходу за автомобилем. Детейлинг подразумевает максимально тщательную и качественную проработку

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

внешнего и внутреннего вида автомобиля до состояния близкого к идеальному и сохранение этого результата на длительный период времени. Для повышения эффективности деятельности детейлинг центра требуется анализировать спрос клиентов на услуги предприятия, для этого необходимы отчёты о востребованности услуг, на основании которых будут приниматься дальнейшие решения руководством (об изменении списка/качества оказываемых услуг).

Целями данной работы являются разработка моделей информационной системы детейлинг центра с учётом требуемых обработки данных для повышения эффективности предприятия и проектирование веб-приложения.

Задачей проекта является программная реализация веб-приложения, проверка его работоспособности.

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

## 1 Анализ технического задания

В данной курсовой работе необходимо создать информационную систему детейлинг центра.

Для разработки такой системы необходимы СУБД для создания базы данных и среда разработки для создания интерфейса и функций по работе с БД.

Разрабатываемое приложение предназначено для использования в детейлинг центре, занимающемся оказанием специализированных процедур по уходу за автомобилем.

Реализуемая программа должна быть разработана с использованием технологии Spring Framework. Spring — фреймворк с открытым исходным кодом, написанный на Java. Технология Spring Framework позволяет разработчикам быстро создавать масштабируемые и надежные приложения, а также использовать преимущества таких технологий, как Spring Boot, Spring Data и Spring Security. Ключевая особенность Spring — в разнообразии возможностей. Это не один фреймворк, а целый набор готовых решений. В его состав входят дополнительные модули и библиотеки, которые совместно работают и регулярно пополняются.

К достоинствам Spring Framework относятся:

1) Универсальность и оптимизация работы: Spring — это масштабная инфраструктура с инструментами буквально на все случаи жизни. В его составе огромное количество модулей, позволяющих решать как стандартные, так и сложные специфические задачи. Все фреймворки пишут, как готовые решения для минимизации рутинных операций. Spring — не исключение. Он предоставляет разработчику целый набор готовых шаблонов и все они экономят время.

2) Большой выбор инструментов: Spring — один из самых разветвленных фреймворков. На его базе можно реализовать практически любое решение.

3) Популярность: Spring широко востребован в среде Java-разработчиков. У этого фреймворка огромное сообщество, исчерпывающая документация и

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



регулярно выходящие дополнения. На нем написаны тысячи крупных проектов, которые требуют поддержки, что делает актуальным изучение Spring.

4) Доступность: Java называют языком программирования для крупных корпораций. У него есть платные и закрытые инструменты. А Spring распространяют бесплатно.

Недостатки:

1) Сложность: Spring — огромная и нагруженная инфраструктура. Перед работой необходимо внимательно прочесть документацию и разобраться в возможностях фреймворка.

2) Долгая настройка: запуск и адаптация Spring к конкретному проекту потребует много времени. Частично это можно решить, используя Spring Boot.

3) Большой объем кода: используя этот фреймворк, разработчики сталкиваются с необходимостью тщательной оптимизации. Это не проблема технологии как таковой, но на практике проекты на Spring могут содержать много лишнего кода и ненужных зависимостей.

Принципы Spring: в основе идеологии этого фреймворка лежит три принципа:

- Универсальность. Если другие фреймворки обычно пишут под определенные решения, то Spring — это фактически платформа для «всего» в Java. На его фундаменте можно написать любые приложения, а его новые модули должны иметь универсальное применение;

- Легкость. Этот фреймворк написан по принципу минимального воздействия. То есть каждый модуль включает в себя только необходимые функции и ничего лишнего. За счет этого готовые решения в Spring проще интегрировать с другими инструментами. А сам код отличается лаконичностью;

- Интегрированная инфраструктура. Сильная часть Spring — связи между компонентами. Все технологии в рамках этого инструмента обеспечены

хорошим взаимодействием друг с другом. Это сделано за счет инверсии контроля — способа внедрения зависимостей.

Spring делает создание веб-приложений быстрым и беспроblemным. Удаляя большую часть шаблонного кода и конфигурации, связанной с веб-разработкой, на выходе получается современная модель веб-программирования, которая упрощает разработку серверных HTML-приложений, REST API и двунаправленных систем на основе событий.

Программный продукт предназначен для автоматизации деятельности детейлинг центра. Администратор реализует CRUD методы обработки данных сотрудников, услуг детейлинг центра. Сотрудник детейлинг-центра управляет записью клиентов на обслуживание, реализует CRUD методы обработки данных клиентов и их автомобилей. Клиент просматривает историю обслуживания своих автомобилей.

Должна быть возможность получения сводных отчётов на основе информации из БД: выручка по услугам за любой временной период, востребованность услуг детейлинг центра, перечень обслуженных автомобилей за любой выбранный период, перечень услуг, предоставленных клиенту по выбранному автомобилю за любой период.

Курсовая работа предполагает выполнение следующих этапов:

- Разработка моделей данных: на этом этапе производится проектирование структуры данных, которые будут использоваться в системе. Это включает в себя определение сущностей, их атрибутов и взаимосвязей, а также создание схем баз данных.

- Проектирование работы системы: в данном разделе разрабатывается общая архитектура системы, включая распределенные компоненты, взаимосвязи между ними и общий механизм взаимодействия.

- Разработка и реализация системы: на основе предыдущих этапов начинается фаза активной разработки, включая написание кода, реализацию функций и компонентов системы.

- Тестирование системы: после завершения разработки проводится тестирование системы, включая проверку корректности ее работы, выявление и устранение возможных ошибок.

Таким образом, для разработки информационной системы детейлинг центра необходимо учесть все требования к базе данных и интерфейсу программы, а также выбрать подходящие инструменты для разработки.

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

## 2 Разработка моделей данных

### 2.1 Концептуальная модель данных

Концептуальная модель хранилища данных представляет собой описание главных (основных) сущностей и отношений между ними. Концептуальная модель является отражением предметных областей, в рамках которых планируется построение хранилища данных.

При проектировании концептуальной модели структурируют данные и выявляют взаимосвязи между ними, без рассмотрения особенностей реализации и вопросов эффективности обработки, поэтому концептуальная модель не является полностью подходящей для дальнейшей разработки, все таблицы должны быть нормализованы для реляционной базы данных. Составленная концептуальная модель представлена на рисунке 1 приложения 1.

### 2.2 Логическая модель данных

Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними.

Концептуальная модель изменяется так, чтобы она могла быть обеспечена конкретной моделью данных.

В результате формируется логическая модель. Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения.

Логическая модель может быть реляционной, иерархической или сетевой.

В качестве способа организации информационной базы выбрана реляционная база данных. Именно такой способ хранения всех данных является наиболее подходящим для проектируемой информационной системы по следующим причинам:

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

- наглядность модели для пользователя: все данные в реляционной модели представлены в табличной форме;
- независимость данных от программного продукта для их обработки;
- реляционные базы данных являются наиболее распространенными среди разработчиков ПО, следовательно, использование этих баз позволит сэкономить время и бюджет на внедрение нового типа БД.

Составленная логическая модель представлена на рисунке 2 приложения 1.

### 2.3 Физическая модель данных

Физические модели данных служат для отображения моделей данных. Основными понятиями модели данных являются поле, логическая запись, логический файл. Слово "логический" введено, чтобы отличать понятия, относящиеся к логической модели данных, от понятий, относящихся к физической модели данных. Основными понятиями физической модели данных, используемыми для представления логической модели данных, являются поле, физическая запись, физический файл. В частности, логическая запись, состоящая из полей, может быть представлена в виде физической записи (из тех же полей), логический файл – в виде физического файла. Имена таблиц и колонок будут сгенерированы на основе сущностей и атрибутов логической модели, учитывая максимальную длину имени и другие синтаксические ограничения, накладываемые СУБД. Если в имени сущности или атрибута встречается пробел, он заменяется на символ «\_».

Физическая модель описывает способ хранения данных в базе данных. В физической модели мы учитываем типы данных, индексы, ограничения целостности и другие технические детали.

Составленная физическая модель представлена на рисунке 3 приложения 1.

## 2.4 Разработка диаграммы прецедентов

Разработка диаграммы прецедентов является одним из ключевых этапов анализа и проектирования системы, особенно в методологии разработки программного обеспечения. Диаграмма прецедентов позволяет идентифицировать и описать функциональные возможности системы, включая взаимодействие между пользователями и системой.

Диаграмма прецедентов представляет собой графическое представление взаимодействия пользователей с системой через выполнение различных прецедентов. Она отражает функциональные требования системы и ее возможности.

Диаграмма прецедентов помогает разработчикам и заинтересованным сторонам понять функциональные требования системы, определить основные актеры и их роли, а также определить взаимодействие между актерами и прецедентами. Она служит основой для дальнейшего проектирования и разработки системы, а также для проверки соответствия системы требованиям пользователей.

На рисунках 4, 5, 6 в приложении 1 представлены диаграммы прецедентов для клиентов, администратора и сотрудника соответственно.

На диаграммах представлены действия, которые могут выполняться пользователями. Эти действия описывают функциональность системы и включают в себя последовательность шагов, выполняемых пользователем.

Функциональность клиента направлена на просмотр актуального списка услуг детейлинг центра и оказанных услуг для их автомобилей. Функциональность сотрудника направлена на работу с данными об автомобилях клиентов и оказанных услугах. Функциональность администратора направлена на работу с данными об актуальных услугах, оказываемых детейлинг центром, и данных о сотрудниках детейлинг центра.

### 3 Проектирование работы системы

В данном разделе содержится информация о проектировании работы “ИС детейлинг центра”, разработанной в текущей курсовой работе. В первую очередь было необходимо определить основные компоненты системы и их взаимодействие.

При проектировании работы системы детейлинг центра нужно уделить особое внимание архитектуре приложения и его функциональности. В соответствии с техническим заданием, система должна быть создана с использованием технологии Spring Framework, в качестве СУБД выбрана Microsoft SQL Server.

Первоначальным этапом проектирования является разработка моделей данных. Для учета клиентов, обслуживаемых автомобилей, оказываемых услуг, сотрудников и оказанных услуг необходимо создать соответствующие сущности и связи между ними. При создании моделей необходимо также предусматривать связи между ними, такие как связь "один ко многим" между клиентами и автомобилями. Для реализации данного этапа можно использовать компонент Spring.Boot для создания моделей.

Далее следует создание CRUD операций для администратора и сотрудника системы. Это позволит администратору управлять данными о сотрудниках и услугах, а сотруднику управлять данными о клиентских автомобилях и оказываемых услугах. В рамках создания CRUD функционала важно предусмотреть возможности добавления, просмотра, обновления и удаления записей.

Для клиента системы необходимо предусмотреть функционал регистрации в системе, просмотра оказанных услуг для автомобилей. Для обеспечения безопасности данных пользователей необходимо реализовать механизм аутентификации и авторизации. Для реализации данного этапа можно использовать компонент Spring.Security для разграничения доступа к функционалу приложения.

Кроме того, система должна предоставлять возможность генерации сводных отчетов на основе данных из базы данных. Эти отчеты могут включать информацию о выручке по услугам за любой временной период, востребованность услуг детейлинг центра, перечень обслуженных автомобилей за любой выбранный период, перечень услуг, предоставленных клиенту по выбранному автомобилю за любой период.

Интерфейс приложения должен быть понятным и удобным для пользователей. Для этого рекомендуется использовать современные принципы дизайна пользовательского интерфейса и обеспечить его дружелюбность и удобство использования.

Система будет использовать базу данных для хранения всех необходимых данных. База данных должна содержать таблицы для учета автомобилей, клиентов, услуг, сотрудников и оказанных услуг.

Клиентская часть будет реализована в отдельном приложении с использованием технологии JavaFX. Она будет отвечать за отображение пользовательского интерфейса и взаимодействие с серверной частью. Серверная часть будет обрабатывать запросы от клиентской части, выполнять CRUD операции с базой данных и генерировать отчеты.

Проектирование работы информационной системы детейлинг центра включает в себя проектирование базы данных, разработку пользовательского интерфейса, создание CRUD операций для администратора, сотрудника. Важно уделить внимание архитектуре приложения, безопасности данных и удобству использования для конечных пользователей.



#### 4 Разработка и реализация системы

В данном разделе будет описан процесс разработки и реализации клиентского и серверного приложений для детейлинг центра.

Клиентское приложение включает в себя следующие компоненты.

Первым компонентом является форма создания учётной записи для клиента детейлинг центра. С помощью этой формы создаются записи в таблицах «Пользователи» и «Клиенты». Данные из первой таблицы будут необходимы для реализации формы, предназначенной для клиентов детейлинг центра. Во второй таблице находятся данные, с которыми работают сотрудники детейлинг центра.

Форма для клиентов является вторым компонентом системы. На ней происходит получение данных из таблицы «Услуги» (список актуальных услуг детейлинг центра), таблицы «Автомобили» (список автомобилей авторизованного пользователя), таблицы «Оказанные услуги» (список оказанных услуг для выбранного автомобиля).

Следующим компонентом системы является форма для работы сотрудника, включающая в себя следующие разделы: список оказанных услуг детейлинг центра, список клиентов, список автомобилей клиентов. Первый раздел необходим для получения информации об оказанных услугах за выбранный период, добавление и изменение записей об оказанных услугах. Данный раздел для работы требует взаимодействия с таблицами «Услуги», «Клиенты», «Автомобили», «Сотрудники». Второй раздел предназначен для отображения списка клиентов детейлинг центра, он взаимодействует только с таблицей «Клиенты». Третий раздел разработан для работы информацией об автомобилях клиентов, он взаимодействует с таблицами «Клиенты» и «Автомобили».

Последним компонентом системы является форма для работы администратора, включающая в себя разделы список услуг детейлинг центра и список сотрудников. Первый раздел используется для управления списком

					МИВлГУ 09.03.04 – 0.012	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

услуг детейлинг центра, работает с таблицей «Услуги», второй – для работы со списком сотрудников, работает с таблицей «Сотрудники».

Переход к необходимой для работы форме происходит автоматически после ввода пользователем данных для авторизации. Это реализовано с помощью таблицы «Пользователи». В случае успешной авторизации в клиентское приложение возвращается тип роли авторизованного пользователя и в зависимости от роли запускается форма. Таким образом реализован разносторонний функционал приложения, зависящий от роли пользователя в системе. Администратор имеет доступ к инструментам управления услугами и сотрудниками детейлинг центра, сотрудник взаимодействует с данными об автомобилях клиентов и оказанных услугах детейлинг центром, а клиентам доступны просмотр списка актуальных услуг детейлинг центра и оказанных услуг для своих автомобилей. Скриншоты разработанного клиентского приложения представлены в приложении 3.

#### 4.1 Руководство программиста

##### Компоненты серверного приложения.

Так как разрабатываемое приложение должно обеспечивать работу с базой данных, первым шагом, основываясь на физической модели, была создана внутренняя структура физической модели в виде Java-классов (Entity) с использованием аннотаций для валидации данных, опираясь на необходимость работы с базой данных.

Были определены строки подключения к базе данных, в файле «application.properties», и указаны параметры, такие как URL базы данных, имя пользователя и пароль.

С помощью Spring Data созданы интерфейсы Repository, наследующиеся от CrudRepository, для всех созданных сущностей, также они дополнены необходимыми методами. Для реализации функционала по работе с методами Repository разработаны Service слои для каждого из созданных Repository.

					МИВлГУ 09.03.04 – 0.012	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

Для взаимодействия с внешними приложениями разработан класс ApiCotroller, использующий аннотацию RestController. Данный класс работает с сущностями Entity через Service слои.

Для демонстрации работы с сервером разработано JavaFX приложение с соответствующей валидацией данных.

По результатам выполнения работы, было проведено тестирование приложения, подтверждающее корректность работы всех компонентов системы.

В результате проделанной работы, программа является полностью функциональной, однако при желании может быть доработана.

Класс UserDTO.java

Разработан для получения данных клиента при регистрации.

Класс ApiController.java

Методы:

userLogin(Authentication auth) – обрабатывает GET-запрос для получения роли пользователя в случае успешной авторизации. Возвращает: (String) роль пользователя в системе.

userRegistry(@RequestBody UserDTO user) – обрабатывает POST-запрос на регистрацию клиента в системе. Возвращает: (String) о результате регистрации.

listOfAutomobiles() – обрабатывает GET-запрос на получение списка всех автомобилей детейлинг центра. Возвращает: Iterable<Automobile>

listOfClients() – обрабатывает GET-запрос на получение списка всех клиентов детейлинг центра. Возвращает: Iterable< Client>

listOfEmployers() – обрабатывает GET-запрос на получение списка актуальных сотрудников детейлинг центра. Возвращает: Iterable< Employer>

listOfServices() – обрабатывает GET-запрос на получение списка актуальных услуг детейлинг центра. Возвращает: Iterable<Service>

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

listOfProvidedServices () – обрабатывает GET-запрос на получение списка оказанных услуг детейлинг центра. Возвращает: Iterable< ProvidedService>

getAutomobile(@PathVariable("id") Long id) – обрабатывает GET-запрос на получение автомобиля по id. Возвращает: Automobile

getClient(@PathVariable("id") Long id) – обрабатывает GET-запрос на получение клиента по id. Возвращает: Client

getEmployer(@PathVariable("id") Long id) – обрабатывает GET-запрос на получение сотрудника по id. Возвращает: Employer

getService(@PathVariable("id") Long id) – обрабатывает GET-запрос на получение услуги по id. Возвращает: Service

getProvidedService(@PathVariable("id") Long id) – обрабатывает GET-запрос на получение оказанной услуги по id. Возвращает: ProvidedService

newAutomobile(@RequestBody Automobile automobile) – обрабатывает POST-запрос на добавление автомобиля. Возвращает: Automobile

newClient(@RequestBody Client client) – обрабатывает POST-запрос на добавление клиента. Возвращает: Client

newEmployer(@RequestBody Employer employer) – обрабатывает POST-запрос на добавление сотрудника. Возвращает: Employer

newService(@RequestBody Service service) – обрабатывает POST-запрос на добавление услуги. Возвращает: Service

newProvidedService(@RequestBody ProvidedService providedService) – обрабатывает POST-запрос на добавление оказанной услуги. Возвращает: ProvidedService

updateAutomobile(@RequestBody Automobile automobile , @PathVariable("id") Long id) – обрабатывает PUT-запрос на изменение автомобиля по id. Возвращает: Automobile

updateClient(@RequestBody Client client , @PathVariable("id") Long id) – обрабатывает PUT-запрос на изменение клиента по id. Возвращает: Client

updateEmployer(@RequestBody Employer employer , @PathVariable("id") Long id) – обрабатывает PUT-запрос на изменение сотрудника по id. Возвращает: Employer

updateService(@RequestBody Service service , @PathVariable("id") Long id) – обрабатывает PUT-запрос на изменение услуги по id. Возвращает: Service

updateProvidedService(@RequestBody ProvidedService providedService , @PathVariable("id") Long id) – обрабатывает PUT-запрос на изменение оказанной услуги по id. Возвращает: ProvidedService

deleteAutomobile (@PathVariable("id") Long id) – обрабатывает DELETE-запрос на удаление автомобиля по id

deleteClient(@PathVariable("id") Long id) – обрабатывает DELETE -запрос на удаление клиента по id

deleteEmployer(@PathVariable("id") Long id) – обрабатывает DELETE-запрос на удаление сотрудника по id

deleteService(@PathVariable("id") Long id) – обрабатывает DELETE-запрос на удаление услуги по id

deleteProvidedService(@PathVariable("id") Long id) – обрабатывает DELETE-запрос на удаление оказанной услуги по id

listOfActualAutomobiles(@PathVariable("phone") String phone) – обрабатывает GET-запрос на получение актуального списка автомобилей владельца с указанным номером телефона phone. Возвращает: Iterable<Automobile>

listOfAutomobiles(@PathVariable("phone") String phone) – обрабатывает GET-запрос на получение списка всех автомобилей владельца с указанным номером телефона phone. Возвращает: Iterable<Automobile>

getAutomobileByGosnumber(@PathVariable("gosnumber") String gosnumber) – обрабатывает GET-запрос на получение автомобиля по гос.номеру gosnumber. Возвращает: Automobile

getClientByPhone(@PathVariable("phone") String phone) – обрабатывает GET-запрос на получение клиента по номеру телефона phone. Возвращает: Client

getEmployerByPhone(@PathVariable("phone") String phone) – обрабатывает GET-запрос на получение сотрудника по номеру телефона phone. Возвращает: Employer

getServiceByName(@RequestBody String name) – обрабатывает POST-запрос на получение услуги по названию name. Возвращает: Service

listOfProvidedServices (@PathVariable("from") LocalDate from,@PathVariable("to") LocalDate to) – обрабатывает GET-запрос на получение списка оказанных услуг детейлинг центра в указанный временной период. Возвращает: Iterable< ProvidedService>

listOfProvidedServices (@PathVariable("id") Long id) – обрабатывает GET-запрос на получение списка оказанных услуг детейлинг центра для конкретного автомобиля. Возвращает: Iterable< ProvidedService>

## 5 Тестирование системы

Целью проведения тестирования является подтверждение реализации требуемой функциональной системы. Случаем, когда тестирование прошло успешно является совпадение с ожидаемым результатом.

В случае, когда приложение требует подтверждения действия пользователя появляется окно с кнопками подтверждения/отмены и уведомлением (информации, на основе которой пользователь должен сделать соответствующий выбор)

При обнаружении ошибки приложением, оно выводит сообщение, содержащее текст ошибки, на экран.

Тестирование было проведено для каждой из разработанных моделей, результаты проверки правильности работы приложения представлены в таблице 1. Демонстрация тестирования отражена в приложении 3.

Таблица 1 – Результаты тестирования разработанного приложения.

Выполненное действие	Полученный результат
Запуск приложения	Отображение стартовой страницы
Нажатие кнопки «Зарегистрироваться»	Открывается форма для регистрации клиента. В случае подтверждения регистрации проверяется заполнение всех полей, соответствие номера телефона шаблону, сравнение пароля в двух полях для ввода пароля. Иначе – закрывается окно регистрации клиента. Если данные введены верно – проверка существует ли пользователь с указанным номером телефона в системе. В случае когда пользователь отсутствует – успешная регистрация, иначе – показ уведомления
Нажатие кнопки «Выход»	Заккрытие приложения
Нажатие кнопки «Подтвердить»	Отправка запроса на сервер, если введены верные логин и пароль – сервер возвращает роль пользователя, и в зависимости от этой роли запускается соответствующая форма. Иначе – показ предупреждения
Заккрытие формы	Заккрытие формы авторизации приведёт к закрытию приложения целиком. Заккрытие форм, появившихся непосредственно после авторизации – прекращение работы под авторизованным пользователем. Заккрытие форм добавления/изменения/удаления информации о чем л. – отмена действия.
Форма для работы клиента	Открывается форма с двумя вкладками – список актуальных услуг детейлинг центра и вкладка с оказанными услугами для выбранного автомобиля авторизованного пользователя

Продолжение таблицы 1:

Выполненное действие	Полученный результат
Переход на вкладку с оказанными услугами и выбор автомобиля	Отображение списка оказанных услуг для выбранного автомобиля
Изменение выбранного автомобиля в поле выбора	Очистка списка оказанных услуг и повторное заполнение этого списка для выбранного автомобиля
Форма для работы администратора	Открывается форма с двумя вкладками – список актуальных услуг детейлинг центра и вкладка с сотрудниками
Нажатие кнопок «Добавить» / «Изменить»	В зависимости от вкладки появляются соответствующие формы для заполнения информации об объекте. Если поля заполнены с учётом предъявляемых требований – в результате подтверждения операции выполняется проверка на возможность выполнения действия, и по её результатам определяется будет ли совершено выбранное действие или показано предупреждение о невозможности его выполнить
Нажатие кнопки «Удалить»	В зависимости от вкладки и выбранного объекта – предложение удалить выбранный объект
Форма для работы сотрудника	Открывается форма с тремя вкладками – список оказанных услуг, список клиентов, список автомобилей выбранного клиента
Нажатие кнопок «Добавить» / «Изменить»	В зависимости от вкладки появляются соответствующие формы для заполнения информации об объекте. Если поля заполнены с учётом предъявляемых требований – в результате подтверждения операции выполняется проверка на возможность выполнения действия, и по её результатам определяется будет ли совершено выбранное действие или показано предупреждение о невозможности его выполнить
Нажатие кнопки «Удалить»	В зависимости от вкладки и выбранного объекта – предложение удалить выбранный объект
Изменение дат в полях с датами	Очистка списка оказанных услуг и повторное заполнение этого списка значениями в выбранный период времени

Результаты, полученные в ходе тестирования разработанного приложения, позволяют сделать заключение в том, что разработанная программа соответствует требованиям технического задания.



## Заключение

В ходе выполнения курсового проекта была разработана распределенная информационная система для автоматизации деятельности детейлинг центра. Разработка распределенной информационной системы для детейлинг центра представляет собой сложную задачу, требующую интеграции различных технических и функциональных компонентов.

В ходе анализа технического задания были выявлены основные требования к системе, необходимые для полноценной и корректной работы данной системы.

На этапе разработки моделей данных была проведена детальная проработка структуры данных системы, что обеспечило основу для эффективного хранения и обработки информации. Проектирование работы системы включило в себя определение распределенной архитектуры, что обеспечит эффективную обработку данных, даже при их распределении по разным узлам системы.

Фаза разработки и реализации системы включала в себя создание ключевых компонентов системы с использованием фреймворка Spring. Этот этап позволил успешно интегрировать различные модули, обеспечив согласованное функционирование системы.

Завершающим этапом было тестирование системы, нацеленное на выявление и устранение возможных ошибок, а также оценку ее производительности. Результаты тестирования подтвердили правильность реализации функционала и эффективность архитектурных решений.

Таким образом, выполнение поставленных задач позволяет сделать вывод о успешной разработке распределенной информационной системы детейлинг центра.

## Список литературы

1) Блох, Дж. Java. Эффективное программирование / Дж. Блох ; перевод В. Стрельцов ; под редакцией Р. Усманов. — 2-е изд. — Саратов : Профобразование, 2019. — 310 с. — ISBN 978-5-4488-0127-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/89870.html>

2) Свистунов, А. Н. Построение распределенных систем на Java : учебное пособие / А. Н. Свистунов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 316 с. — ISBN 978-5-4497-0940-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102045.html>

3) Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов. — Москва : СОЛОН-Р, 2016. — 336 с. — ISBN 5-93455-134-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/90352.html>

4) Кириченко, А. В. Динамические сайты на HTML, CSS, Javascript И Bootstrap. Практика, практика и только практика / А. В. Кириченко, Е. В. Дубовик. — Санкт-Петербург : Наука и Техника, 2018. — 272 с. — ISBN 978-5-94387-763-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/77578.html>

## Приложение 1

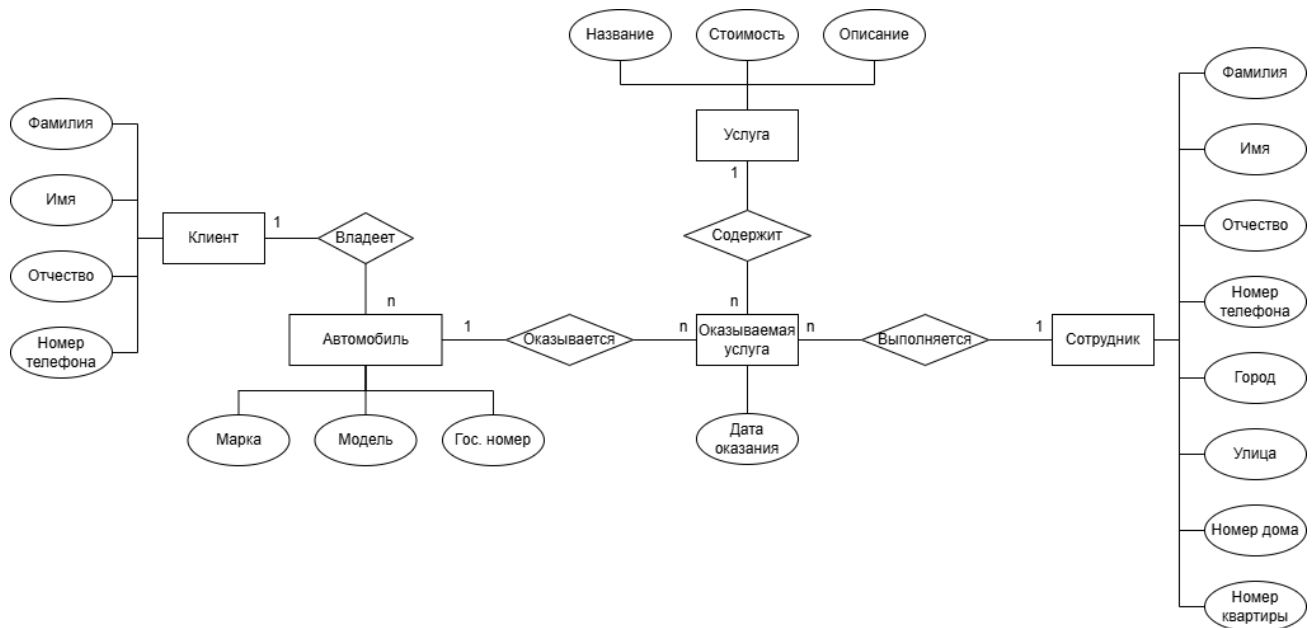


Рисунок 1 – Концептуальная модель данных



Рисунок 2 – Логическая модель данных

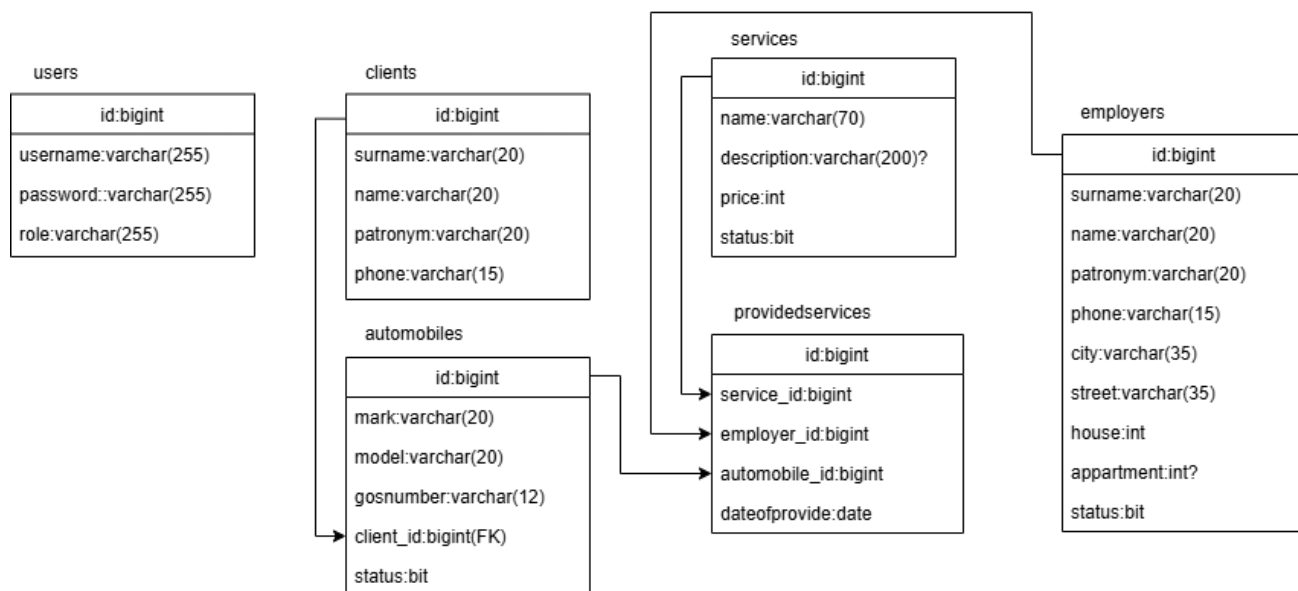


Рисунок 3 – Физическая модель данных



Рисунок 4 – Диаграмма прецедентов пользователя «Клиент»



Рисунок 5 – Диаграмма прецедентов пользователя «Администратор»



Рисунок 6 – Диаграмма прецедентов пользователя «Сотрудник»

## Приложение 2

Для подробного ознакомления с данным приложением можно использовать ссылку на репозиторий данного проекта: серверное приложение - <https://github.com/Mejiroy/spring-kurosovaya>



QR-код для перехода на репозиторий с серверным приложением

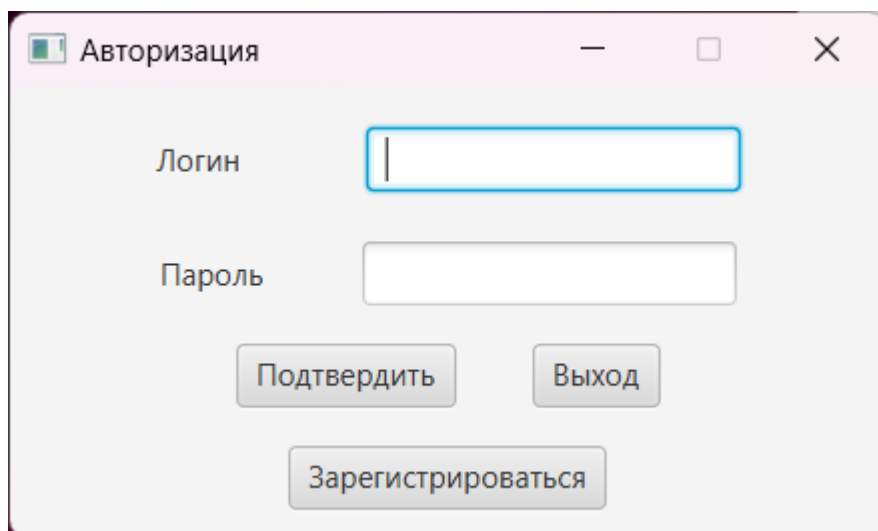
Клиентское приложение - <https://github.com/Mejiroy/kurosovaya-front>



QR-код для перехода на репозиторий с клиентским приложением

					МИВлГУ 09.03.04 – 0.012	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

### Приложение 3



Авторизация

Логин

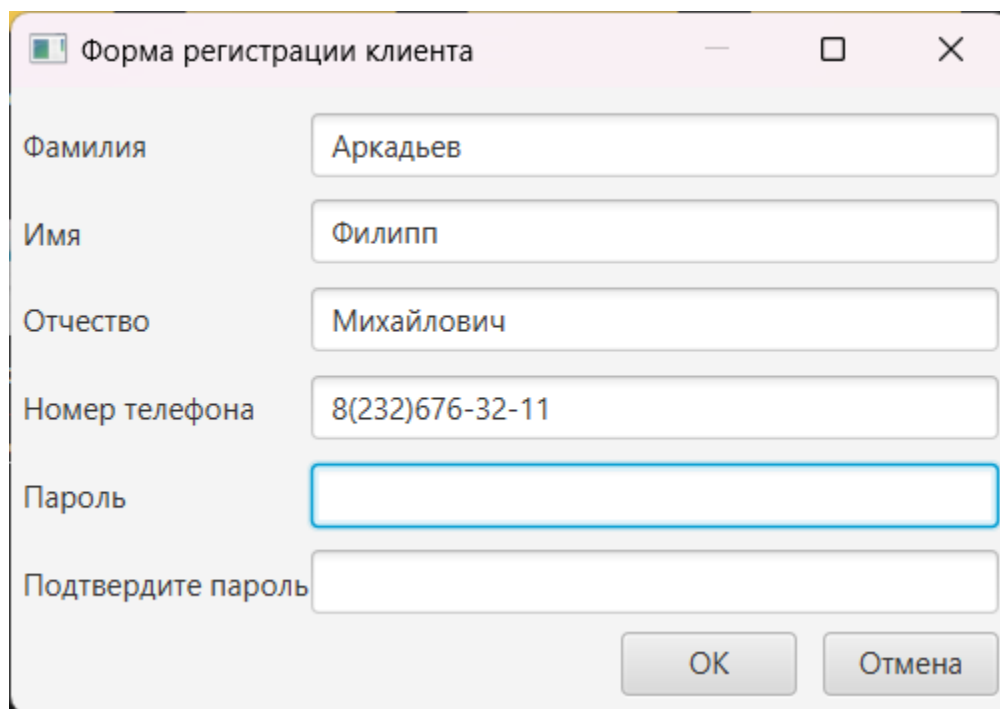
Пароль

Подтвердить

Выход

Зарегистрироваться

Рисунок 1 – Стартовая форма приложения



Форма регистрации клиента

Фамилия

Имя

Отчество

Номер телефона

Пароль

Подтвердите пароль

OK

Отмена

Рисунок 2 – Форма регистрации клиента

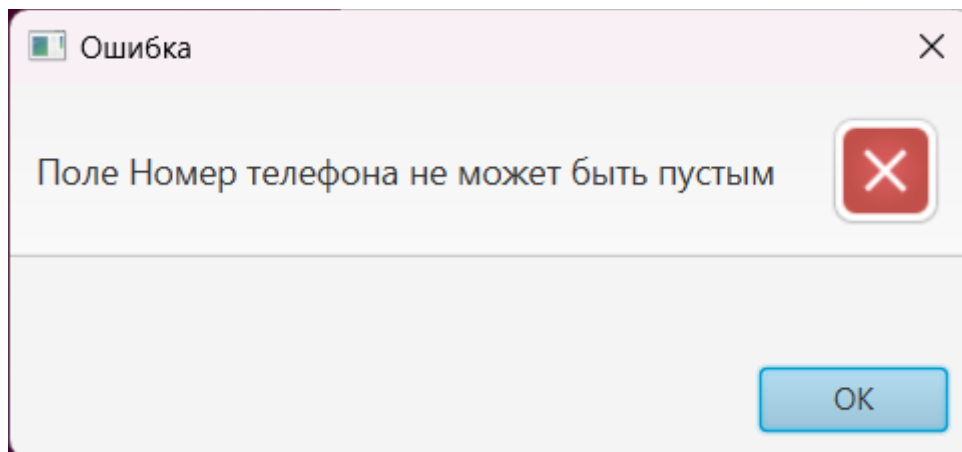


Рисунок 3 – Попытка регистрации без заполнения всех необходимых полей

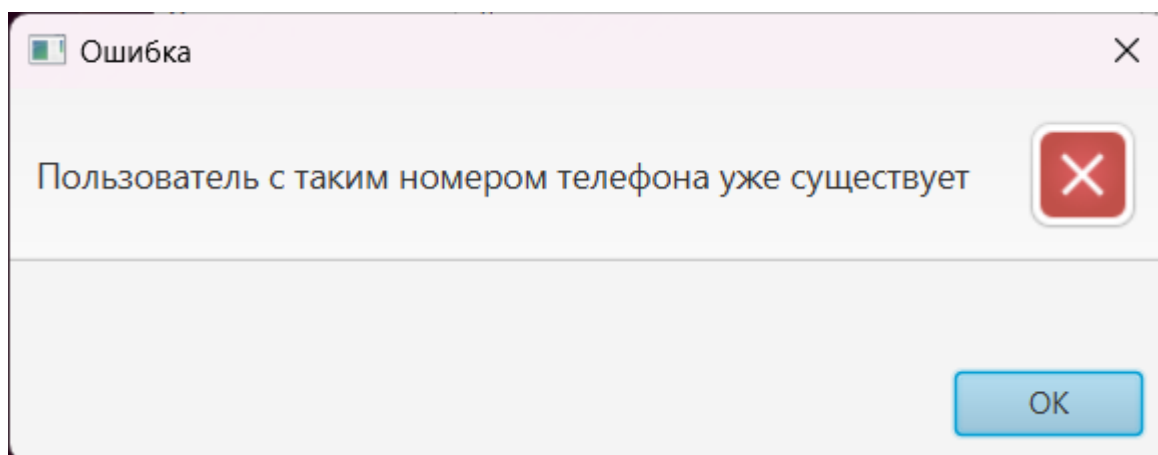


Рисунок 4 – Попытка регистрации на номер телефона, использующийся другим пользователем

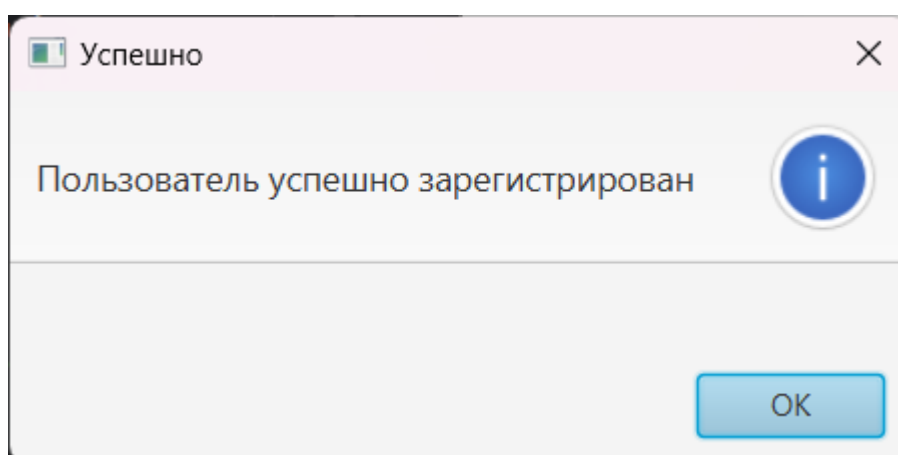


Рисунок 5 – Результат успешной регистрации

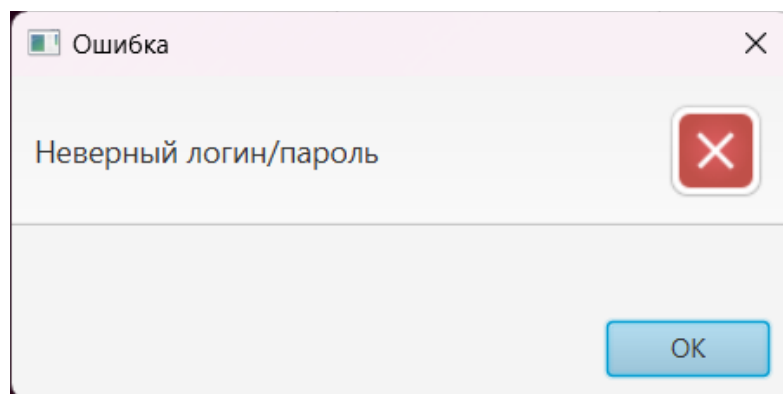


Рисунок 6 – Попытка авторизации с неверными логином/паролем

Название	Стоимость	Описание
Мойка автомобиля	600	
Оклейка автомобиля антигравийной пленкой	12500	Антигравийная полиурет...
Покраска кузова автомобиля	40000	

Рисунок 7 – Форма для пользователя с ролью клиент (список актуальных услуг детейлинг центра)

Автомобиль: BMW M5 P444MP456rus

Назва	Службника	Стоимость	Дата оказания
Мойка автомобиля	Артёмовна	500	2020-02-19
Мойка автомобиля	Артёмовна	500	2024-02-19

Рисунок 8 – Список автомобилей и оказанных ему услуг для авторизованного пользователя



Форма для работы сотрудника

Список оказанных услуг    Список клиентов    Список автомобилей клиентов

Название	Клиент	Автомобиль	Сотрудник	Стоимость	Дата оказания
Мойка автомобиля	Ермилов Михаил Анатольевич 8(123)456-78-90	BMW M5 P444MP456rus	Зубова Альбина Артёмовна 8(777)434-22-11	500	2020-02-19
Мойка автомобиля	Ермилов Михаил Анатольевич 8(123)456-78-90	BMW M2 P444MP457rus	Карасёв Эдуард Михайлович 8(777)434-22-12	600	2021-02-19
Мойка автомобиля	Ермилов Михаил Анатольевич 8(123)456-78-90	BMW M2 P444MP457rus	Зубарев Аркадий Петрович 8(444)233-33-51	600	2022-02-19
Мойка автомобиля	Ермилов Михаил Анатольевич 8(123)456-78-90	BMW M5 P444MP456rus	Зубова Альбина Артёмовна 8(777)434-22-11	500	2024-02-19

От: 01.01.2020    До: 25.12.2024    Добавить    Изменить    Удалить

Рисунок 9 – Форма для пользователя с ролью сотрудник (список оказанных услуг детейлинг центром за выбранный период)

Форма для работы сотрудника

Список оказанных услуг    Список клиентов    Список автомобилей клиентов

Фамилия Имя Отчество	Номер телефона
Баранов Алексей Михайлович	8(123)456-78-91
Ермилов Михаил Анатольевич	8(123)456-78-90
Ермилов Михаил Анатольевич	8(123)456-78-93
Карамазов Валерий Леонидович	8(123)456-78-99

Рисунок 10 – Список клиентов детейлинг центра

Форма для работы сотрудника

Список оказанных услуг    Список клиентов    Список автомобилей клиентов

Клиент: Ермилов Михаил Анатольевич 8(123)456-78-90

Марка	Модель	Гос.номер
BMW	M2	P444MP457rus
BMW	M3	P444MP458rus
BMW	M4	P444MP459rus

Добавить    Изменить    Удалить

Рисунок 11 – Список автомобилей выбранного клиента

Информация об автомобиле

Марка: HONDA

Модель:

Гос.номер: B888TB42rus

OK    Отмена

Рисунок 12 – Форма для добавления/изменения данных об автомобиле клиента

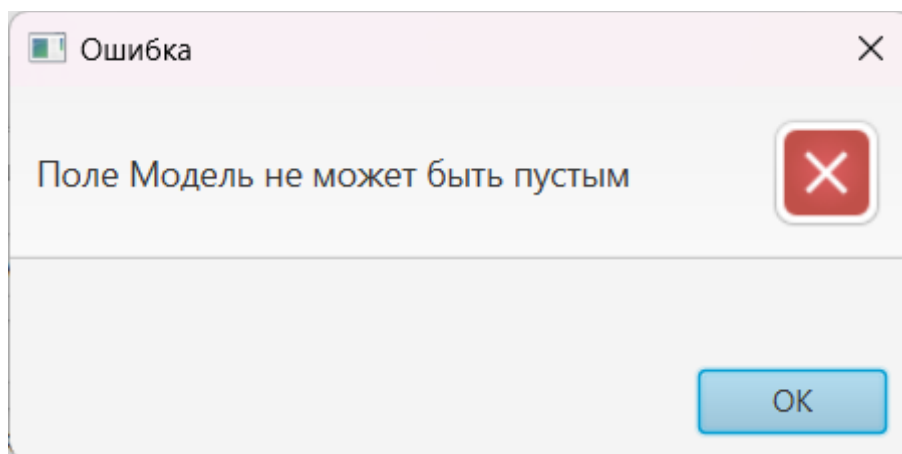


Рисунок 13 – Попытка добавления/изменения автомобиля без заполнения полей

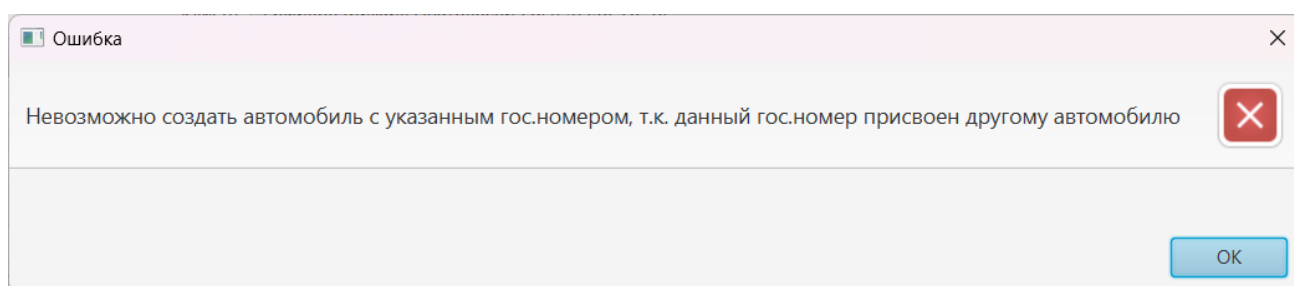


Рисунок 14 – Попытка добавления/изменения автомобиля с указанием гос.номера, присвоенного другому автомобилю

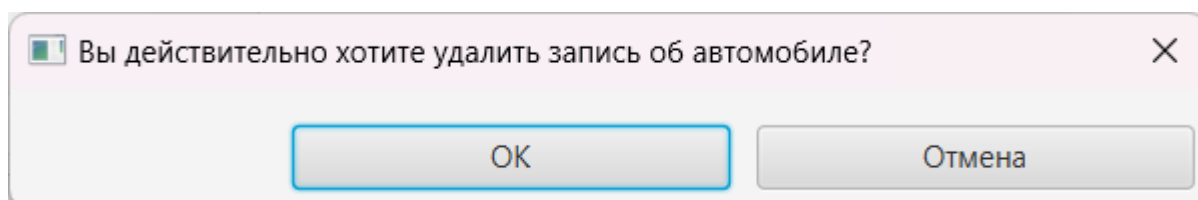


Рисунок 15 – Форма подтверждения удаления записи об автомобиле

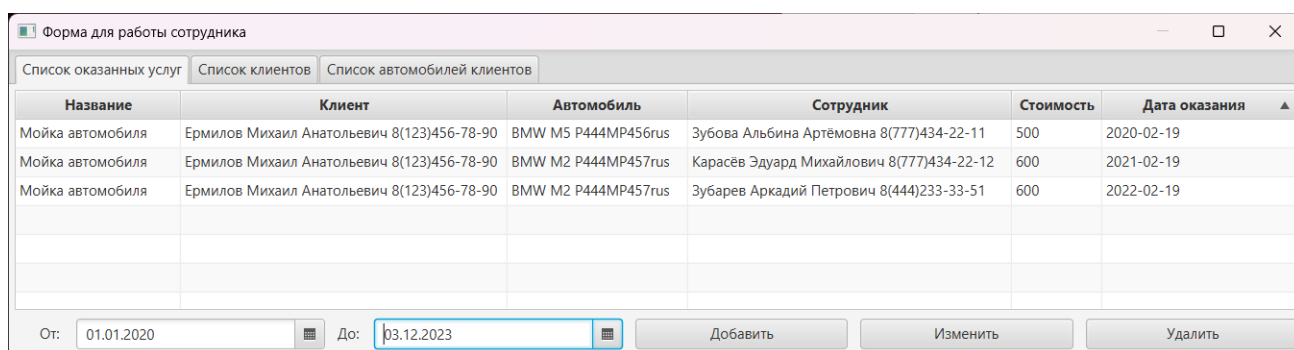


Рисунок 16 – Выборка оказанных услуг в изменённый временной период

Информация об оказанной услуге

Клиент: Ермилов Михаил Анатольевич 8(123)456-78-93

Автомобиль: AUDI A4 A234PM33rus

Услуга: Оклейка автомобиля антигравийной пленкой

Сотрудник: Зубова Альбина Артёмовна 8(777)434-22-11

Дата оказания: 25.12.2024

OK Отмена

Рисунок 17 – Форма добавления/изменения оказанной услуги

Форма для работы администратора

Список услуг Список сотрудников

Название	Стоимость	Описание
Мойка автомобиля	600	
Оклейка автомобиля антигравийной пленкой	12500	Антигравийная полиуретановая пленка SunTek PPF с...
Покраска кузова автомобиля	40000	

Добавить Изменить Удалить

Рисунок 18 – Форма для пользователя с ролью администратор (список актуальных услуг детейлинг центра)

Информация об услуге

Название: Оклейка автомобиля антигравийной пленкой

Стоимость: 600

Описание:

OK Отмена

Рисунок 19 – Форма добавления/изменения информации об услуге

Ошибка

Невозможно изменить название выбранной услуги, т.к. введенное название соответствует названию другой услуги

OK

Рисунок 20 – Попытка добавления/изменения услуги с указанием названия, присвоенного другой услуге

Форма для работы администратора

Список услуг Список сотрудников

Фамилия Имя Отчество ▲	Номер телефона ▲	Адрес(Город, улица, дом, квартира) ▲
Зубарева Альбина Михайлович	8(444)233-33-52	Муром, Муромская, 22
Зубарев Аркадий Петрович	8(444)233-33-51	Москва, Московская, 19
Зубова Альбина Артёмовна	8(777)434-22-11	Муром, Муромская, 14, 1
Карасёв Эдуард Михайлович	8(777)434-22-12	Навашино, Московская, 16, 1

Добавить Изменить Удалить

Рисунок 21 – Список сотрудников детейлинг центра

Информация о сотруднике

Фамилия: Зубова

Имя: Альбина

Отчество: Артёмовна

Номер телефона: 8(444)233-33-51

Город: Муром

Улица: Муромская

Дом: 14

Квартира: 1

OK Отмена

Рисунок 22 – Форма добавления/изменения информации о сотруднике

Ошибка

Невозможно изменить номер выбранного сотрудника, т.к. введенный номер занят другим сотрудником

OK

Рисунок 23 – Попытка добавления/изменения сотрудника с указанием номера, который используется другим сотрудником