**федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ» (КНИТУ-КАИ)**

**ОТДЕЛЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**Д И П Л О М Н Ы Й П Р О Е К Т**

**Тема:** Разработка информационной системы для клиентов автомойки с использованием платформы .NET Framework

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель**  преподаватель |  |  |  | А. Э. Баязитова |
| *(должность)* |  | *(подпись)* |  | *(И.О. Фамилия)* |
| **Обучающийся**  4433 |  |  |  | А. М. Шавалиев |
| *(группа)* |  | *(подпись)* |  | *(И.О. Фамилия)* |

**Специальность** 09.02.07 «Информационные системы и программирование»

**Казань 2022** **г.**

**АННОТАЦИЯ**

Тема дипломной работы: «Разработка информационной системы для клиентов автомойки с использованием платформы .NET Framework».

Цель разработки дипломного проекта – создание функционирующих модулей менеджера и клиента программной системы учета выполненной работы автомойки.

Для достижения данной цели были использованы язык разметки гипертекста HTML и язык каскадных стилей CSS, а также платформа разработки веб-приложений ASP.NET.

Разработанный программный продукт в качестве веб-сервиса и десктопного приложения в дальнейшем может использоваться в сети автомоек «Золотой теленок»

Объем дипломной работы составляет 72 страниц, в состав которых входят: 54 рисунков, 10 таблиц. В процессе разработки дипломного проекта было использовано 20 литературных источников.

Оглавление

[ВВЕДЕНИЕ 7](#_Toc104936733)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 9](#_Toc104936734)

[**1.1** **Описание целевой организации** 9](#_Toc104936735)

[**1.2** **Существующие аналоги программного решения** 10](#_Toc104936736)

[**1.3** **Обзор существующих бизнес-процессов организации** 11](#_Toc104936737)

[**1.4** **Обзор и анализ средств разработки** 12](#_Toc104936738)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ 19](#_Toc104936739)

[**2.1 Постановка задачи** 19](#_Toc104936740)

[**2.2 Концептуальная модель программной системы** 19](#_Toc104936741)

[**2.3 Логическая модель программной системы** 30](#_Toc104936742)

[**2.4 Диаграмма последовательности** 34](#_Toc104936743)

[**2.5 Диаграмма состояний** 40](#_Toc104936744)

[**2.6 Диаграмма классов** 43](#_Toc104936745)

[**2.7 Проектирование интерфейса приложения** 44](#_Toc104936746)

[**2.8 Концептуальное проектирование базы данных** 47](#_Toc104936747)

[**2.9 Логическое проектирование базы данных** 49](#_Toc104936748)

[ГЛАВА 3. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ 52](#_Toc104936749)

[**3.1 Руководство пользователя 52**](#_Toc104936750)

[**Требования к программному обеспечению 52**](#_Toc104936751)

[**Серверная часть 53**](#_Toc104936752)

[**Клиентская часть 53**](#_Toc104936753)

[**Требования к техническому обеспечению 53**](#_Toc104936754)

[**Серверная часть 53**](#_Toc104936755)

[**Клиентская часть 54**](#_Toc104936756)

[**Подготовка к работе. 54**](#_Toc104936757)

[**Знакомство с АИС «Золотой теленок» 54**](#_Toc104936758)

[**Страница журнала 55**](#_Toc104936759)

[**Описание операций 55**](#_Toc104936760)

[**Работа с журналом 55**](#_Toc104936761)

[**Просмотр журнала 56**](#_Toc104936762)

[**Добавление записи в журнал 56**](#_Toc104936763)

[**Удаление записи с журнала 57**](#_Toc104936764)

[**Редактирование записи журнала 57**](#_Toc104936765)

[**Работа с сотрудниками 57**](#_Toc104936766)

[**Просмотр сотрудников 58**](#_Toc104936767)

[**Добавление нового сотрудника в журнал 58**](#_Toc104936768)

[**Удаление сотрудника из списка 58**](#_Toc104936769)

[**Редактирование данных о сотруднике; 59**](#_Toc104936770)

[**Работа с услугами 59**](#_Toc104936771)

[**Просмотр услуг 59**](#_Toc104936772)

[**Добавление услуги 60**](#_Toc104936773)

[**Удаление услуги из списка 60**](#_Toc104936774)

[**Редактирование услуги 60**](#_Toc104936775)

[**Работа с машинами 61**](#_Toc104936776)

[**Просмотр списка машин 61**](#_Toc104936777)

[**Добавление новой машины 62**](#_Toc104936778)

[**Удаление машины из списка 62**](#_Toc104936779)

[**Редактирование информации о машинах 63**](#_Toc104936780)

[**Аварийные ситуации 63**](#_Toc104936781)

[**Рекомендации по освоению 63**](#_Toc104936782)

[**ЗАКЛЮЧЕНИЕ 68**](#_Toc104936783)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 69**](#_Toc104936784)

[**ПРИЛОЖЕНИЕ 71**](#_Toc104936785)

# ВВЕДЕНИЕ

В последнее время электронно-вычислительные устройства очень плотно вошли в современную жизнь большинства людей на планете. Еще полвека тому назад человек и представить не мог, что компьютеры, на тот момент являющиеся огромной ценностью для человечества и занимающие по размерам целые комнаты, будут находиться в каждом доме современного человека. Во множество электронно-вычислительных устройств входят: компьютеры, ноутбуки, планшеты, а также различного вида смартфоны.

Более двадцати лет назад людям, которые хотели бы узнать какую-либо информацию, приходилось бы тратить личное время на поиски требуемой информации во множестве источников. В настоящее время компьютеры позволяют систематизировать процесс получения информации обо всем, что существует в мире, и обеспечить доступ к информации буквально в любой момент времени.

Сегодня компьютеры стали неотъемлемой частью не только дома в быту, но и в сфере услуг. Главным направлением перестройки менеджмента и его радикального усовершенствования, приспособления к современным условиям стало массовое использование новейшей компьютерной и телекоммуникационной техники, формирование на ее основе высокоэффективных информационно-управленческих технологий. Особое значение имеет внедрение информационного менеджмента, значительно расширяющее возможности использования компаниями информационных ресурсов. Развитие информационного менеджмента связано с организацией системы обработки данных и знаний, последовательного их развития до уровня интегрированных автоматизированных систем управления, охватывающих по вертикали и горизонтали все уровни и звенья производства и сбыта.

В современных условиях эффективное управление представляет собой ценный ресурс организации, наряду с финансовыми, материальными, человеческими и другими ресурсами. Следовательно, повышение эффективности управленческой деятельности становится одним из направлений совершенствования деятельности предприятия в целом. Наиболее очевидным способом повышения эффективности протекания трудового процесса является его автоматизация.

Автоматизированная информационная система «Золотой теленок» предназначена для клиентов и менеджера предприятия, клиенты с помощью веб-сайта могут легко записаться на услуги автомойки, а менеджер с десктопного приложения ведет учет всех технических моментов предприятия.

# ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## **Описание целевой организации**

Информационная система реализуется для автомоек — предприятий, осуществляющих мойку автомобилей и оказывающие сопутствующие услуги (чистку салона, полировку, удаление битумных пятен, мойку двигателя и др.).

Зачастую автомойки располагаются на автозаправочных станциях, рядом с автомагазинами и станциями технического обслуживания автомобилей. В связи со значительным объёмом расходуемых моющих средств автомойки являются потенциальными загрязнителями окружающей среды. В связи с этим, как правило, государственные органы требуют оснащать автомойки системами регенерации и очистки.

Бывают мойки самообслуживания (владелец самостоятельно моет автомобиль), а также мойки, где операции с автомобилем осуществляются обслуживающим персоналом — автомойщиками и механизированные автомойки, где процесс мойки осуществляется автоматически оборудованием автомойки.

Предприятие, для которого реализуется данная система – это небольшая автомойка «Золотой теленок», которой владеет индивидуальный предприниматель по адресу Лесная ул., 27, село Актаныш, Россия. Автомойка имеет два бокса: для размещения машин во время оказания услуг и кассовая зона, где администратор встречает клиентов. После осуществления мойки клиент расплачивается за услугу в кассе и получает чек. Клиент во время обслуживания машины находится в комнате отдыха.

За рабочие сутки услугами автомойки пользуются в среднем 20-30 клиентов.

Потребителями данного программного решения являются непосредственные работники, а именно менеджер организации. Он принимает клиентов и производит все расчеты по дневной выручке. Клиенты же могут оставлять заявки через веб-сайт.

## **Существующие аналоги программного решения**

У каждой организации имеются свои программные решения для реализации основных бизнес-процессов. Стандартный функционал для сети автомоек – это возможность ведения журнала клиентов и предоставленных услуг для дальнейшего подсчёта оклада работников и чистой выручки. Некоторые сети автомоек имеют в наличии веб-сайт для записи на определенную услугу.

Рассмотрим преимущества и недостатки информационной системы сети автомоек «Авто Сити».

Преимущества:

* Имеет удобный и понятный для клиента веб-сайт с возможностью просмотра цен на те или иные услуги на каждом из точек в городе;
* Присутствие окна для просмотра акции, отзывов, контактов и т. п.

Недостатки:

* Через сайт невозможно записаться на услуги организации, для этого необходимо скачивать мобильное приложение, потребителю это может быть неудобно.

Интерфейс страницы с обзором услуг автомойки «Авто Сити» представлен на рисунке 1.

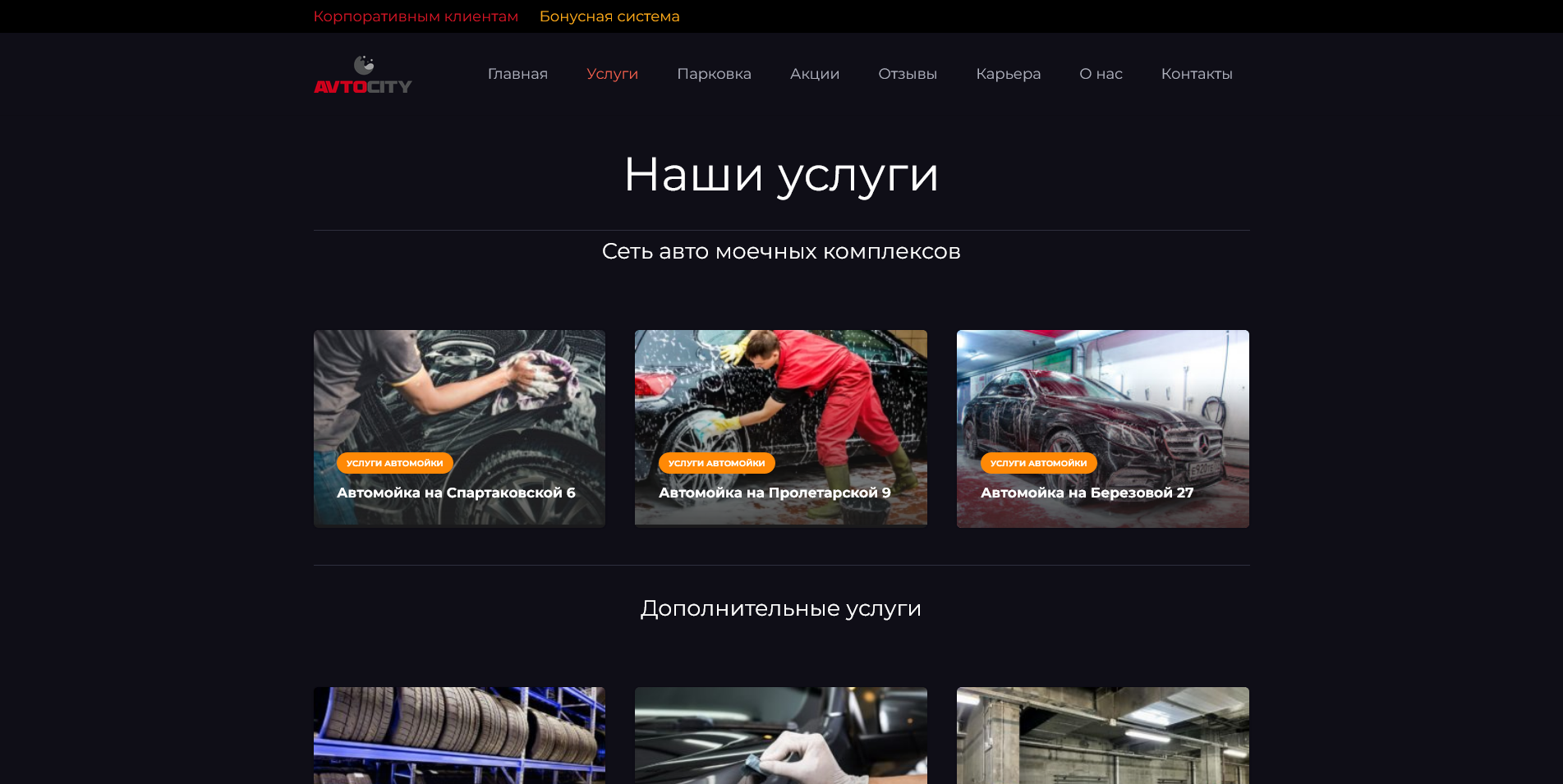


Рисунок 1 – Интерфейс страницы услуг автомойки «Авто Сити»

Рассмотрим преимущества и недостатки информационной системы сети автомоек «Мойдодыр».

Преимущества:

* Имеет удобный и понятный для клиента веб-сайт;
* Имеется возможность просмотра цен на те или иные услуги, просмотра акции, специальных предложений, информации о организации, контакты и т.д.

Недостатки:

* На услуги невозможно записаться онлайн через сайт, для этого необходимо перейти на мессенджер «Telegram», если у потребителя не окажется данного мессенджера, то уже онлайн запись невозможен.

Интерфейс страницы с обзором услуг автомойки «Мойдодыр» представлен на рисунке 2.



Рисунок 2 – Интерфейс страницы услуг автомойки «Мойдодыр»

## **Обзор существующих бизнес-процессов организации**

Администратор каждый день ведет бумажный учет всех поступивших клиентов. При поступлении новой машины на оказание услуг записываются следующие данные: модель машины, работник выполняющую услугу, наименование услуги и цена.

В бумажных носителях хранится информация о нагрузке работников, их график работы и расчет выручки.

При введения бумажного учета необходимо учитывать человеческий фактор, который может привести к разным последствиям: утеря журнала с записями, намеренное уничтожение журналов и т. д. Кроме человеческого фактора бумажным носителям угрожают пожары, наводнения и другие стихийные бедствия.

## **Обзор и анализ средств разработки**

* + 1. **Обзор интегрированных средств разработки**

В сфере разработки программного обеспечения и веб-страниц существует огромное количество разнообразных и универсальных средств разработки, начиная от стандартного блокнота и заканчивая текстовыми редакторами, например,

Sublime Text – это текстовый редактор, чаще всего предназначенный для программирования на языке Python. Стандартный интерфейс приложения без использования каких-либо плагинов и дополнений представлен на рисунке ниже (см. Рисунок 3*Рисунок* ). Данный текстовый редактор является схожим по функционалу со всеми иными редакторами, но имеет существенный недостаток: плагины, которые установлены по умолчанию в Sublime Text, невозможно отключить.

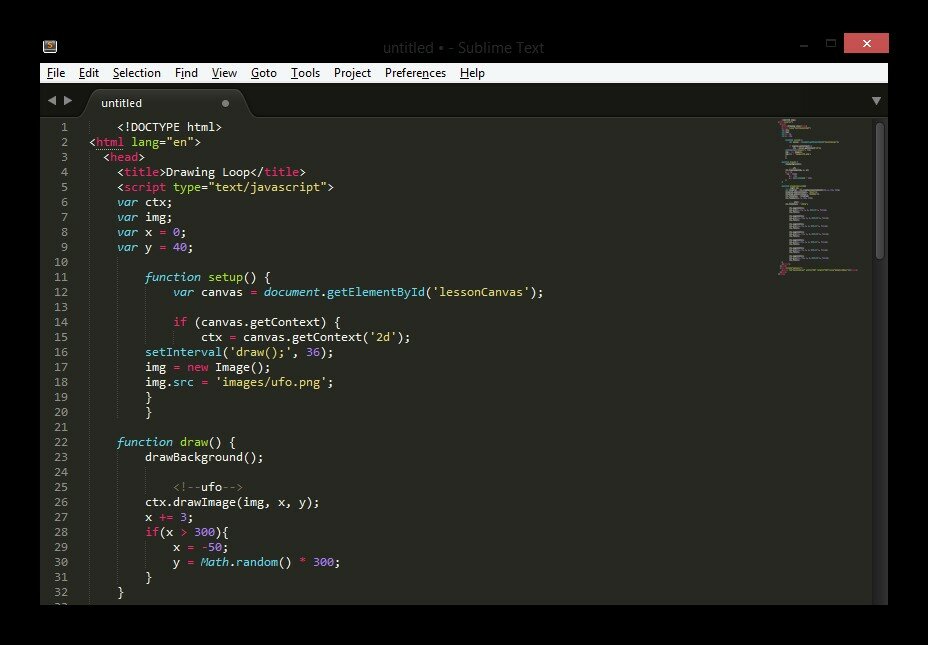


Рисунок 3 – Интерфейс «Sublime Text»

Notepad или же Notepad++ – это, как и остальные текстовые редакторы, универсальная интегрированная среда разработки, в данном случае, которая способна распознать любой синтаксис большого числа языков программирования. Функционалом данное приложение практически не отличается от своих аналогов, однако по сравнению с ними, имеет относительно простой и понятный интерфейс, что позволяет ускорить выполнение работы пользователями.

Atom – это многофункциональный текстовый редактор, который по сравнению с иными редакторами, позволяет легко и быстро установить все необходимые расширения для повышения быстродействия написания кода, и в последствии, разработки веб-сервиса. Также интегрированная среда разработки Atom имеет удобный для восприятия интерфейс, однако для обычных пользователей, мало работающих с программными средами разработки, данный интерфейс может негативно отражаться на быстродействии выполнения работы. Функционал программы поддерживает выбор цветовой гаммы не только задней и боковой панелей приложения, но и исходного кода, написанного пользователем. Интерфейс среды разработки Atom с предустановленными плагинами для удобства работы представлен на рисунке ниже (см. Рисунок 4).

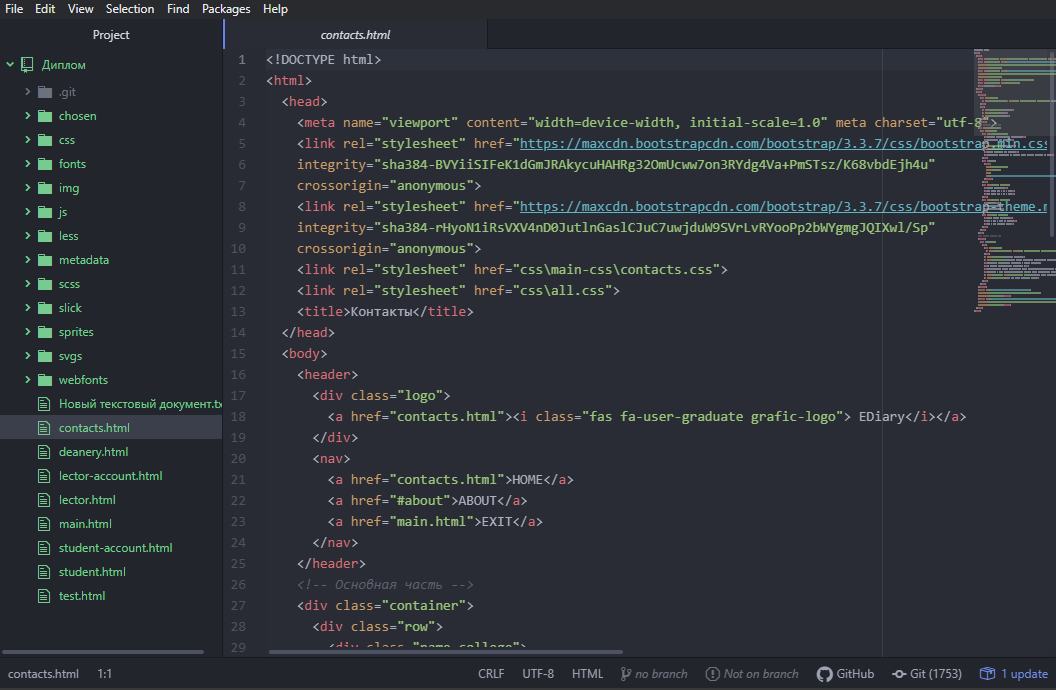


Рисунок 4 - Интерфейс «Atom»

В качестве основной интегрированной среды разработки для создания веб-сервиса на начальном этапе до разработки функционала была использована именно среда разработки Atom, так как ее темный вид интерфейса и цветовая палитра текста являются более приятными для восприятия глаз, чем любые другие аналоги текстовых редакторов. Также в данном приложении легко ориентироваться в текстовых файлах большого размера, при использовании соответствующих горячих клавиш для поиска необходимой информации в коде и т.п.

В качестве среды разработки для реализации и создания функционала веб-сервиса была использована платформа разработки веб-приложений ASP.NET, входящая в платформу .NET Framework. Такая среда разработки позволяет специалисту точно и без появления различных проблем внедрять программный код для поддержки функционала веб-сервиса прямо в исходный код верстки или, иными словами, дизайна веб-страницы. На рисунке ниже представлен стандартный интерфейс Microsoft Visual Studio с фрагментом внедрения языка программирования C# в программный код дизайна веб-страницы (см. Рисунок 5).

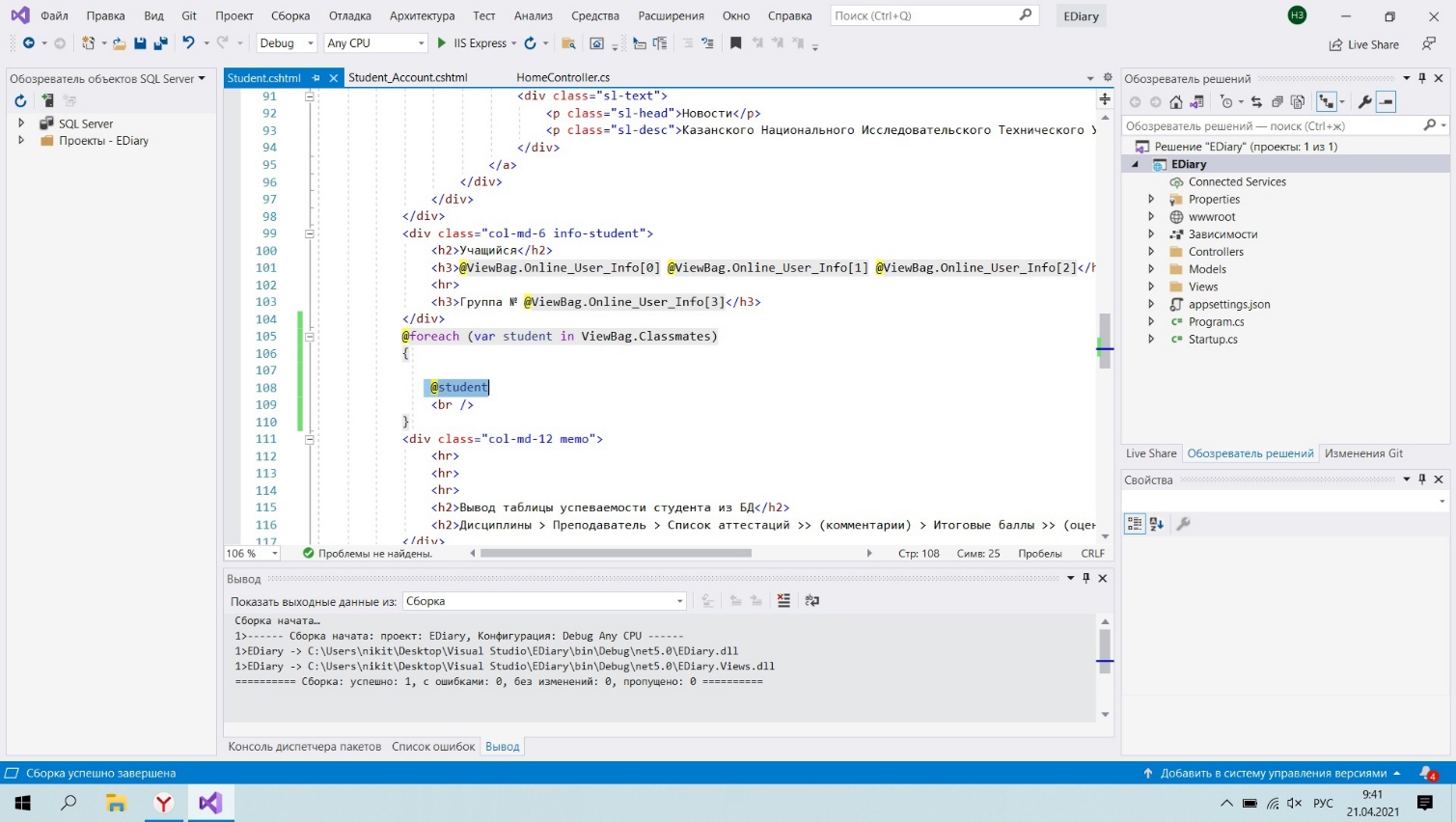


Рисунок 5 - Интерфейс интегрированной среды разработки

* + 1. **Обзор средств реализации верстки страницы**

В настоящее время на рынке клиент–серверных приложений, помимо использования различных языков программирования для разработки веб-проектов, например, PHP, Python, TypeScript, использующийся как расширение JavaScript, и прочие, существуют лишь два основных языка, без которых ни одна страница в сети Интернет не будет отображаться так, как задумывает пользователь или заказчик.

HyperText Markup Language, или же сокращенно, HTML – является общепринятым языком разметки гипертекста, позволяющим открывать HTML–страницы. Страницы открываются браузерами, при этом обмениваясь с сервером информацией по протоколу HTTP или HTTPS, в зависимости от степени защищенности, в виде простого текста или с использованием шифрования. HTML является основой и фундаментом всех веб-страниц, который хранит в себе информацию о выбранных автором шрифтах, кодировках, стилях, цвете текста и т.п. HTML совместим со множеством различных компьютерных платформ.

Cascading Style Sheets (сокращенно, CSS – каскадные таблицы стилей) – это формальный язык, предназначенный для преобразования внешнего вида веб-страницы на сайте в удобную для восприятия пользователем форму. Как правило, язык CSS используют для описания интернет–страниц, уже написанных на языках HTML, либо XHTML. Данный язык задает стиль веб-страницы. При помощи CSS происходит разработка дизайна, верстка модели сайта для различных устройств, называемая адаптивной версткой.

При разработке и организации дизайна веб-страниц для облегчения этого процесса используется платформа Bootstrap. Bootstrap – это общедоступный фреймворк, который включает в себя HTML, CSS и язык программирования JavaScript. Данная платформа предназначена для организации быстрого, удобного и адаптивного интерфейса веб-сервиса.

Вышеперечисленные языки разметки гипертекста и стиля совместно составляют основу любой веб-страницы в сети Интернет. Далее автор по своему усмотрению выбирает любой из имеющихся языков программирования, чтобы реализовать требуемые функции.

Язык разметки HTML и формальный язык стилей CSS являются обязательными языками для разработки веб-страниц. В совокупности они организуют основу интерфейса и дизайна веб-страницы, которая позволяет реализовать следующие возможности:

* возможность управления любыми документами в едином блоке;
* доступность функции настройки отображения веб-страницы для всех современных устройств.
  + 1. **Обзор языков программирования и средств разработки и управления базами данных**

Вышеупомянутые языки программирования, такие как PHP, Python, JavaScript, TypeScript, использующийся как расширение JavaScript, Visual C# и другие современные языки, в совместном использовании с языком разметки HTML и языком каскадных стилей CSS образуют практически готовый и работоспособный веб-сервис для работы различных пользователей. К использованию языков при разработке веб-проектов добавляется использование средств разработки баз данных, например, SQL, phpMyAdmin, PostrgeSQL, mongoDB, orientDB и другие.

Язык программирования JavaScript является схожим по реализованному функционалу с PHP – это язык сценариев или же скриптов, с помощью которых разрабатываются интерактивные веб-страницы. В браузере для JavaScript доступно всё, что связано с манипулированием веб-страницами, взаимодействием с пользователем и веб-сервером. JavaScript позволяет программисту выполнять следующие функции:

* добавление нового HTML-кода на страницу, изменение существующего содержимое, модификация стилей;
* возможность реагирования на действия пользователя, щелчки мыши, перемещения указателя, нажатия клавиш;
* отправление сетевых запросов на удалённые сервера, скачивание и загрузка файлов при помощи технологи ajax-запросов;
* возможность запоминать данные на стороне клиента, при помощи local storage.

Также использование JavaScript в разработке веб-сервиса позволяет реализовать следующий ряд возможностей, которые другие языки программирования либо не способны реализовать, либо делают эти возможности трудоемкими:

* создание скриптов для выполнения на стороне сервера;
* создание скриптов для выполнения в командной строке;
* создание приложений GUI, выполняющихся на стороне клиента.

Язык программирования JavaScript является языком, который позволяет веб-сервису обрести свойство интерактивности, придать веб-странице требуемые анимации. Однако для разработки полноценного веб-проекта одного языка программирования недостаточно. Для реализации всех необходимых функций в модулях реализуемой системы был использован язык программирования C Sharp при использовании платформы разработки веб-сервисов ASP.NET.

C Sharp является объектно-ориентированным языком, разработанным в начале 2000-х годов группой инженеров компании Microsoft. Данный язык программирования позволяет создать не только консольные приложения и тому подобные, но и позволяет организовать функциональную часть веб-сервисов, или так называемый back-end. Back-end – это программно-аппаратная часть сервиса, которая отвечает за осуществление функционирования внутренней части веб-сайта.

В программировании для хранения данных существует огромное количество баз данных и систем управления базами данных. В рамках разработки веб-страниц пользуются популярностью такие базы данных, как SQL, phpMyAdmin, PostrgeSQL, mongoDB, orientDB и т.п. Для реализации базы данных приложения дипломной работы будет использована система управления реляционными базами данных Microsoft SQL Server. Данный выбор обусловлен простотой использования при разработке схемы базы данных и наличием адаптеров и коннекторов для взаимодействия с платформой .NET.

В разработке информационной системы была реализована объектно-ориентированная технология доступа к данным Entity Framework, которая является встроенной в платформу разработки .NET Framework от компании Microsoft. Entity Framework предоставляет возможность взаимодействия с объектами данных при помощи языка Entity SQL, который практически неотличим от языка программирования SQL, предназначенного для модификации и управления данными в базе данных реляционного типа.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

## **2.1 Постановка задачи**

Целью данного дипломного проекта является разработка информационной системы для клиентов автомойки с использованием платформы .NET Framework.

Для достижения поставленной цели необходимо реализовать следующие задачи:

1. Провести анализ предметной области;

2. Выбрать средства разработки и реализации программных модулей;

3. Спроектировать и создать дизайн программных модулей, обладающих следующими свойствами: удобство, понятность, простота в использовании, полная адаптивность на различных электронных устройствах;

4. Спроектировать архитектуру программной системы;

5. Создать базу данных для хранения всей необходимой информации;

6. Разработать функциональные модули программной системы в соответствии с выбранными средствами разработки;

7. Выполнить тестирование программного продукта с целью проверки полной работоспособности системы и выявления соответствия ранее заявленным требованиям;

8. Выполнить разработку полной технической документации для дальнейшего сопровождения и помощи в эксплуатации пользователями разработанного программного продукта.

## **2.2 Концептуальная модель программной системы**

Создание любой программной системы всегда состоит из нескольких этапов разработки. Одним из основных этапов разработки является этап моделирования, который предшествует этапу создания системы. Продуманные и созданные модели программной системы позволяют обеспечить правильное взаимодействие и понимание работы между разработчиками системы внутри команды, а также непосредственно между разработчиками и заказчиком проекта. Модель строится на основных и дополнительных требованиях заказчика к программной системе, затем заказчик может провести проверку на соответствие созданной модели с его требованиями к системе. Модель программной системы должна в полной мере отображать свойства разрабатываемой системы.

Для описания модели создаваемой программной системы разработчики используют технологию моделирования при помощи унифицированного языка моделирования UML. UML позволяет использовать графический язык отображения спецификаций, проектирования и документирования системы.

Взаимодействие между пользователями или акторами при разработке программных систем отображают при помощи соответствующих диаграмм прецедентов, или иными словами, диаграмм вариантов использования. Такие диаграммы в полной мере должны отражать функции системы, которые заказчик отметил обязательными. Диаграммы прецедентов обычно называют use-case диаграммами.

На use-case диаграммах изображаются следующие элементы:

* акторы или пользователи – группа лиц, взаимодействующих в разрабатываемой системе;
* варианты использования (прецеденты) – сервисы и процессы, которые программная система способна предоставлять акторам;
* отношения между элементами диаграммы;
* комментарии.

Любые программные системы проектируются с учётом того, что в процессе своей работы они будут использоваться людьми или будут взаимодействовать с другими системами. Сущности, с которыми взаимодействует система в процессе своей работы, называются акторами. Каждый актор ожидает, что система будет вести себя строго определённым и предумышленным образом. Обычно на графических диаграммах прецедентов акторы изображаются человеком в схематичном виде.

Описание акторов модулей разрабатываемой системы представлено в таблице 1

Таблица 1 – Выявление акторов

| **Акторы** | **Краткое описание** |
| --- | --- |
| Менеджер | Пользователь, который обладает следующими функциональными возможностями:   * просмотр клиентских заявок на услуги; * одобрение заявок; * просмотр журнала записей; * добавление, редактирование и удаление записей; * просмотр журнала услуг; * добавление, редактирование и удаление услуг; * просмотр списка сотрудников; добавление, редактирование и удаление сотрудников; * просмотр нагрузок сотрудников; просмотр машин; * добавление, редактирование и удаление машин. |
| Клиент | Пользователь, который обладает следующими функциональными возможностями:   * оставить заявку на услугу; * написать в службу поддержку для получения обратной связи; * просмотр журнала услуг. |

При разработке любой системы и ПО необходимо провести построение диаграмм прецедентов, которая в дальнейшем позволит описать пользовательские требования, требования к взаимодействую между другими системами, а также позволит создать модель взаимодействия людей и существующих компаний в реальной жизни.

Выявленные варианты использования для ранее указанных акторов представлены в таблице 2.

Таблица 2 – Варианты использования

| **Основной актор** | **Наименование** | **Формулировка** |
| --- | --- | --- |
| Менеджер | Просмотр клиентских заявок на услуги | Менеджер может просматривать список заявок, поступивших от клиентов |
| Менеджер | Одобрение заявки | Менеджер может одобрить заявку для будущей работы с клиентом |
| Менеджер | Просмотр журнала записей | Менеджер может просматривать журнал записей проделанной работы |
| Менеджер | Добавление записи | Менеджер может новые записи в журнал |
| Менеджер | Редактирование записи | Менеджер может редактировать записи в журнале |
| Менеджер | Удаление записи | Менеджер может удалять записи из журнала |
| Менеджер | Просмотр журнала услуг | Менеджер может просматривать журнал предоставляемых автомойкой услуг |
| Менеджер | Добавление услуги | Менеджер может добавлять новые услуги в журнал |
| Менеджер | Редактирование услуги | Менеджер может редактировать услуги в журнале |
| Менеджер | Удаление услуги | Менеджер может удалять услуги с журнала |
| Менеджер | Просмотр списка сотрудников | Менеджер может просматривать список всех сотрудников, работающих в данной сети |
| Менеджер | Добавление сотрудника | Менеджер может добавлять новых сотрудников в систему |
| Менеджер | Редактирование сотрудника | Менеджер может редактировать сотрудника в системе |
| Менеджер | Удаление сотрудника | Менеджер может удалять сотрудника из системы |
| Менеджер | Просмотр машин | Менеджер может просматривать список машин и их классификации |
| Менеджер | Добавление машины | Менеджер может новые машины в систему |
| Менеджер | Редактирование машины | Менеджер может редактировать наименование модели и марки машины |
| Менеджер | Удаление машины | Менеджер может удалять машины из системы |
| Клиент | Просмотр журнала услуг | Клиент может просматривать список предоставляемых автомойкой услуг |
| Клиент | Запись на услугу | Клиент может записаться на услугу через специальную форму |
| Клиент | Сообщение в службу поддержки | Клиент может обратиться с вопросами и предложениями в службу поддержки |

На основе выявленных вариантов использования для акторов на Рисунок представлена диаграмма прецедентов или use-case диаграмма.



Рисунок 6 – Диаграмма вариантов использования программной системы

В таблице 3 показан полный шаблон варианта использования «Добавить машину».

Таблица 3 – Описание варианта использования «Добавить машину»

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название:** | | ТС-1 Добавить машину | |
| **Автор:** | Айдар | **Дата создания** | 26.04.2022 |
| **Основное действующее лицо:** | Менеджер | **Дополнительное действующее лицо:** | - |
| **Описание:** | | Пользователь может добавить новую машину | |
| **Триггер:** | | При нажатии кнопки «Добавить машину» | |
| **Предварительные условия:** | | PRE-1. Открыт список машин | |
| **Выходные условия:** | | POST-1. Машина появляется на главном экране | |
| **Нормальное направление развития варианта использования:** | | **Добавить машину**   1. Открыть список машин; 2. Нажать кнопку «Добавить»; 3. Ввести модель и марку машины; 4. Сохранить машину. | |
| **Альтернативное направление развития варианта использования:** | | **Редактировать машину**   1. Открыть список машин; 2. Нажать на нужную машину; 3. Нажать «Редактировать»; 4. Изменить марку или модель машины; 5. Сохранить машину. | |
| **Исключения:** | | 1. **Не указано название** 2. При сохранении машины не введено название. | |
| **Приоритет:** | | Низкий | |
| **Частота использования:** | | Примерно раз в год | |
| **Бизнес-правила:** | | **-** | |
| **Специальные требования:** | | **-** | |
| **Предположения:** | | **-** | |

В таблице 4 показан полный шаблон варианта использования «Добавить сотрудника».

Таблица 4 – Описание варианта использования «Добавить сотрудника»

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название:** | | ТС-2 Добавить сотрудника | |
| **Автор:** | Айдар | **Дата создания** | 26.04.2022 |
| **Основное действующее лицо:** | Менеджер | **Дополнительное действующее лицо:** | - |
| **Описание:** | | Пользователь может добавить нового сотрудника | |
| **Триггер:** | | При нажатии кнопки «Добавить сотрудника» | |
| **Предварительные условия:** | | PRE-1. Открыт список сотрудников | |
| **Выходные условия:** | | POST-1. Сотрудник появляется на главном экране | |
| **Нормальное направление развития варианта использования:** | | **Добавить машину**   1. Открыть список сотрудников 2. Нажать кнопку «Добавить» 3. Ввести фамилию имя отчество и номер телефона сотрудника 4. Сохранить сотрудника | |
| **Альтернативное направление развития варианта использования:** | | **Редактировать машину**   1. Открыть список сотрудников 2. Нажать на нужного сотрудника 3. Нажать «Редактировать» 4. Изменить фамилию имя отчество или номер телефона сотрудника 5. Сохранить машину | |
| **Исключения:** | | 1. **Не указаны все данные** 2. При сохранении сотрудника не введено имя | |
| **Приоритет:** | | Низкий | |
| **Частота использования:** | | Примерно раз в пол год | |

В таблице 5 показан полный шаблон варианта использования «Добавить услугу».

Таблица 5 – Описание варианта использования «Добавить услуги»

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название:** | | ТС-3 Добавить услугу | |
| **Автор:** | Айдар | **Дата создания** | 26.04.2022 |
| **Основное действующее лицо:** | Менеджер | **Дополнительное действующее лицо:** | - |
| **Описание:** | | Пользователь может добавить новую услугу | |
| **Триггер:** | | При нажатии кнопки «Добавить услугу» | |
| **Предварительные условия:** | | PRE-1. Открыт список услуг | |
| **Выходные условия:** | | POST-1. Услуга появляется на главном экране | |
| **Нормальное направление развития варианта использования:** | | **Добавить услугу**   1. Открыть список услуг; 2. Нажать кнопку «Добавить»; 3. Ввести наименование, описание и цену услуги; 4. Сохранить услугу. | |
| **Альтернативное направление развития варианта использования:** | | **Редактировать услугу**   1. Открыть список услуг; 2. Нажать на нужную услугу; 3. Нажать «Редактировать»; 4. Изменить наименование, описание и цену услуги; 5. Сохранить услугу. | |
| **Исключения:** | | 1. **Не указаны все данные** 2. При сохранении услуги не введено наименование. | |
| **Приоритет:** | | Низкий | |
| **Частота использования:** | | Примерно раз в год | |

## **2.3 Логическая модель программной системы**

При разработке приложений необходимо учитывать не только варианты использования, но и конкретные алгоритмы поведения приложения при исполнении актором той или иной функциональной возможности. Такие алгоритмы поведения в совместном использовании позволяют разработчикам системы организовать логическую модель работы приложения или системы. При создании такой модели происходит организация схемы работы системы, которая предусматривает все сценарии поведения акторов или пользователей.

Пользователь в роли клиента при входе на веб-сайт автомойки встречает главная страница с формой для отправки заявки на услугу автомойки. При нажатии на кнопку связи с поддержкой будет открыто новое окошко для ввода сообщения и отправки.

Схема работы программной системы для клиентов автомойки от лица клиента представлена на Рисунок .

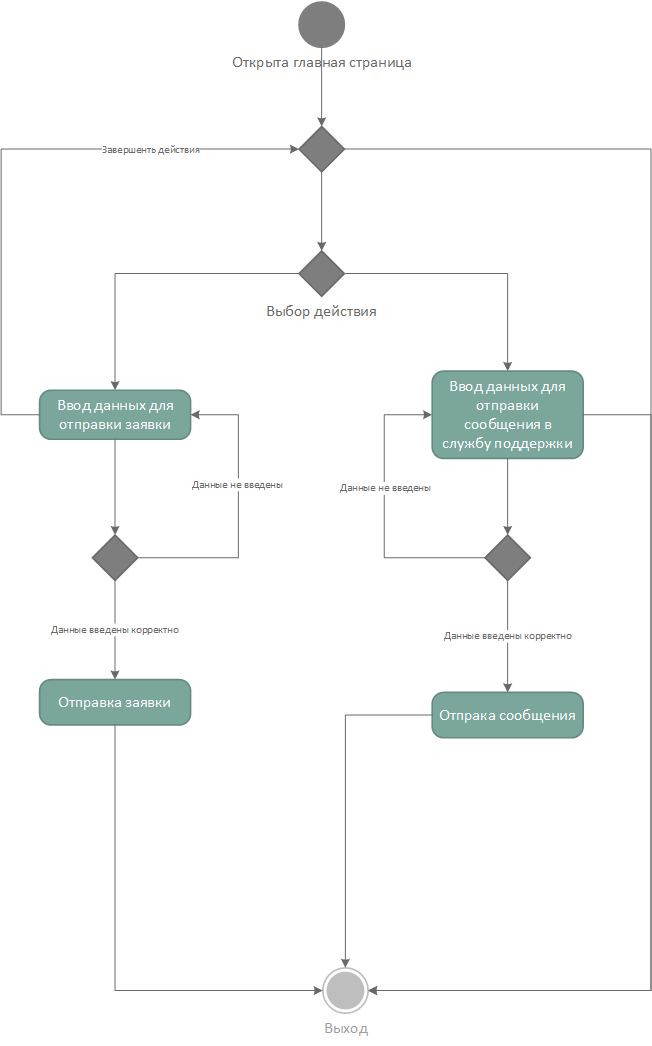


Рисунок 7 – Схема работы программной системы от лица клиента

Пользователь в роли менеджера при входе в приложение встречает просто окно авторизации для ввода пароля. Авторизовавшись, менеджер перенаправляется на главное меню приложения, на которой отображены основные элементы навигации, необходимые для перехода между окнами приложения. Далее менеджер может выбрать дальнейшие действия в приложении. При нажатии на кнопку «Заявки» менеджер переходит на страницу с заявками от клиентов, где можно выбрать один из двух вариантов: отклонить или принять заявку. Кнопкой «Назад» можно вернуться на главное меню приложения.

При нажатии на кнопку «Журнал» менеджер переходит на страницу с записями работ, где можно добавить новую запись с помощью специального окна добавления, а также можно отредактировать и удалить запись. Кнопка «Назад» возвращает менеджера обратно на главное меню.

При нажатии на кнопку «Сотрудники» менеджер переходит на страницу с сотрудниками данной сети, где так же можно добавить нового сотрудника в сеть благодаря специальному окну добавления, редактировать и удалить сотрудника из сети. На этой же странице можно посмотреть нагрузку для каждого сотрудника. Кнопка «Назад» возвращает менеджера обратно на главное меню.

При нажатии на кнопку «Услуги» менеджер переходит на страницу с услугами, которые предоставляет данная автомойка, с помощью специального окна можно добавить новую услуги, отредактировать услугу и удалить. Кнопка назад возвращает менеджера обратно на главное меню.

При нажатии на кнопку «Машины» менеджер переходит на страницу со списком машин, на которые распространяются услуги данной сети. Так же с помощью специального окна можно добавить новую машину, отредактировать и удалить машину. Кнопка назад возвращает менеджера обратно на главное меню. Схема работы программной системы учета успеваемости студентов от лица преподавателя представлена на **Ошибка! Источник ссылки не найден.**.

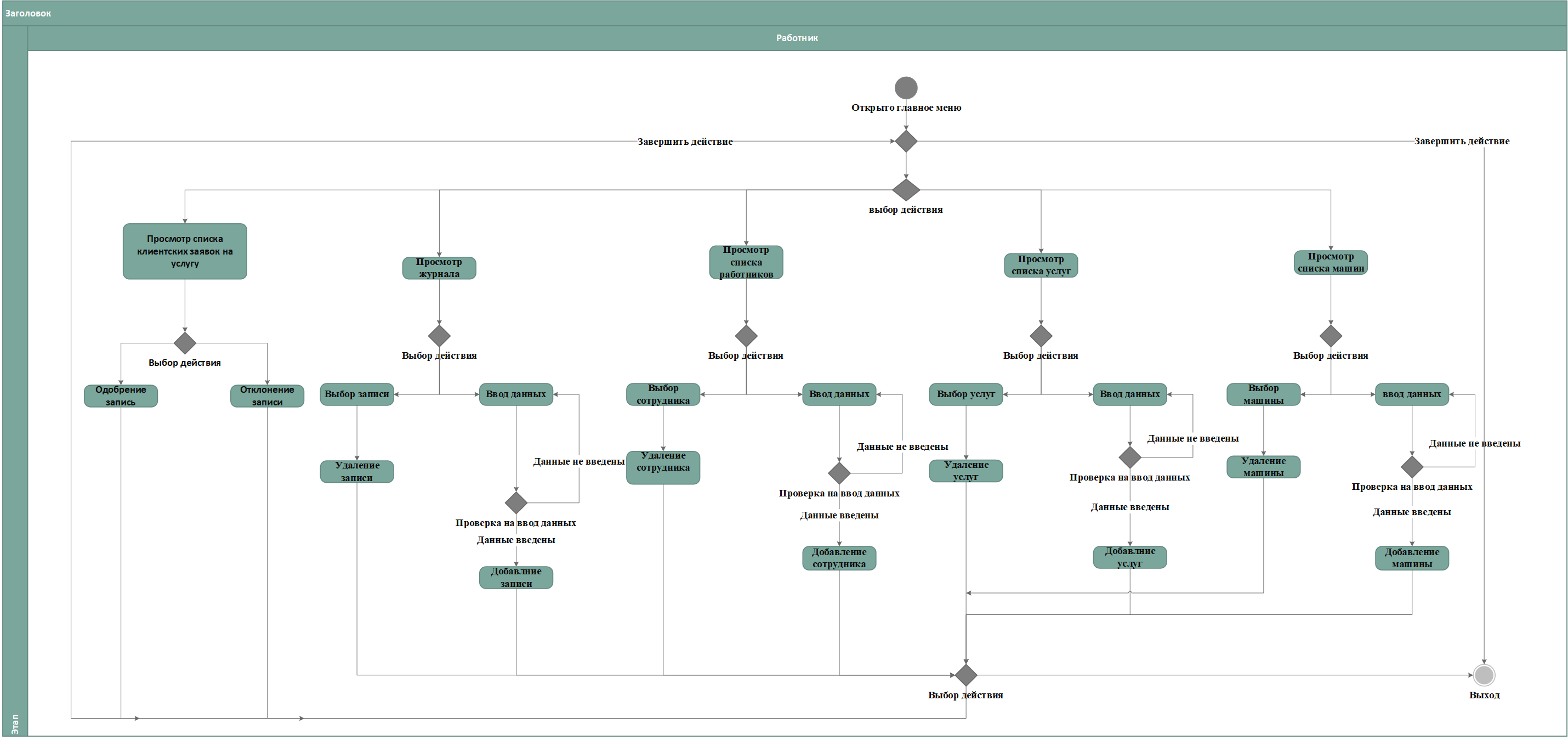


Рисунок 8 – Схема работы программной системы от лица Администратора

## **2.4 Диаграмма последовательности**

Диаграмма последовательности — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента.

Диаграммы последовательности хороши для представления взаимодействия объектов, но не очень подходят для точного определения поведения.

Диаграммы последовательности показывают взаимодействие, представляя каждого участника вместе с его линией жизни (lifeline), которая идет вертикально вниз и упорядочивает сообщения на странице. Сообщения также следует читать сверху вниз.

На рисунке 9 представлена диаграмма последовательности для прецедента «Просмотр клиентских заявок на услуги»

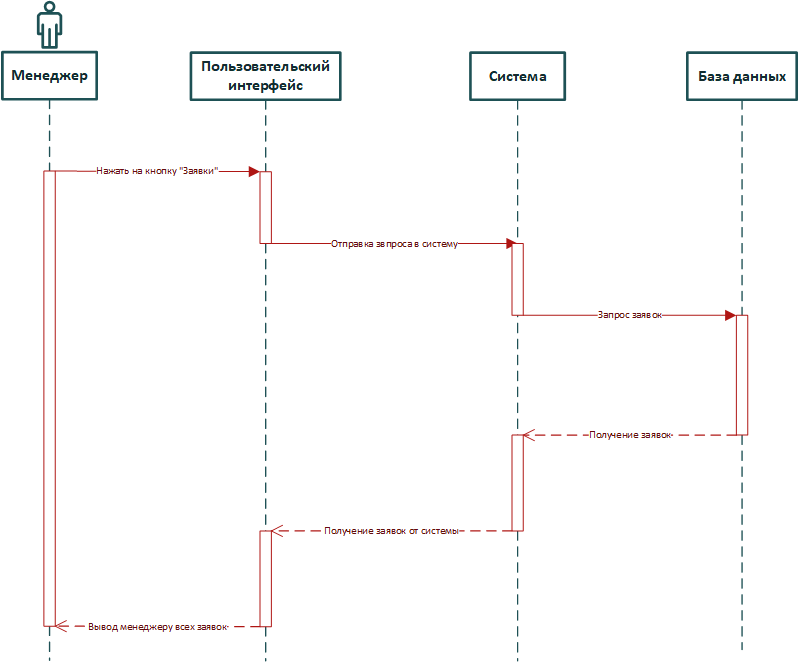


Рисунок 9 – Диаграмма последовательности для прецедента

«Просмотр клиентских заявок на услуги»

На рисунке 10 представлена диаграмма последовательности для прецедента «Просмотр журнала записей».

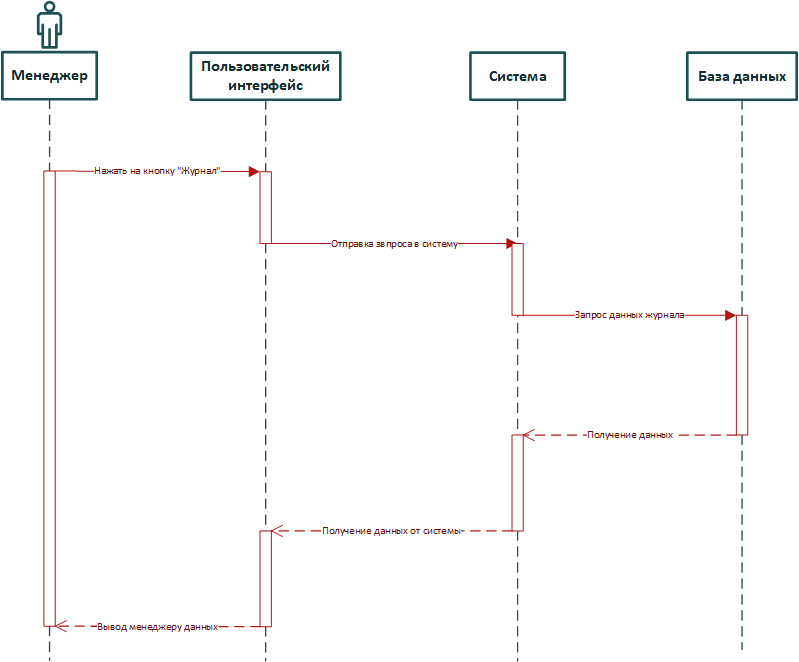


Рисунок 10 – Диаграмма последовательности для прецедента

«Просмотр журнала записей»

На рисунке 11 представлена диаграмма последовательности для прецедента «Просмотр журнала услуг»

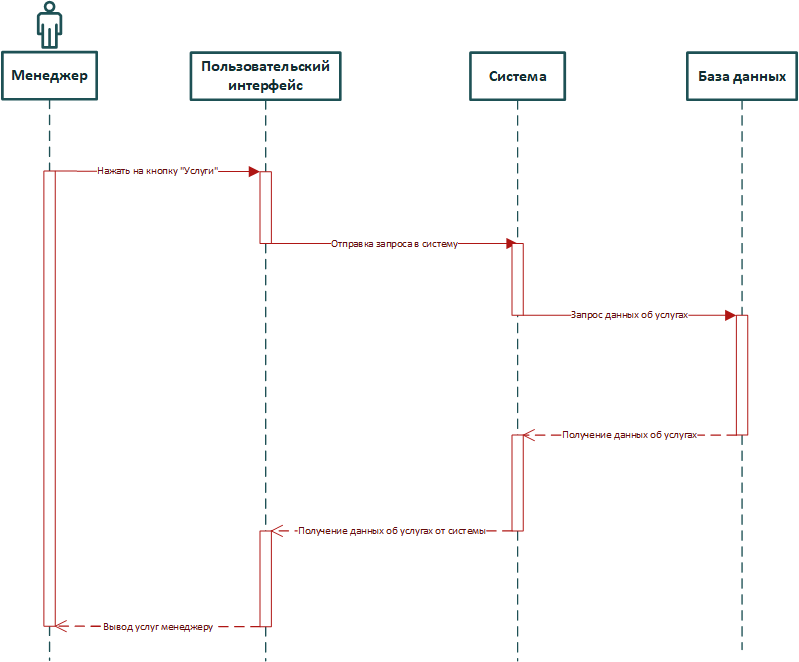
****

Рисунок 11 – Диаграмма последовательности для прецедента

«Просмотр журнала услуг»

На рисунке 12 представлена диаграмма последовательности для прецедента «Просмотр списка сотрудников».

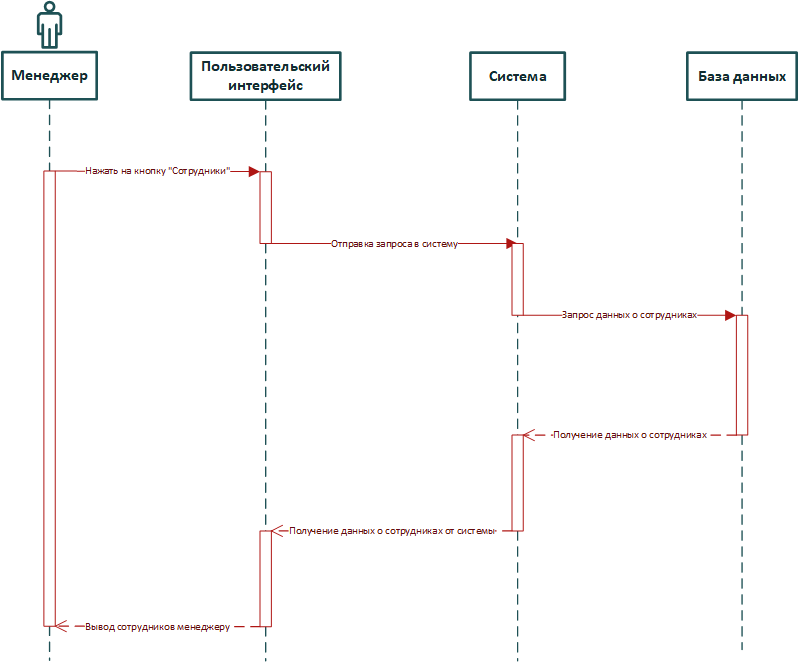
****

Рисунок 12 – Диаграмма последовательности для прецедента

«Просмотр списка сотрудников»

На рисунке 13 представлена диаграмма последовательности для прецедента «Просмотр машин».

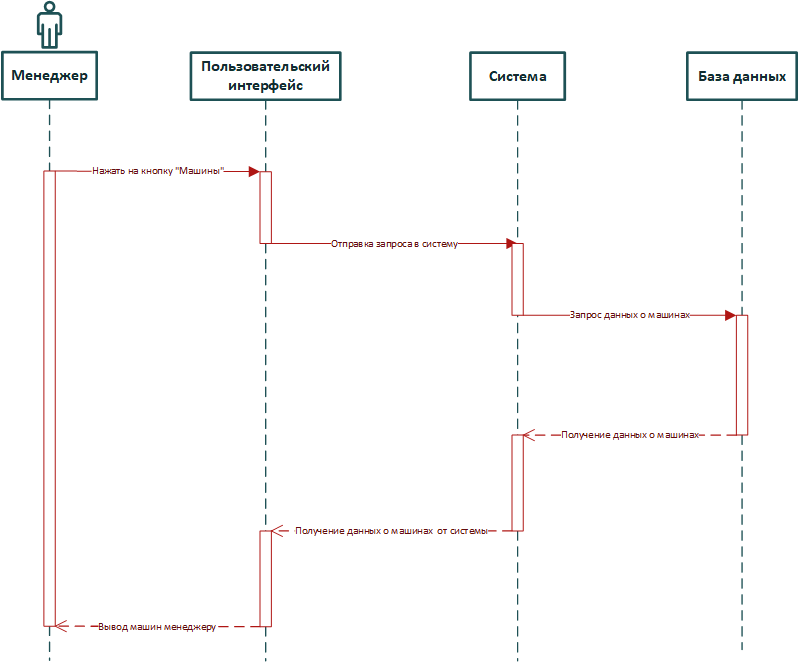


Рисунок 13 – Диаграмма последовательности для прецедента

«Просмотр машин»

На рисунке 14 представлена диаграмма последовательности для прецедента «Просмотр нагрузки сотрудника».

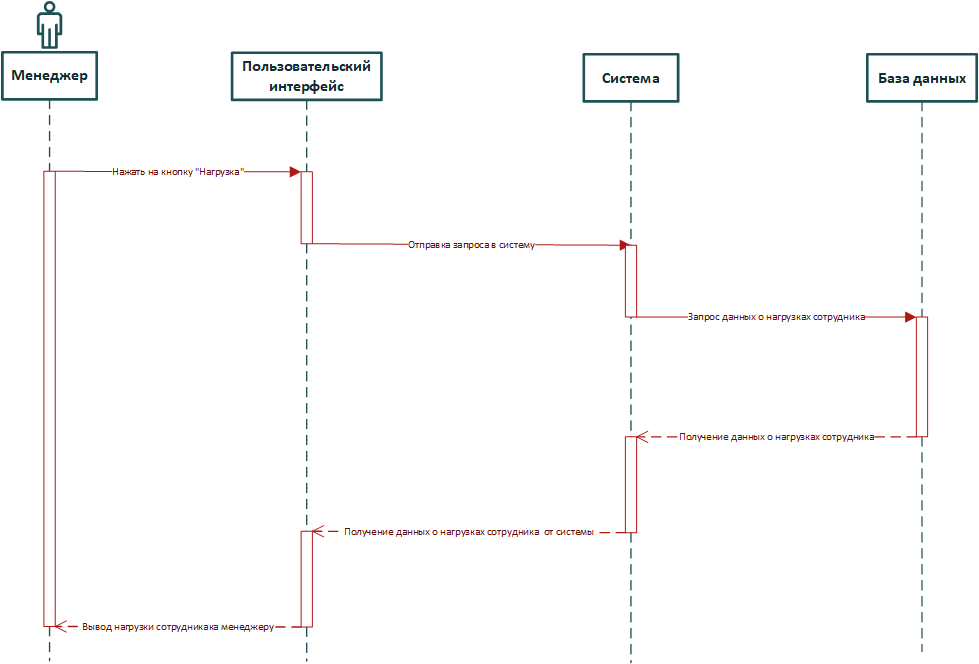


Рисунок 14 – Диаграмма последовательности для прецедента

«Просмотр нагрузки сотрудника»

На рисунке 15 представлена диаграмма последовательности для прецедента «Одобрение заявки».

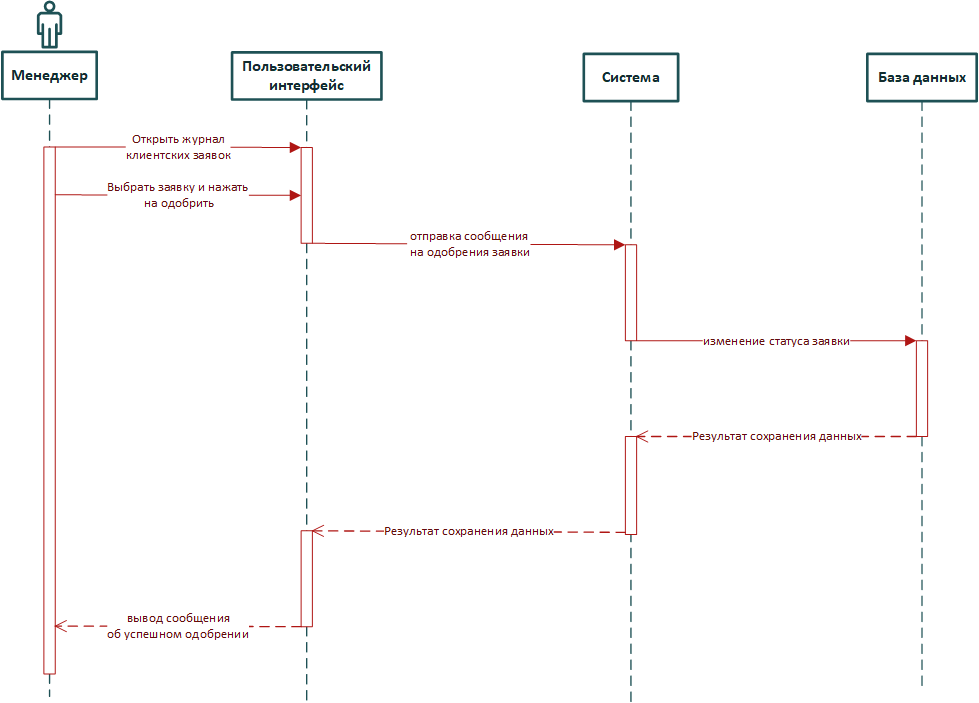


Рисунок 15 – Диаграмма последовательности для прецедента

«Одобрение заявки»

На рисунке 16 представлена диаграмма последовательности для прецедента «Добавление записи».

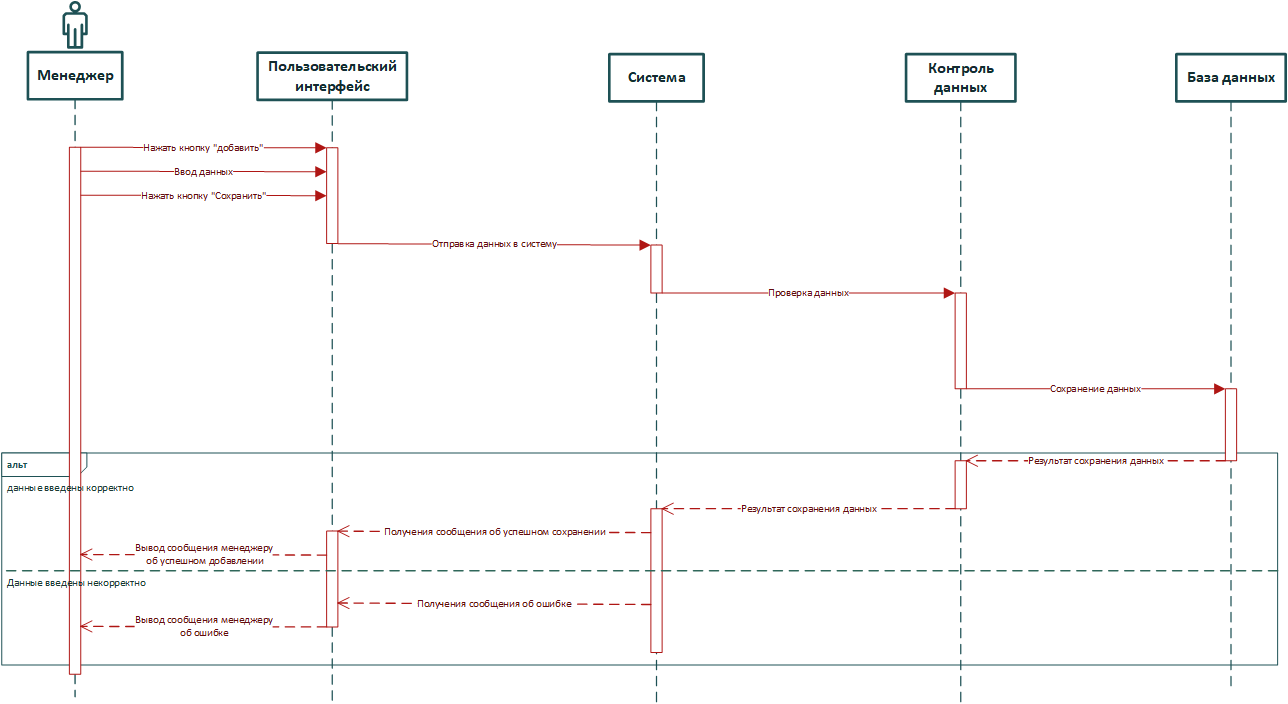


Рисунок 16 – Диаграмма последовательности для прецедента

«Добавление записи»

На рисунке 17 представлена диаграмма последовательности для прецедента «Редактирование записи»

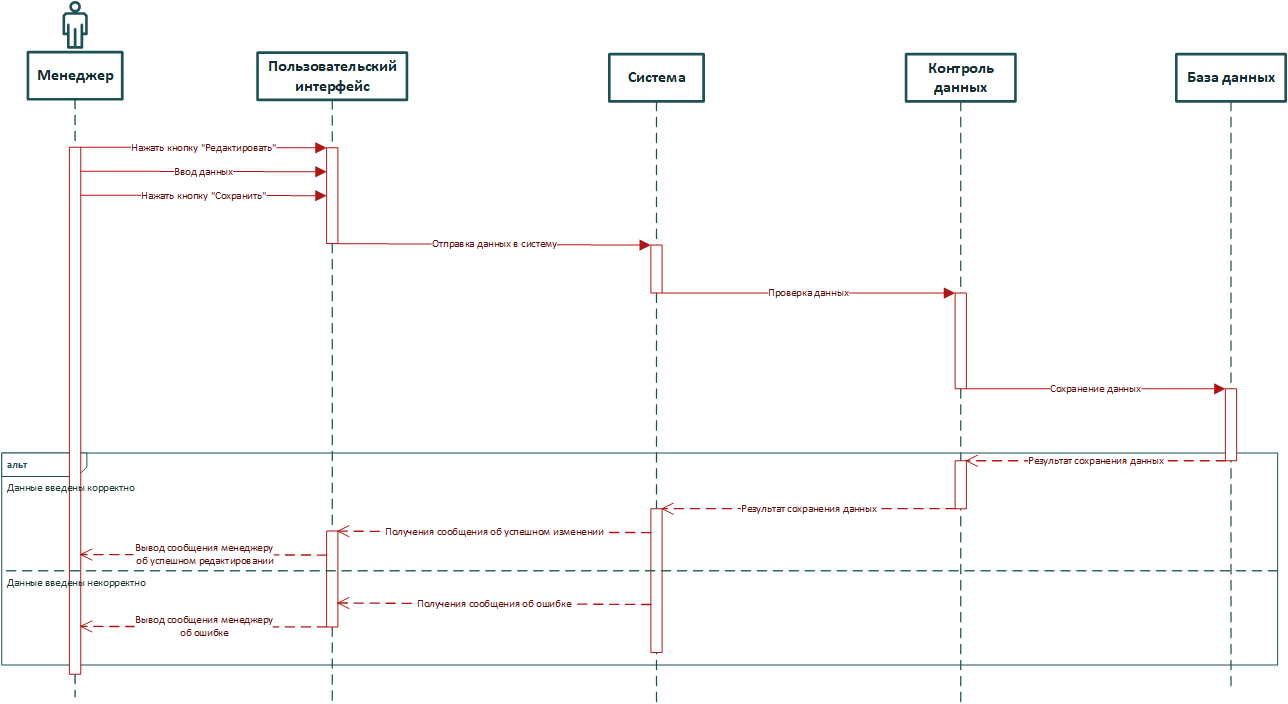


Рисунок 18 – Диаграмма последовательности для прецедента «Редактирование записи»

На рисунке 19 представлена диаграмма последовательности для прецедента «Удаление записи».

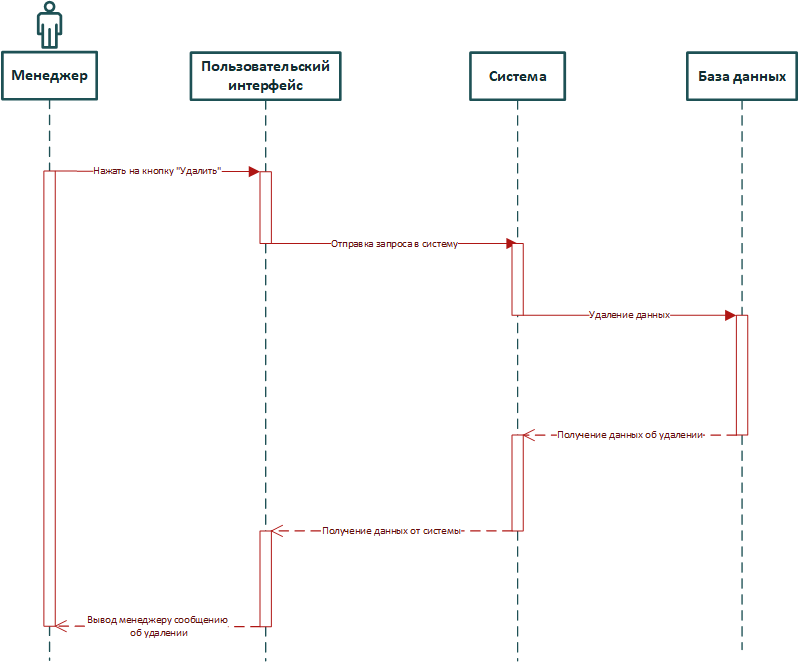


Рисунок 19 – Диаграмма последовательности для прецедента

«Удаление записи»

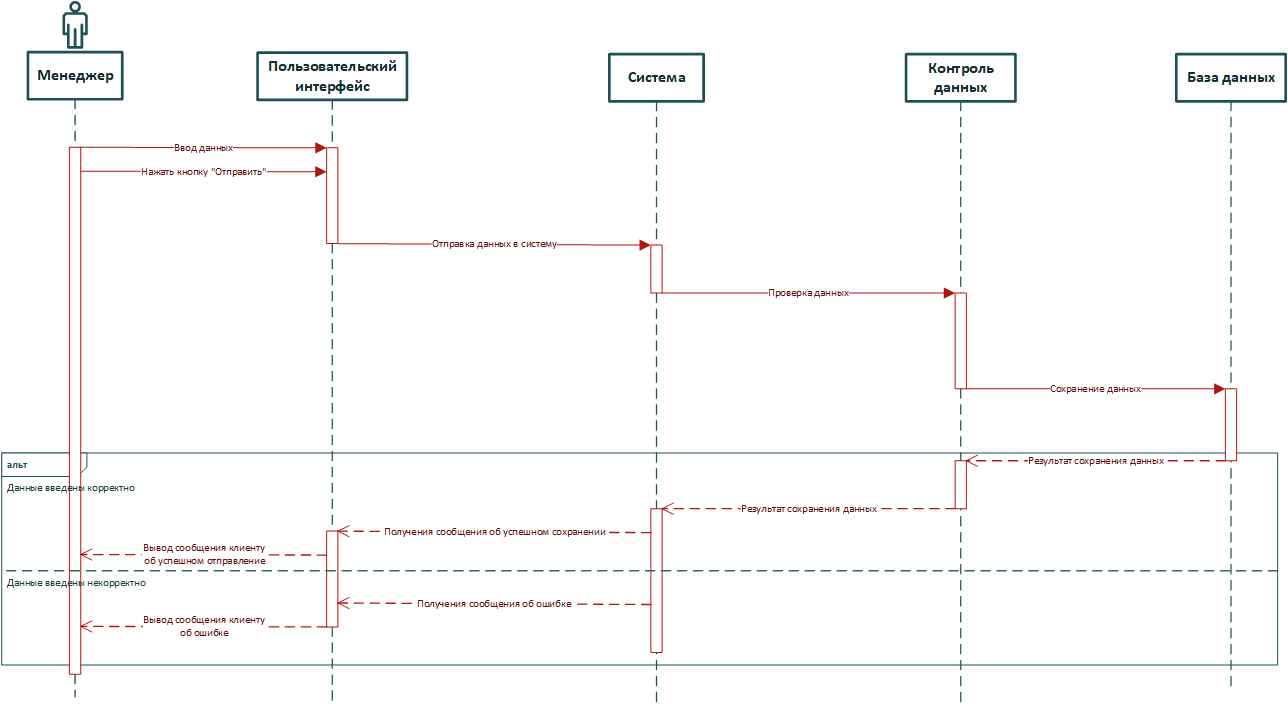


Рисунок 20 – Диаграмма последовательности для прецедента

«Записаться на услугу»

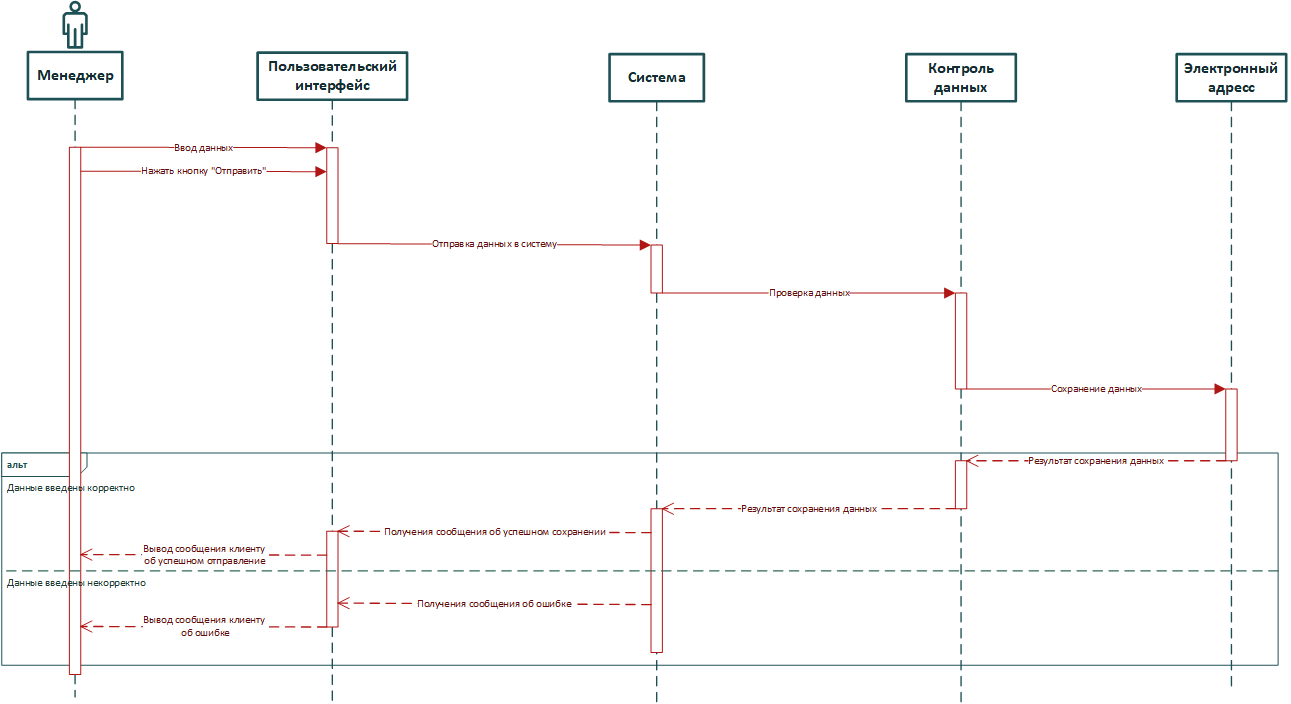


Рисунок 20 – Диаграмма последовательности для прецедента

«Написать в службу поддержки»

## **2.5 Диаграмма состояний**

Диаграммы состояний (state machine diagrams) – это известная технология описания поведения системы. В объектно-ориентированных подходах мы рисуем диаграмму состояний единственного класса, чтобы показать поведение одного объекта в течение его жизни.

Диаграмма описывает все возможные состояния, в которые может попасть конкретный объект, и то, как состояние объекта изменяется в результате событий, которые наступают для объекта. Состояние может быть прервано в результате наступления некоторого события

Список основных действий включает следующие значения:

• entry - действие, которое выполняется в момент входа в данное состояние (входное действие);

• exit - действие, которое выполняется в момент выхода из данного состояния (выходное действие);

• do - выполняющаяся деятельность ("do activity") в течение всего времени, пока объект находится в данном состоянии.

Построим диаграмму состояний для актора «Администратор». Для каждого прецедента добавляем внутренние активности, то есть определяем действия, которые могут возникнуть в рассматриваемом состоянии.

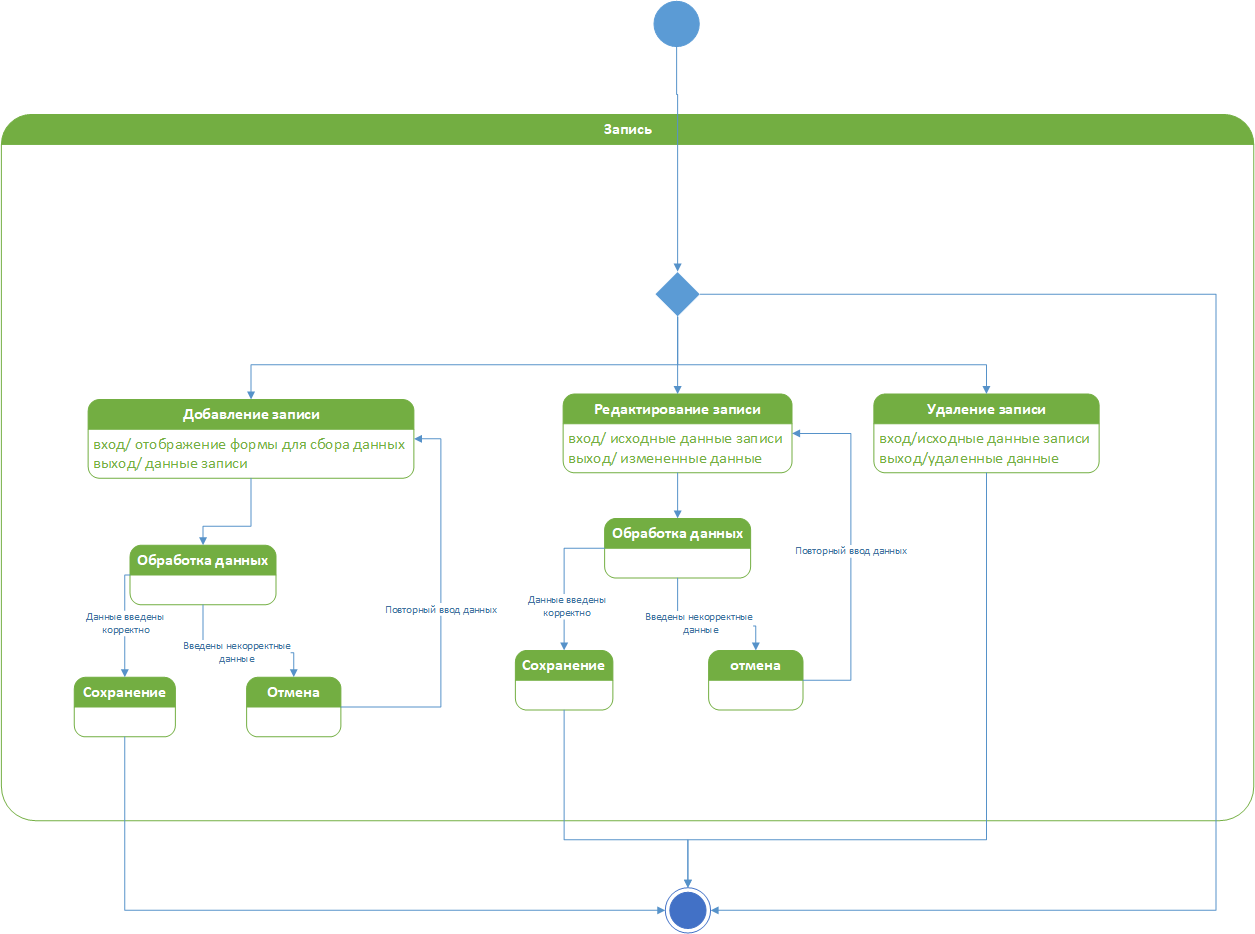
****

Рисунок 21 – Диаграмма состояний для прецедентов «Добавление, редактирование и удаления записи журнала»

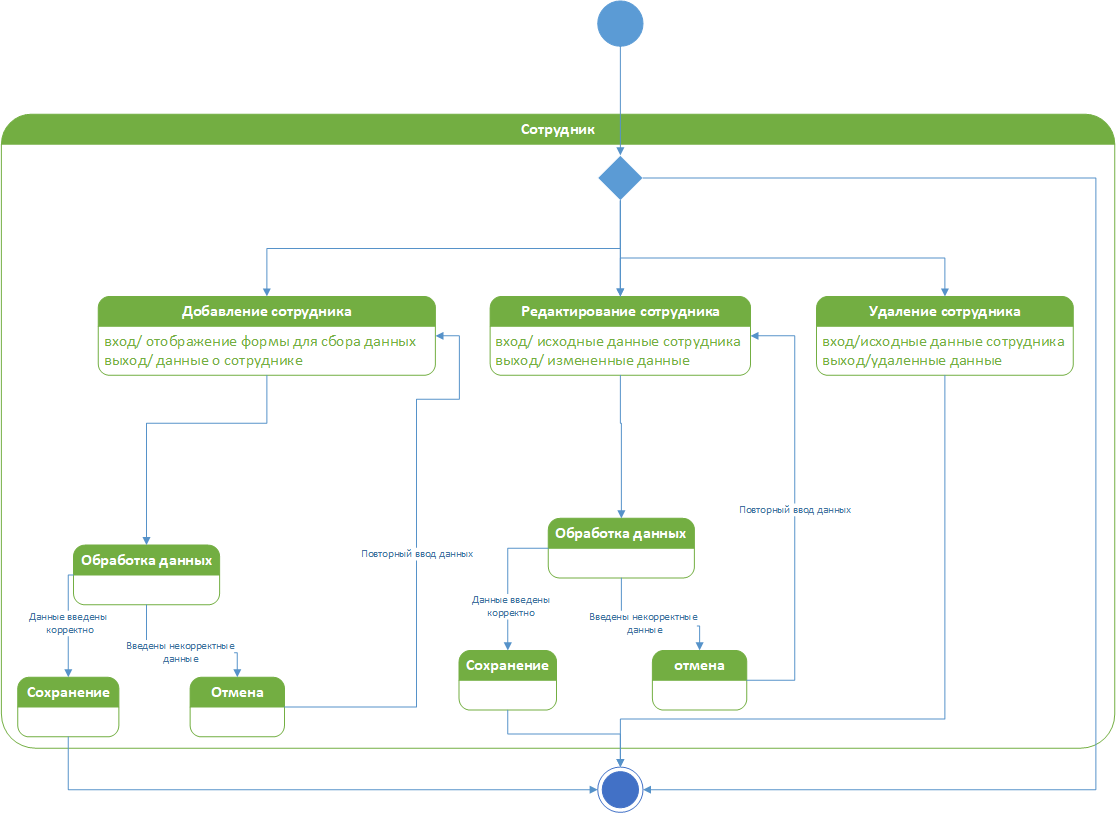
****

Рисунок 22 – Диаграмма состояний для прецедентов «Добавление, редактирование и удаления сотрудника»

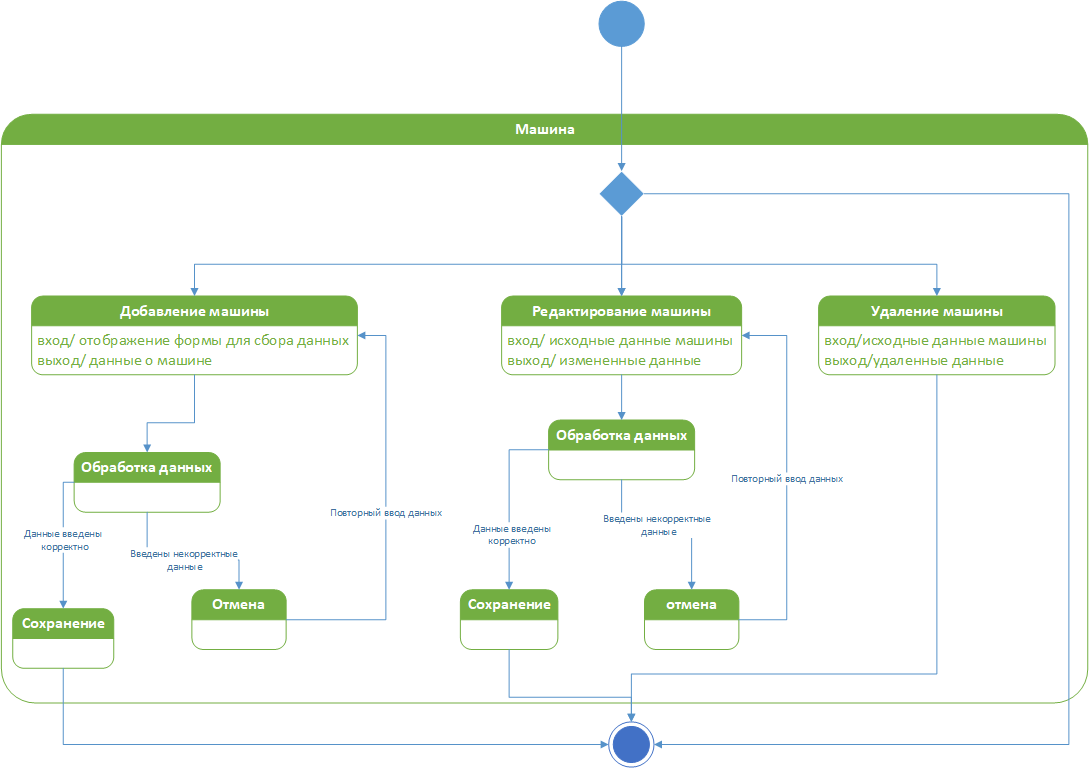


Рисунок 23 – Диаграмма состояний для прецедентов «Добавление, редактирование и удаления машины»

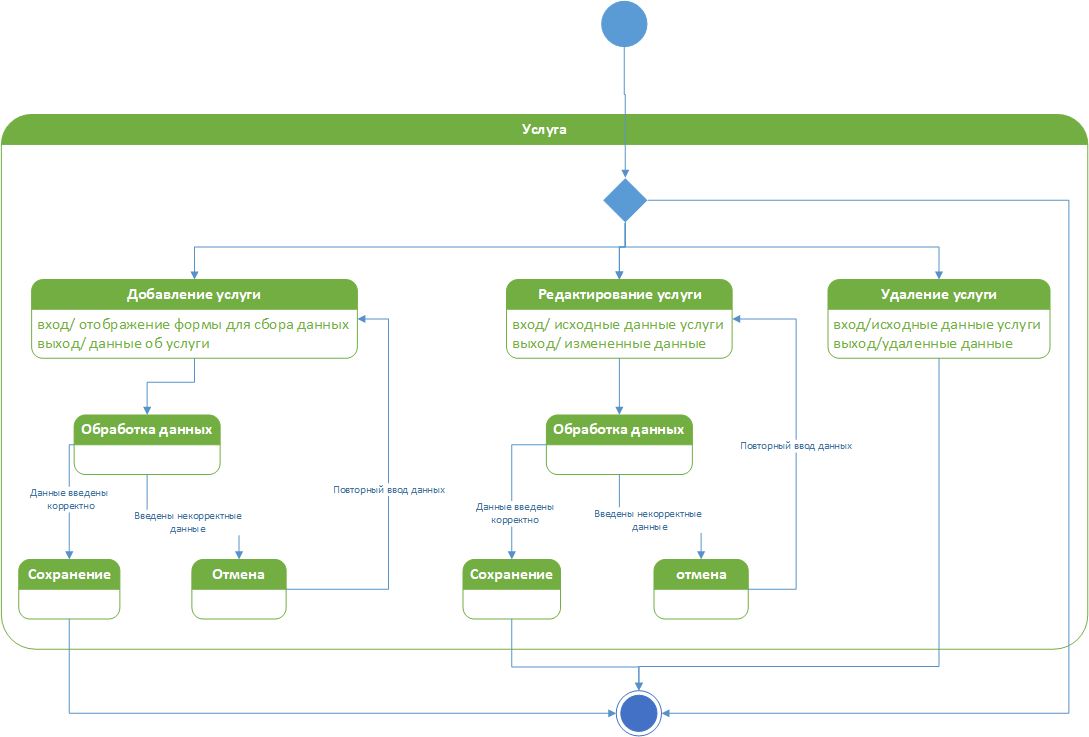


Рисунок 24 – Диаграмма состояний для прецедентов «Добавление, редактирование и удаления услуги»

## **2.6 Диаграмма классов**

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами

На диаграмме класс изображается в виде прямоугольника, разделенного на три части: имя класса (на английском языке), его атрибуты и его операции. В качестве классов часто выступают сущности, использованные при построении диаграммы сущность-связь.

Также, ключевым элементом диаграммы является атрибут, описывающий свойство в виде строки текста внутри прямоугольника класса. В зависимости от детализации диаграммы, синтаксис для атрибута может включать наименование, тип, значение по умолчанию и др.

В рисунке 25 представлена диаграмма классов для проектируемой системе.

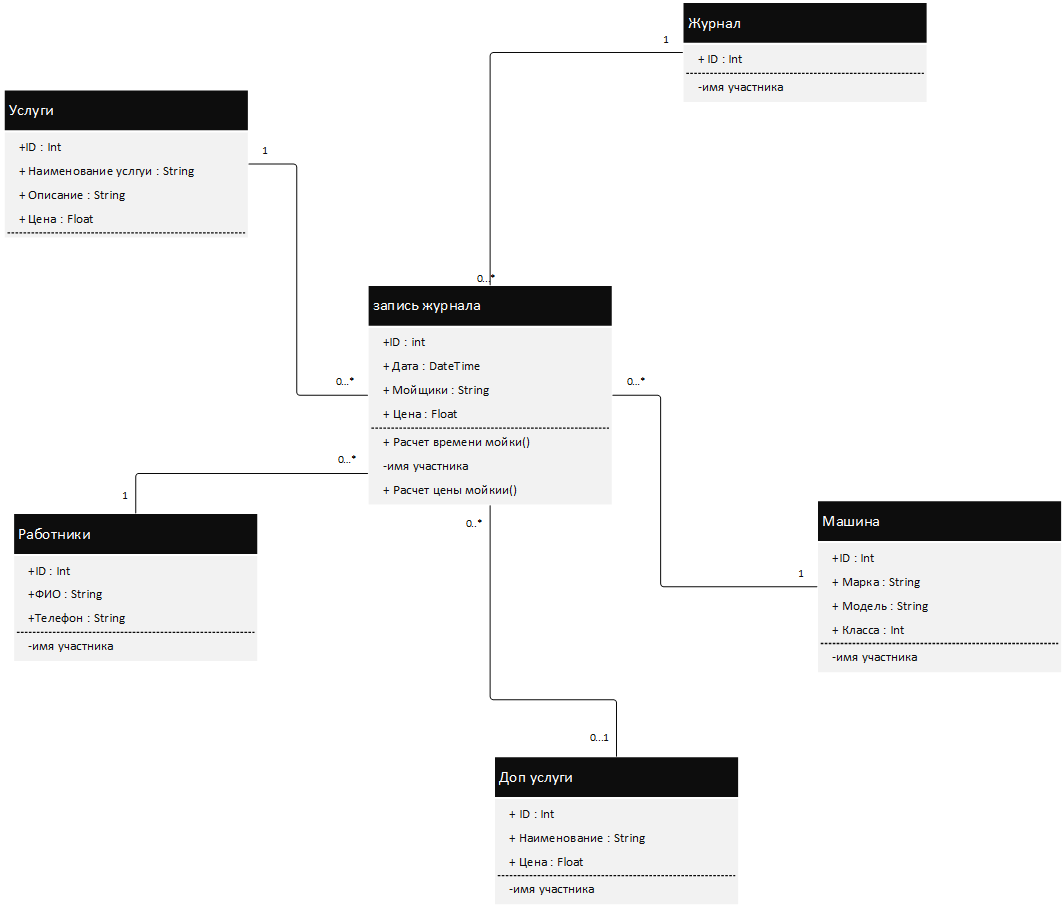


Рисунок 25 – Диаграмма классов

## **2.7 Проектирование интерфейса приложения**

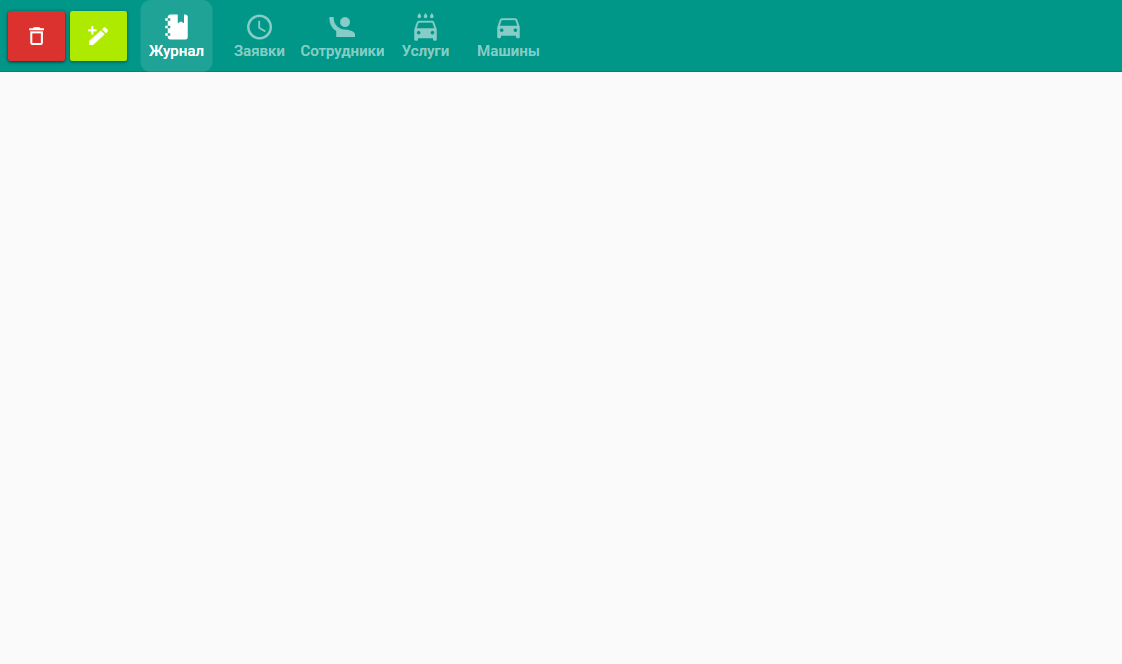


Рисунок 26 – «Окно вывода журнала»

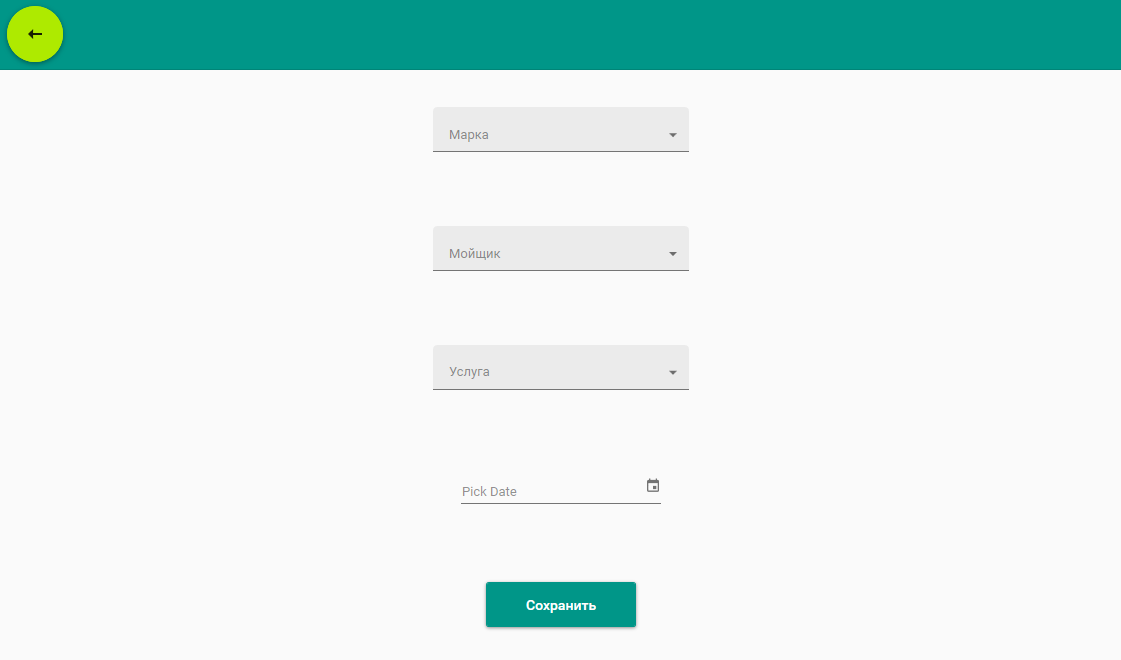


Рисунок 27 – «Окно добавления записи в журнал»

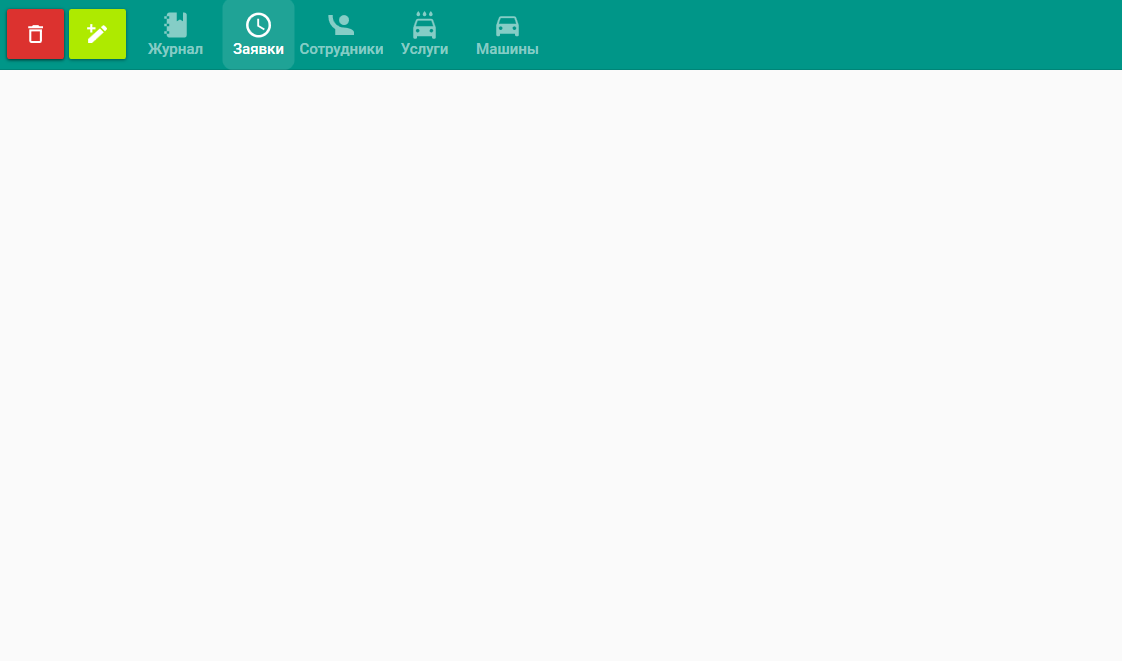


Рисунок 28 – «Окно вывода клиентских записей»

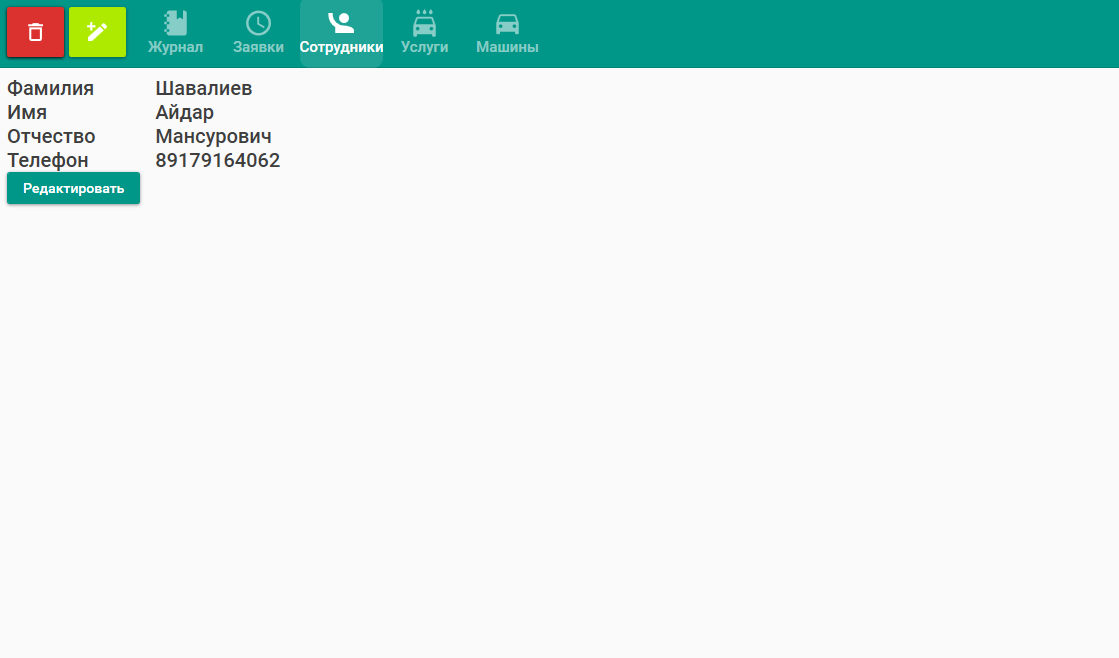


Рисунок 29 – «Окно со списками сотрудников»

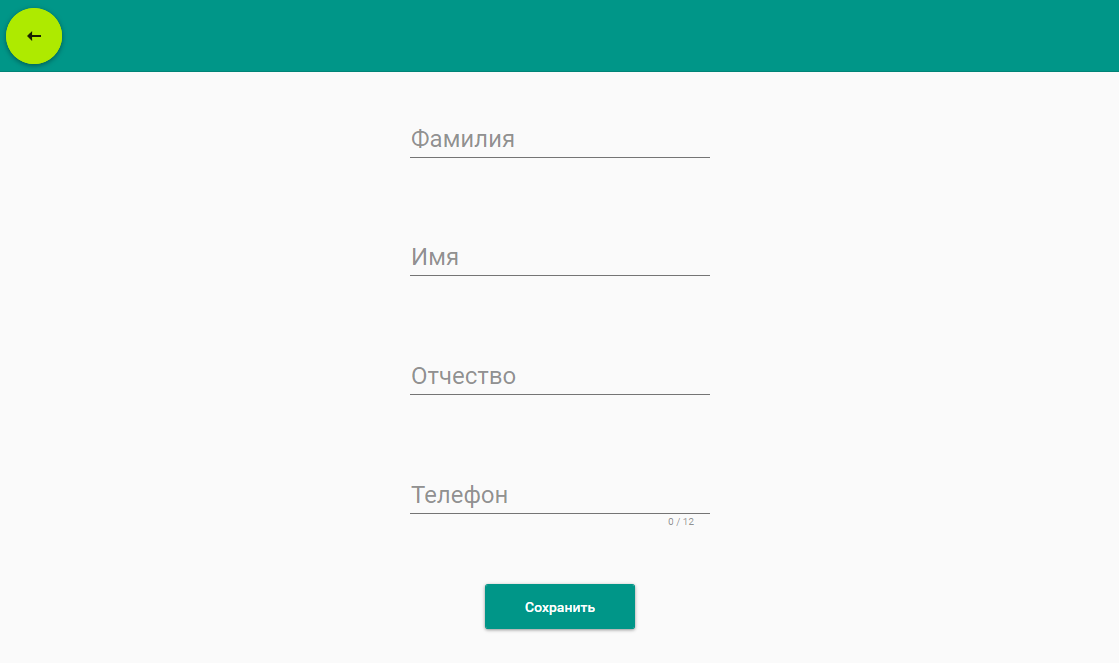


Рисунок 30 – «Окно добавления сотрудника»

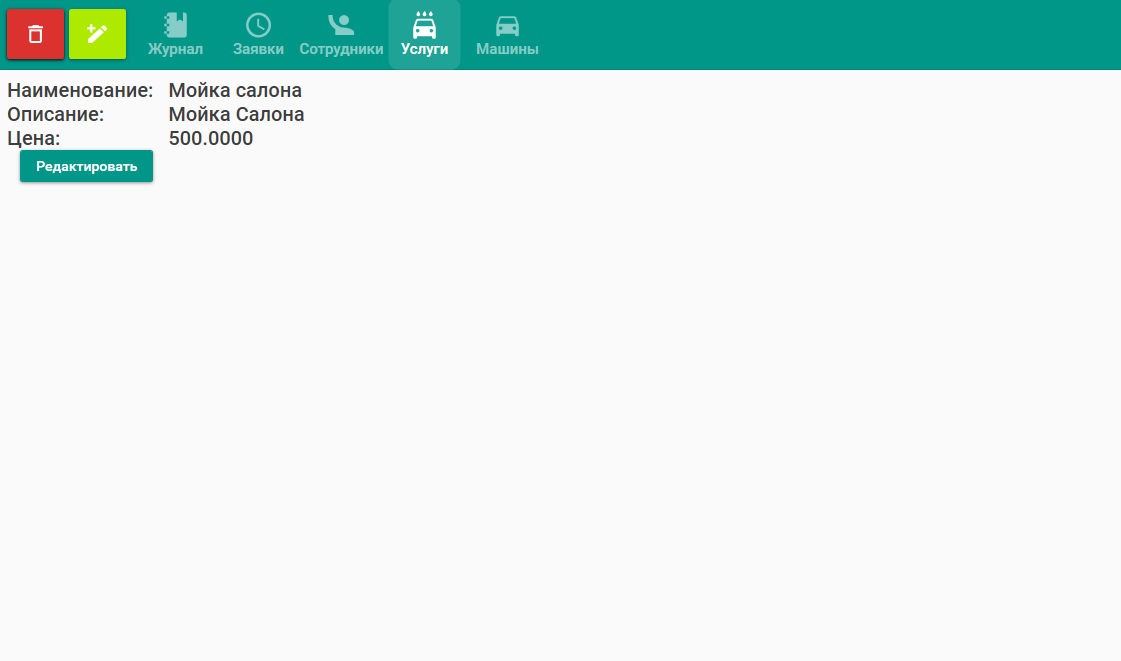


Рисунок 31 – «Окно со списками услуг»

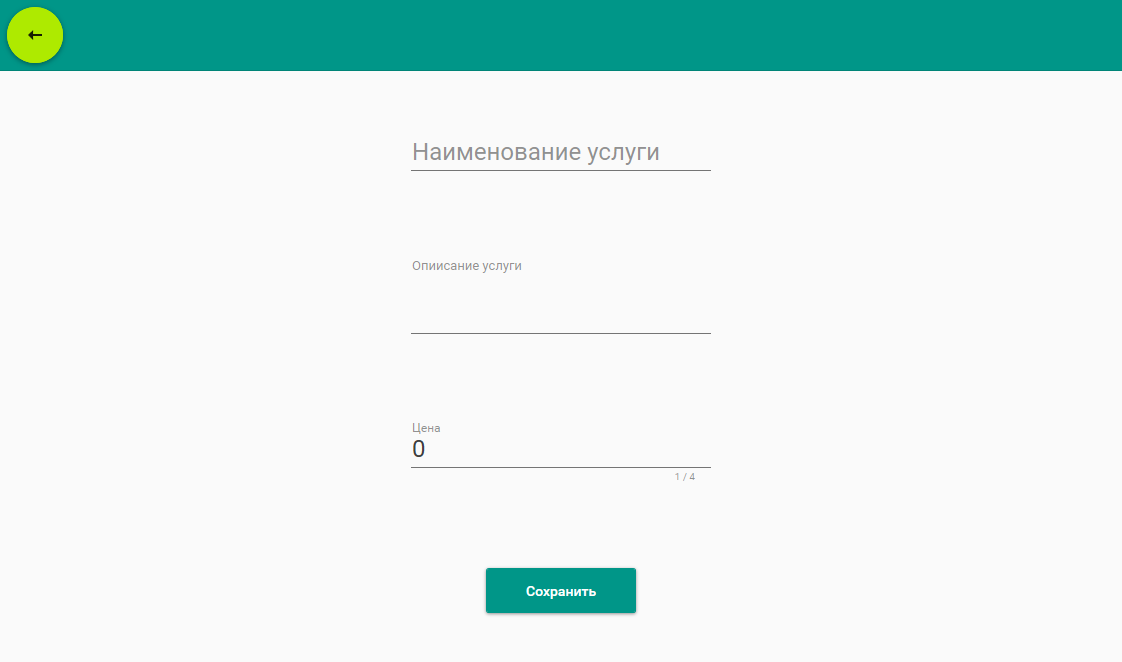


Рисунок 32 – «Окно добавления услуг»

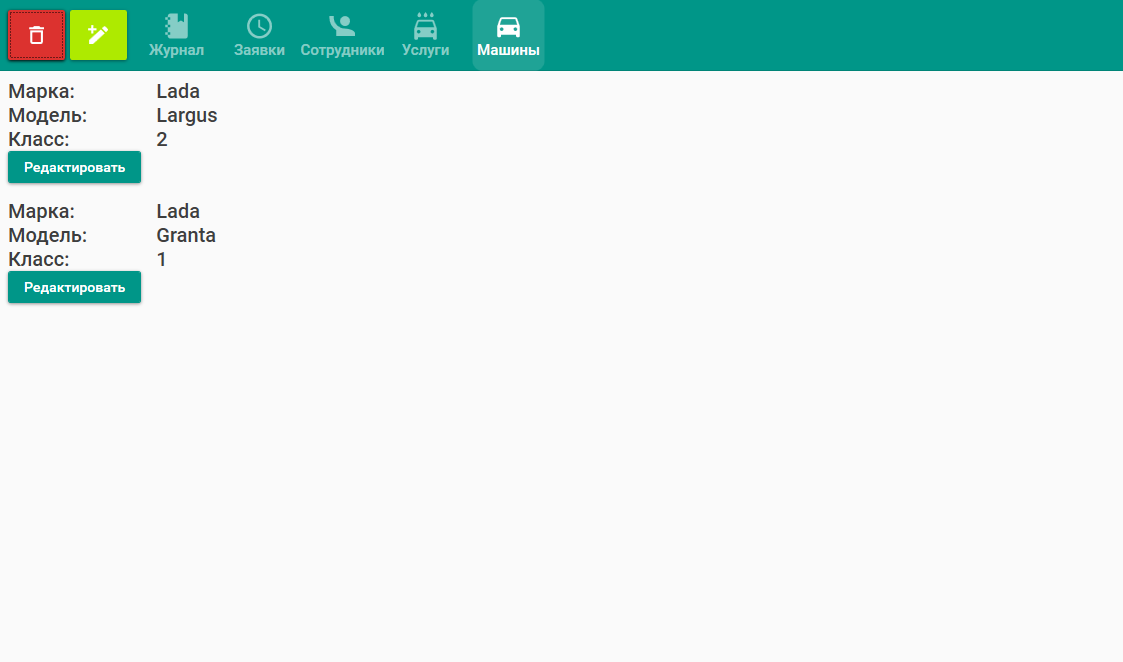


Рисунок 33 – «Окно со списками машин»

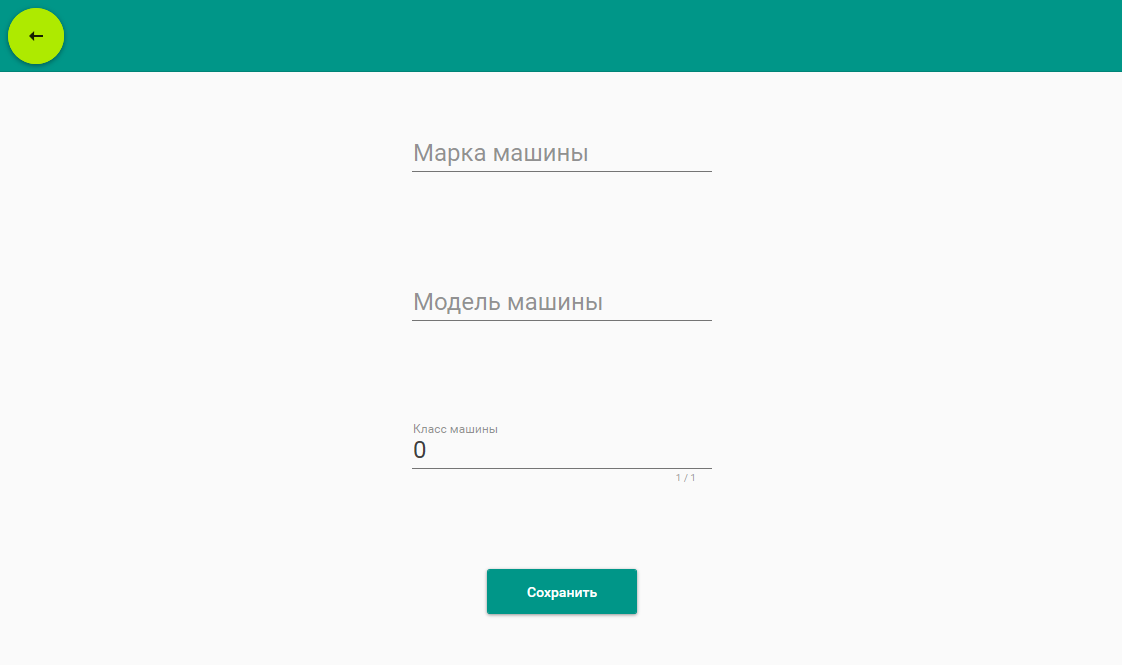


Рисунок 34 – «Окно добавления машин»

## **2.8 Концептуальное проектирование базы данных**

Диаграмма отношений сущностей (ERD) – это визуальное представление **различных сущностей внутри системы и их взаимосвязи друг с другом.** ERD – модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Основные шаги построения ER-диаграммы:

1. Цель и область применения: Определите цель и область применения для анализа и моделирования.
2. Сущности: Определите сущности, которые включены в эту предметную область. Когда будете готовы, начинайте рисовать их в прямоугольниках и подписывать существительными.
3. Связи: Определите, как все сущности связаны между собой. Нарисуйте линии между ними для обозначения связей и подпишите их.
4. Атрибуты: Следующий уровень – добавление ключевых атрибутов сущностей.
5. Кратность (мощность) связей: Покажите, где связь один-к-одному (1:1), где один-ко-многим (1:M), а где многие-ко-многим (M:M).

Сущность (entity) — класс реальных или виртуальных однотипных объектов, информацию о которых необходимо хранить в базе данных.

Связь (relationship) — ассоциация между сущностями. Для облегчения понимания диаграммы следует добавлять названия связей.

Атрибуты предназначены для описания сущности (свойства сущности).

Ключ — это один или несколько атрибутов, уникально определяющих сущность.

На рисунке 35 представлена ERD – диаграмма, описывающая все сущности и связи между ними для проекта разрабатываемой системы.

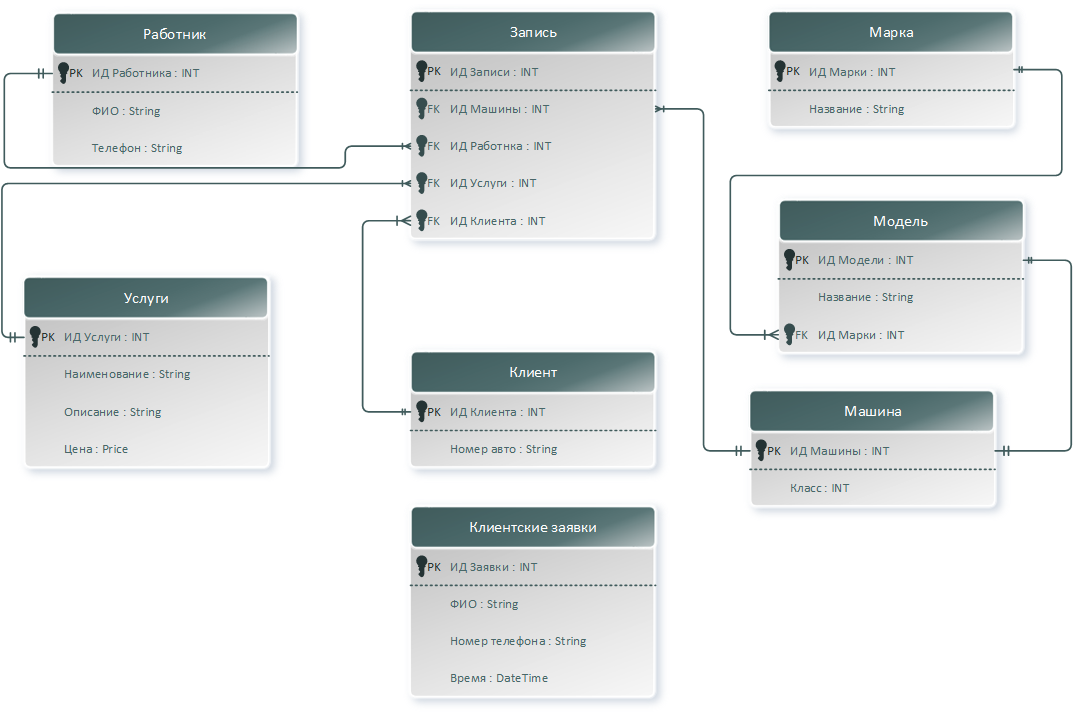


Рисунок 35 – Диаграмма сущность-связь

## **2.9** Логическое проектирование базы данных

Для ER-диаграммы, созданной в предыдущем параграфе, был разработан словарь данных – набор информации, описывающий, какой тип данных хранится в базе данных, их формат, структуру и способы использования данных. Структура словаря данных должна соответствовать разработанной концептуальной схеме. В документе отражены ограничения, определенные в таблицах, включая первичные ключи, отношения внешнего ключа с другими таблицами и ненулевые ограничения, необходимые пояснения и комментарии к неоднозначным полям.

Разработанный словарь данных представлен в таблице 6.

Таблица 6 – Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Словарь Данных | | | | |
| Работник | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Работников | INT | Y | Autoincrement |
|  | ФИО | NVARCHAR (100) | Y |  |
|  | Телефон | NVARCHAR (15) | Y |  |
| Услуги | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Услуги | INT | Y | Autoincrement |
|  | Наименование | NVARCHAR (50) | Y |  |
|  | Описание | NVARCHAR (1000) | Y |  |
|  | Цена | PRICE (20) | Y |  |
| Запись | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Записи | INT | Y | Autoincrement |
| FK | ИД Машины | INT | Y | Внешний ключ на Машины |
| FK | ИД Работника | INT | Y | Внешний ключ на Работники |
| FK | ИД Услуги | INT | Y | Внешний ключ на Услуги |
| FK | ИД Клиента | INT | Y | Внешний ключ на Клиенты |
| Клиент | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Клиента | INT | Y | Autoincrement |
|  | Номер Авто | NVARCHAR (5) | Y |  |
| Марка | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Марки | INT | Y | Autoincrement |
|  | Название | NVARCHAR (200) | Y |  |
| Модель | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Модели | INT | Y | Autoincrement |
|  | Название | NVARCHAR (200) | Y |  |
| FK | ИД Марки | INT | Y | Внешний ключ на Марка |
| Машины | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Машины | INT | Y | Autoincrement |
|  | Класс | INT | Y |  |
| Модель | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Модели | INT | Y | Autoincrement |
|  | Название | NVARCHAR (200) | Y |  |
| FK | ИД Марки | INT | Y | Внешний ключ на Марка |
| Клиентские заявки | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED? | NOTES |
| PK | ИД Заявки | INT | Y | Autoincrement |
|  | ФИО | STRING | Y |  |
|  | Номер телефона | STRING | Y |  |
|  | Время | DATETIME | Y |  |

# ГЛАВА 3. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

# Руководство пользователя

Настоящий документ является руководством пользователя по эксплуатации автоматизированной информационной системы «Разработка автоматизированной информационной системы для администрирования сети автомоек» (далее по тексту – АИС «Золотой теленок») для введения учета выручки. В данном руководстве приводится следующая информация:

− Введение.

− Назначение и условия применения.

− Подготовка к работе.

− Описание операций.

− Аварийные ситуации.

− Рекомендации по освоению.

Настоящий документ разработан в соответствии с РД 50-34.698-90 «Автоматизированные системы. Требования к содержанию документов».

Термины, определения используемые сокращения.

|  |  |
| --- | --- |
| **Обозначение** | **Описание** |
| АИС | Автоматизированная информационная система |
| БД | База данных |
| ПО | Программное обеспечение |
| ЯП | Языки программирования |
| СУБД | Система управления базами данных |
| SQL | Structured Query Language – Язык структурированных запросов |
| VS | Visual studio – среда разработки |

# Требования к программному обеспечению

Данный раздел содержит следующие подразделы:

* Серверная часть;
* Клиентская часть.

# Серверная часть

Таблица 7 – Требования к конфигурации программного обеспечения серверной части

|  |  |
| --- | --- |
| **Компонент** | **Конфигурация** |
| Операционная система | Microsoft Windows 7/8/8.1/10/11 |
| СУБД | MS SQL Server 18.9.2 |
| Общесистемное ПО | Microsoft .Net SDK 5.0.401 (64x) |

# Клиентская часть

Таблица 8 – Требования к конфигурации программного обеспечения клиентской части

|  |  |
| --- | --- |
| **Компонент** | **Конфигурация** |
| Операционная система | Microsoft Windows 7/8/8.1/10/11 |
| Общесистемное ПО | Microsoft .Net SDK 5.0.401 (64x) |
| Общесистемное ПО | Браузер |

# Требования к техническому обеспечению

Данный раздел содержит следующие подразделы:

* Серверная часть;
* Клиентская часть.

# Серверная часть

Аппаратное обеспечение сервера БД должно удовлетворять следующим техническим требованиям:

Таблица 9 – Требования к конфигурации аппаратного обеспечения серверной части

|  |  |
| --- | --- |
| **Компонент** | **Минимальная конфигурация** |
| Процессор | Intel Celeron G5920 или AMD Athlon X4 |
| Оперативная память | 4 Гб SDRAM |
| Жесткий диск | 160 Гб SATA |
| Видеоадаптер | Встроен в системную плату |
| Сетевая плата | Ethernet 100 Мб |
| Дополнительное оборудование | Монитор 800\*600 точек |

# Клиентская часть

Для работы с АИС «Золотой теленок» рабочие станции пользователей должны удовлетворять следующим минимальным требованиям:

* Просмотр и введения учета выполненных услуг
* Просмотр и введения кадрового учета
* Просмотр и введения учета предоставляемых услуг
* Просмотр и введения списка машин

Таблица 10 – Требования к конфигурации аппаратного обеспечения клиентской части

|  |  |
| --- | --- |
| **Компонент** | **Минимальная конфигурация** |
| Процессор | Intel Celeron G5920 или AMD Athlon X4 |
| Оперативная память | 4 Гб SDRAM |
| Жесткий диск | 1 Гб |
| Видеоадаптер | Встроен в системную плату |
| Сетевая плата | Ethernet 100 Мб |
| Дополнительное оборудование | Монитор 800\*600 точек |

Подготовка к работе.

Перед началом работы с АИС «Золотой теленок» необходимо установить соответствующее программное обеспечение.

# Знакомство с АИС «Золотой теленок»

Перед тем как приступить к работе с АИС «Золотой теленок», необходимо ознакомиться со следующей информацией:

* Страница журнала;
* Страница клиентских заявок;
* Страница с сотрудниками
* Страница с услугами;
* Страница с машинами
* Страница добавления

# Страница журнала

При запуске приложения начальной страницей открывается журнал.

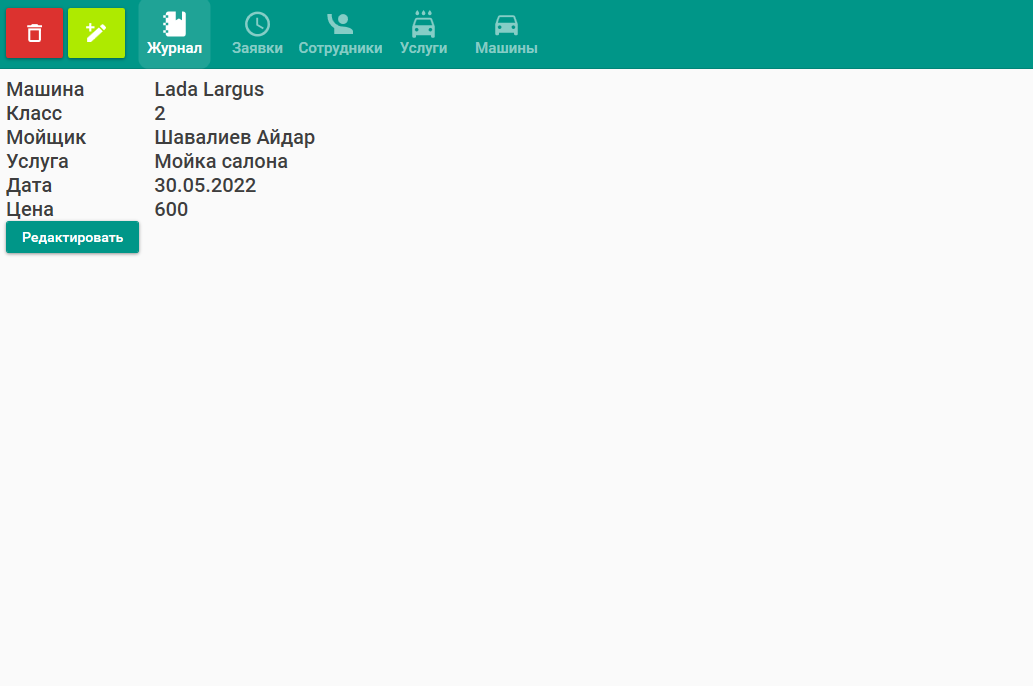


Рисунок 36 – «Страница журнала»

Описание операций

Данный раздел содержит описание следующих подразделов:

* Работа с журналом;
* Работа с заявками;
* Работа с сотрудниками;
* Работа с услугами;
* Работа с машинами;

# Работа с журналом

Данный подраздел содержит описание следующих пунктов:

* Просмотр журнала;
* Добавление записи в журнал;
* Удаление записи с журнала;
* Редактирование записи журнала;

# Просмотр журнала

Страница журнала отображается при запуске приложения. В журнале выводится информация о том для какой машины какой сотрудник выполнил ту или иную услугу, и информация в какое время и на какую сумму была выполнена услуга.

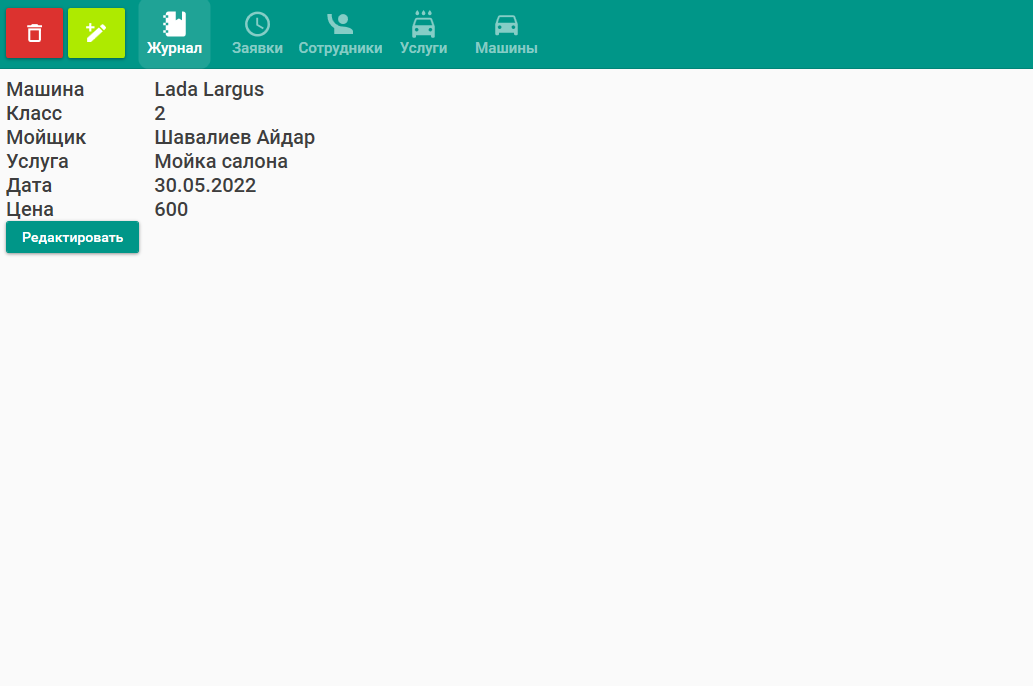


Рисунок 37 – «Страница журнала»

# Добавление записи в журнал

Для перехода к добавлению записи необходимо нажать на кнопку «Добавить» на странице журнала. Далее откроется страница добавления. Чтобы добавить новую запись необходимо выбрать все необходимы значения в полях и нажать на кнопку сохранить.

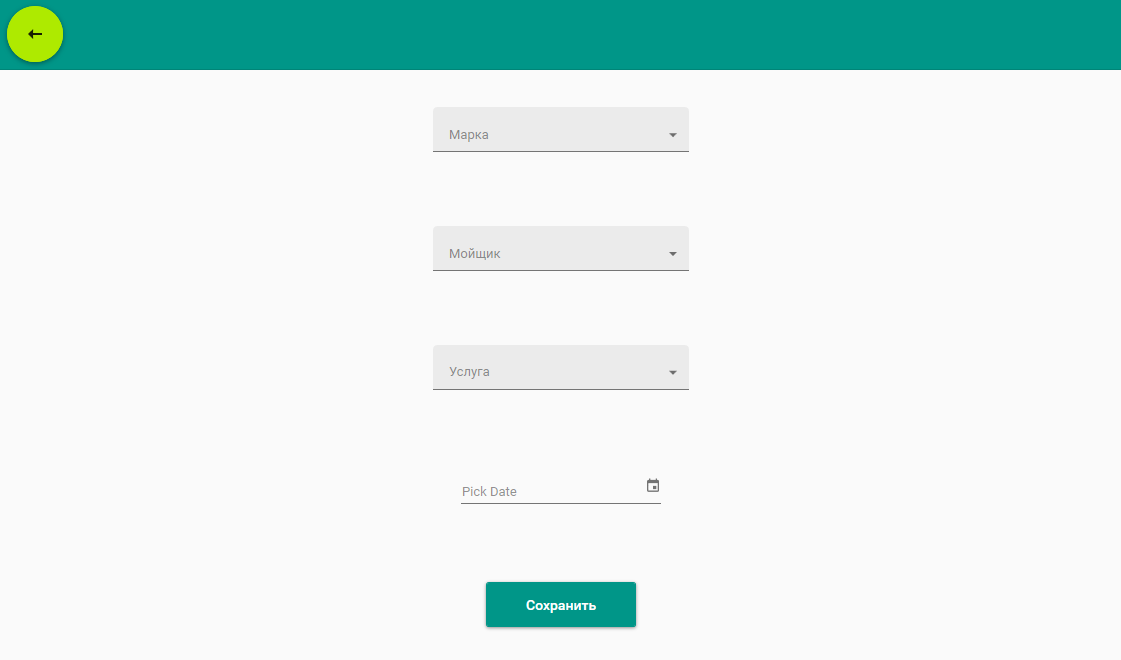


Рисунок 38 – «Окно добавления записи в журнал»

# Удаление записи с журнала

Для удаления записи необходимо выполнить несколько действий:

* Нажать на нужную запись ;
* Нажать на кнопку «Удалить»;

# Редактирование записи журнала

Для перехода к редактированию записи необходимо нажать на кнопку «Редактировать» рядом с нужной записью на странице журнала. Далее откроется страница редактирования. Чтобы редактировать запись нужно в тех полях где необходимо обновить данные выбрать новые значения и нажать сохранить.

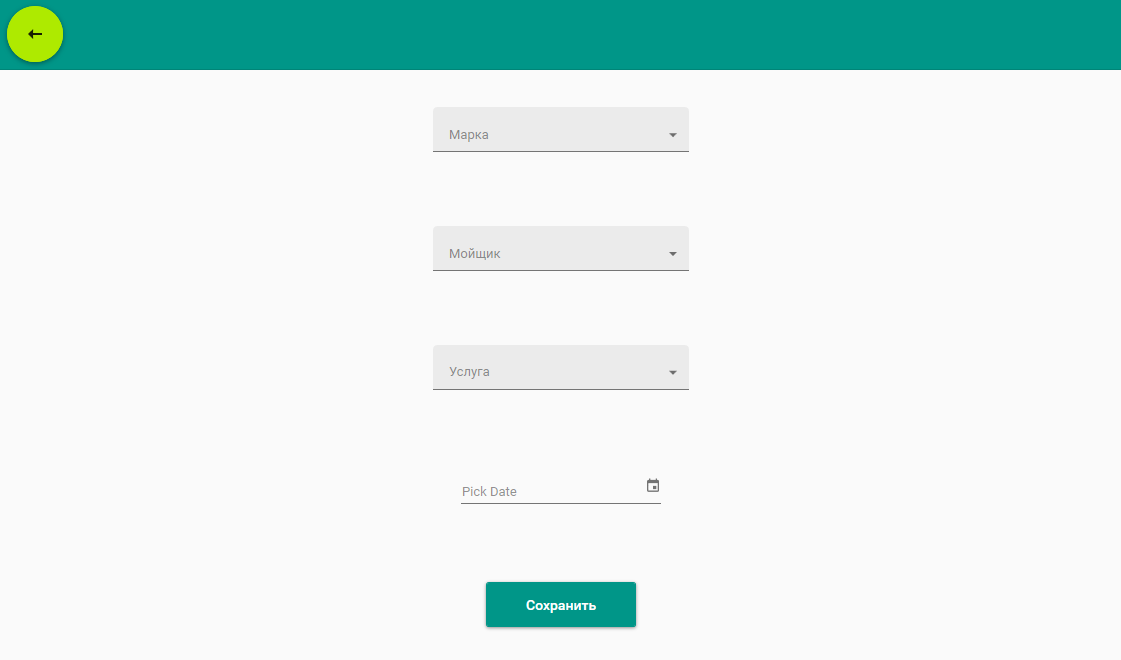


Рисунок 39 – «Окно редактирования записи журнала»

# Работа с сотрудниками

Данный подраздел содержит описание следующих пунктов:

* Просмотр сотрудников;
* Добавление нового сотрудника в список;
* Удаление сотрудника из списка;
* Редактирование данных о сотруднике;

# Просмотр сотрудников

При нажатии на кнопку «Сотрудники» отобразится страницы со списком всех сотрудников работающих в автомойке. У каждого сотрудника выводится фамилия, имя, отчество и номер телефона.

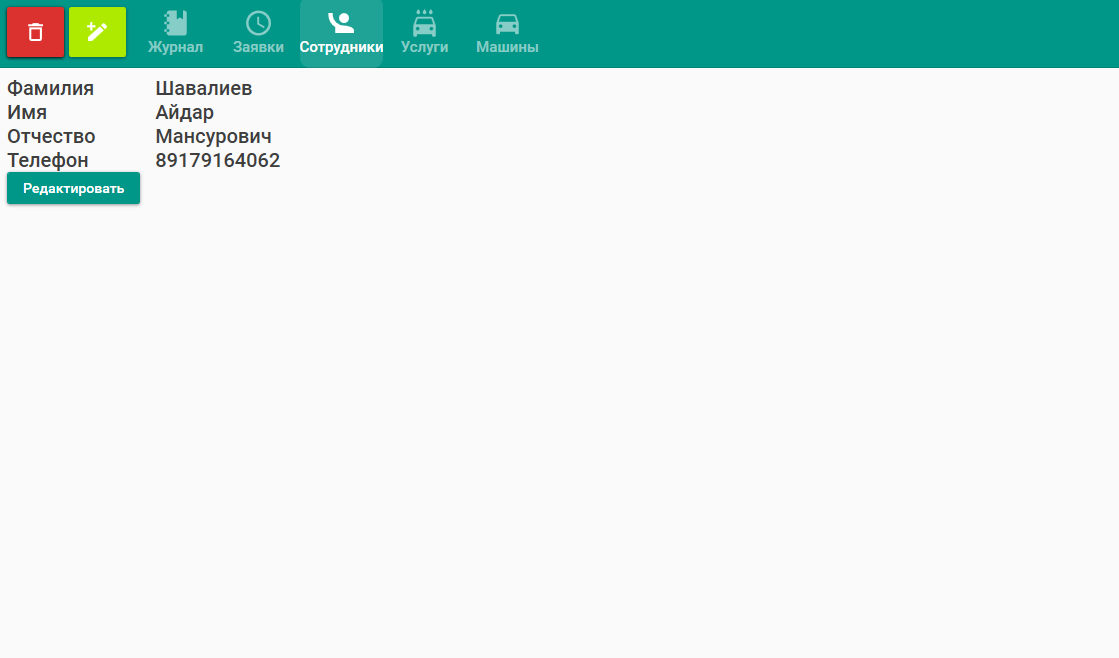


Рисунок 40 – «Окно со списками сотрудников»

# Добавление нового сотрудника в журнал

Для перехода к добавлению нового сотрудника необходимо нажать на кнопку «Добавить» на странице сотрудников. Далее откроется страница добавления. Чтобы добавить сотрудника необходимо ввести данные в полях и нажать на кнопку сохранить.

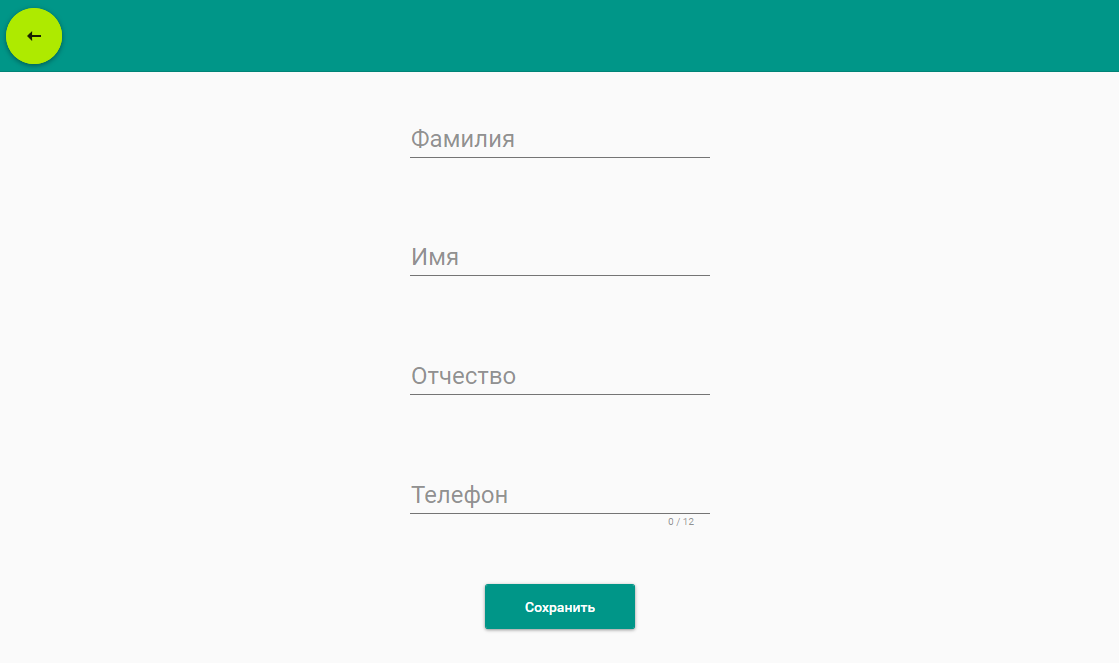


Рисунок 41 – «Окно добавления сотрудника»

# Удаление сотрудника из списка

Для удаления сотрудника необходимо выполнить несколько действий:

* Нажать на нужного сотрудника;
* Нажать на кнопку «Удалить»;

# Редактирование данных о сотруднике;

Для перехода к редактированию данных сотрудника необходимо нажать на кнопку «Редактировать» рядом с нужным сотрудником на странице. Далее откроется страница редактирования. Чтобы редактировать нужно в тех полях где необходимо обновить данные прописать новые значения и нажать сохранить.

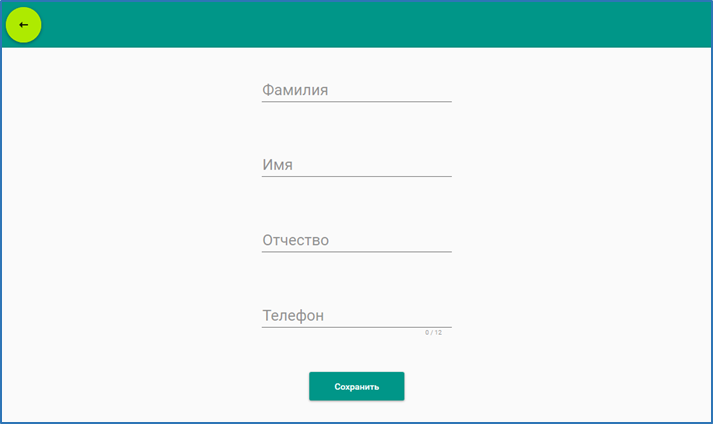


Рисунок 42 – «Окно редактирования данных о сотруднике»

# Работа с услугами

Данный подраздел содержит описание следующих пунктов:

* Просмотр услуг;
* Добавление новой услуги в список;
* Удаление услуги из списка;
* Редактирование информации об услугах;

# Просмотр услуг

При нажатии на кнопку «Услуги» отобразится страницы со списком всех услуг которые предоставляет данная автомойка. Выводится наименование услуги, подробное ее описание и цена относящийся к первому классу машин.

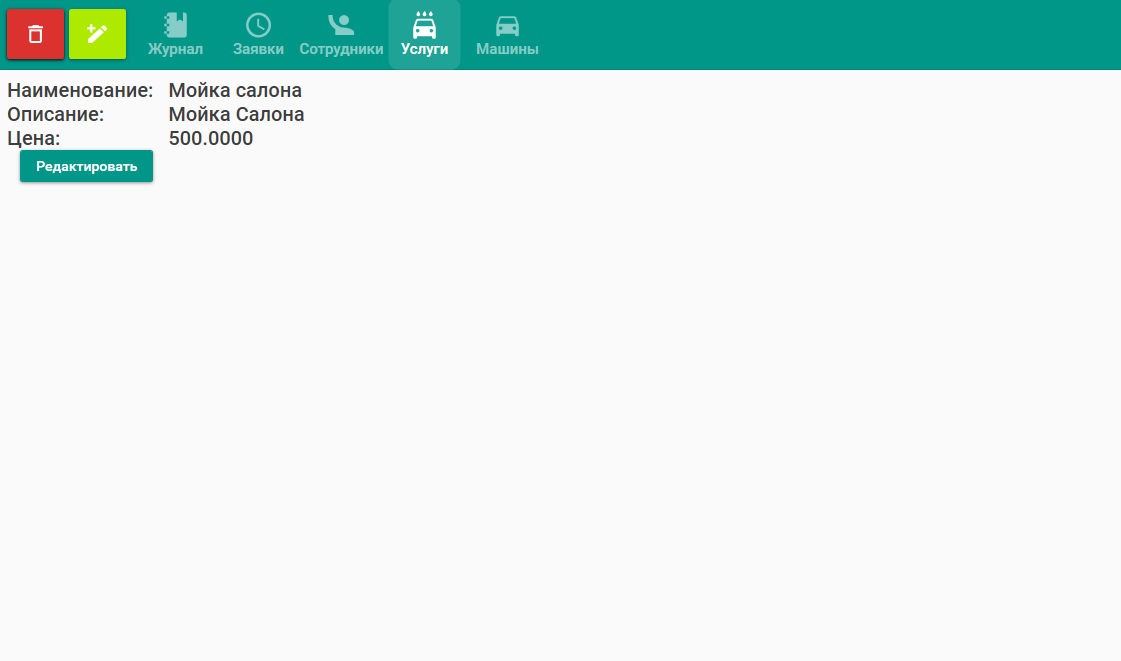


Рисунок 43 – «Окно со списками сотрудников»

# Добавление услуги

Для перехода к добавлению новой услуги, необходимо нажать на кнопку «Добавить» на странице услуг. Далее откроется страница добавления. Чтобы добавить услугу, необходимо ввести данные в полях и нажать на кнопку сохранить.

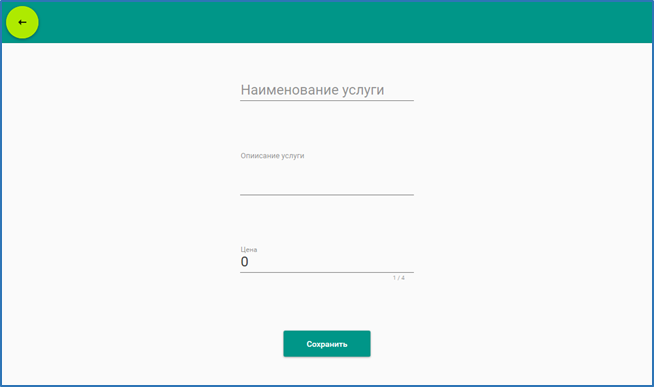


Рисунок 44 – «Окно добавления услуги»

# Удаление услуги из списка

Для удаления услуги необходимо выполнить несколько действий:

* Нажать на нужную услугу;
* Нажать на кнопку «Удалить»;

# Редактирование услуги

Для перехода к редактированию услуг необходимо нажать на кнопку «Редактировать» рядом с нужной услугой на странице. Далее откроется страница редактирования. Чтобы редактировать, нужно в тех полях где необходимо обновить данные прописать новые значения и нажать сохранить.

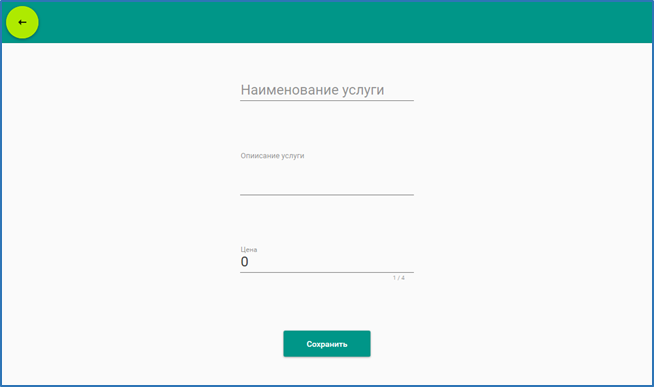


Рисунок 45 – «Окно редактирование услуги»

# Работа с машинами

Данный подраздел содержит описание следующих пунктов:

* Просмотр списка машин;
* Добавление новой машины в список;
* Удаление машин из списка;
* Редактирование информации о машинах;

# Просмотр списка машин

При нажатии на кнопку «Машины» отобразится страница со списком всех машин, которым предоставляются услуги данной автомойки. Выводится марка машины, ее модель и класс.

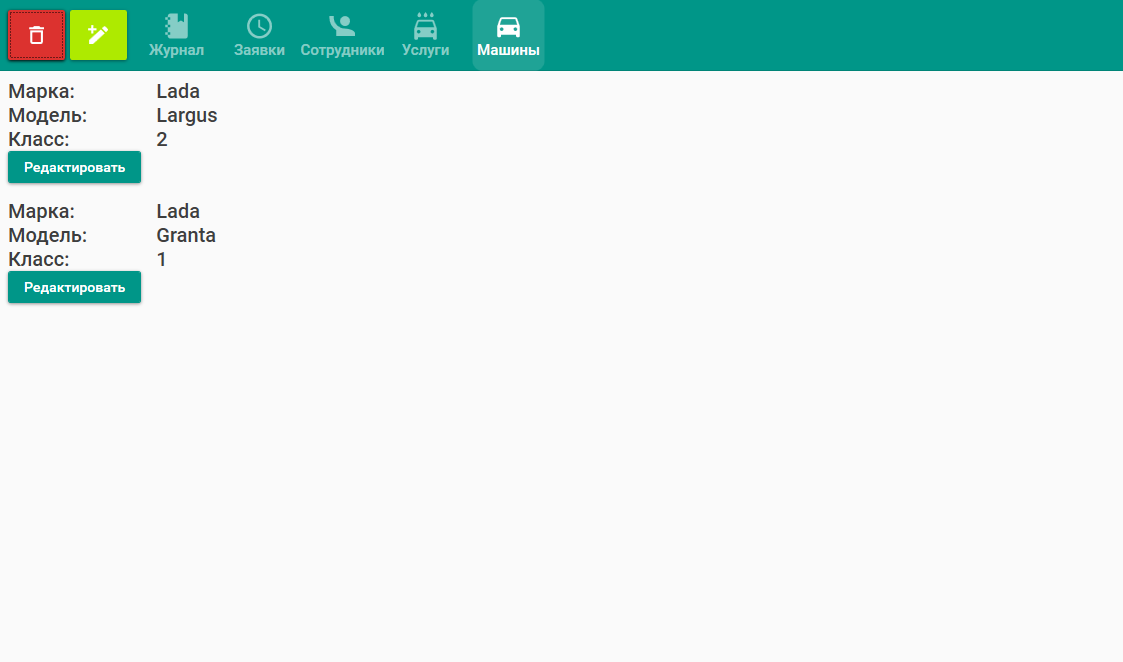


Рисунок 46 – «Окно со списками машин»

# Добавление новой машины

Для перехода к добавлению новой машины, необходимо нажать на кнопку «Добавить» на странице машин. Далее откроется страница добавления. Чтобы добавить машину, необходимо ввести данные в полях и нажать на кнопку сохранить.

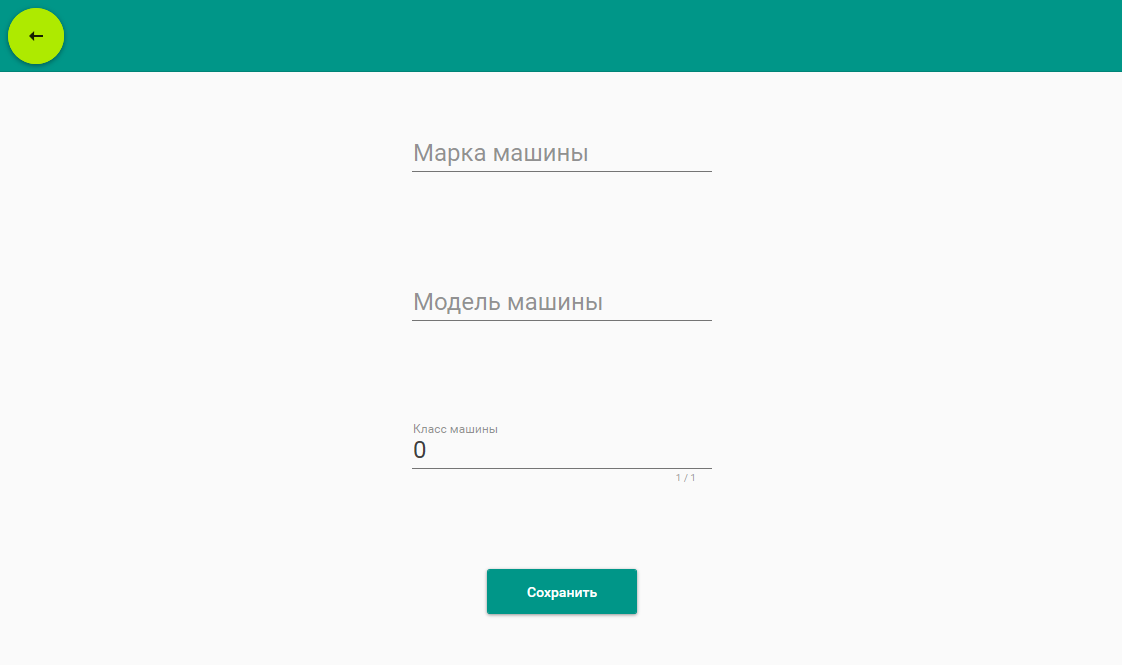


Рисунок 47 – «Окно добавления машин»

# Удаление машины из списка

Для удаления машины необходимо выполнить несколько действий:

* Нажать на нужную машину;
* Нажать на кнопку «Удалить»;

# Редактирование информации о машинах

Для перехода к редактированию машин необходимо нажать на кнопку «Редактировать» рядом с нужной машиной на странице. Далее откроется страница редактирования. Чтобы редактировать, нужно в тех полях где необходимо обновить данные прописать новые значения и нажать сохранить.

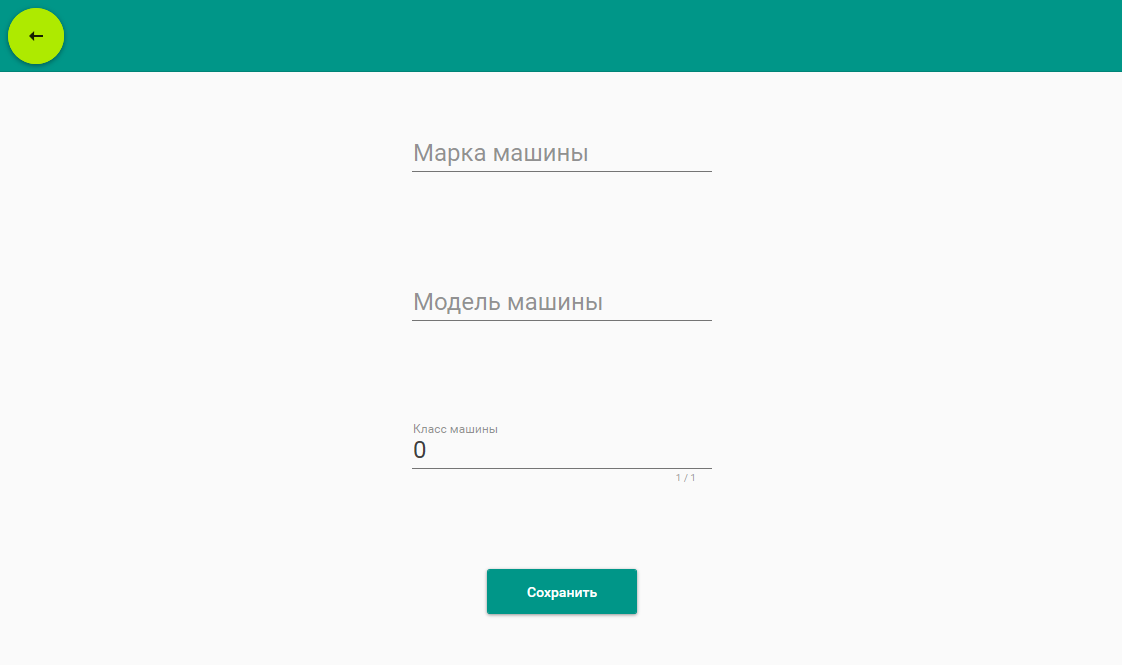


Рисунок 48 – «Окно редактирования машин»

Аварийные ситуации

АИС «Золотой теленок» является сложным программным модулем, работа с которым может привести к следующим аварийным ситуациям:

* Не заполнены поля для заполнения – для устранения данной ошибки необходимо заполнить указанные поля.
* Системные сообщения – для устранения данной ошибки необходимо обратиться к разработчику Подсистемы.

# Рекомендации по освоению

Для успешной работы с АИС «Золотой теленок» необходимо:

* получить навыки работы с операционной системой Windows;
* ознакомиться с данным руководством пользователя.
  1. **Руководство программиста**

Данная программа подключена к MSSMS с помощью .NET Framework, для получения контекста из базы данных используется класс ZTDBEntities.cs

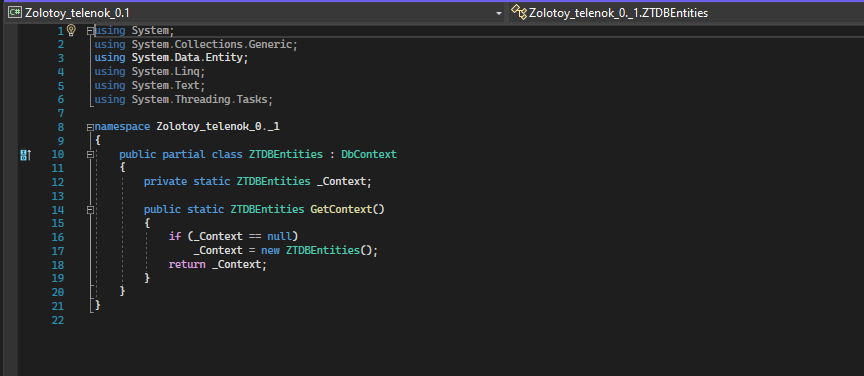


Рисунок Х 49 «Класс ZTDBEntities.cs»

* 1. **Руководство по установке программного обеспечения**

Чтобы установить приложение на рабочий компьютер необходимо запустить инсталлятор «Золотой теленок»

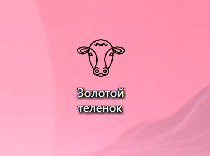


Рисунок 50 – «Инсталлятор»

Далее откроется окно выбора папки установки фалов приложения

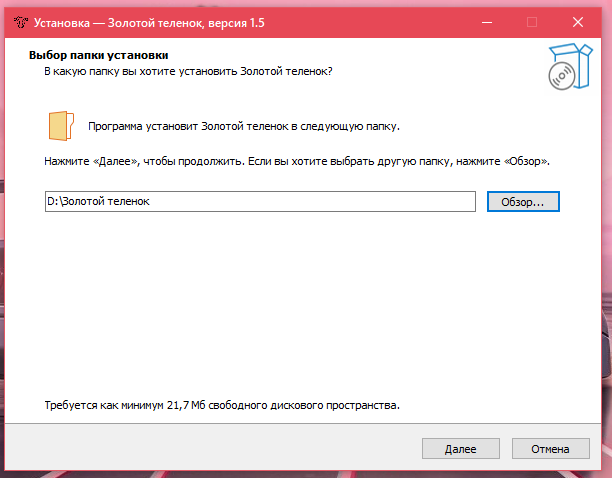


Рисунок 51 – «Окно выбор папки установки»

Далее откроется окно выбора дополнительных задач где мы можем выбрать дополнительные задачи, который выполняются при установке.

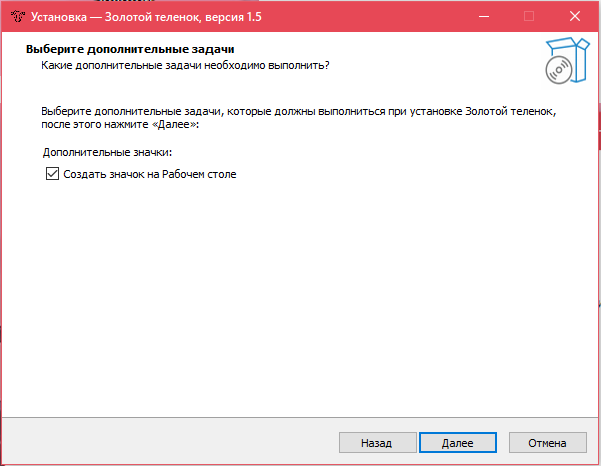


Рисунок 52 – «окно дополнительных задач»

Далее необходимо нажать на кнопку установить и приложение установится на рабочий компьютер.

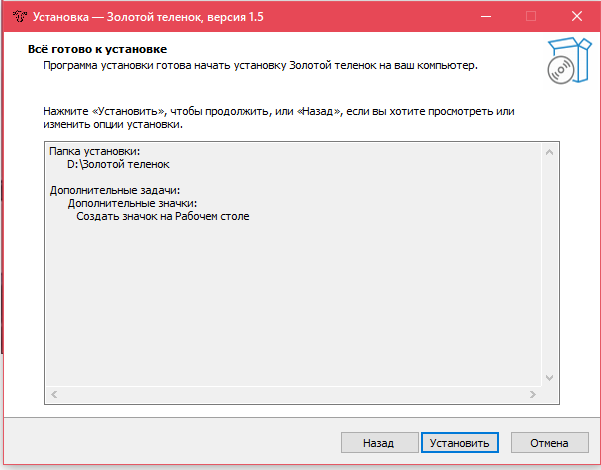


Рисунок 53 – «Окно уведомления о готовности к установке»

После установки нам выводится сообщение о завершении установки, приложение готова к запуску.

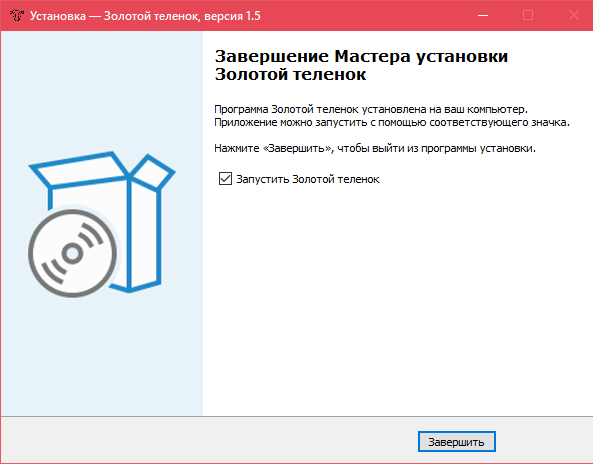


Рисунок 54 – «Окно сообщения о завершении установки»

# ЗАКЛЮЧЕНИЕ

В результате изучения и анализа предметной области разрабатываемой программной системы, а именно области применения и сферы работы . Переход от бумажной документации к электронному обороту снизит риск потери.

В рамках разработки дипломного проекта были достигнуты нижеперечисленные результаты:

* изучен теоретический материал по разработке веб-сервисов, веб-дизайна и WPF приложения
* проведен анализ существующих аналогов;
* проведен анализ, ознакомление и выбор средств разработки веб-сервисов и десктопного приложения;
* спроектирована архитектура разработанной программной системы учета выполненной работы автомойки: схемы работы веб-сервиса, различные UML-диаграммы;
* спроектирован и разработан удобный и понятный интерфейс программной системы;
* разработана база данных для хранения всех необходимых данных о записях журнала, клиентах, сотрудниках, услугах и машинах.
* разработаны модули менеджера программной системы учета выполненной работы автомойки, которые включают в себя следующие функциональные возможности: введения журнала, введения кадрового состава, управления списками заявок клиентов, списками услуг и машин.
* разработаны модули клиента программной системы учета выполненной работы автомойки, которые включают в себя следующие функциональные возможности: возможности оставить клиента заявку на услугу в определенное время, а также в случае возникновения проблем обратится в службу технической поддержки.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Робсон Э., Фримен Э. Изучаем HTML, XHTML и CSS. Изд. Питер, 2019 – 720 с.

2. Макфарланд Д. Новая большая книга CSS. Изд. Питер, 2019 – 719 с.

3. Макфарланд Д. JavaScript и jQuery. Исчерпывающее руководство. Изд. Эксмо, 2016 – 880 с.

4. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. Изд. Питер, 2019 – 816 с.

5. HtmlBook – блог-справочник по HTML5 [Электронный ресурс]: самоучитель HTML, справочник HTML, HTML5. URL: http://htmlbook.ru/html

6. Metanit – блог, содержащий информацию обо всем, что связано с разработкой веб-сайтов при помощи ASP.NET [Электронный ресурс]: URL: https://metanit.com/sharp/

7. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. Изд. Эксмо, 2020 – 480 с.

8. Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. Изд. Питер, 2017 – 272 с.

9. Крокфорд Д. Как устроен JavaScript. Изд. Питер, 2019 – 304 с.

10. Браун Э. Изучаем JavaScript. Руководство по созданию современных веб-сайтов. Изд. Вильямс, 2017 – 363 с.

11. Дакетт Д. Javascript и jQuery. Интерактивная веб-разработка. Изд. Эксмо, 2017 – 640 с.

12. Хавербеке М. Выразительный JavaScript. Современное веб-программирование. Изд. Питер, 2019 – 480 с.

13. Д. Флэнаган. JavaScript. Карманный справочник. Изд. Вильямс, 2019 – 320 с.

14. Learn JavaScript – блог, содержащий информацию обо всех основах языка программирования JavaScript [Электронный ресурс]: современный учебник по изучению языка программирования JavaScript. URL: https://learn.javascript.ru/

15. LearnJS – блок, содержащий информацию обо всех принципах построения кода на языке программирования JavaScript. [Электронный ресурс]: учебник, самоучитель по JS. URL: https://learnjs.ru/

16. Д. Пэкетт, Д. Чамберс, С. Тиммс. ASP.NET Core. Разработка приложений. Изд. Питер, 2017 – 464 с.

17. ASP.NET – блог, описывающий работу с ASP.NET [Электронный ресурс]: Веб-фреймворк с открытым исходным кодом для .NET. URL. https://dotnet.microsoft.com/apps/aspnet

18. Васильев А.Н. Программирование для начинающих на C#. Основные сведения. Изд. Эксмо, 2019 – 592 с.

19. Мартишин С., Симонов В., Храпченко М. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench. Изд. Форум, 2019 – 160 с.

20. Заяц А., Васильев Н. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript. Изд. Лань, 2021 – 120 с.

# ПРИЛОЖЕНИЕ