

**МИНИСТЕРСТВО ЭНЕРГЕТИКИ, ПРОМЫШЛЕННОСТИ И СВЯЗИ
СТАВРОПОЛЬСКОГО КРАЯ**

Государственное бюджетное профессиональное образовательное учреждение
«Ставропольский колледж связи имени Героя Советского Союза В.А.
Петрова» (ГБПОУ СКС)

Цикловая комиссия вычислительной техники

КУРСОВАЯ РАБОТА

**по ПМ.02 «Осуществление интеграции программных модулей»
МДК 02.01 «Технология разработки программного обеспечения»**

Тема: Разработка информационной системы для технического осмотра
автомобилей

Выполнил Евсеенко Илья Николаевич
(Ф. И. О.,
_____ 4 курс ИП193 _____
курс, № группа)

Руководитель
Лукьянова Светлана Павловна

Ставрополь, 2023

**«Ставропольский колледж связи
имени Героя Советского Союза В.А. Петрова»**

УТВЕРЖДАЮ

Председатель цикловой комиссии

_____/И.В. Ерёмина

« ____ » _____ 2023 г.

**ЗАДАНИЕ
для курсовой работы (проекта)**

Студенту Евсеевко Илья Николаевич, 4 курс, ИПП193

Руководитель Лукьянова Светлана Павловна

Тема _____ Разработка информационной системы для технического осмотра автомобилей _____

№ п/п	Разделы, подразделы и их содержание	Сроки выполнения
1	Введение	00.00.2023
2	Теоретические основы разработки программного обеспечения	00.00.2023
3	Техническое задание на разработку	00.00.2023
4	Функциональная модель информационной системы	00.00.2023
5	Объектно-ориентированное проектирование системы	00.00.2023
6	Создание информационной системы	00.00.2023
7	Тестирование программного продукта	00.00.2023
8	Техническая документация	00.00.2023
9	Заключение	00.00.2023
10	Список использованных источников	00.00.2023
11	Приложения	00.00.2023

Преподаватель-руководитель курсовой работы (проекта)

_____/С.П. Лукьянова/

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	4
2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ	5
3 ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ ИНФОРМАЦИОННОЙ СИСТЕМЫ	6
4 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ СИСТЕМЫ	9
4.1 Модели вариантов использования системы	9
4.2 Диаграмма классов.....	9
4.3 Диаграммы деятельности	10
4.4 Диаграмма последовательности	12
4.5 Диаграмма кооперации.....	12
4.6 Диаграмма состояний	13
5 СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	15
5.1 Разработка интерфейса программного продукта.....	15
5.2 Разработка программного кода системы	17
5.3 Справочная система.....	18
6 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА	19
7 ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ	21
Заключение	22
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	24
ПРИЛОЖЕНИЕ А.....	26
ПРИЛОЖЕНИЕ Б	37
Приложение В	40

ВВЕДЕНИЕ

В современном мире информационные системы используются повсеместно. Данные системы позволяют автоматизировать процессы производства, развивают привлекательность продукта для бизнеса. В работе часто используются базы данных, которые позволяют работать с большими объемами данных.

Основной фактор – эффективность работы с данными, и информационные системы позволяют решать задачи предприятия, обрабатывать потоки данных. У любой информационной системы есть интерфейс взаимодействия, архитектура баз данных и вывод информации.

В рамках курсового проектирования ведется разработки программы для технического осмотра автомобилей.

Область применения программы – оценка и учет состояния автомобиля. Программа применяется для ведения системы контроля состояния автомобилей с возможностью поиска автомобиля по госномеру и возможностью изменения данных.

Основные требования к проекту – удобство использования на ноутбуках для выездного осмотра или осмотра на месте, данные, введенные пользователем должны автоматически сохраняться в базу данных, программа должна позволять пользователю создавать новые записи в базу данных, их изменение и удаление ненужных записей, а также вывод информации списком, отображении данных конкретного автомобиля и поиск записей по госномеру.

Актуальность разработки заключается в том, что программа для учета автомобиля будет удобна для составления базы данных клиентов станций технического обслуживания для учета и оценки проведения работ и удобной делегации работ между сотрудниками.

Цель курсового проекта: используя современные средства проектирования и разработки реализовать все этапы жизненного цикла программных продуктов.

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для того, чтобы создать информационную систему, надо понять, как происходит работа станции технического обслуживания или автосервиса.

Сотрудник СТО при осмотре автомобиля обращается к информационной системе и вводит госномер и данные о состоянии автомобиля. Таким образом производится диагностика автомобиля.

Система работает с шаблонными данными, которые представляют ввод данных о каждой детали автомобиля. Таким образом сотруднику СТО не надо помнить, в какой последовательности надо проводить осмотр автомобиля, а можно действовать по алгоритму. Система оптимизирует шаги осмотра для каждого из типа автомобиля, будь то легковой автомобиль, или грузовой. Система так же позволит удобно просматривать данные о каждом автомобиле любому сотруднику СТО.

В системе будет встроен поиск по госномеру для поиска автомобиля в базе данных. Так же система будет оснащена модулем для составления отчета и накладных, что позволит составлять оценочную стоимость ремонта.

Основными пользователями программы будут клиенты программы и администратор баз данных.

2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ

Техническое задание (ТЗ) — это документ, определяющий объем, цели, результаты и требования проекта. Его цель - установить общее понимание между заказчиком и подрядчиком и гарантировать, что проект будет завершен вовремя, в рамках бюджета и к удовлетворению заказчика.

В процессе проектирования информационной системы документооборота в юридической фирме было создано и утверждено техническое задание, обеспечивающее четкое понимание целей и требований проекта. Документ ТЗ включает подробное описание масштаба проекта, функциональности, требований к производительности, сроков и бюджета. Это важный документ, который служит основой для планирования проекта, мониторинга и контроля. Использование ТЗ гарантирует, что все заинтересованные стороны имеют общее понимание проекта, что имеет решающее значение для успеха проекта.

Техническое задание на разработку программного продукта «Информационная система для технического осмотра автомобилей» приведено в приложении А.

3 ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Основная деятельность автосервиса заключается в диагностике и ремонте легковых автомобилей. Таким образом, контекстная диаграмма данной информационной системы будет содержать только единственную работу «Диагностика автомобиля».

В соответствии с методом IDEF0 для любой работы необходимо определить входные данные, выходные данные, управление и механизм, которые изображаются на диаграмме стрелками:

- Входные данные: госномер автомобиля и техническое состояние автомобиля
- Выходные данные: отчеты в виде накладных о диагностике автомобиля и предварительные расчеты о стоимости ремонта.
- Управление: нормативные документы.
- Механизм: сотрудники (менеджеры, мастера и работники по ремонту) и заказчик.

Контекстная диаграмма (рисунок 1) имеет уровень A0. Это самый высокий уровень абстракции для данной задачи, выражающий точку зрения любого внешнего субъекта на деятельность редакции.

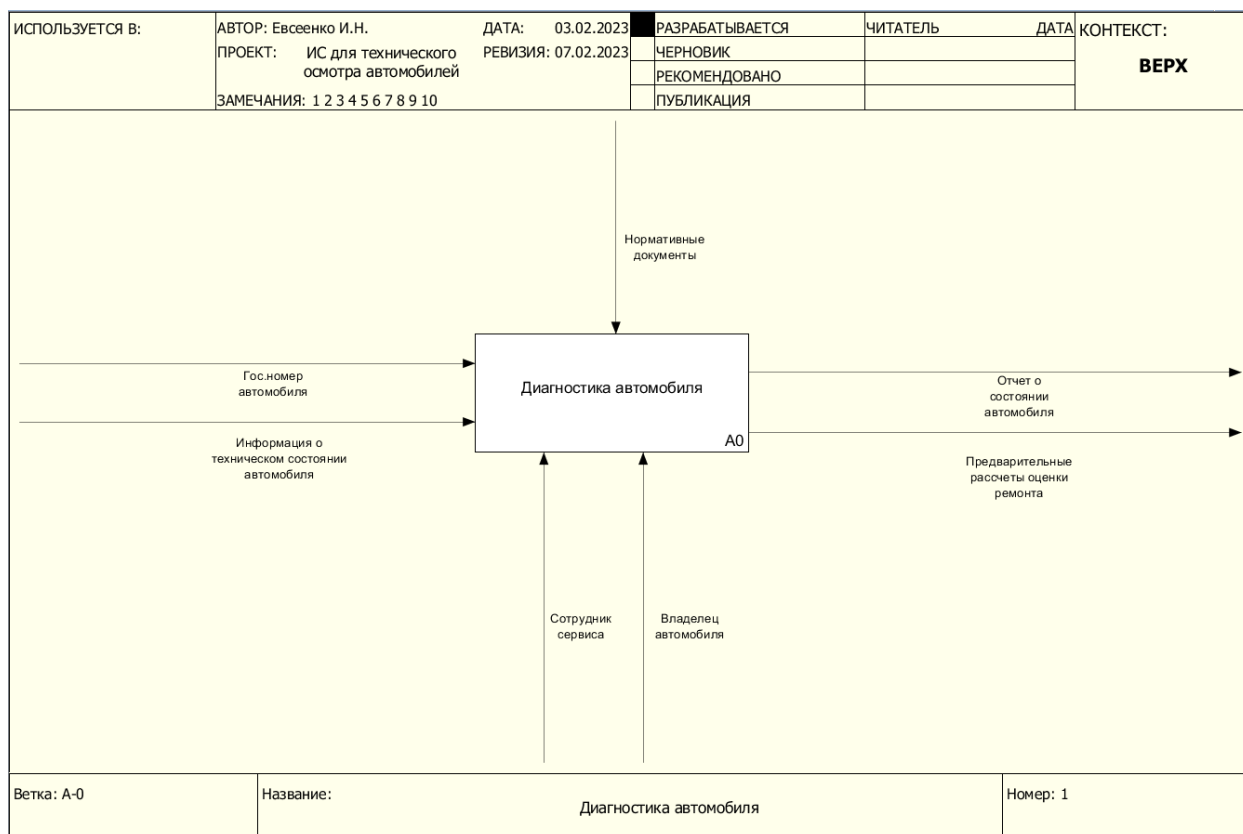


Рисунок 1 – Контекстная диаграмма информационной системы

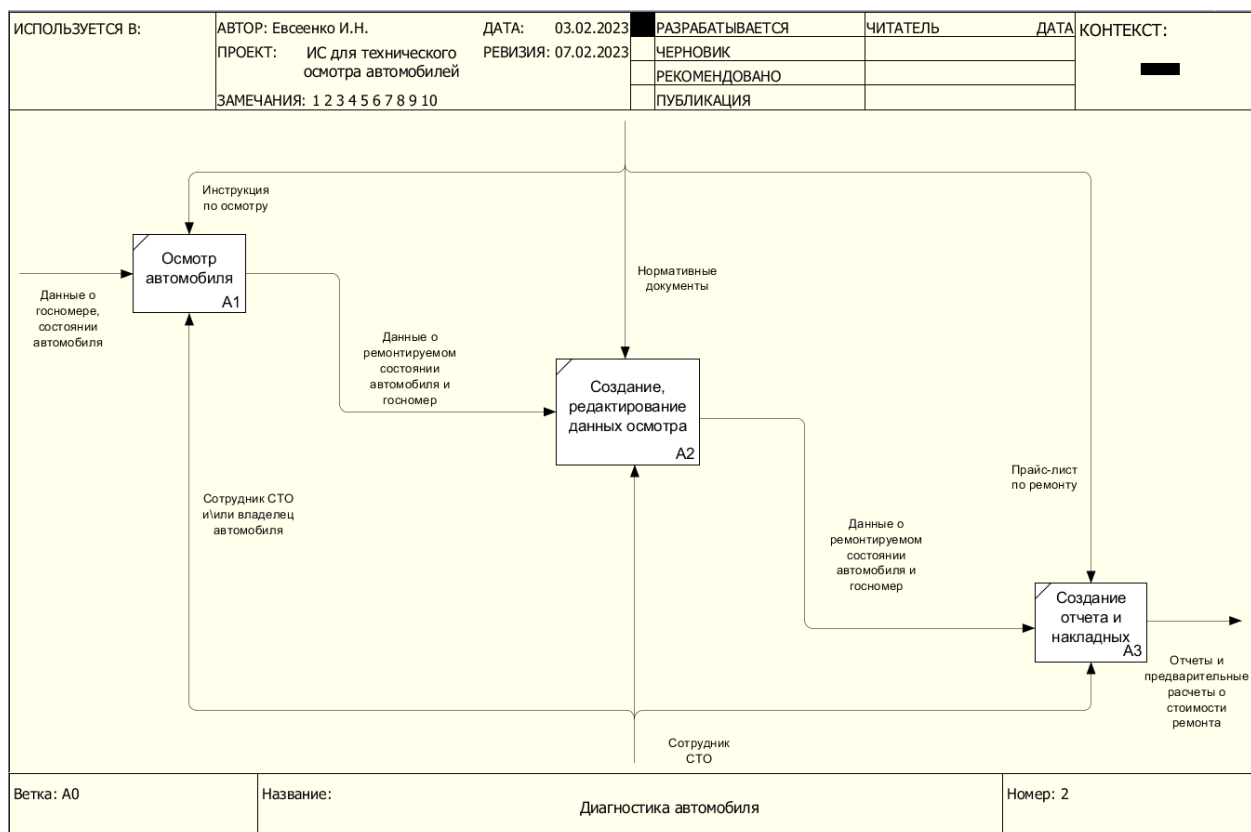


Рисунок 2 – Функциональная диаграмма первого уровня

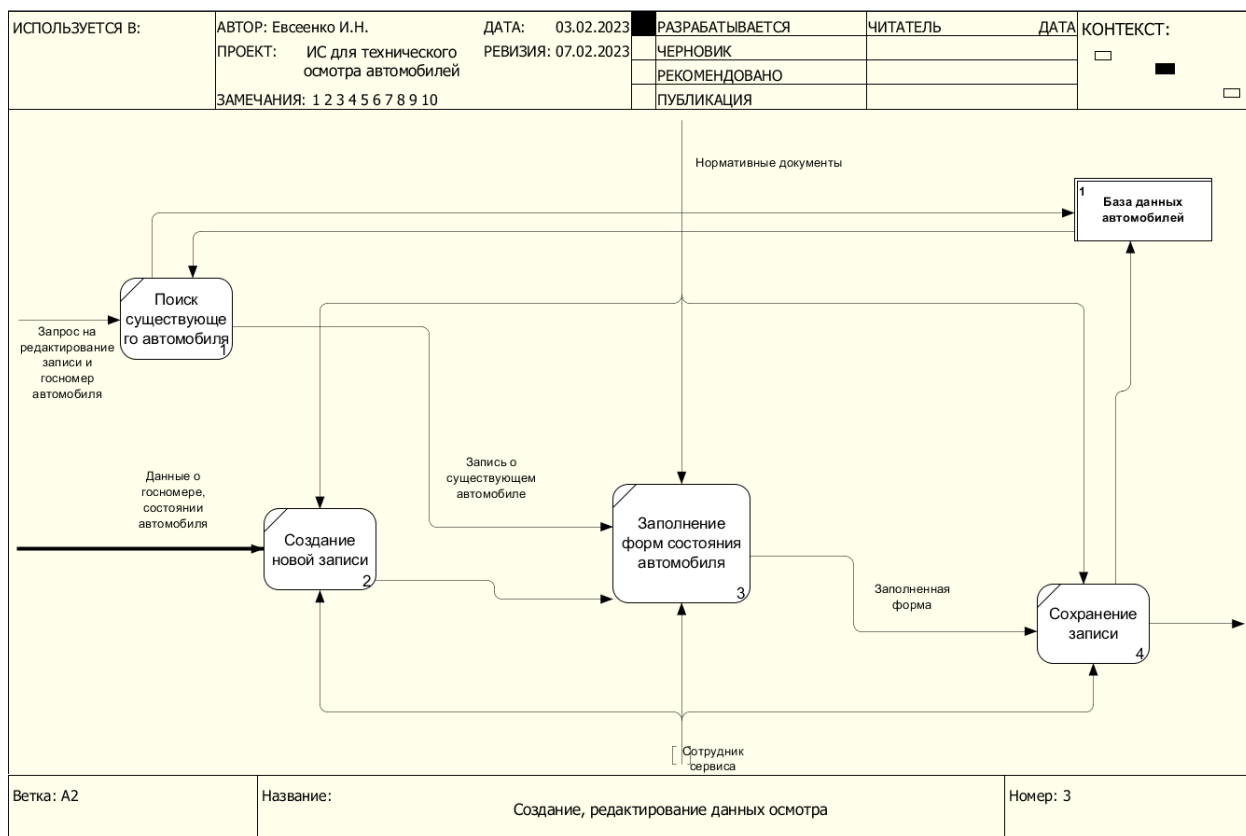


Рисунок 3 – DFD диаграмма для информационной системы функции
«Создание, редактирование и осмотр автомобиля»

4 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ СИСТЕМЫ

4.1 Модели вариантов использования системы

В ходе анализа для проектируемой информационной системы было выделено одно действующее лицо. Для каждого из них были выделены прецеденты.

Полученная диаграмма вариантов использования ИС «Технического осмотра автомобилей» показана на рисунке 4.

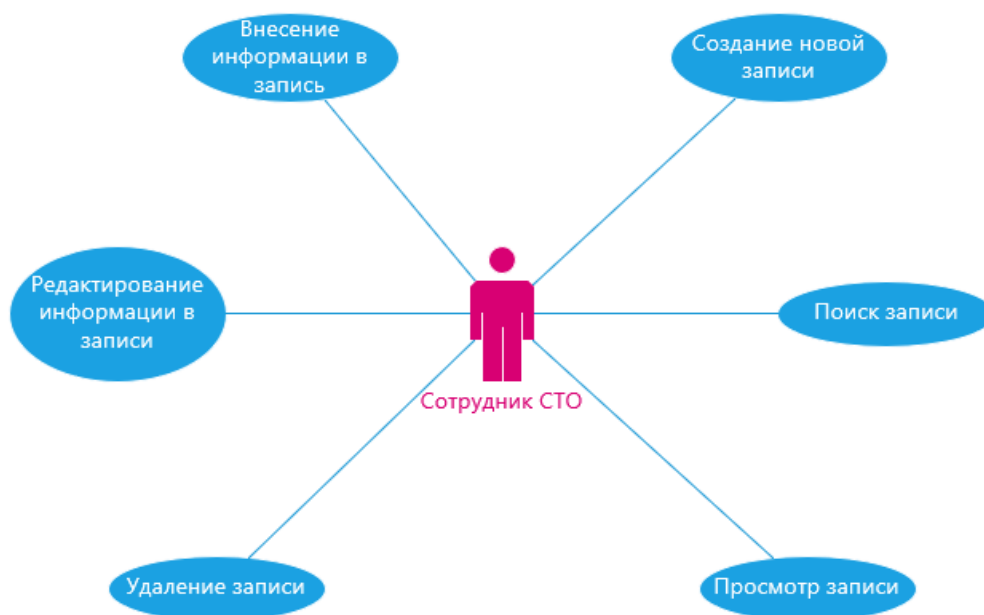


Рисунок 4 – Диаграмма вариантов использования информационной системы

В данной диаграмме один актер. Он использует всё рабочее пространство ИС, имеет возможности создания, редактирования, удаление и просмотра записей об автомобилях.

4.2 Диаграмма классов

Основной диаграммой проекта является диаграмма классов. Статистическая модель организации данных. Основная диаграмма классов представлена на рисунке 5.

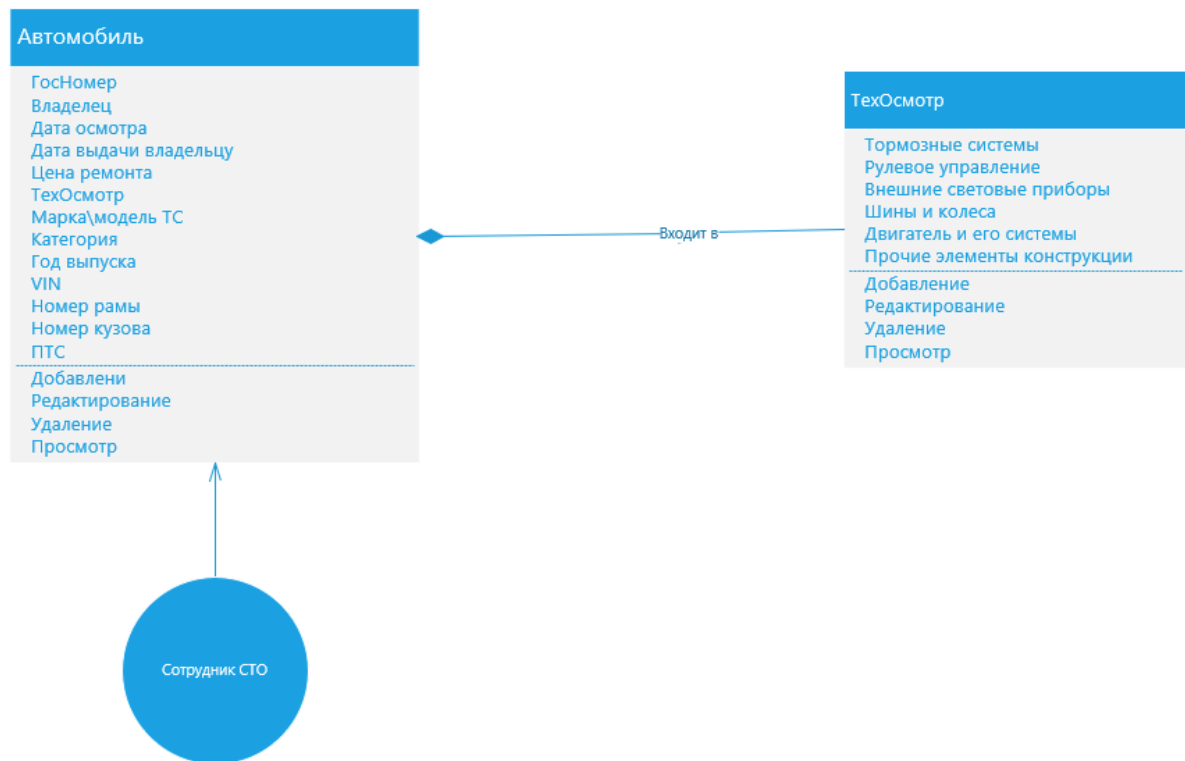


Рисунок 5 – Диаграмма классов

В данной диаграмме выделены классы использования данных и их классификация. Для удобного представления, созданы два класса – информация об автомобиле и его технического осмотра для составления предварительной информации об автомобиле при клиенте для составления накладных, а техосмотр будет проводиться самостоятельно сотрудником СТО.

4.3 Диаграммы деятельности

Диаграмма деятельности, составленная для информационной системы, представлена на рисунке 6.

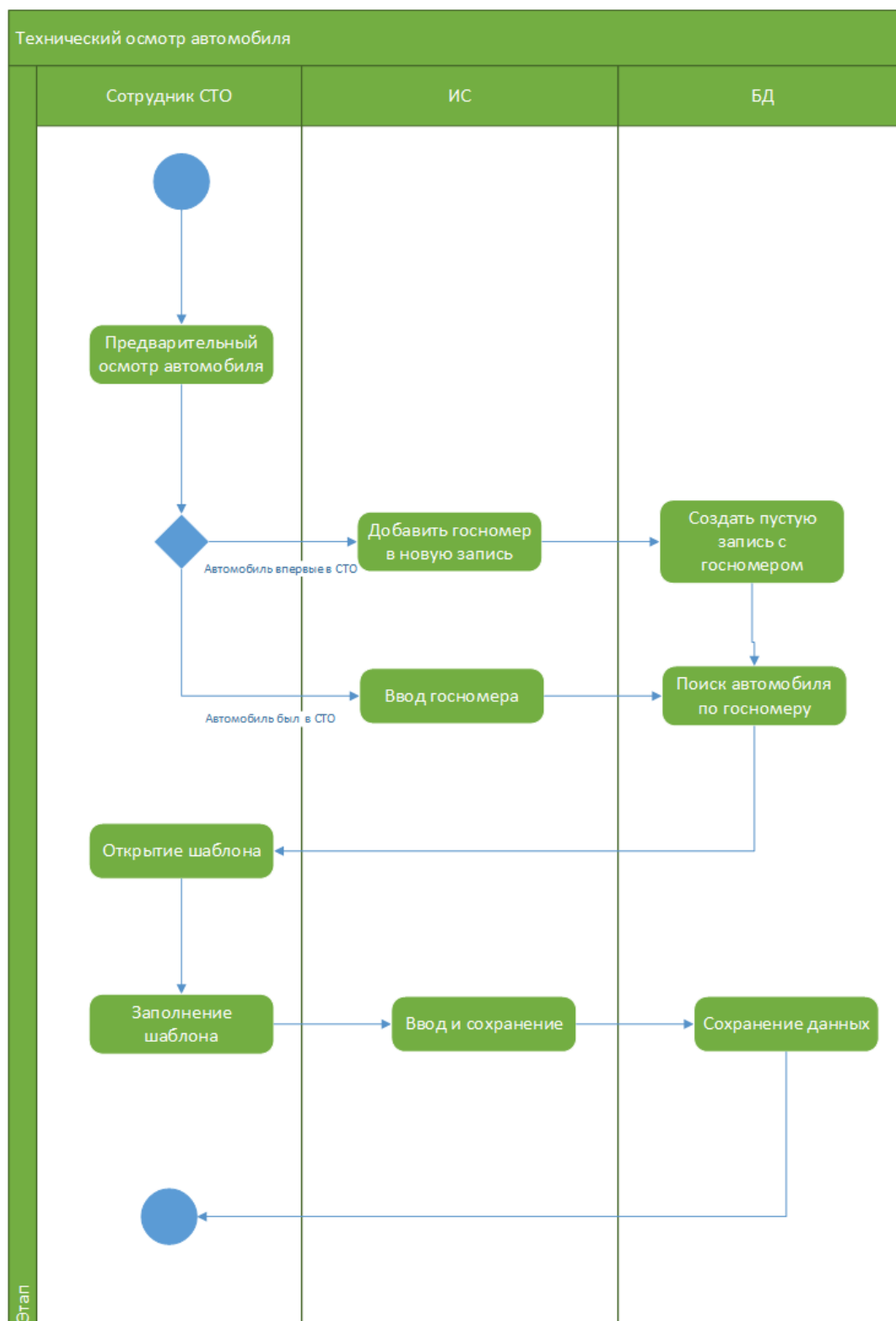


Рисунок 6 – Диаграмма деятельности

В данной диаграмме используются инструменты взаимодействия между пользователем, ИС и базой данных. В местах создания и поиска автомобиля используется автоматизированные переходы между базой

данных. Т.е. при создании нового автомобиля будет произведен его поиск в базе данных (проверка создания) и откроется шаблон.

4.4 Диаграмма последовательности

Диаграмма последовательности составленная для варианта использования «...», представлена на рисунке 7.

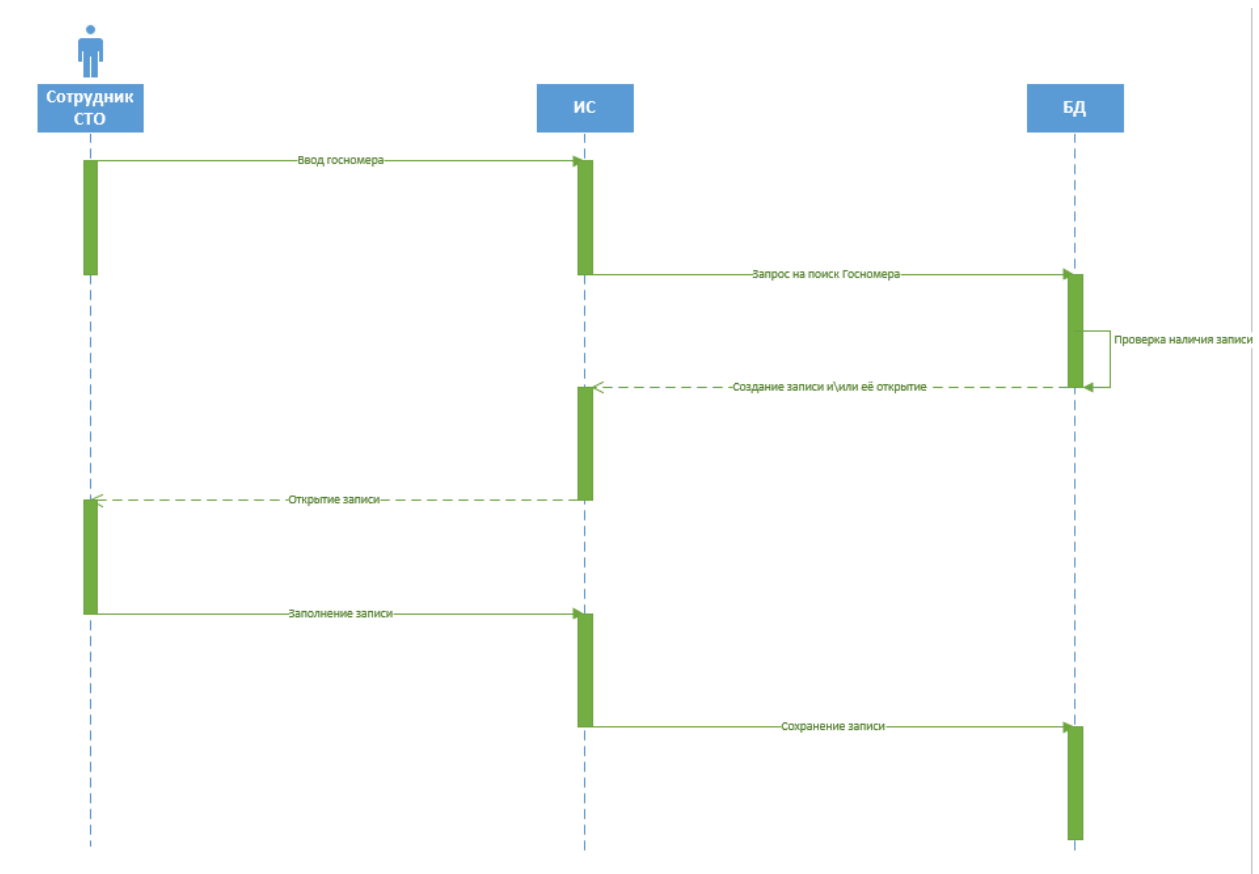


Рисунок 7 – Диаграмма последовательности

В данной диаграмме представлены последовательности создания записи об автомобиле или поиске существующей и её редактировании.

4.5 Диаграмма кооперации

Диаграмма кооперации составленная для варианта использования ИС «технического осмотра автомобилей», представлена на рисунке 8.

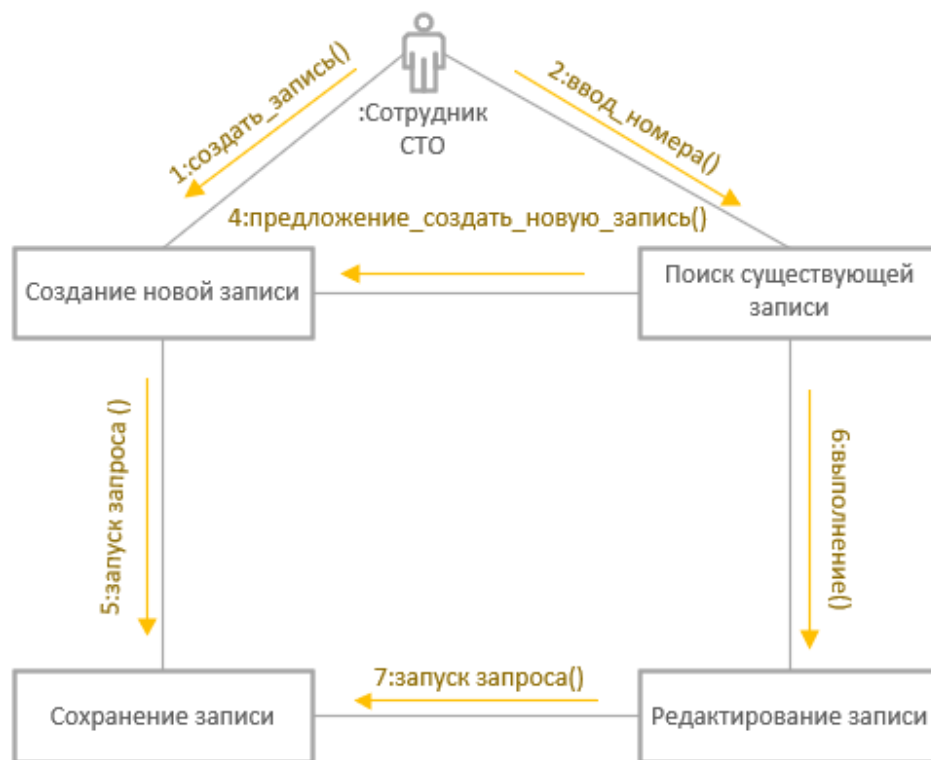


Рисунок 8 – Диаграмма кооперации

В данной диаграмме представлены функции, которые ведут к модулям активации. При выполнении функции выполняется задача, последующая данной функции. Можно сказать, что ИС модульная

4.6 Диаграмма состояний

Диаграмма состояний составленная для варианта использования ИС «технического осмотра автомобилей», представлена на рисунке 9.

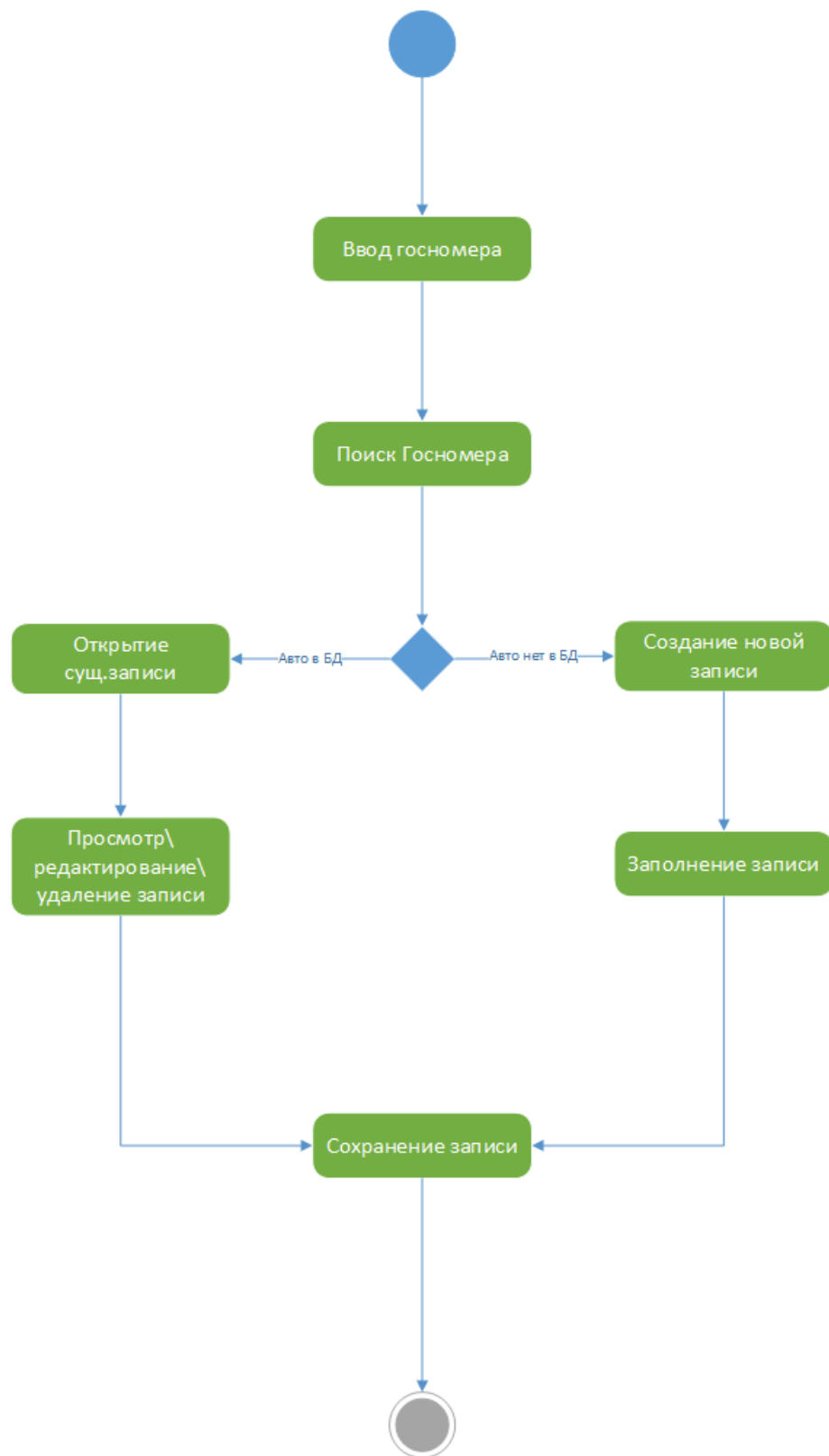


Рисунок 9 – Диаграмма состояний

В данной диаграмме показаны последовательности проявления системы с ветвлением. При различных условиях будут выполняться разные действия.

5 СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

5.1 Разработка интерфейса программного продукта

Для создания программного интерфейса и информационной системы важно разработать базу данных, поскольку она служит основой для хранения и организации данных. В базе данных хранится вся информация, необходимая для работы системы, например, учетные записи пользователей, информация о клиентах, карточки автомобиля.

Разработка базы данных включает в себя несколько этапов, в том числе проектирование схемы базы данных, создание таблиц и связей, а также заполнение базы данных данными. Разработка схемы включает в себя определение типов данных, которые необходимо хранить, взаимосвязей между данными и того, как данные будут организованы в таблицы. После разработки схемы создаются таблицы и устанавливаются связи между ними.

База данных «CarSTO» служит основой информационной системы, предоставляя необходимые данные для правильного функционирования программного интерфейса и других компонентов системы.



Рисунок 10 – Схема базы данных «CarSTO»

В ходе разработке было принято решение использовать в системе графический интерфейс.

Графический интерфейс удобнее использовать, т.е. он нагляднее представляет объекты данных в визуальный интерфейсах, таких как кнопки, формы, окна и т.п. Обычному пользователю удобнее будет пользоваться графическим интерфейсом, потому что он так привык. Такие интерфейсы сочетают удобство использования и наглядность.

На рисунке 11 представлен интерфейс разработанного программного продукта.

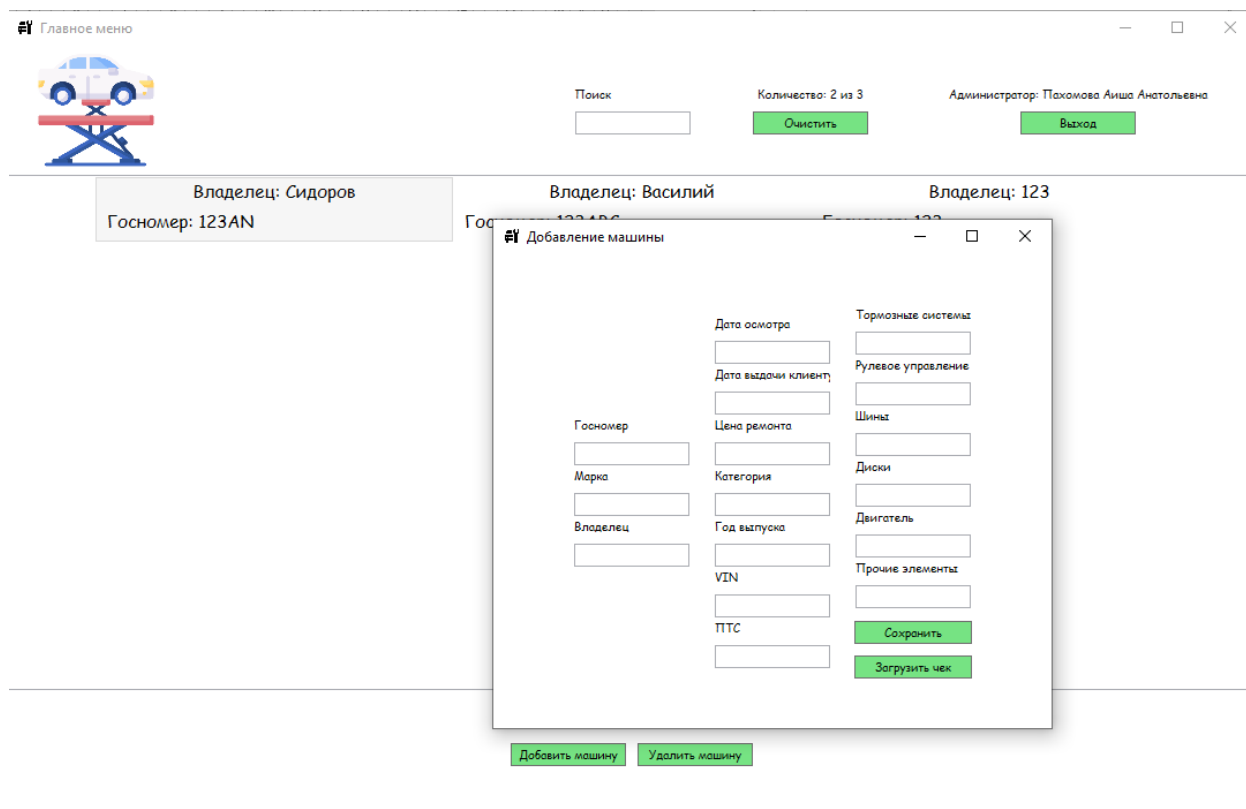


Рисунок 11 – Интерфейс приложения «Информационной системы учета технического осмотра автомобилей»

5.2 Разработка программного кода системы

Разработка приложения производилась на платформе .Net Microsoft на языке программирования C#.

C#-это язык программирования, разработанный Microsoft, который широко используется для разработки приложений в среде .NET. C# имеет ряд особенностей, которые делают его хорошим выбором для разработки интерфейса информационной системы, в том числе:

1. Объектно-ориентированный: предназначен для поддержки создания сложных программных систем.
2. Простота в освоении: относительно простой язык для изучения.
3. Кроссплатформенность: можно использовать для разработки приложений, которые могут работать на разных платформах, что делает его универсальным языком для разработки интерфейсов. (Данная система написана на WFP – фреймворке для Windows, не портируемом на другие ОС,

но используя Uno или MAUI можно написать кроссплатформенный GUI для всех ОС)

4. Интегрированная среда разработки (IDE): используется в сочетании с Microsoft Visual Studio, мощной IDE, которая предоставляет ряд инструментов и функций для разработки и отладки приложений.

5. Используя ООП, можно создать хорошо расширяемую и масштабируемую систему.

6. Используя паттерны программирования, можно привлекать в проект новых программистов, т.к. используя паттерны программирования можно добиться универсальности и читаемости кода.

Часть исходного кода приложения представлена в приложении Б.

5.3 Справочная система

Справочная система для приложения была разработана при помощи специализированной программы Dr.Explain. Файл справки, представленный на рисунке 11, был прикреплен к разработанному приложению.

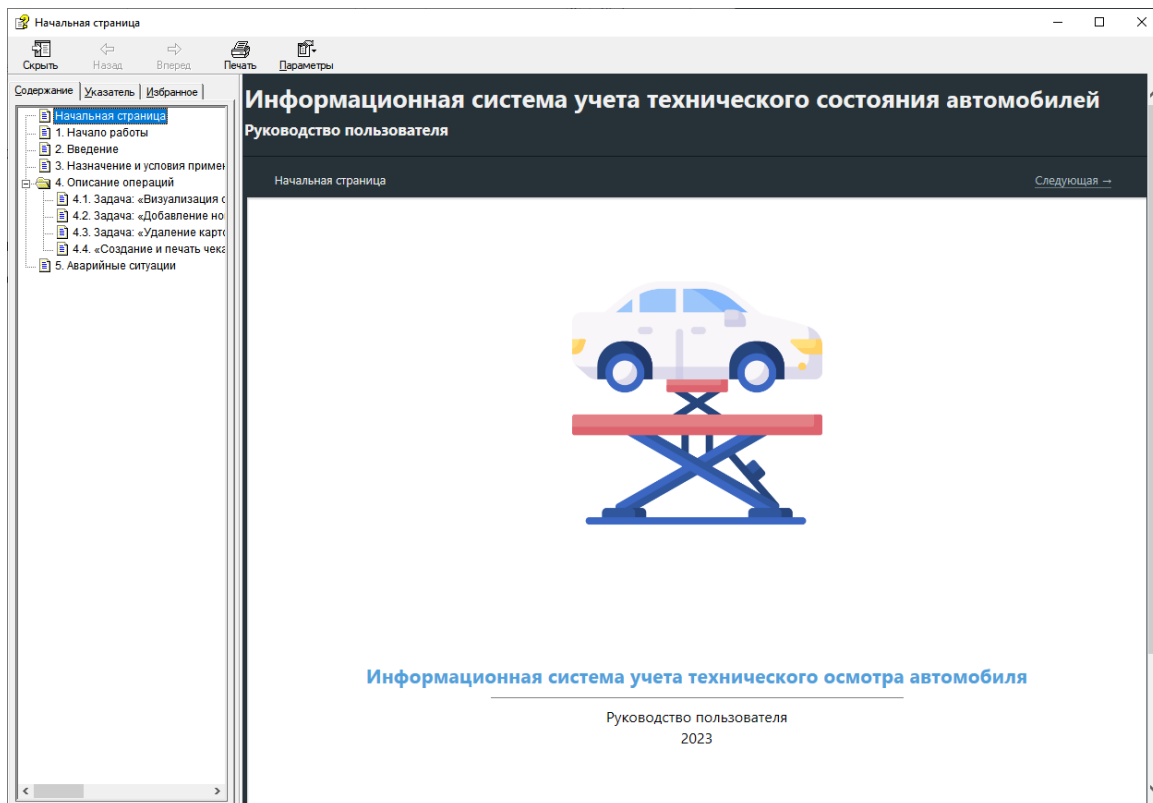


Рисунок 12 – Стартовое окно файла справки

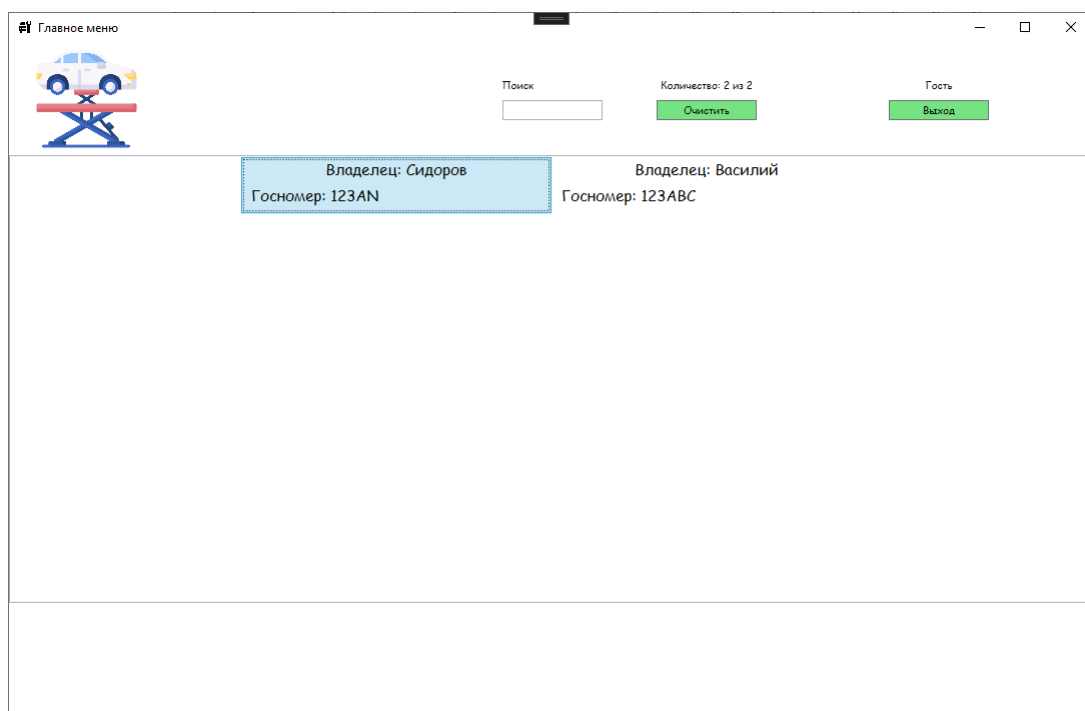
6 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

1. Метод функционального тестирования: этот метод тестирования проверяет, соответствуют ли функциональные требования системы ее спецификации. Тест-кейс: в форме добавления автомобиля не указать ненулевые значения (это госномер и владелец). Ожидаемый результат – появление диалогового окна с оповещением “Укажите госномер/владельца”. Результат тестирования: Появляется диалоговое окно “Укажите госномер” когда поле госномера пустое и “Укажите владельца” когда поле госномера заполнено, а поле владельца пустое.

The screenshot shows a web application window titled "Добавление машины" (Add car). The form contains several input fields for car details: "Госномер" (License plate), "Марка" (Brand), "Владелец" (Owner), "Дата осмотра" (Inspection date), "Дата выдачи клиенту" (Date issued to client), "Цена ремонта" (Repair price), "Тормозные системы" (Brake systems), "Рулевое управление" (Steering), "Шины" (Tires), "Двигатель" (Engine), "Дополнительные элементы" (Additional elements), and "ПТС" (Vehicle passport). There are two green buttons at the bottom: "Сохранить" (Save) and "Загрузить чек" (Load receipt). A modal dialog box is open in the center, titled "Укажите госномер" (Specify license plate), with an "OK" button.

2. Метод тестирования безопасности: этот метод тестирования проверяет, насколько безопасно и защищено система от несанкционированного доступа. Тест-кейс: попробовать удалить карточку автомобиля неавторизованным пользователем. Ожидаемый результат – отсутствие кнопки “Добавить автомобиль” и “Удалить автомобиль”.

Результат тестирования: кнопки “Добавить автомобиль” и “Удалить автомобиль” отсутствуют в форме при неавторизованным пользователем.



3. Метод тестирования совместимости: этот метод тестирования проверяет, насколько хорошо система работает с другими приложениями и системами. Тест-кейс: запуск программы в Windows 11. Ожидаемый результат: запуск программы.

Результат тестирования: программа завершилась с ошибкой “Не установлен .net 6.0 runtime” и открыла браузер, где можно скачать дистрибутив .net 6.0 runtime. После установки .net 6.0 runtime программа CarSTO.exe запустилась.

7 ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

Руководство пользователя — это документ, содержащий инструкции по использованию программной системы или приложения. Его цель — помочь пользователям понять систему и ее функциональные возможности, а также направить их в выполнении конкретных задач или операций в системе.

При подготовке Руководства пользователя следует руководствоваться нормативными документами, такими как руководства по стилю и стандартные рабочие процедуры, чтобы обеспечить последовательность и ясность представления информации. Кроме того, Руководство пользователя должно соответствовать всем применимым законам или правилам, связанным с системой программного обеспечения и ее использованием.

Следует отметить, что в процессе разработки системы документооборота, также было создано Руководство пользователя, дающее рекомендации по использованию системы.

В процессе разработки была разработана программная документация на систему, включающая в себя техническое задание на разработку (Приложение А), руководство пользователя (Приложение В).

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы была разработана информационная система для технического осмотра автомобилей в Станции технического обслуживания (далее СТО), позволяющая обеспечить централизованный и эффективный способ хранения, управления и отслеживания информации о состоянии автомобилей.

В результате применения данной информационной системы будет повышена производительность, уменьшено количество ошибок и улучшена совместная работа членов команды.

Недостатки проекта:

1. Отсутствие четких целей: проект может не иметь четких и четко определенных целей, что может затруднить определение того, достигает ли система своих целей.

2. Неадекватный пользовательский ввод: проект мог не получить достаточного вклада от пользователей системы управления документами, таких как юристы, помощники юристов и административный персонал, что может привести к тому, что система не будет соответствовать их потребностям.

3. Недостаточное тестирование системы. Возможно, проект не провел достаточного тестирования системы управления документами для выявления и устранения любых ошибок, уязвимостей в системе безопасности или других проблем.

4. Ограниченная масштабируемость: проект мог не учитывать потенциальный рост и расширение юридической фирмы, что может ограничить масштабируемость системы управления документами.

Возможные пути дальнейшего улучшения:

1. Четкое определение цели. Команда проекта должна четко определить цели системы управления информацией, в том числе то, что она должна выполнять, и как будет измеряться успех.

2. Вовлечение пользователей в процесс разработки. Команда проекта должна вовлекать пользователей системы управления информацией о состоянии автомобилей в процесс разработки, в том числе собирать их мнения и отзывы, чтобы убедиться, что система отвечает их потребностям и удобна для пользователя.

3. Проведение тщательного тестирования. Команда проекта должна провести тщательное тестирование системы управления информацией о состоянии автомобилей перед ее развертыванием, включая функциональное тестирование, тестирование производительности и тестирование безопасности, чтобы убедиться, что она надежна, безопасна и работает должным образом.

4. Планирование масштабируемости: проектная группа должна планировать потенциальный рост и расширение СТО, разрабатывая масштабируемую и адаптируемую к изменяющимся потребностям и требованиям систему управления информацией о состоянии автомобилей. Это может включать использование облачных решений или внедрение других технологий, обеспечивающих простоту масштабирования.

5. Обеспечение обучения и поддержки: команда проекта должна обеспечить надлежащее обучение и поддержку пользователей системы управления документами, чтобы убедиться, что они знают, как использовать ее эффективно и действенно. Это может включать создание руководств пользователя, проведение учебных занятий и предложение постоянной поддержки и помощи.

Также стоит отметить, что разработка системы управления информацией о состоянии автомобилей является важным шагом на пути к обеспечению эффективного и действенного управления информацией о состоянии автомобилей в СТО. Хорошо продуманная система управления информацией о состоянии автомобилей может значительно повысить общую производительность и конкурентоспособность СТО.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование: учебное пособие для спо / С. В. Белугина. — 3-е изд., стер. — Санкт-Петербург: Лань, 2022. — 312 с. — ISBN 978-5-8114-9817-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/200390>
2. Зверева В. П., Назаров А. В. Сопровождение и обслуживание программного обеспечения компьютерных систем. Учебник, -М.: Издательский центр «Академия», 2018. - 256с.
3. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для спо / Т. М. Зубкова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 252 с. — ISBN 978-5-8114-9556-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/200462>
4. Макарова, Н.В. Основы программирования: учебник / Макарова Н.В., Нилова Ю.Н., Зеленина С.Б., Лебедева Е.В. — Москва: КноРус, 2021. — 451 с. — ISBN 978-5-406-03394-4. — URL: <https://book.ru/book/936582>
5. Попов, А.А. Эргономика пользовательских интерфейсов в информационных системах: учебное пособие / Попов А.А. — Москва: КноРус, 2021. — 304 с. — ISBN 978-5-406-07634-7. — URL: <https://book.ru/book/938669>
6. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем. Учебник, -М.: Издательский центр «Академия», 2019. - 384с.
7. Руденко, О.Г. Системный анализ и проектирование информационных систем: учебное пособие / Руденко О.Г., Кожухова С.В. — Москва: ИНФРА-М, 2021. — 216 с.
8. Грэхем, И. Основы разработки программного обеспечения: учебник / Грэхем И., Спенсер Д., Блэквелл Дж. — Москва: ДМК Пресс, 2021. — 512 с.

9. Котеров, Д.Г. РНР 7 в подлиннике: учебное пособие / Котеров Д.Г.
– Москва: ДМК Пресс, 2021. – 1024 с.

10. Борисов, А.И. Базы данных: учебное пособие / Борисов А.И.,
Булычев Д.Ю., Никитин Е.А. – Москва: Издательство Юрайт, 2021. – 352 с.

ПРИЛОЖЕНИЕ А

Техническое задание к программе «Информационная система для технического осмотра автомобилей»

Содержание

1. Введение
 - 1.1. Наименование программы
 - 1.2. Назначение и область применения
2. Требования к программе
 - 2.1. Требования к функциональным характеристикам
 - 2.2. Требования к надежности
 - 2.2.1. Требования к обеспечению надежного функционирования программы
 - 2.2.2. Время восстановления после отказа
 - 2.2.3. Отказы из-за некорректных действий оператора
3. Условия эксплуатации
 - 3.1. Климатические условия эксплуатации
 - 3.2. Требования к квалификации и численности персонала
 - 3.3. Требования к составу и параметрам технических средств
 - 3.4. Требования к информационной и программной совместимости
 - 3.4.1. Требования к информационным структурам и методам решения
 - 3.4.2. Требования к исходным кодам и языкам программирования
 - 3.4.3. Требования к программным средствам, используемым программой
 - 3.4.4. Требования к защите информации и программ
 - 3.5. Специальные требования
4. Требования к программной документации
 - 4.1. Предварительный состав программной документации
5. Технико-экономические показатели
 - 5.1. Экономические преимущества разработки
6. Стадии и этапы разработки
 - 6.1. Стадии разработки
 - 6.2. Этапы разработки
 - 6.3. Содержание работ по этапам
7. Порядок контроля и приемки
 - 7.1. Виды испытаний
 - 7.2. Общие требования к приемке работы

1. Введение

Цель проекта - создать информационную систему для учета информации о техническом состоянии автомобилей. Система должна позволять сотрудникам СТО производить диагностику автомобилей и сохранять результаты в базу данных.

1.1. Наименование программы

Наименование программы: "Разработка информационной системы для технического осмотра автомобилей"

1.2. Назначение и область применения

Назначение информационной системы для учета технического состояния автомобилей заключается в автоматизации процесса осмотра автомобилей на станции технического обслуживания или автосервисе. Система позволяет сотрудникам СТО быстро и удобно вводить данные о состоянии автомобиля, осуществлять диагностику и составлять отчеты и накладные на ремонт.

Основной целью разработки информационной системы является упрощение и оптимизация процесса осмотра автомобилей, улучшение качества обслуживания автовладельцев и повышение эффективности работы сотрудников СТО.

Область применения информационной системы охватывает работу станций технического обслуживания и автосервисов, где производится осмотр, диагностика и ремонт автомобилей. Система может быть использована для учета технического состояния как легковых автомобилей, так и грузовых. Она может быть применена как в небольших автосервисах, так и в крупных сетях СТО.

2. Требования к программе

2.1. Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- Ввод данных о состоянии автомобиля с помощью шаблонных данных для каждой детали автомобиля.
- Оптимизация процесса осмотра автомобиля для каждого типа автомобиля, включая легковые и грузовые.
- Поиск автомобиля в базе данных по государственному номеру.
- Просмотр данных о каждом автомобиле для сотрудников СТО.
- Составление отчетов и накладных для клиентов, включая оценочную стоимость ремонта.

2.2. Требования к надежности

- Быстродействие: Система должна обеспечивать быстрое время ответа на запросы пользователей и максимально быстрый доступ к информации о состоянии автомобиля.
- Безопасность: Система должна обеспечивать конфиденциальность и защиту данных клиентов.
- Надежность: Система должна обеспечивать высокую степень надежности, чтобы избежать потери данных в случае сбоев в работе системы.

2.2.1 Требования к обеспечению надежного функционирования программы

Для обеспечения надежного функционирования информационной системы для учета технического состояния автомобилей необходимо уделять внимание следующим требованиям:

1. Надежность и устойчивость системы:

- Гарантия сохранности данных в случае сбоев системы или отключения электропитания;
- Устойчивость системы к вирусам, хакерским атакам и другим угрозам безопасности.

2. Производительность:

- Быстрый доступ к данным;
- Обеспечение высокой скорости работы системы даже при одновременном доступе нескольких пользователей.

3. Совместимость:

- Система должна быть совместима с различными операционными системами;

4. Удобство использования:

- Интуитивно понятный и простой интерфейс;
- Возможность обучения пользователей.

5. Поддержка и обновления:

- Регулярное обновление системы для устранения ошибок и уязвимостей;
- Техническая поддержка и помощь пользователям.

2.2.2. Время восстановления после отказа

Для обеспечения надежной работы информационной системы необходимо установить время восстановления после отказа. Время восстановления - это время, которое потребуется для восстановления работы системы после ее сбоя или отказа. Чем меньше это время, тем быстрее

система сможет вернуться в рабочее состояние и продолжить свою работу.

Для минимизации времени восстановления необходимо предусмотреть следующие меры:

1. Регулярное резервное копирование данных системы, которое позволит быстро восстановить данные после сбоя.
2. Регулярное тестирование системы на предмет выявления возможных проблем и уязвимостей.
3. Разработка плана восстановления после сбоя, включая инструкции для персонала, ответственного за восстановление системы, и тестирование этого плана на регулярной основе.
4. Использование надежного оборудования и программного обеспечения, которые не будут выходить из строя в случае сбоев или отказов.
5. Непрерывное мониторинг системы для выявления сбоев и отказов, а также быстрого реагирования на них.

Время восстановления после отказа может быть различным в зависимости от масштаба сбоя и причин его возникновения, однако необходимо стремиться к тому, чтобы оно было как можно меньше.

2.2.3. Отказы из-за некорректных действий оператора

Один из основных источников отказов в информационных системах - это некорректные действия операторов. Они могут случайно удалить или изменить данные, нарушить конфигурацию системы, вызвать перегрузку серверов и другие проблемы. Для уменьшения риска отказов из-за некорректных действий операторов необходимо предусмотреть следующие меры:

1. Обучение операторов - обучение персонала и настройка процедур использования системы помогут уменьшить число случайных ошибок.

2. Разграничение прав доступа - предоставление доступа к системе только необходимому количеству пользователей и разделение прав доступа по уровню.
3. Определение процедур - определение процедур использования системы, таких как создание резервных копий, контроль версий, проведение тестирования перед внесением изменений в систему.
4. Ограничение возможностей - ограничение возможностей операторов, например, запрет на удаление важных файлов или изменение конфигурации системы без специального разрешения.
5. Мониторинг и журналирование - создание системы журналирования и мониторинга, которые позволяют отслеживать действия операторов и выявлять потенциальные проблемы.
6. Проведение аудита - регулярный аудит системы, который позволит выявлять нарушения в использовании системы и принимать меры по их устранению.
7. Использование автоматизированных средств - использование программного обеспечения, которое автоматически обнаруживает и устраняет ошибки, может значительно снизить риск возникновения проблем.

Применение этих мер поможет уменьшить число отказов в информационной системе, вызванных некорректными действиями операторов.

3. Условия эксплуатации

3.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации

3.2. Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный администратор и конечный пользователь программы — оператор.

Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

- а) задача поддержания работоспособности технических средств;
- б) задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;
- в) задача установки (инсталляции) программы.
- г) задача создания резервных копий базы данных.

3.3. Требования к составу и параметрам технических средств

3.3.1. В состав технических средств должен входить персональный компьютер, выполняющий роль клиента и сервера, включающий в себя:

- 3.3.1.1. процессор core 2 duo, не менее;
- 3.3.1.2. оперативную память объемом, 256 Мбайт, не менее;
- 3.3.1.3. свободного пространства на жестком диске, 1 Гигабайт, не менее;
- 3.3.1.4. операционную систему Windows 10 или Windows 11;

3.4. Требования к информационной и программной совместимости

3.4.1. Требования к информационным структурам и методам решения

База данных работает под управлением Microsoft SQL Server. Используется многопоточный доступ к базе данных. Необходимо обеспечить одновременную работу с программой с той же базой данных модулей экспорта внешних данных.

3.4.2. Требования к исходным кодам и языкам программирования
Дополнительные требования не предъявляются

3.4.3. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 10 или Windows 11 и Microsoft SQL Server 10

3.4.4. Требования к защите информации и программ

Требования к защите информации и программ не предъявляются

3.5. Специальные требования

Специальные требования к данной программе не предъявляются

4. Требования к программной документации

4.1. Предварительный состав программной документации

Состав программной документации должен включать в себя:

- 4.1.1. техническое задание;
- 4.1.2. программу и методики испытаний;
- 4.1.3. руководство оператора;

5. Техничко-экономические показатели

5.1. Экономические преимущества разработки

Разработка информационной системы для учета информации о техническом состоянии автомобилей может принести несколько экономических преимуществ:

1. Оптимизация процессов: система позволит оптимизировать процессы диагностики и обслуживания автомобилей, что увеличит производительность сотрудников СТО и сократит время нахождения автомобилей на ремонте.
2. Снижение расходов: система позволит снизить расходы на хранение бумажной документации и уменьшить вероятность ошибок при заполнении бумажных документов.
3. Увеличение точности диагностики: система позволит увеличить точность диагностики автомобилей, что приведет к более качественному обслуживанию и сокращению количества повторных обращений клиентов в сервис.
4. Улучшение управления данными: система позволит улучшить управление данными и быстро находить необходимую информацию о состоянии автомобилей, что повысит эффективность работы сотрудников СТО.
5. Повышение удовлетворенности клиентов: благодаря быстрому и качественному обслуживанию, а также возможности отслеживать состояние своего автомобиля в реальном времени, клиенты будут более довольны сервисом и вернуться снова в этот автосервис.

6. Стадии и этапы разработки

6.1. Стадии разработки

Разработка должна быть проведена в три стадии:

1. разработка технического задания;
2. рабочее проектирование;
3. внедрение.

6.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;
2. разработка программной документации;
3. испытания программы.

На стадии внедрения должен быть выполнен этап разработки подготовка и передача программы

6.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;
2. определение и уточнение требований к техническим средствам;
3. определение требований к программе;
4. определение стадий, этапов и сроков разработки программы и документации на неё;
5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена

разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение и методики испытаний;
2. проведение приемо-сдаточных испытаний;
3. корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

7. Порядок контроля и приемки

7.1. Виды испытаний

Приемо-сдаточные испытания должны проводиться на объекте Заказчика в оговоренные сроки.

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком Программы и методик испытаний.

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний

7.2. Общие требования к приемке работы

На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывает Акт приемки-сдачи программы в эксплуатацию.

ПРИЛОЖЕНИЕ Б

Исходный программный код информационной системы «Учета технического осмотра автомобилей»

```
using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;

namespace CarSTO.Models;

public partial class CarSTOContext : DbContext
{
    public ICommand ExitCommand { get; set; }

    private bool CanExitCommandExecute(object p)
    {
        return true;
    }

    private void OnExitCommandExecuted(object p)
    {
        Window window = Application.Current.Windows[0];

        LoginWindow loginWindow = new LoginWindow();
        LoginViewModel loginViewModel = new LoginViewModel();
        loginWindow.DataContext = loginViewModel;

        window.Close();
        loginWindow.Show();
    }

    public ICommand ClearCommand { get; set; }

    private bool CanClearCommandExecute(object p)
    {
        return true;
    }

    private void OnClearCommandExecuted(object p)
    {
        FilterSelectedIndex = -1;
        SortSelectedIndex = -1;
        SearchText = "";
    }

    public ICommand DeleteCommand { get; set; }

    private bool CanDeleteCommandExecute(object p)
    {
        if ((Cars)p != null)
```

```

        {
            return true;
        }
        else {
            return false;
        }
    }

private void OnDeleteCommandExecuted(object p)
{
    using (CarSTOContext db = new CarSTOContext())
    {

        if (CanDeleteProduct( (Cars)p ))
        {
            if ((Cars)p != null)
            {
                if (MessageBox.Show($"Вы точно хотите удалить
{((Cars)p).Nazvanie}", "Внимание!",
                MessageBoxButton.YesNo,
                MessageBoxImage.Question) == DialogResult.Yes)
                {
                    db.Car.Remove((Cars)p);
                    db.SaveChanges();
                    MessageBox.Show($"Товар {
((Cars)p).Nazvanie} удален!");

                    CurrentProducts = db.Car.ToList();
                    CountProducts = $"Количество:
{CurrentProducts.Count()} из {db.Car.ToList().Count}";
                }
            }
        }
    }

}

public ICommand SaveProductCommand { get; set; }

private bool CanSaveProductCommandExecute(object p)
{
    return true;
}

private void OnSaveProductCommandExecuted(object p)
{
    using (CarSTOContext db = new CarSTOContext())
    {
        if (CurrentProduct.Id == null)
        {
            MessageBox.Show("Добавляем машину?");
            try
            {
                db.Car.Add(CurrentProduct);
                db.SaveChanges();
                MessageBox.Show("Продукт успешно добавлен!");
            }
            catch { }
        }
    }
}

```

```

        CountProducts = $"Количество:
{CurrentProducts.Count()} из {db.Car.ToList().Count}";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.InnerException.ToString());
    }
}
else
{
    try
    {
        db.Car.Update(CurrentProduct);
        db.SaveChanges();
        MessageBox.Show("Продукт успешно обновлен!");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
    }
}
}
}

```

```

    protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
    =>
optionsBuilder.UseSqlServer("Server=(localdb)\\mssqllocaldb;Database=C
arSTO;Trusted_Connection=True;");

```

```

    partial void OnModelCreatingPartial(ModelBuilder modelBuilder);
}

```


ПРИЛОЖЕНИЕ В

Руководство пользователя

Разделы руководства пользователя:

1. Введение.
2. Назначение и условия применения.
3. Подготовка к работе.
4. Описание операций.
5. Аварийные ситуации.
6. Рекомендации по освоению.

1. Введение

1.1. Область применения

Требования настоящего документа применяются при:

- предварительных комплексных испытаниях;
- опытной эксплуатации;
- приемочных испытаниях;

1.2. Краткое описание возможностей

Информационная системы для технического осмотра автомобилей (далее, система) предназначена для контроля и учета списка автомобилей, составления технического осмотра автомобилей

Администратор и менеджер могут редактировать данные об автомобиле, добавлять новые карточки автомобилей и удалять старые. Реализована система поиска. Пользователь может иметь учетную запись в системе.

1.3. Уровень подготовки пользователя

Пользователь информационной системы для технического осмотра автомобилей должен иметь опыт работы с компьютером. Для управления системой необходимо прочитать руководство пользователя.

2. Назначение и условия применения информационной системы для технического осмотра автомобилей

Информационная системы для технического осмотра автомобилей предназначена для сбора информации о техническом составляющем автомобиля.

Работа системы возможна всегда, когда есть необходимость в получении информации об автомобилях.

Работа системы доступна всем пользователям с установленными правами доступа.

3. Подготовка к работе

3.1. Состав и содержание дистрибутивного носителя данных

Для работы с системой необходимо следующее программное обеспечение:

1. ОС Windows 10 и выше

3.2. Порядок загрузки данных и программ

Перед началом работы с системой на рабочем месте пользователя необходимо выполнить следующие действия:


1. Скачать дистрибутив любым доступным способом
2. Распакуйте zip папку в удобное место на жестком диске
3. Запустите CarSTO.exe в папке

3.3. Порядок проверки работоспособности

Для проверки доступности системы с рабочего места пользователя необходимо выполнить следующие действия:

1. Запустить CarSTO.exe
2. Зайти под своей учетной записью или учетной записью гостя

Авторизация



Логин


Пароль

Авторизация

Гость

В верхнем правом углу отобразится имя пользователя или Гость

Главное меню



Поиск

Количество: 2 из 2

Очистить

Администратор: Пахомова Аиша Анатольевна

Выход

Владелец: Сидоров
Госномер: 123AN

Владелец: Василий
Госномер: 123ABC

Добавить машину

Удалить машину

В случае если приложение система не запускается, то следует обратиться в службу поддержки.

4. Описание операций

4.1. Выполняемые функции и задачи

Система выполняет функции и задачи, приведенные в таблице ниже:

Функции	Задачи	Описание
Отображение списка автомобилей	Визуализация списка	В ходе выполнения данной задачи пользователю системы предоставляется возможность работы со списком автомобилей, описания, технической информации об автомобиле
Добавление автомобиля в список	Добавление нового автомобиля или редактирование текущего с целью изменить данные об автомобиле	В ходе выполнения данной задачи пользователю системы предоставляется возможность создать новую карточку автомобиля, добавить к ней описание, марку, госномер, техническую информацию и т.п.
Удаление автомобиля из списка	Удаление автомобиля из списка	В ходе выполнения данной задачи пользователю системы предоставляется возможность удаления карточки автомобиля из списка.
Создание и печать чека	Создание PDF файла чека и отправка его в печать	В ходе выполнения данной задачи пользователю предоставляется возможность создать PDF файл чека карточки автомобиля и отправить его автоматически на печать

4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач

Задача: «Визуализация списка»

Операция 1: Зайти в учетную запись системы

Условия, при соблюдении которых возможно выполнение операции:

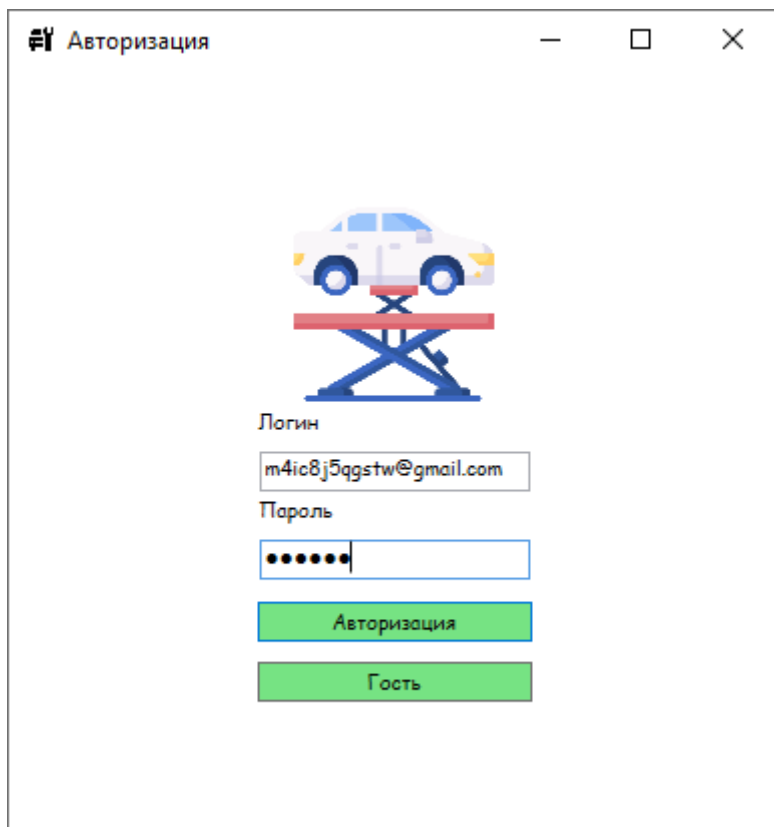
1. Компьютер пользователя подключен к корпоративной сети.
2. База данных доступна.
3. Система функционирует в штатном режиме.

Подготовительные действия:

На компьютере пользователя необходимо выполнить дополнительные настройки, приведенные в п. 3.2 настоящего документа.

Основные действия в требуемой последовательности:

1. Запустить CatSTO.exe
2. В открывшемся окне в поле «Логин» ввести имя пользователя, в поле «Пароль» ввести пароль пользователя и нажать на кнопку Авторизация или нажать на кнопку Гость.



Операция 2: Дождаться загрузки списка карточек автомобилей

The screenshot shows a web application window titled "Главное меню" (Main Menu). In the top left corner, there is a logo of a car on a lift. The top right corner displays the user's role "Администратор: Таханова Аиша Анатольевна" and a "Выход" (Logout) button. Below the header, there is a search bar labeled "Поиск" and a status indicator "Количество: 2 из 2" with an "Очистить" (Clear) button. The main content area displays a list of cars in a table. The first row shows "Владелец: Сидоров" and "Госномер: 123AN". The second row shows "Владелец: Василий" and "Госномер: 123ABC". At the bottom of the window, there are two buttons: "Добавить машину" (Add car) and "Удалить машину" (Delete car).

Владелец	Госномер
Сидоров	123AN
Василий	123ABC

Заключительные действия:

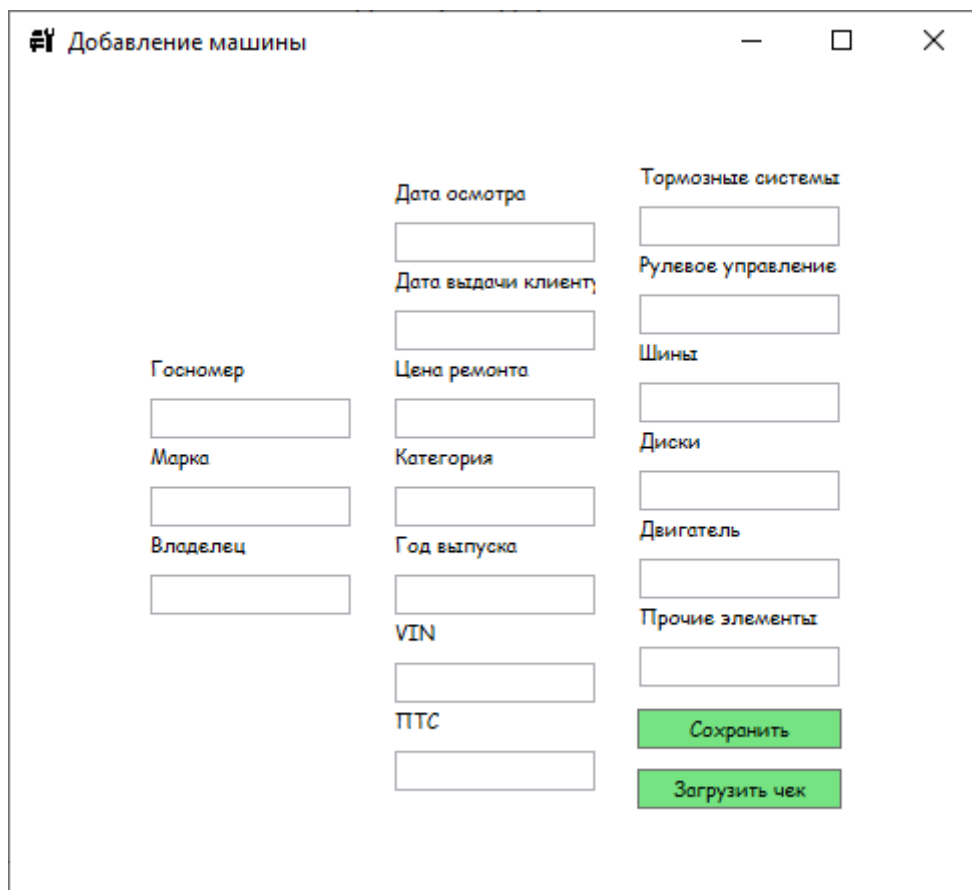
После завершения работы с можно выйти из системы с помощью кнопки Выйти или закрыть текущее окно

Задача: «Добавление новой карточки автомобиля или редактирование текущей с целью изменить данные об автомобиле»

Операция 1: Зайти в учетную запись системы как из предыдущего пункта, но для добавления карточки нужна учетная запись менеджера или администратора.

Операция 2: Добавление информации об автомобиле

1. Нажмите на кнопку “Добавить автомобиль” – откроется новое окно
2. Заполните необходимые поля



Добавление машины

Госномер

Марка

Владелец

Дата осмотра

Дата выдачи клиенту

Цена ремонта

Категория

Год выпуска

VIN

ПТС

Тормозные системы

Рулевое управление

Шины

Диски

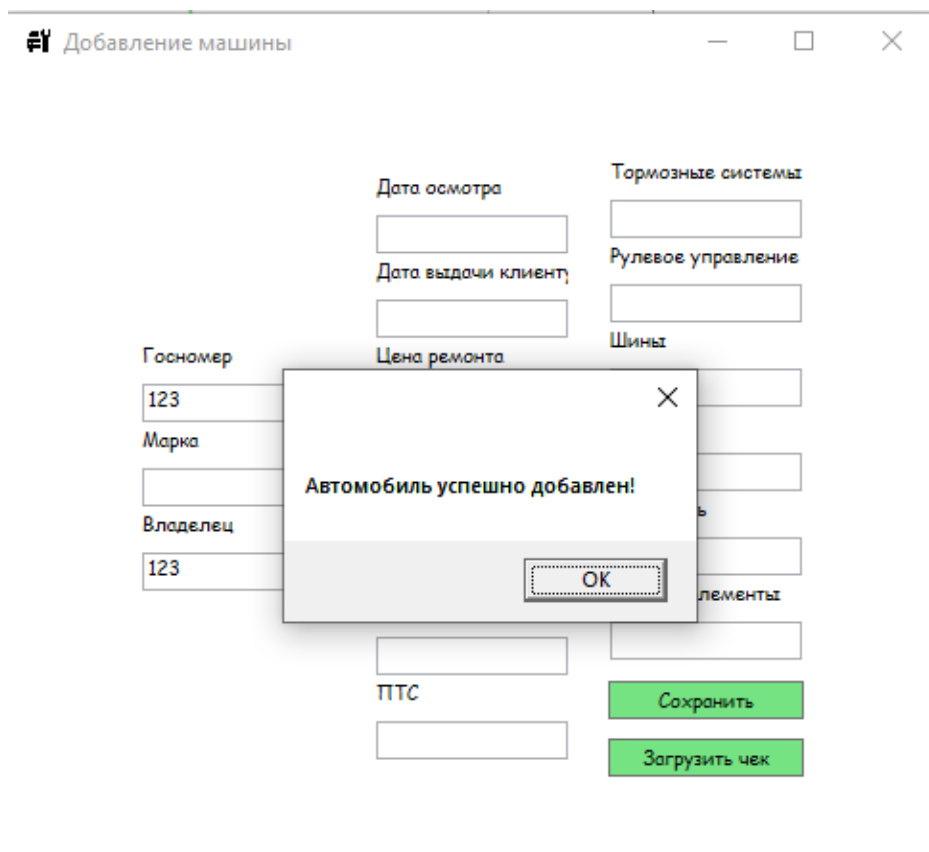
Двигатель

Прочие элементы

Сохранить

Загрузить чек

3. После заполнения данными нажмите кнопку Сохранить и убедитесь что система сообщит вам об успешности операции



Добавление машины

Госномер

123

Марка

Владелец

123

Дата осмотра

Дата выдачи клиенту

Цена ремонта

Категория

Год выпуска

VIN

ПТС

Тормозные системы

Рулевое управление

Шины

Диски

Двигатель

Прочие элементы

Сохранить

Загрузить чек

Автомобиль успешно добавлен!

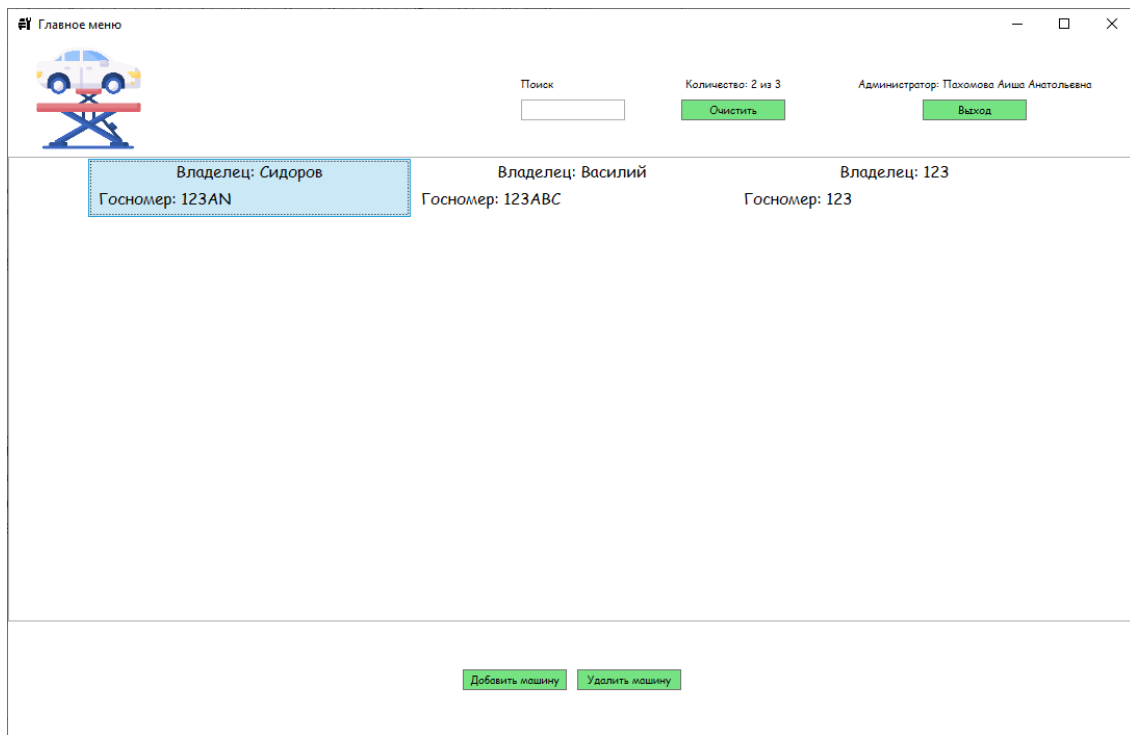
OK

Откроется окно редактирования.

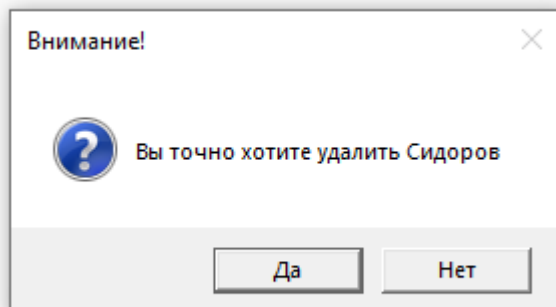
Заключительные действия:

Задача: «Удаление карточки автомобиля из списка»

Операция 2: Выберите карточку автомобиля из списка (она выделится)



Нажмите на кнопку Удалить автомобиль и подтвердите действие в диалоговом окне



Заключительные действия:

После завершения работы убедитесь, что автомобиль удален из списка.

Задача: «Создание и печать чека»

Операция 1: Зайти в учетную запись системы как из предыдущего пункта, но для добавления автомобиля нужна учетная запись менеджера или администратора.

Операция 2: Дважды нажмите на карточку автомобиля из списка

Главное меню

Поиск

Количество: 2 из 3

Очистить

Администратор: Пахомова Аиша Анатольевна

Выход

Владелец: Сидоров

Госномер: 123AN

Владелец: Василий

Госномер: 123ABC

Владелец: 123

Госномер: 123

Добавить машину

Удалить машину

Откроется окно редактирования карточки

Редактирование Сидоров

Госномер

123AN

Марка

Владелец

Сидоров

Дата осмотра

Дата выдачи клиенту

Цена ремонта

Категория

Год выпуска

VIN

ПТС

Тормозные системы

Рулевое управление

Шины

Диски

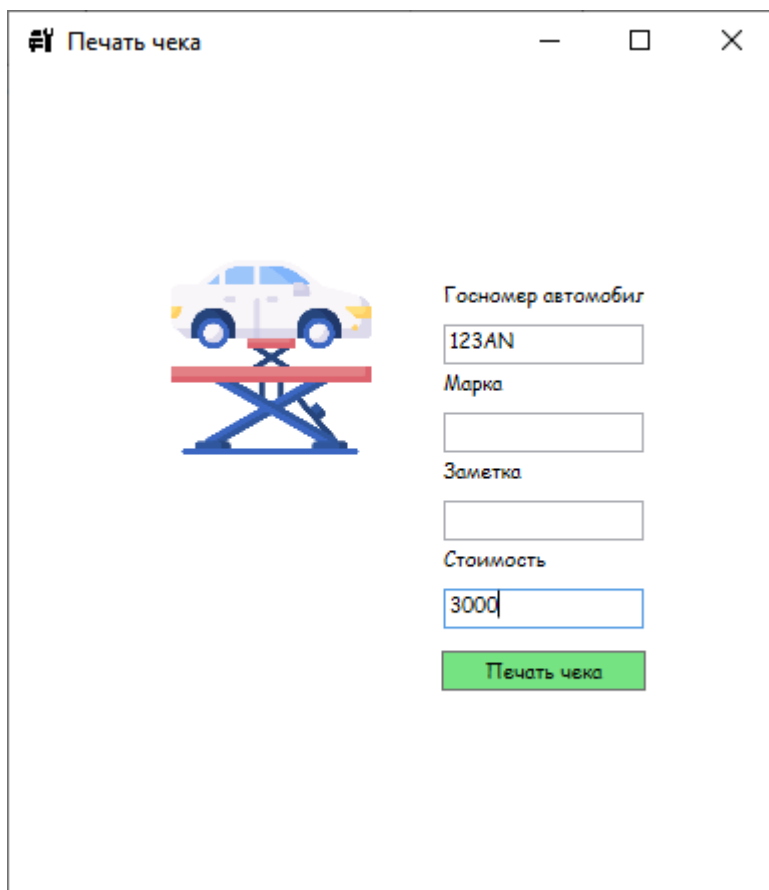
Двигатель

Прочие элементы

Сохранить

Загрузить чек

Нажмите на кнопку Загрузить чек



Печать чека

Госномер автомобиля
123AN

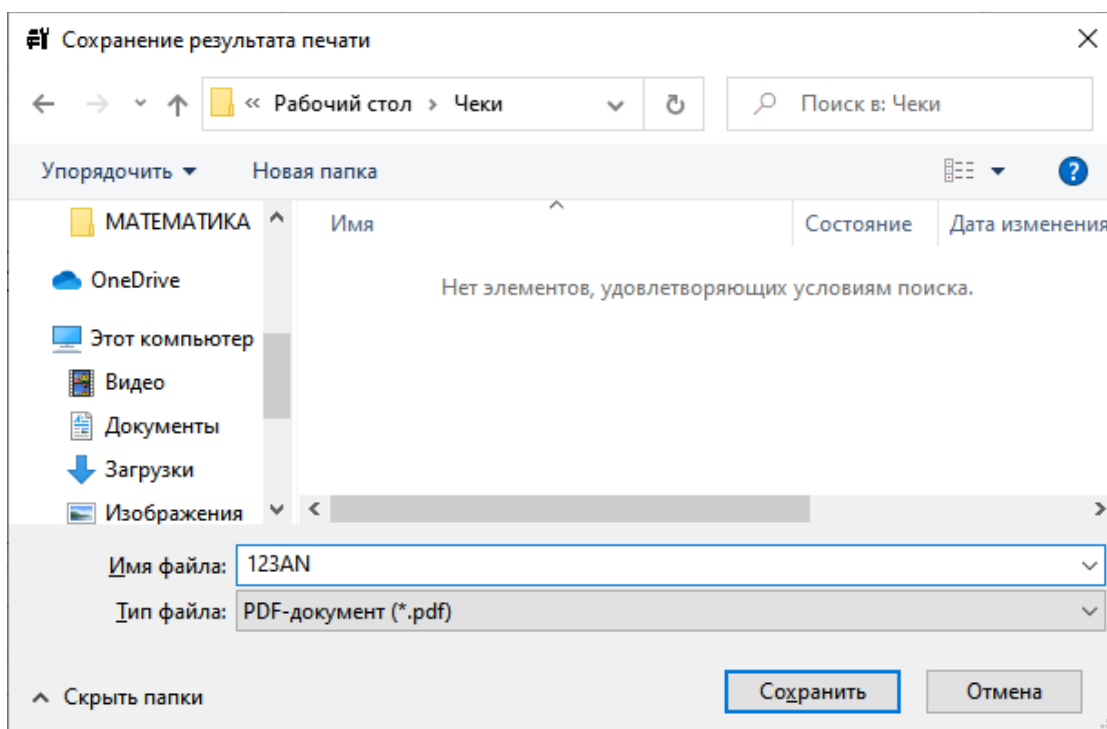
Марка

Заметка

Стоимость
3000

Печать чека

Заполните или измените данные и нажмите на Печать чека



Сохранение результата печати

← → ↕ ⬅ « Рабочий стол » Чеки ➡ ↻ 🔍 Поиск в: Чеки

Упорядочить ▾ Новая папка

Имя Состояние Дата изменения

Нет элементов, удовлетворяющих условиям поиска.

Имя файла: 123AN

Тип файла: PDF-документ (*.pdf)

Скрыть папки Сохранить Отмена

Выберите, в какую папку сохранить чек и укажите название файла и нажмите кнопку Сохранить.

Заключительные действия:

После завершения работы убедитесь, что в трее Windows появился значок принтера, а в указанной выше папке появился файл чека, который Вы сохранили. Принтер при этом отправил печать чека в очередь и при правильной настройке принтера администратором, распечатает чек.



5. Аварийные ситуации

В случае возникновения ошибок при работе системы, не описанных ниже в данном разделе, необходимо обращаться к сотруднику подразделения технической поддержки либо к ответственному Администратору.