Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

Ликино-Дулевский политехнический колледж – филиал ГГТУ

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Дипломный проект

Разработка приложения по приему запросов от клиентов организации

ООО «Новый Коммунальный Стандарт»

Выполнил:

Серебрянников Евгений Александрович\_\_\_\_\_\_\_\_\_\_\_

студент группы ИСП.20А\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по специальности

09.02.07 Информационные системы и программирование

очной формы обучения

Руководитель:

Селивёрстова Ольга Михайловна\_\_\_\_\_\_\_\_\_

Оценка:

\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулево

2024 год

**Содержание**

[Введение 3](#_Toc168386973)

[1. Разработка системного проекта 7](#_Toc168386974)

[1.1. Назначение разработки 7](#_Toc168386975)

[1.2. Требования к функциональным характеристикам 7](#_Toc168386976)

[1.3. Требования к надёжности и безопасности 8](#_Toc168386977)

[1.4. Требования к составу и параметрам технических средств 9](#_Toc168386978)

[1.5. Требования к информационной и программной совместимости 9](#_Toc168386979)

[2. Разработка технического проекта 10](#_Toc168386980)

[2.1. Обоснование выбора Case – средств, Case -технологии 10](#_Toc168386981)

[2.2. Проектирование модели данных 10](#_Toc168386982)

[2.3. Детальное проектирование интерфейса 13](#_Toc168386983)

[2.4. Функциональная схема 16](#_Toc168386984)

[3. Реализация 17](#_Toc168386985)

[3.1. Обоснование выбора средств разработки 17](#_Toc168386986)

[3.2. Руководство системного программиста 17](#_Toc168386987)

[3.3. Руководство программиста 21](#_Toc168386988)

[3.4. Руководство пользователя 24](#_Toc168386989)

[4.Тестирование и отладка 30](#_Toc168386990)

[4.1. Виды тестирования (у каждого свой вид) 30](#_Toc168386991)

[4.2. Отладка приложения 30](#_Toc168386992)

[5. Расчет базовой себестоимости разрабатываемого продукта 33](#_Toc168386993)

[5.1. Исходные данные 33](#_Toc168386994)

[5.2. Расчеты затрат на выполнение программы 33](#_Toc168386995)

[5.3. Расчёт отчислений на социальное страхование и обеспечение 34](#_Toc168386996)

[5.4. Расчёт базовой себестоимости компьютерного продукта 34](#_Toc168386997)

[Заключение 37](#_Toc168386998)

[Список литературы 38](#_Toc168386999)

[Приложения 40](#_Toc168387000)

**Введение**

Управление заявками — это процесс, который включает в себя обработку, контроль и анализ обращений пользователей, направленных на удовлетворение их потребностей и повышение уровня обслуживания. Этот процесс играет важную роль в современном бизнесе, так как он помогает организациям эффективно взаимодействовать с клиентами, партнёрами и сотрудниками.

Основные задачи управления заявками:

* Обработка входящих обращений: сбор, регистрация и классификация заявок.
* Рассмотрение и принятие решений: определение приоритетов, назначение ответственных исполнителей и сроков выполнения.
* Контроль выполнения: мониторинг хода работ, корректировка планов и ресурсов при необходимости.
* Завершение и закрытие заявок: проверка результатов, оценка качества выполнения и предоставление обратной связи.

Для успешного управления заявками организациям необходимо использовать специализированные инструменты и системы, которые помогут автоматизировать процессы, улучшить коммуникацию между участниками и повысить эффективность работы.

**Актуальность и значимость рассматриваемой проблемы**

Актуальность и значимость проблемы управления заявками обусловлены рядом факторов, которые делают эту область критически важной для многих организаций и предприятий. Вот некоторые из причин, по которым управление заявками является актуальным и значимым:

* улучшение обслуживания клиентов: эффективное управление заявками позволяет организации быстро и адекватно реагировать на запросы клиентов, что повышает уровень удовлетворенности и лояльности клиентов.
* повышение эффективности работы: систематизированный подход к управлению заявками помогает оптимизировать рабочие процессы, сократить время обработки заявок и уменьшить вероятность ошибок.
* обеспечение прозрачности процессов: управление заявками позволяет отслеживать статус каждой заявки, что обеспечивает прозрачность для клиентов и сотрудников, а также помогает в управлении ресурсами.
* аналитика и принятие решений: данные, собранные в процессе управления заявками, могут быть использованы для анализа тенденций, выявления проблемных областей и принятия обоснованных управленческих решений.
* конкурентное преимущество: организации, которые эффективно управляют заявками, могут предложить более быстрые и качественные услуги, что дает им конкурентное преимущество на рынке.

**Цель исследования** – проанализировать бизнес-процессы, происходящие в организации «Новый Коммунальный Стандарт», собрать материал теоретической и практической направленности для формирования банка данных, обработки и хранения поступающей информации, разработать приложение, которое автоматизирует рабочие процессы технического администратора.

**Задачи:**

1. Изучение литературы, относящейся к предметной области;
2. Формирование требований к разрабатываемому программному продукту;
3. Разработка программного продукта;
4. Расчет базовой стоимости программного продукта;
5. Разработка полной и понятной документации руководства пользователя и программиста.

**Методы**:

1. SQL Server – для создания базы данных
2. Visual Studio – для создания приложения.
3. Draw.io – для проектирования предметной области.
4. Word – для написания пояснительной записки
5. Excel – для расчета экономической части

**Результатом работы** является рабочее приложение по приему запросов от клиентов организации ООО «Новый Коммунальный Стандарт».

**Предлагаемая работа** состоит из введения, пяти разделов, заключения и приложения. Во введении обосновывается актуальность темы, формулируются цель, задачи и практическая значимость работы.

Первый раздел посвящена описанию разработки, составу выполняемых функций, а также техническим и программным требованиям.

Второй раздел является разработкой технического проекта в Case-средствах. Описаны программы для разработки, созданы схемы в Draw.io, определены бизнес-процессы, описан алгоритм работы программы, а также визуальный интерфейс.

Третий раздел описывает этапы реализации программного продукта. Показаны таблицы базы данных, написаны руководство программиста и руководство пользователя, описывающие программу, методы обращения к программному продукту.

Четвертый раздел посвящен тестированию и отладке программного продукта. Описаны тестовый сценарии, показаны ошибки, которые возникали при разработке приложения.

Пятый раздел описывает расчет базовой стоимости продукта. Произведен расчет затрат на выполнение программы, расчет отчислений на социальное страхование и обеспечение, расчёт отчислений на социальное страхование и обеспечение Расчёт базовой себестоимости компьютерного продукта.

В заключении описываются основные результаты выпускной квалификационной работы.

Приложение содержит *программный модуль страницы «Заявки», используемых в разработанной системе, а также документы Word и статистики в Excel.*

Общий объем работы 45 страницы. Список литературы содержит 12 источников.

**1. Разработка системного проекта**

**1.1. Назначение разработки**

**Наименование приложения:** программа по приему запросов от клиентов организации ООО «Новый Коммунальный Стандарт»

**Цель приложения:** программа предназначена для оптимизации процессов обработки запросов на предоставление услуг и упрощения ввода данных. Она помогает систематизировать информацию о запросах и обеспечивает лёгкий доступ к ней для дальнейшего использования. Также программа упрощает процесс вывода данных на печать в формате документа Word, что значительно ускоряет работу технических администраторов, имеющих дело с большим количеством запросов.

**1.2. Требования к функциональным характеристикам**

**Требования к функциональным возможностям приложения:**

Программа представляет собой систему управления заявками на оказание услуги и информацией о клиентах. Пользователь имеет возможность просматривать, редактировать и удалять информацию о заявках и клиентах. Для удобства работы с данными, предусмотрена сортировка таблиц по основным полям (например, по дате заявки или по статусу работы) и фильтрация по типам заявок, статусам работы и видам услуг.

Кроме того, пользователь может добавлять и удалять информацию в архиве, который служит для хранения устаревших данных или информации о завершенных заявках. Для оформления заявок и договоров предусмотрена функция вывода на печать, которая позволяет создавать документы в удобном для печати формате.

Для анализа данных и создания статистики по работе мастеров и по количеству купленных услуг предусмотрена возможность экспорта информации в формат Excel. Это позволяет пользователю проводить анализ эффективности работы сервиса, выявлять тренды и оптимизировать процессы.

Важным элементом программы является система авторизации пользователей. Каждый пользователь должен иметь уникальный логин и пароль для доступа к данным. Это обеспечивает безопасность информации и контролирует доступ к конфиденциальным данным.

Таким образом, программа предоставляет широкие возможности для управления заявками на оказание услуги и клиентской информацией, обеспечивая удобство работы с данными, контроль доступа и возможность анализа результатов.

**Взаимодействие с другими приложениями**

Экспорт данных из приложения в файл Excel представляет собой процесс загрузки данных из электронной таблицы приложения в файл Excel

Привязка базы данных SQL к приложению требует создания модели Entity Framework для взаимодействия с данными.

**Взаимодействие с пользователем**

**Технический администратор**

Технический администратор может просматривать, редактировать и удалять информацию о заявках и клиентах, сортировать таблицы по основным полям или фильтровать по типам заявок, статусам работы и услугам для удобного просмотра, добавлять и удалять информацию в архиве, выводить на печать договоры по оказанию услуги и бланки заявок, создавать статистические отчёты в Excel и авторизоваться в системе для доступа ко всем этим функциям.

**1.3. Требования к надёжности и безопасности**

Авторизация через логин и пароль служит эффективным средством защиты современного приложения, поскольку предоставляет возможность контроля доступа к функциональности и данным приложения, тем самым предотвращая неавторизованное использование и потенциальное злоупотребление ресурсами приложения. Это достигается за счет того, что пользователи должны пройти процесс аутентификации, то есть подтвердить свою личность, путем ввода уникального логина (имени пользователя) и пароля, известного только им. Без успешной авторизации через логин и пароль пользователь не сможет получить доступ к функциям и данным приложения, что обеспечивает их защиту от несанкционированного использования и воздействия.

**1.4. Требования к составу и параметрам технических средств**

Таблица №1 «Минимальные требования»

|  |  |
| --- | --- |
| Процессор | Intel(R) Core (TM) i5-4460 CPU @ 3.20GHz |
| Оперативная Память | 8 ГБ, DDR4, DIMM, 2133 МГц; |
| Разрешение экрана | 1920x1080 |
| Размер экрана | От 20 дюймов |
| Устройства ввода | Мышь, клавиатура |
| Дисковое пространство | 1 Гб |
| Подключаемые ус-ва | Принтер |
| Операционная система | Windows 10 и выше |

**1.5. Требования к информационной и программной совместимости**

Для корректной работы программы необходимо:

Windows 10 - рекомендуемая операционная система

Visual Studio 2022 - оптимальная среда разработки программ

Microsoft SQL Server Management Studio - необходим для разработки базы данных

Microsoft Excel 2016 - хорошо подходит для анализа данных и создания статистики

Microsoft Word 2019 - удобный инструмент для создания документов, таких как договоры и бланки запросов.

**2. Разработка технического проекта**

**2.1. Обоснование выбора Case – средств, Case -технологии**

CASE-средства (Computer Aided Software Engineering) — это инструмент, который позволяет автоматизировать процесс разработки информационной системы и программного обеспечения. Основная цель применения CASE-средств — сокращение времени и затрат на разработку информационных систем и повышение их качества.

Draw.io — это бесплатный интернет-сервис, который помогает создавать схемы различных алгоритмов и процессов, прототипы и инфографику, проектировать структуры любого типа и сложности.

**Основные преимущества Draw.io:**

1. Простота использования. Сервис интуитивно понятен, все функции и инструменты легко найти и начать работу.
2. Облегчённый интерфейс. В нём можно быстро собрать нужный продукт.
3. Большой набор готовых элементов-шаблонов — около 500

**2.2. Проектирование модели данных**

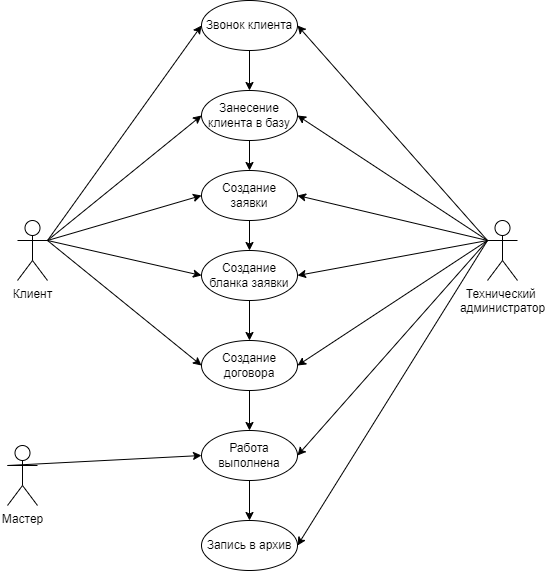


Рис. 1 «Диаграмма прецедентов»

Таблица №2 «Бизнес-процессы»

|  |  |
| --- | --- |
| **Номер бизнес-процесса** | **Название бизнес-процесса** |
| 1\_Зк | Звонок клиента |
| 2\_Зквб | Занесение клиента в базу |
| 3\_Сз | Создание заявки |
| 4\_Сбз | Создание бланка заявки |
| 5\_Сд | Создание договора |
| 6\_Рв | Работа выполнена |
| 7\_Зва | Запись в архив |

**Словесный алгоритм 5\_СД «Создание договора»**

1) Технический администратор предоставляет макет договора, разработанный в соответствии с требованиями заказчика.

2) Клиент внимательно изучает предоставленный макет договора, обращая внимание на условия сотрудничества, обязанности сторон, сроки выполнения работ и порядок оплаты.

3) Если в макете договора обнаружены ошибки или неточности, клиент сообщает об этом техническому администратору, который вносит необходимые исправления.

4) Клиент заполняет личные данные в соответствующих полях договора, указывая своё имя, контактные данные и реквизиты для оплаты.

5) Технический администратор повторно проверяет исправленный макет договора, чтобы убедиться в отсутствии ошибок и неточностей.

6) Технический администратор подписывает договор от имени компании-исполнителя, используя печать и подпись организации.

7) Клиент подписывает договор, подтверждая своё согласие с условиями сотрудничества и обязуясь оплатить услуги технического администратора в соответствии с указанным порядком и сроками оплаты.

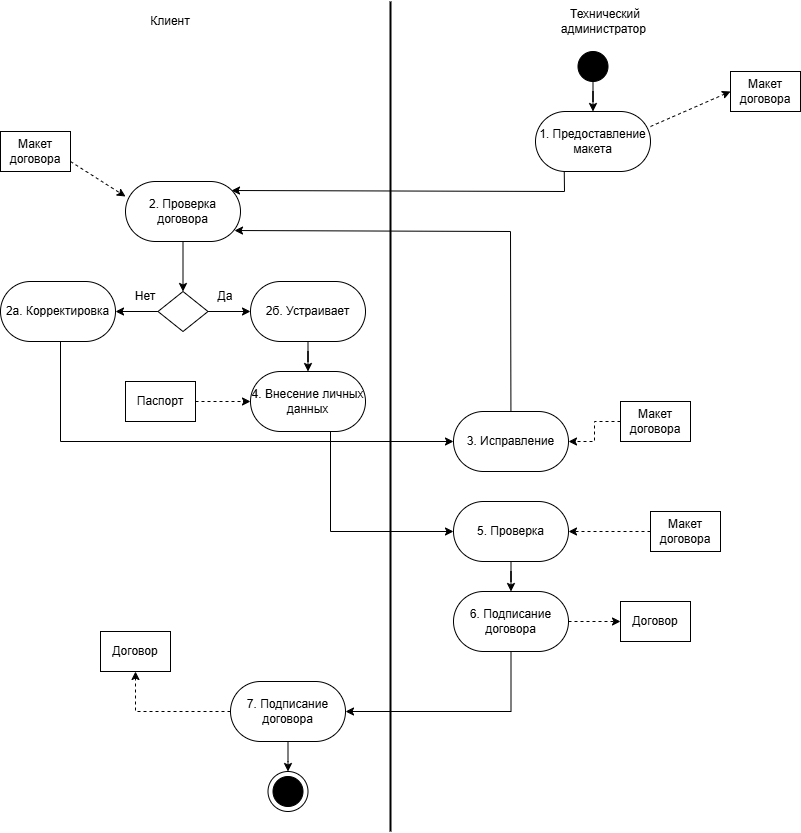


Рис. 2 «Диаграмма действий»

Таблица №3 «Таблица описания операций «Создание договора \_5Сд»»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Диаграмма и**  **номер операции**  **на**  **диаграмме** | **Операция** | **Исполнитель** | **Как часто** | **Входящие**  **документы** | **Исходящий**  **документ** |
| 5\_Сд\_1 | Предоставление макета | Технический администратор | по мере необходимости | - | Макет договора |
| 5\_Сд\_2 | Проверка договора | Клиент | по мере необходимости | Макет договора | - |
| 5\_Сд\_2а | Корректировка | Клиент | по мере необходимости | - | - |
| 5\_Сд\_2б | Устраивает | Клиент | по мере необходимости | - | - |
| 5\_Сд\_3 | Исправление | Технический администратор | по мере необходимости | Макет договора | - |
| 5\_Сд\_4 | Внесение личных данных | Клиент | по мере необходимости | Паспорт | - |
| 5\_Сд\_5 | Проверка | Технический администратор | по мере необходимости | Макет договора | - |
| Продолжение таблицы №3 «Таблица описания операций «Создание договора \_5Сд»» | | | | | | |
| 5\_Сд\_6 | Подписание договора | Технический администратор | по мере необходимости | - | Договор |
| 5\_Сд\_7 | Подписание договора | Клиент | по мере необходимости | - | Договор |

**2.3. Детальное проектирование интерфейса**

Многооконный интерфейс является спецификацией для приложений, которые обрабатывают документы в Microsoft Windows. Спецификация описывает структуру окон и пользовательский интерфейс, который позволяет пользователю работать с несколькими документами внутри одного приложения.

Элемент интерфейса — это примитив графического интерфейса пользователя, имеющий стандартный внешний вид и выполняющий стандартные действия.

Windows Presentation Foundation (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 3 «Авторизация»

Изображение выглядит как текст, программное обеспечение, число, снимок экрана

Автоматически созданное описание

Рис. 4 «Заявки»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 5 «Клиенты»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 6 «Архив»

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рис. 7 «Добавление заявки»

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 8 «Добавление клиента»

Изображение выглядит как текст, снимок экрана, число, линия

Автоматически созданное описание

Рис. 9 «Добавление заявления в архив»

**2.4. Функциональная схема**

**Изображение выглядит как снимок экрана, текст, черно-белый, графический дизайн

Автоматически созданное описание**

Рис. 10 «Блок схема работы приложения»

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

Рис. 11 «Структура программы»

**3. Реализация**

**3.1. Обоснование выбора средств разработки**

Visual Studio — это полнофункциональная интегрированная среда разработки (IDE), которая подходит для создания программного обеспечения для различных платформ, включая .NET или .NET Core, iOS, Android через Xamarin и целевые объекты, поддерживающие C#

SQL Server Management Studio (SSMS). Бесплатная графическая среда для управления инфраструктурой SQL Server, разработанная компанией Microsoft. С помощью SSMS можно разрабатывать базы данных, выполнять инструкции T-SQL, а также администрировать SQL Server.

**3.2. Руководство системного программиста**

Системный программист - специалист, занимающийся разработкой, поддержкой и оптимизацией системного программного обеспечения, которое обеспечивает работу компьютерных систем и сетей. Он проектирует и внедряет драйверы, утилиты и другие компоненты, необходимые для корректного функционирования оборудования и приложений.

Основные задачи системного программиста включают:

* создание и обновление системного обеспечения;
* определение подходящих методов и инструментов для моделирования, анализа работы и построения цифровых механизмов;
* обеспечение надёжности, производительности и безопасности эксплуатации информационных систем;
* консультирование администрации по вопросам применения технических и цифровых средств;
* установка, настройка и оптимизация программного обеспечения;
* проектирование и внедрение цифровых средств;
* подключение, тестирование и замена внешних устройств;
* ведение баз данных;
* проведение антивирусных мероприятий;
* сопровождение договоров со сторонними организациями, предоставляющими услуги оснащения;
* администрирование внешних и внутренних информационных каналов;
* устранение аварийных ситуаций;
* обучение сотрудников основам информационной грамотности.

Проверка работоспособности функции «Авторизация» проводится следующим образом (таблица №4).

Таблица №4 «Проверка авторизации»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Запустить программу «Klient» после этого нужно ввести логин и пароль на страницы авторизации. | Успешный запуск программы |
| 2 | **Проверка отображения информации пользователю ПО**  После успешной авторизации мы переходим страницу «Заявки». | Отображение страницы заявки |

Проверка работоспособности функции «Добавление данных» проводится следующим образом (таблица №5).

Таблица №5 «Проверка добавления данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Нажимаем ПКМ по таблице и выбираем в контекстном меню функцию «Создать заявку». | Открытие окна с добавлением данных  Изображение выглядит как текст, Шрифт, снимок экрана  Автоматически созданное описание |
| 2 | **Проверка добавления информации**  После нажатия на кнопку и открытия окна «Добавления» вводим информацию в пустые строки ввода  Нажимаем кнопку «Добавить» | Успешный ввод данных в пустые поля, сохранение информации и её отображение в таблице с данными    Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание |

Проверка работоспособности функции «Редактирование данных» проводится следующим образом (таблица №6).

Таблица №6 «Проверка редактирования данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Нажимаем ПКМ по нужной записи и выбираем в контекстном меню функцию «Редактировать». | Открытие окна редактирование записей  Изображение выглядит как текст, снимок экрана, Шрифт  Автоматически созданное описание  Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт  Автоматически созданное описание |
| 2 | **Проверка редактирования информации**  После нажатия на кнопку и открытия окна редактирования записей  Изменяем информацию и нажимаем «Изменить» | Успешное изменение информации, сохранение и отображение информации в таблице с данными  Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание |

Проверка работоспособности функции «Удаление данных» проводится следующим образом (таблица №7).

Таблица №7 «Проверка удаления данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Нажимаем ПКМ по записи или нескольким записям в таблице данных и по нажимаем на кнопку «Удалить» | Успешное выделение записей и вывод сообщения об удалении данных |
| Продолжение таблицы №7 «Проверка удаления данных» | | |
| 2 | **Проверка удалении информации**  После нажатия на кнопку и открытия всплывающего сообщения  Подтверждаем удаление данных, нажав кнопку Да | Закрытие всплывающего сообщения и удаление данных из таблицы с данными  Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описаниеИзображение выглядит как снимок экрана, текст, Прямоугольник, линия  Автоматически созданное описание |

**3.3. Руководство программиста**

В дипломной работе используются такая технология как Entity Framework Core (EF Core) — это объектно-ориентированная, лёгкая и расширяемая технология доступа к данным от Microsoft. Она является частью платформы Entity Framework и разработана с учётом особенностей работы с данными в современных приложениях.

EF Core предоставляет разработчикам возможность работать с данными независимо от типа хранилища, используя универсальные API. Это позволяет упростить процесс разработки и повысить эффективность работы с данными.

Технология Entity Framework Core поддерживает множество различных систем баз данных, таких как SQL Server, MySQL, PostgreSQL, Oracle и другие. Она также совместима с различными платформами .NET, включая Windows Forms, консольные приложения, WPF, UWP и ASP.NET Core.

Использование Entity Framework Core позволяет разработчикам сосредоточиться на бизнес-логике своего приложения. Это делает процесс разработки более быстрым и эффективным, а также снижает вероятность ошибок, связанных с работой с базами данных.

Таблица №8 «Словарь данных»

|  |  |  |  |
| --- | --- | --- | --- |
| **ключ** | **поле** | **обязательное** | **примечание** |
| **Клиент** | | | |
| Первичный | IdKlient | Да | Идентификатор |
|  | FIO | Нет | ФИО клиента |
|  | Phone | Нет | Телефон |
| Продолжение таблицы №1 «Словарь данных» | | | |
| **Исполнитель** | | | |
| Первичный | IdPerformers | Да | Идентификатор |
|  | FIO | Нет | ФИО исполнителя |
|  | Email | Нет | Почта |
|  | Phone | Нет | Телефон |
| **Заявки** | | | |
| Первичный | IdApplications | Да | Идентификатор |
|  | DateEnd | Нет | Дата принятия заявки |
| Внешний | IdKlient | Нет | Внешний ключ к таблице клиент |
|  | Object | Нет | Адрес |
|  | IdStatus | Нет | Статус |
| Внешний | IdPerformers | Нет | Внешний ключ к таблице исполнитель |
|  | Kolvo | Нет | Количество |
|  | Opisanie | Нет | Описание |
|  | DateEnding | Нет | Дата работы |
| Внешний | IdRequest | Нет | Внешний ключ к таблице запрос |
| **Статус** | | | |
| Первичный | IdStatus | Да | Идентификатор |
|  | NameStatus | Нет | Статус |
| **Архив** | | | |
| Первичный | IdSuccess | Да | Идентификатор |
|  | FIOklient | Нет | Клиент |
|  | KolvoS | Нет | Количество |
|  | ObjectS | Нет | Адрес |
|  | OpisanieS | Нет | Описание |
|  | NameTypeS | Нет | Тип услуги |
|  | DateEndingS | Нет | Дата работы |
|  | DateEndS | Нет | Дата принятия заявки |
|  | StatS | Нет | Статус |
|  | NameTemaS | Нет | Услуга |
|  | FIOMast | Нет | Мастер |
|  | FIOper | Нет | Исполнитель |
| **Запрос (Regust)** | | | |
| Первичный | IdRequest | Да | Идентификатор |
| Внешний | IdTema | Нет | Внешний ключ к таблице услуга |
| Внешний | IdType | Нет | Внешний ключ к таблице тип услуги |
| Внешний | IdMaster | Нет | Внешний ключ к таблице мастер |
| **Услуга** | | | |
| Первичный | IdTema | Да | Идентификатор |
|  | NameTema | Да | Услуга |
|  | Pay | Да | Цена |
| **Мастер** | | | |
| Первичный | IdMaster | Да | Идентификатор |
|  | FIO | Да | ФИО |
|  | PhoneMat | Да | Телефон |
| **Тип услуги** | | | |
| Первичный | IdType | Да | Идентификатор |
|  | NameType | Да | Тип услуги |

**Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный

Автоматически созданное описание**

Рис. 12 «ER- диаграмма»

**3.4. Руководство пользователя**

При запуске программы (исполняемый «.exe» файл) открывается страница «Авторизация», на которой необходимо ввести логин и пароль, предоставленные вам разработчиком.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рис. 13 «Авторизация»

**Страница «Заявки»:**

После корректного введения логина и пароля, попадаем на страницу с перечнем заявок. Здесь представлена вся информация о текущих заявках, доступны переходы на другие страницы, возможность фильтрации по датам, есть поиск по услуге. Здесь же указано общее количество заявок в списке и контекстное меню с функциями: обновление, редактирование, архивация, удаление, создание договора и заявок, создание статистиков в Excel. Также доступны фильтры по типу услуги, названию услуге и статусу выполнения.

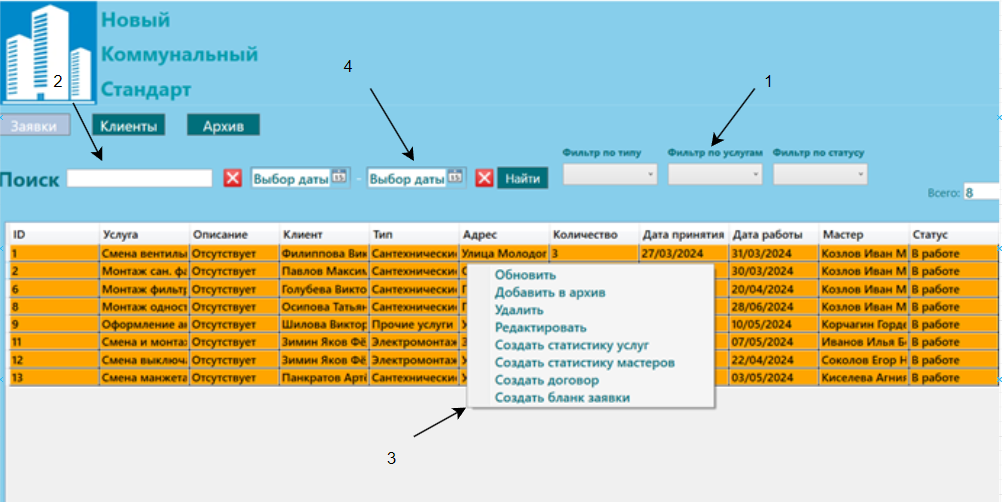


Рис. 14 «Заявки»

**Страница «Клиенты»:**

Страница «Клиенты» содержит полный список клиентов с их именами и номерами телефонов. Она также предлагает функцию поиска по ФИО клиента и позволяет переходить по другим страницам, отображает количество клиентов. Дополнительно, возможно добавление новых клиентов в базу. Через контекстное меню доступны операции редактирования, обновления, удаления, а также создание заявки для выбранного клиента.

Изображение выглядит как текст, снимок экрана, программное обеспечение, диаграмма

Автоматически созданное описание

Рис. 15 «Клиенты»

**Страница «Архив»:**

Страница «Архив» содержит список. Здесь представлена информация о выполненных заявках, доступны переходы на другие страницы, возможность фильтрации по датам, есть поиск по ФИО клиента. Тут указано общее количество заявок в списке и контекстное меню с функциями: обновление и удаление. Также доступны фильтры по типу услуги, услуге и статусу выполнения.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 16 «Архив»

**Фильтр:**

Чтобы задействовать фильтр по типу услуги, по услугам или по статусу нужно нажать на любой из выпадающих списков и выбрать нужный параметр (см.рис.14, п.1).

Если нужно использовать фильтр по датам нужно указать начальную и конечную дату. Затем нажать кнопку «Найти». (см.рис.14, п.4).

**Поиск:**

Для поиска нужной записи нужно введите фамилию, имя и отчество клиента в соответствующие поле (см.рис.14, п.2).

**Добавление клиента:**

Можно добавить клиента. Для этого нужно перейти на страницу «Клиенты» и нажать на зеленый крестик. Это приведет на страницу добавления клиента, где необходимо внести данные (см.рис.15, п.2).

**Обновить:**

Функция "Обновить" используется для обновления данных и проверки дат работ на их актуальность. Она предупреждает о недействительности записей. Если нужно его использовать надо вызвать контекстное меню используя правую кнопку мыши (ПКМ) и нажать на кнопку «Обновить» (см.рис.14, п.3).

**Добавление в архив:**

Чтобы использовать эту функцию, надо нажать на нужную запись и правой кнопкой мыши выбрать опцию «Добавить в архив». Это действие перенесет на страницу добавления (см.рис.14, п.3).

**Удалить:**

Функция «Удалить» позволяет избавиться от выбранных вами записей. Для ее использования необходимо правой кнопкой мыши нажать на нужную запись и затем нажать на кнопку «Удалить» (см.рис.14, п.3).

**Редактировать:**

Чтобы отредактировать запись, сначала выберите ее. Затем, чтобы вызвать контекстное меню, используйте правую кнопку мыши (ПКМ). Найдите и нажмите на кнопку «Редактировать». Это позволит внести изменения в выбранную запись (см.рис.14, п.3).

**Экспорт Договора по оказанию услуги или бланка заявки в Word:**

Чтобы отправить отчет в Word, выполните следующие действия:

1. Выделите нужную запись.

2. Вызовите контекстное меню, нажав правую кнопку мыши (ПКМ).

3. В появившемся меню выберите нужную опцию: «Создать договор» или «Создать заявку».

4. После выбора опции автоматически откроется Word с уже готовым шаблоном, где вы можете ввести необходимую информацию. (см.рис.14, п.3).

**Создание статистики в Excel:**

1. Вызовите контекстное меню, нажав правую кнопку мыши (ПКМ).

2. В появившемся меню выберите нужную опцию: «Создать статистику услуг» или «Создать статистику мастеров».

3. После выбора опции автоматически откроется Excel с уже готовым шаблоном. (см.рис.14, п.3).

**Создание заявки:**

Чтобы создать заявку, сначала выберите нужного клиента в системе. Затем кликните по его имени правой кнопкой мыши и затем нужно нажать на кнопку «Создать заявку». Это активирует контекстное меню, через которое перейдет на страницу создания заявки (см.рис.15, п.1).

Таблица №9 «Сообщения пользователю»

|  |  |
| --- | --- |
| Попытка входа в программу с неверными данными  Изображение выглядит как текст, снимок экрана, Шрифт, символ  Автоматически созданное описание | Сообщение об успешном входе в программу  . **Изображение выглядит как текст, снимок экрана, Шрифт, число  Автоматически созданное описание** |
| Попытка найти период времени с не указанными датами | Сообщение об успешном удалении. |
| Продолжение таблицы №9 «Сообщения пользователю» | |
| **Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание** | **Изображение выглядит как снимок экрана, текст, линия, Шрифт  Автоматически созданное описание** |
| Подтверждение удаления записи.  **Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание** | Сообщение об успешном редактировании заявления.  **Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание** |
| Сообщение о том, что такой клиент уже существует.  **Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание** | Попытка добавления записи в архив с неверным статусом.  **Изображение выглядит как текст, снимок экрана, символ, Шрифт  Автоматически созданное описание** |
| Успешное добавление записи в архив.  **Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание** | Успешное добавление нового клиента.  **Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание** |
| Сообщение о том, что дата работы уже прошло. | Сообщение о том, что нужно указать логин и пароль. |

**4.Тестирование и отладка**

**4.1. Тестовая документация**

Тестовая документация является неотъемлемой частью процесса тестирования программного обеспечения (ПО). Она обеспечивает системный и документированный подход к тестированию, гарантируя, что тестирование проводится всесторонне, эффективно и в соответствии с требованиями.

Тестовая документация включает в себя различные документы, которые описывают стратегии тестирования, тестовые сценарии, тестовые данные и ожидаемые результаты. Основными типами тестовой документации являются:

План тестирования: описывает цели, область, стратегию и график тестирования.

Тестовые сценарии: подробные инструкции о том, как выполнять конкретные тесты.

Тестовые данные: данные, используемые для выполнения тестов.

Ожидаемые результаты: результаты, которые должны быть получены при выполнении тестов.

Теория тестовой документации основана на следующих принципах:

Полнота: тестовая документация должна охватывать все аспекты тестируемого ПО.

Точность: документация должна быть точной и недвусмысленной.

Проверяемость: тестовые сценарии написаны таким образом, чтобы их можно было легко проверить.

Повторяемость: тесты должны быть воспроизводимыми, чтобы иметь возможность получить согласованные результаты при повторном выполнении.

Управляемость: тестовая документация должна быть организована и поддерживаться в актуальном состоянии.

**4.2. Отладка приложения**

Строка присваивает новое значение переменной \_currentItem

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

Рис. 17 «Ошибка»

Изображение выглядит как текст, электроника, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 18 «Исправление»

Участок кода с подсчётом записей, работающий некорректно:

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Рис. 19 «Ошибка»

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Рис. 20 «Исправление»

Участок кода с фильтром записей, работающий некорректно:

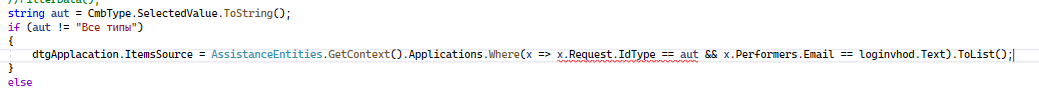


Рис. 21 «Ошибка»



Рис. 22 «Исправление»

Не выводит список услуг в Combox



Рис. 23 «Ошибка»



Рис. 24 «Исправление»

Тестирование программного обеспечения является важной и неотъемлемой частью процесса разработки программного обеспечения. Оно направлено на выявление и устранение дефектов, обеспечение качества и соответствия требованиям. Существует множество различных подходов и методов тестирования ПО.

Ручное тестирование предполагает, что тестировщик вручную выполняет различные сценарии тестирования, взаимодействуя с программой и проверяя ее функциональность, производительность, удобство использования и другие аспекты. Это позволяет охватить широкий спектр тестовых сценариев, но требует значительных временных и трудовых затрат.

Автоматизированное тестирование, напротив, подразумевает использование специальных инструментов и скриптов, которые выполняют тестовые сценарии без участия человека. Это позволяет повысить эффективность, скорость и точность тестирования.

Тестовый сценарий №1

|  |  |
| --- | --- |
| **Тестовый пример #** | Klient |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | На странице «Заявки» выберите нужную вам запись, нажмите ПКМ и нажмите «Создание договора». |
| **Краткое изложение теста** | Должен выводиться документ Word |
| **Этапы теста** | 1. Запустите программу через Visual Studio.  2. Попадаем на страницу «Авторизации».  3. Вводим логин и пароль от исполнителя  4. Выбираем нужную запись, нажимаем ПКМ и нажимаем «Создание договора». |
| **Тестовые данные** | Документ Word |
| **Ожидаемый результат** | Должен появиться документ Word с готовым шаблоном |
| **Фактический результат** | Выводит ошибку и пустой документ Word |
| **Статус** | Незачет. |
| **Предварительное условие** | Установите новую версию Word 2019. |
| **Постусловие** | Выдоится документ Word c готовым шаблоном |
| **Примечания/комментарии** | Пользователь должен обязательно выбрать нужную ему запись иначе будет шаблон с пустыми данными |

Тестовый сценарий №2

|  |  |
| --- | --- |
| **Тестовый пример #** | Klient |
| **Приоритет тестирования** | Низкое |
| **Заголовок/название теста** | На странице «Заявки» нажмите правую кнопку мыши по списку, нажимаем на «Фильтр по типу» и затем с выпадающего списка выбираем любой тип. |
| **Краткое изложение теста** | Должен выводиться список заявок с параметром который вы выбрали и количество записей будет изменяться. |
| **Этапы теста** | 1. Запустите программу через Visual Studio.  2. Попадаем на страницу «Авторизации».  3. Вводим логин и пароль от исполнителя  4. Нажимаем правую кнопку мыши по списку, далее нажимаем на «Фильтр по типу» и затем с выпадающего списка выбираем любой тип. |
| **Тестовые данные** | Заявки, количество записей. |
| **Ожидаемый результат** | Если вы выбрали тип, то должен выводиться список заявок с типом который вы выбрали и правильное количество записей. |
| **Фактический результат** | Список заявок с нужным типом выведен и правильно считает количество записей. |
| **Статус** | Зачет. |
| **Предварительное условие** | - |
| **Постусловие** | Список заявок с нужным параметром будет выводиться и количество записей |
| **Примечания/комментарии** | Пользователь может выбирать любые фильтры или сортировки. Программа должна правильно работать. |

**5. Расчет базовой себестоимости разрабатываемого продукта**

**5.1. Исходные данные**

Затраты времени и занятость специалистов соответствуют трудоемкости и сложности этапов создания приложения представлены в таблице № 5.

Таблица №10 «Исходные данные»

|  |  |
| --- | --- |
| **Этапы разработки программного продукта** | **Величина затрат рабочего времени, часы** |
| Анализ поставленной задачи | 6 |
| Проектирование интерфейса | 10 |
| Разработка и проектирование БД | 9 |
| Разработка интерфейса | 12 |
| Программирование | 16 |
| Оптимизация и рефакторинг | 20 |
| Тестирование | 23 |
| Отладка | 26 |
| Итого | 122 |

**5.2. Расчеты затрат на выполнение программы**

Создание компьютерной программы связанно со значительными затратами времени и соответственно большим удельным весом в себестоимости таких затратах, как оплата труда.

В работе по проектированию и разработке программы участвовал 1 специалист – разработчик с окладом – 21000 руб.

Расчет оплаты специалистов производиться исходя из дневной тарифной ставки.

Тарифная ставка — в трудовом праве фиксированный размер оплаты: труда работника за выполнение нормы труда определённой сложности за единицу времени.

Фонд оплаты труда- суммарные денежные средства, израсходованные в течение определённого периода времени на заработную плату, включая: стимулирующие оплаты и дополнительную оплату.

Расчёт чистых затрат на выполнение программы рассчитывается из затрат на заработную плату разработчику, умноженную на количество часов разработки и суммирование с прочими затратами на разработку, такие как электричество.

Затраты на заработную плату разработчику = 21000/(21\*8)\* 122= 15250 руб.

**5.3. Расчёт отчислений на социальное страхование и обеспечение**

Обычный размер ставки - для наемного работника - составляет 30 %. Пенсионный фонд Российской Федерации - 22 %

Фонд социального страхования - 2,9 %

Фонды обязательного медицинского страхования - 5,1 %

Итого: 30 %

Таблица №11 «Расчет отчислений на социальное страхование и обеспечение»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Должность работника** | **Пенсионный фонд РФ – ПФР (22%), руб.** | **Фонд социального страхования РФ –ФССРФ (2,9%), руб.** | **ФОМС (5,1%), руб.** | **Итого отчислений на социальное страхование и обеспечение (ОСО), руб.** |
| 1 | Разработчик | 3355 | 442,25 | 777,75 | 4555 |

Формулы расчета для таблицы №6:

ПФР = ОТ \* 22% = 15250 руб. \* 22% = 3355 руб.

ФСС РФ = ОТ \* 2,9% = 15250 руб. \* 2,9% = 442,25 руб.

ФОМС = ОТ \* 5,1% = 15250 руб. \* 5,1% = 777,75 руб.

ОСО = ПФР + ФССРФ + ФОМС = 3355 руб. + 442,25 руб. + 777,75 руб. = 4575 руб.

**5.4. Расчёт базовой себестоимости компьютерного продукта**

Себестоимость — это текущие затраты организации на производство и реализацию продукции, выраженные в денежной форме.

Себестоимость продукции является качественным показателем, так как она характеризует уровень использования всех ресурсов, находящихся в распоряжении организации.

Косвенные расходы — это затраты, связанные с производством и реализацией продукции (работ, услуг), которые нельзя напрямую отнести к изготовлению продукции (работ, услуг) и можно учесть в расходах в том периоде, когда они понесены.

К косвенным расходам можно отнести, например, затраты на услуги связи, аренду офиса и т.д.

Самое главное отличие прямых расходов от косвенных расходов в том, что сумма косвенных расходов в полном объеме относится к расходам текущего отчетного (налогового) периода, а прямых — к расходам текущего периода по мере реализации товаров, работ, то есть с учетом остатков незавершенного производства.

Исключением являются случаи, когда деятельность организации связана с оказанием услуг. Данные налогоплательщики вправе относить сумму прямых расходов отчетного (налогового) периода в полном объеме на уменьшение доходов от производства и реализации данного отчетного (налогового) периода без распределения на остатки незавершенного производства.

Относить ли расходы к прямым или косвенным расходам компания определяет отдельно для каждого производственного цикла.

Если те или иные ресурсы согласно технологическим регламентам не включены в производственный цикл, не являются его неотъемлемой частью, то затраты на них можно учитывать в составе косвенных расходов.

**Перечень косвенных расходов**

Косвенные расходы — это расходы, которые нельзя прямо отнести на конкретные виды продукции (работ, услуг).

К косвенным расходам относятся:

административно-управленческие расходы;

расходы на отопление и освещение помещений;

расходы на страхование;

расходы на содержание общехозяйственного персонала;

амортизационные отчисления и расходы на ремонт основных средств управленческого и общехозяйственного назначения;

арендная плата за помещения общехозяйственного назначения;

расходы по оплате информационных, аудиторских, консультационных и т. п. услуг;

расходы, связанные со сбытом продукции:

другие аналогичные по назначению управленческие расходы.

Таблица №12 «Расчёт базовой себестоимости»

|  |  |  |
| --- | --- | --- |
| **№** | **Статьи затрат** | **Сумма в рублях** |
| 1 | Фонд оплаты труда (ФОТ) | 19825 |
| 2 | Косвенные расходы (КР) 10% от ФОТ | 1982,5 |
| 3 | Расходы на продажу (РП) (ФОТ + КР) \* 10% | 2180,75 |
| 4 | Полная себестоимость (ПС) | 23988,25 |

Формулы расчета для таблицы №6:

ФОТ = ОТ + ОСО = 15250 руб. + 4575 руб. = 19825 руб.

В данной организации косвенные расходы составляют 10% от ФОТ.

КР = ФОТ \*10% = = 19825 руб. \* 10% = 1982,5 руб.

РП = (ФОТ + КР) \* 10% = (19825 руб. + 1982,5 руб.) \* 10% =2180,75 руб.

ПС = ФОТ + КР + РП = 19825 руб. + 1982,5 руб. + 2180,75 руб. = 23988,25 руб.

Итоговая стоимость разработки приложения для технического администратора организации ООО «Новый Коммунальный Стандарт» по приему заявок от клиентов составит 23988,25 р

**Заключение**

В рамках данной работы проведено комплексное исследование бизнес-процессов организации «Новый Коммунальный Стандарт было выявлено, что автоматизация работы технического администратора может значительно увеличить эффективность его деятельности.

В результате разработки программного продукта «Управление заявками» была создана база данных на SQL Server, приложение на Visual Studio.

Программное обеспечение «Управление заявками» предоставляет техническому администратору ряд преимуществ, способствующих повышению эффективности его работы.

Централизованное управление рабочим процессом: программное обеспечение обеспечивает единую платформу для создания, обработки и отслеживания заявок, что позволяет техническому администратору эффективно управлять потоками задач, распределять обязанности между мастерами и контролировать сроки выполнения.

Улучшение взаимодействия с клиентами: программное обеспечение «Управление заявками» позволяет создавать единые профили пользователей, собирать информацию о истории их обращений, что позволяет предоставлять более персонализированную и эффективную поддержку.

Сортировка и фильтрация данных: возможность сортировать и фильтровать информацию по различным критериям (статус, тип заявки, дата создания) позволяет администраторам быстро находить необходимую информацию и оперативно реагировать на запросы пользователей, сокращая время обработки заявок.

Анализ производительности и отчетность: возможность создания статистических отчетов в формате Excel предоставляет ценную информацию о динамике работы отдела IT-поддержки и создание договоров по оказанию услуги и бланка заявки, позволяя оптимизировать процессы.

Прозрачность и контроль: выгрузка договоров и заявок на печать обеспечивает четкое документирование всех действий и взаимодействий, повышает прозрачность работы и облегчает контроль со стороны руководства.

В результате использования программного обеспечения «Управление заявками» технические администраторы получают следующие выгоды: повышение производительности, улучшение качества обслуживания, упрощение документооборота.

В целом, программное обеспечение «Управление заявками» является инструментом для технических администраторов, позволяющим им эффективно управлять рабочим процессом. Это инструмент, который не только повышает производительность работы, но и способствует достижению стратегических целей организации.

**Список литературы**

1. ГОСТ 34.602-89 Автоматизированные системы управления. Техническое задание на создание автоматизированной системы.
2. ГОСТ 19.503-79 Единая система программной документации. Технический проект.
3. ГОСТ 19.504-79 Единая система программной документации. Рабочая документация.
4. ГОСТ 19.505-79 Единая система программной документации. Текст программы.
5. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программного обеспечения.
6. ГОСТ Р 51232-2019 Информационные технологии. Технологический процесс разработки и сопровождения программного обеспечения

**Приложения**

Приложение 1 «Договор»

Изображение выглядит как текст, письмо, Шрифт, снимок экрана

Автоматически созданное описание

Приложение 2 «Бланк заявки»

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

Приложение 3 «Статистика услуг»

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

Приложение 4 «Статистика мастеров»

**Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание**

Код страницы «Заявки»

public PageApplication()

{

InitializeComponent();

// Получаем данные из свойства приложения

string data = (App.Current as App).Data;

// Устанавливаем данные в TextBox

loginvhod.Text = data;

string login = data;

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(z => z.Performers.Email == login).ToList();

System.Int32 customerCount = AssistanceEntities.GetContext().Applications.Where(z => z.Performers.Email == login).Count();

countzapicei.Text = customerCount.ToString();

var listform = AssistanceEntities.GetContext().Type.Select(x => x.NameType).Distinct().ToList();

var listform2 = AssistanceEntities.GetContext().Status.Select(x => x.NameStatus).Distinct().ToList();

var listform3 = AssistanceEntities.GetContext().Tema.Select(x => x.NameTema).Distinct().ToList();

CmbType.Items.Add("Все типы");

foreach (string formm in listform)

{

CmbType.Items.Add(formm);

}

CmbStatus.Items.Add("Все статусы");

foreach (string formm in listform2)

{

CmbStatus.Items.Add(formm);

}

CmbUslugi.Items.Add("Все услуги");

foreach (string formm in listform3)

{

CmbUslugi.Items.Add(formm);

}

loginvhod.Visibility = Visibility.Hidden;

}

private void MenuUpdate\_Click(object sender, RoutedEventArgs e)

{

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).ToList();

System.Int32 customerCount = AssistanceEntities.GetContext().Applications.Where(z => z.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = customerCount.ToString();

DateTime currentDate = DateTime.Now;

foreach (Applications row in dtgApplacation.Items)

{

DateTime workDate = Convert.ToDateTime(row.DateEnding);

TimeSpan difference = currentDate - workDate;

int daysDifference = difference.Days;

if (daysDifference > 10)

{

MessageBox.Show($"ID {row.IdApplications} Дата работы давно прошла, рекомендуется обновить информацию.", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

}

private void MenuDel\_Click(object sender, RoutedEventArgs e)

{

var salesForRemoving = dtgApplacation.SelectedItems.Cast<Applications>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {salesForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

AssistanceEntities.GetContext().Applications.RemoveRange(salesForRemoving);

AssistanceEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).ToList();

System.Int32 customerCount = AssistanceEntities.GetContext().Applications.Where(z => z.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = customerCount.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void dateselect\_Click(object sender, RoutedEventArgs e)

{

if (Picker1.SelectedDate != null && Picker2.SelectedDate != null)

{

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.DateEnd >= Picker1.SelectedDate && x.DateEnd <= Picker2.SelectedDate && x.Performers.Email == loginvhod.Text).ToList();

int count = AssistanceEntities.GetContext().Applications.Where(x => x.DateEnd >= Picker1.SelectedDate && x.DateEnd <= Picker2.SelectedDate && x.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = count.ToString();

}

else

{

MessageBox.Show("Укажите даты");

return;

}

}

private void findtema\_TextChanged(object sender, TextChangedEventArgs e)

{

if (findtema.Text != "")

{

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Request.Tema.NameTema.ToLower().Contains(findtema.Text.ToLower()) && x.Performers.Email == loginvhod.Text).ToList();

int count = AssistanceEntities.GetContext().Applications.Where(x => x.Request.Tema.NameTema.ToLower().Contains(findtema.Text.ToLower()) && x.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = count.ToString();

}

else

{

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).ToList();

System.Int32 customerCount = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = customerCount.ToString();

}

}

private void CmbType\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string aut = CmbType.SelectedValue.ToString();

if (aut != "Все типы")

{

int cod = AssistanceEntities.GetContext().Type.First(x => x.NameType == aut).IdType;

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Request.IdType == cod && x.Performers.Email == loginvhod.Text).ToList();

int count = AssistanceEntities.GetContext().Applications.Where(x => x.Request.IdType == cod && x.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = count.ToString();

}

else

{

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).ToList();

int count1 = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = count1.ToString();

}

}

private void CmbMaster\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

string aut = CmbUslugi.SelectedValue.ToString();

if (aut != "Все услуги")

{

int cod = AssistanceEntities.GetContext().Tema.First(x => x.NameTema == aut).IdTema;

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Request.IdTema == cod && x.Performers.Email == loginvhod.Text).ToList();

int count = AssistanceEntities.GetContext().Applications.Where(x => x.Request.IdTema == cod && x.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = count.ToString();

}

else

{

dtgApplacation.ItemsSource = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).ToList();

int count1 = AssistanceEntities.GetContext().Applications.Where(x => x.Performers.Email == loginvhod.Text).Count();

countzapicei.Text = count1.ToString();

}

}