ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «БУГУЛЬМИНСКИЙ МАШИНОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ»

|  |  |
| --- | --- |
|  | Допущен к защите  Зам.директора по УР  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.В. Жакупова  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

**ВЫПУСКНАя КВАЛИФИКАЦИОННАЯ РАБОТА**

|  |
| --- |
| **ВКР.09.02.05.747.008** |
|  |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

|  |  |
| --- | --- |
| Выполнил студент  \_\_\_\_\_\_\_\_\_\_\_(Д.И. Казаков)  (ФИО) | Руководитель:  \_\_\_\_\_\_\_\_\_\_\_(Л.И. Дмитриева)  (ФИО) |
| Консультанты  \_\_\_\_\_\_\_\_\_\_\_(Е.М. Рамазанова)  (ФИО)  \_\_\_\_\_\_\_\_\_\_\_(М.М. Зайнутдинова)  (ФИО) | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Председатель ГЭК |
| Нормоконтролёр  \_\_\_\_\_\_\_\_\_\_\_( М.М. Зайнутдинова )  (ФИО) |
| \_\_\_\_\_\_\_\_\_\_\_(С.А. Мех)  (ФИО) «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |

Бугульма 2021 г.

Содержание

[Введение 4](#_Toc10375611)

[1 Теоретический раздел 7](#_Toc10375612)

[1.1 Обследование предприятия 7](#_Toc10375613)

[1.2 Состояние проблемы 7](#_Toc10375614)

[1.3 Методы решения проблемы 12](#_Toc10375615)

[1.4 Обоснование выбранного метода 13](#_Toc10375616)

[2 Раздел программной документации 17](#_Toc10375617)

[2.1 Спецификация 18](#_Toc10375618)

[2.2 Техническое задание 19](#_Toc10375619)

[2.3 Описание программы 25](#_Toc10375621)

[2.4 Текст программы 32](#_Toc10375622)

[2.5 Руководство программиста 47](#_Toc10375623)

2.6 [Руководство оператора 53](#_Toc10375624)

[3 Расчётно-экономический раздел 63](#_Toc10375625)

[3.1 Расчёт амортизационных отчислений 63](#_Toc10375626)

[3.2 Расчет себестоимости разработки 65](#_Toc10375627)

[3.3 Затраты на электроэнергию 67](#_Toc10375628)

[3.5 Расчет цены программы 68](#_Toc10375629)

[3.6 Оценка экономической эффективности 69](#_Toc10375630)

[4 Организация техники безопасности и охраны труда 71](#_Toc10375631)

[4.1 Общие положения 71](#_Toc10375632)

[4.1 Требования к помещениям для эксплуатации ПК 73](#_Toc10375633)

[4.2 Требования к организации и оборудованию рабочих мест с ПК 75](#_Toc10375634)

[4.3 Требования к организации режима работы оператора ПК 76](#_Toc10375635)

[Заключение 78](#_Toc10375636)

[Приложение А 79](#_Toc10375637)

[Приложение Б 80](#_Toc10375638)

[Список использованных источников 81](#_Toc10375639)

[Ссылочные нормативные документы 82](#_Toc10375640)

# Введение

На сегодняшний день любая организация на определенной стадии своего развития сталкивается с вопросом о необходимости автоматизации. Предприятия, решившие внедрить систему автоматизированного учета своей деятельности, руководствуются желанием упростить уже существующий комплекс операций учета за счет оптимизации документооборота и сокращения трудозатрат персонала компании. Профессиональная система автоматизации является надежным инструментом, который необходим для постоянного контроля и отслеживания работы любой компании. Она позволяет быстро и качественно обслужить клиентов, оптимизировать работу с поставщиками, наладить прозрачный управленческий, бухгалтерский и финансовый учет, эффективно построить работу с персоналом, вести гибкую дисконтную политику, своевременно принимать ответственные решения и многое другое.

При выборе системы автоматизации необходимо сформулировать требования к ней и сравнить имеющиеся на рынке продукты. Далее следует оценить потенциал компании-разработчика, ведь система автоматизации приобретается не на один год, и в этом случае нужно выбрать надёжного партнёра. Важно приобрести комплексную систему управления. Недостаточно только функций автоматического учёта и контроля. Система должна содержать ряд готовых бизнес-процессов, необходимых для эффективного и результативного управления предприятием. Сегодня можно с уверенностью говорить о том, что автоматизация – это не дань моде, а объективная необходимость, и среди предпринимателей IT-бизнеса есть полное понимание, на чём можно сэкономить, а без чего невозможно организовать стабильную работу компании.

Ремонт компьютеров и комплектующих, мобильных телефонов и т.д. – это производственный процесс. Внедрение автоматизированной системы управления этим процессом ведёт к повышению его эффективности, а следовательно, к увеличению прибыльности компании в целом.

В качестве темы выпускной квалификационной работы (далее ВКР) была выбрана тема «Разработка, внедрение и адаптация информационной системы сервисного центра по ремонту бытовой техники» для ООО «Компьютерные системы». Выбор темы по заведению и учёту заказов в сервисной мастерской связан с тем, что основная причина автоматизации сервисного центра – это улучшение качества обслуживания, чтобы больше зарабатывать. Когда все процессы налажены и упрощены, можно больше времени тратить на развитие и привлечение клиентов

Данное программное обеспечение по учёту заказов на ремонт в сервисном центре предназначена для решения задач:

1. Учёта заказов на ремонт.
2. Формирования отчётов.

Программный продукт автоматизирует технологический процесс ведения заказов на ремонт, чтобы обеспечить работу самого сервисного центра, сократить временные затраты по учёту заказов на ремонт и обслуживание, а также составления отчётов по выполненным работам.

Объектом исследования выступает компания ООО «Компьютерные системы».

Предметом изучения является составление и учёт заказов на ремонт и обслуживание.

Целью выпускной квалификационной работы является автоматизировать процесс сервисной компании по ремонту бытовой техники в ООО «Компьютерные системы» к 15 июня 2020 года.

Для реализации данной цели были поставлены следующие задачи:

* ознакомиться с объектом исследования ООО «Компьютерные системы»;
* основательно изучить предмет исследования по составлению заказов на ремонт техники, а также выявить недостатки ведения сервисного документооборота;
* внести предложения по совершенствованию заведения заявок в сервисной компании ООО «Компьютерные системы».

ВКР состоит из введения, четырёх основных разделов, заключения, списка использованных источников и приложения.

В первом разделе рассматривается изучаемая предметная область и оценивается существенное состояние автоматизированной области. Было проведено изучение характеристик предприятия, а также характеристика его подразделений и видов деятельности. Производилось обоснование необходимости автоматизации и постановка задачи, а также анализ существующих разработок.

Во втором разделе составляется программная документация на данный программный продукт по учёту книжного фонда в библиотеке. Также разрабатывается документация по руководству программиста и оператора.

Третий раздел посвящен экономическим расчётам. В этом разделе рассчитывается заработная плата для программиста-разработчика, сумма амортизационных отчислений, затраты на электроэнергию и затраты на разработку самого программного обеспечения.

В последнем, четвёртом разделе, описывается техника безопасности и охрана труда на предприятии.

В заключении приедены результаты выполнения работы.

В приложении вынесена схема потоков данных и алгоритм работы программы.

ВКР состоит из 84 страниц, 22 рисунка, 14 формул, 17 таблиц и 2 приложения.

# 1 Теоретический раздел

## Обследование предприятия

Деятельность «Компьютерных систем» прежде всего, направлена на обслуживание частных лиц и небольших организаций, не имеющих для этого нужных навыков (специалистов), оборудования или времени. На начальный период деятельности фирмы планируется предоставлять следующий перечень услуг:

* ремонт и обслуживание не гарантийных компьютеров и периферии.

Смысл такой услуги, как ремонт, понятен и не требует описания. Под обслуживанием понимается: плановые профилактические работы, настройка аппаратной части, обновление микропрограмм BIOS, чистка, смазка вентиляторов и кулеров, замена, заправка и ремонт картриджей для струйных, лазерных и матричных принтеров, а также копировальных аппаратов.

* настройка программного обеспечения.

Не каждый пользователь персонального компьютера в состоянии грамотно установить и (или) настроить то или иное программное обеспечение. Фирма предлагает решение данной проблемы специалистами за невысокую плату. Возможна как установка программного обеспечения заказчика, так и нашего. В последнем случае стоимость возрастает на цену лицензии, если требуемое программное обеспечение не является бесплатным.

Дополнительные услуги:

* компьютерный набор и верстка.

Достаточно распространенной проблемой для частного лица (и не только) является необходимость преобразования рукописного текста в электронный вид. При этом не каждый имеет необходимую технику, свободное время и навыки быстрой работы с клавиатурой. Компьютер-сервис предлагает решение этой проблемы в максимально короткие сроки. Также предлагается верстка – подготовка набранного текста для печати в любом стиле на любом формате носителя.

* сканирование, распознавание и ксерокопия.

Предлагается сканирование любых графических материалов на формате до А4 включительно в любом качестве. Кроме того предоставляется услуга сканирования печатного текста с последующим распознаванием (переводом в электронный вид). Ксерокопия осуществляется без использования копировального аппарата – для этого оригинал сканируется и печатается на лазерном принтере.

* распечатка на лазерном принтере (монохромная).

Название говорит само за себя. Распечатка возможна как материала, принесенного заказчиком, так и набранного или отсканированного нашей фирмой.

* печать цифровых фотографий и цветных материалов.

Осуществляется на специальном струйном фотопринтере с применением специализированной бумаги повышенного качества. Качество отпечатка не ниже, чем у фотографии с обычной пленки, полученной в мини фотолаборатории. Возможна распечатка как с электронного носителя, принесенного клиентом, так и перепечатывание оригинала после его сканирования и редактирования.

* запись CD - R и CD - RW дисков.

Для надежного резервного хранения информации в последнее время выгодно использовать записываемые и перезаписываемые компакт диски. Но для этого требуется специальные устройства, которые пока недостаточно распространены. «Компьютерные системы» предлагает запись и перезапись информации заказчика на компакт диски. Предполагается использование, как дисков клиента, так и наших компактов. В последнем случае стоимость услуги несколько снижается.

* поиск в Internet определенной тематической информации.

Данная услуга предполагает получение из Internet информации, располагающейся по определенному адресу, который предоставляет клиент. Более сложная разновидность этого сервиса – поиск информации по определенной тематике. Например, книг, статей, рефератов, программ и так далее. Возможен поиск в присутствии клиента под его руководством.

* разработка web сайтов.

«Компьютерные системы» предлагает создание Web-страничек различной степени сложности для организаций и частных лиц. Также при желании клиента возможно получение доменного имени любого уровня.

* написание несложных программных продуктов.

Нередко организации или частному лицу требуется получить некое несложное программное средство, которое невозможно приобрести, например, потому что его не существует в готовом виде. Единственный выход в такой ситуации – написать такую программу на заказ, что можно осуществить, обратившись в «Компьютерные системы».

* бесплатная помощь в приобретении компьютеров, периферии и другой сложной техники, связанной с информационными технологиями.

Приобретать сложную дорогую технику желательно с профессионалом в данной области, если, конечно, покупатель сам не является таковым. В противном случае даже при учете честного продавца некомпетентный покупатель рискует приобрести неудачный товар. Наша фирма поможет решить эту проблему абсолютно бесплатно для клиента. При этом доход образуется в виде комиссионных с фирмы-продавца.

Все вышеописанные 12 услуг можно разделить на 3 вида деятельности, которые подлежат лицензированию:

1. Ремонт и обслуживание компьютерной техники;
2. Работа с информацией;
3. Разработка программных продуктов.

## 1.2 Состояние проблемы

В последнее время информационные технологии стали неотъемлемой частью нашей жизни. Экономические информационные системы, связанные с представлением и обработкой информации для всех уровней управления экономическими объектами, приобретают особую важность в общественной жизни. На данный момент невозможно представить какую-либо организацию, не применяющую компьютерных технологий. Это обусловлено и тем, что государственные структуры требуют обязательных отчётов в электронном виде, следовательно, необходима систематизированная информация.

Документооборот является важнейшей составной частью при оформлении документов, так как ошибки отрицательно сказываются на операциях учёта сервисного центра.

Исходя из современных требований, предъявляемых к качеству работы предприятия, нельзя не отметить, что эффективная работа его всецело зависит от уровня оснащения компании информационными средствами.

Таким образом, для автоматизации и облегчения рутинной работы при составлении отчётов по заказам на ремонт, принимаются программы, которые позволяют хранить информацию о технологических процессах и использования его в нужных целях.

Актуальность данной темы заключается в том, что создание информационной системы для автоматизации деятельности сервисного центра, можно рассматривать как неотъемлемый элемент развития данного предприятия, что способствует упрощению работы обслуживающего персонала и более ускоренный процесс обслуживания цифровой техники.

Целью создания программного продукта по созданию заявок в сервисном центре является автоматизация заведения заявок на ремонт и их обслуживание, которая позволит улучшить эффективность выполнения процессов, происходящих в сервисном центре.

Разрабатываемое приложение содержит следующую информационную структуру:

1. Входные данные:

* данные о клиентах;
* виды работ (услуг);
* заказы клиентов.

1. Выходные данные:

* прайс-лист на услуги и работы;
* договор на выполнение работ сервисным центром с перечнем выполняемых работ;
* акт выполненных работ;
* отчёты по выполненным работам.

## 1.3 Методы решения проблемы

Процесс создания автоматизированной информационной системы по созданию заявок в сервисном центре включает в себя следующие этапы постановки задач:

Таблица 1 – Основные этапы создания программы

|  |  |  |
| --- | --- | --- |
| № | Задачи | Количество дней |
| 1 | Изучение предметной области, постановка задачи, определение целей и технологий разработки | 1 |
| 2 | Написание формального технического задания на автоматизированную систему | 2 |
| 3 | Создание графического интерфейса для всех вкладок программы | 3 |
| 4 | Написание основного функционала программы | 9 |
| 5 | Наполнение базы данных программы контентом | 1 |
| 6 | Тестирование программы на предприятии, с целью выявления ошибок в корректности работы | 1 |
| 7 | Исправление и доработка основного функционала программы | 2 |
| 8 | Обучение сотрудников пользованию программой | 1 |
| 9 | Внедрение программы на производство | 1 |
| Итого | | 21 |

Программа по созданию заявок на ремонт бытовой техники позволяет вести создавать заявку на ремонт или обслуживание той или иной техники по прайс-листу. Клиент, обращаясь в мастерскую, сдаёт ту технику, которую необходимо отремонтировать или провести диагностику. Администратор программы заводит клиента в базу данных и создаёт на его имя заказ, после распечатывает ему акт приёмки заказа. Программное обеспечение оснащено клиентской частью, поэтому клиент может удалённо видеть, в каком статусе находится его заказ. Когда он видит, что статус «Готов», он может подходить в фирму и забирать отремонтированную технику.

Таблица 2 – Основные характеристики программы

|  |  |  |
| --- | --- | --- |
| № | Назначения | Наличие в программе |
| 1 | Наличие удобного пользовательского интерфейса | Да |
| 2 | Наличие адаптируемого дизайна программы | Да |
| 3 | Платный/бесплатный | Бесплатный |
| 4 | Совместимость с аппаратным и программным обеспечением компьютера | Да |
| 5 | Кроссплатформенный | Нет |
| 6 | Наличие лицензии на программный продукт | Да |
| 7 | Удовлетворяет потребностям заказчика | Да |
| 8 | Умение программы формировать отчёт | Да |

## 1.4 Обоснование выбранного метода

Microsoft Visual Studio – это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом.

Функциональная структура среды включает в себя:

* редактор исходного кода, который включает множество дополнительных функций, как автодополнение IntelliSense, рефракторинг кода;
* отладчик кода;
* редактор форм, предназначенный для упрощённого конструирования графических интерфейсов;
* веб-редактор;
* дизайнер классов;
* дизайнер схем баз данных.

Преимущества IDE-среды Visual Studio:

* [пересмотренные основы](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#redefined-fundamentals). Новые возможности настройки позволяют сократить время установки и выполнять установку любого компонента в любое время;
* [быстродействие и производительность](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#performance-and-productivity). Мы уделили внимание новым и современным возможностям по разработке мобильных, облачных и классических приложений. Теперь Visual Studio запускается и реагирует быстрее, а также использует меньше памяти;
* [разработка облачных приложений с помощью Azure](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#cloud-app-development-with-azure). Встроенный набор инструментов Azure позволяет без проблем создавать ориентированные на облако приложения на базе Microsoft Azure.Visual Studio упрощает настройку, сборку, отладку, упаковку и развертывание приложений и служб в Azure;
* [разработка приложений для Windows](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#windows-app-development). Используются шаблоны универсальной платформы Windows в Visual Studio 2017, чтобы разработать единый проект для всех устройств под управлением Windows 10 – персональных компьютеров, планшетов, телефонов, игровых консолей Xbox, очков HoloLens, Surface Hub и многих других;
* [разработка мобильных приложений](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#mobile-app-development). Совершенствуются проекты и получаются результаты быстрее с Xamarin, который объединяет многоплатформенные требования к мобильности, используя одноядерную базу кода и набор навыков;
* [кроссплатформенная разработка](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#cross-platform-development). Без проблем доставляются программные обеспечения для любой целевой платформы. Распространяются процессы DevOps на SQL Server с помощью средств работы с данными Redgate и безопасно автоматизируются развертывания базы данных в Visual Studio. Или используются .NET Core для создания приложений и библиотек, которые в неизменном виде выполняются в операционных системах Windows, Linux и macOS;
* [разработка игр](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#games-development). Средства Visual Studio для Unity (VSTU) позволяют использовать Visual Studio для создания сценариев игр и редакторов на языке C#, а затем использовать его мощный отладчик для поиска и исправления ошибок;
* [разработка для сценариев ИИ](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017#ai-development)[[1]](#footnote-1). Благодаря инструментам Visual Studio Tools for AI получаются эффективные средства для оптимизации инновационных разработок в сфере искусственного интеллекта на основе среды Visual Studio. Решения для создания, тестирования и развертывания ИИ и глубинного обучения легко интегрируются с машинным обучением Azure и предоставляют широкие возможности для экспериментов.

C# – является последним из широко распространенных языков программирования, впитавший в себя весь имеющийся опыт и вобравший лучшие стороны существующих языков программирования, при этом является специально созданным для работы в NET. Сама архитектура NET продиктовала ему объектно-ориентированную направленность.

В Visual Studio выделаются такие особенности, как возможность объявлять несколько классов в одном файле, из чего следует синтаксическая поддержка иерархической системы пространств имен. Из вещей, включенных в спецификацию языка, но не являющихся чисто "программистскими" необходимо отметить возможность использования комментариев в формате XML. Если комментарии отвечают специально описанной структуре, компилятор по ним может сгенерировать единый XML-файл документации.

Архитектурой проекта могут определяться локальные атрибуты, которые будут связанны с любыми элементами языка – классами, интерфейсами и т.д.

Преимущества среды разработки Microsoft Visual Studio 2017 Community:

* 1. Бесплатная;
  2. Кроссплатформенная;
  3. Объектно-ориентированный;
  4. Наличие всплывающих подсказок в коде;
  5. Удобный графический интерфейс.

Для разработки модели базы данных использовалась СУБД[[2]](#footnote-2) Access 2010. Приложение Microsoft Access 2010 исключительно удобно в использовании благодаря готовым шаблонам и эффективным средствам, которые сохраняют свою возможность по мере увеличения объёма данных.

Access 2010 позволяет эффективно работать с данными даже пользователям, не являющимися специалистами по базам данных. Кроме того, новые веб-базы данных расширяют возможности приложения Access, упрощая отслеживание сведения, их совместное использование и составление отчётов.

Преимущества СУБД Microsoft Access 2010:

* простой и понятный интерфейс;
* СУБД MS Access общеизвестна и очень популярна в настольных ПК;
* безграничные возможности экспорта данных, в том числе и хранение самой базы в MySQLServer;
* невысокая цена;
* широкий выбор конструкторов для построения форм, отчётов и запросов;
* широкие возможности импорта данных;
* возможность создания пароля в базе данных;
* редактор SQL.

Таким образом, для создания выпускной квалификационной работы была выбрана среда разработки Visual Studio 2017 Community, язык написания C# и приложение для разработки модели базы данных Microsoft Access 2010.

**2** **Раздел программной документации**

# Спецификация

046.БМТ.09.02.05.747.008

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Примечание |
|  |  |  |
|  | Документация |  |
|  |  |  |
| 046. БМТ.09.02.05.747.008.90 | Техническое задание |  |
| 046. БМТ.09.02.05.747.008.12 | Текст программы |  |
| 046. БМТ.09.02.05.747.008.13 | Описание программы |  |
| 046. БМТ.09.02.05.747.008.33 | Руководство программиста |  |
| 046. БМТ.09.02.05.747.008.34 | Руководство оператора |  |
|  |  |  |
|  | Компоненты |  |
|  |  |  |
| Setup «k-system» | Приложение для установки программного обеспечения по составлению заказов в сервисном центре |  |
| k-system.mdb | База данных |  |
| Справка.chm | Руководство пользователю |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Министерство образования и науки Республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
|  | Утверждаю  Руководитель работы  Дмитриева Л.И.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Техническое задание**

Лист утверждения

**046.БМТ.09.02.05.747.008.90.01-1-ЛУ**

|  |  |
| --- | --- |
| Нормоконтролер  Зайнутдинова М.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. | Исполнитель  Казаков Д.И.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

2021 г.

Министерство образования и науки республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
| **УтверждеН**  **046.БМТ.09.02.05.747.008.90.01-ЛУ** |  |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Техническое задание**

**046. БМТ.09.02.05.747.008.90**

**Листов 3**

2021 г.

## Аннотация

В техническом задании на разработку программного обеспечения по заведению заявок на ремонт бытовой техники в сервисном центре описываются, какие задачи были поставлены во время проектирования программного продукта «k-system», для чего она предназначена, а также имеющиеся входные и выходные данные при работе с программой.

Для создания программного продукта по заведению заявок на ремонт бытовой техники в сервисном центре, необходимо разработать следующий функционал:

* разграничение прав доступа;
* возможность распечатки акта приёмки товара на ремонт и акта выдачи оборудования;
* возможность общаться с администратором системы по электронной почте;
* получение информации клиентом, о статусе его работы.

Входные данные:

* данные о клиентах;
* виды работ (услуг).

Выходные данные:

* прайс-лист на услуги и работы;
* акт приёмки оборудования на ремонт;
* акт выполненных работ;
* отчёты по выполненным работам.

Учётные записи имеют следующие привилегии:

* 1. Раздел администрирования:
* заведение клиента в базу;
* создание заказа клиента;
* заведение вида работы и типа ремонта в прайс-лист;
* отчётность по заказам на клиента;
* общение с клиентом по электронной почте.

2. Раздел сотрудника:

* просмотр прайс-листа услуг сервисной компании;
* просмотр заказа;
* общение с администратором по электронной почте.

Важным критерием при создании программного продукта является надёжность и безопасность всей информации, которая будет содержаться в программе.

Требования к ПК для эксплуатации программного продукта по учёту книжного фонда в библиотеке:

* поддержка операционной системы Windows 7/8/8.1/10;
* оперативная память 1Гб;
* свободное место на жёстком диске 1 Гб;
* видеоадаптер, совместимый с Direct 9 и поддерживающий разрешение экрана 1024×768 точек и выше;
* клавиатура;
* мышь.

Необходимое программное обеспечение для бесперебойной работы программного продукта по заведению заявок на ремонт бытовой техники:

* Microsoft Office Access версии 2010 – СУБД, на которой создана база данных;
* Microsoft Office Word версии 2010 – для печати акта приёмки и акта сдачи заказа;
* NET. Framework версии 4.6.1 – для корректной работы программы.

При вводе в эксплуатацию программного обеспечения по заведению заявок на ремонт бытовой техники программа должна иметь возможность формировать заказ на определённого клиента, печати актов на приём заказов, актов на выдачу заказов, составлению отчётности по заказам.

Для планирования сроков сдачи проекта, был составлен календарный план, в котором представлены все задачи на программный продукт и временные рамки выполнения работ.

Таблица 3 – Календарный план проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Задачи | Дата начала | Дата конца | Продолжительность (дни) |
| 1 | Составление технического задания | 18.05.2021 | 19.05.2021 | 1 |
| 2 | Разработка алгоритма программы | 18.05.2021 | 21.05.2021 | 3 |
| 3 | Разработка интерфейса программы | 20.05.2021 | 21.05.2021 | 1 |
| 4 | Разработка базы данных | 21.05.2021 | 22.05.2021 | 1 |
| 5 | Написание функционала | 22.05.2021 | 30.05.2021 | 8 |
| 6 | Тестирование | 30.05.2021 | 31.05.2021 | 1 |
| 7 | Исправление ошибок | 31.05.2021 | 01.06.2021 | 1 |
| 8 | Составление документации на проект | 01.06.2021 | 02.06.2021 | 1 |
| 9 | Внедрение программы на предприятие | 02.06.2021 | 03.06.2021 | 1 |
| 10 | Обучение людей | 03.06.2021 | 04.06.2021 | 1 |
| 11 | Сопровождение проекта | 04.06.2021 | 06.06.2021 | 2 |
| Итого | | | | 21 |

Министерство образования и науки Республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
|  | Утверждаю  Руководитель работы  Дмитриева Л.И.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_2020 г. |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Описание программы**

Лист утверждения

**046.БМТ.09.02.05.747.008.13.01-1-ЛУ**

|  |  |
| --- | --- |
| Нормоконтролер  Зайнутдинова М.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. | Исполнитель  Казаков Д.И.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

2021 г.

Министерство образования и науки республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
| **УтверждеН**  **046. БМТ.09.02.05.747.008.13.01-1-ЛУ** |  |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Описание программы**

**046. БМТ.09.02.05.747.008.13**

**Листов 5**

2021 г.

Аннотация

В подразделе «Описание программы» описывается система и программное обеспечение, с помощью которого создавался программный продукт, минимальные требования для корректной работы программы, входные и выходные данные, а также представлена информационная модель базы данных программного обеспечения по заведению заявок по ремонту бытовой техники.

Работа в сервисном центре включает комплекс операций, связанных с заведения заявок на ремонт и ведения отчётности. Все эти операции и составляют бизнес план любого сервисного центра.

Целью улучшения работы сервисного центра является создание программного продукта по заведению заявок на ремонт бытовой техники в сервисном центре.

Основные задачи, которые будет выполнять программа:

* составление заявок на ремонт;
* учёт заказов;
* формирование актов.

Данное программное обеспечение предназначено для решения задач, которые возникают в сервисном центре по заведению заказов на ремонт. Приложение должно автоматизировать технологический процесс заведения заявок на ремонт и мониторинг клиентом своих заказов.

Программа по заведению заявок на ремонт в сервисном центре разрабатывается в среде Microsoft Visual Studio 2017 Community на компьютере с 64-разрядной операционной системой Windows 10 Professional.

Основные аппаратные характеристики персонального компьютера, на котором проходил тест работы программы:

* процессор: Intel(R) Core(TM) i3-7100U CPU @2.40GHz 2.40GHz;
* оперативная память: 4,00 ГБ;
* дисковое пространство: 1 ТБ;
* разрешение монитора: 1920×1080;
* видеокарта Nvidia Geforce;
* клавиатура;
* мышь.

А также программные характеристики:

* операционная система Windows 10 Professional;
* Microsoft Access версии 2010;
* Microsoft Visual Studio версии 2017;
* Microsoft Word версии 2010.

Разрабатываемая автоматизированная информационная система содержит следующую информационную структуру:

Входные данные:

* данные о клиентах;
* данные о менеджерах;
* расходные материалы и запчасти;
* виды работ (услуг);
* данные о работниках и их специализациях;
* информация об организации

Выходные данные:

* прайс-лист на услуги и работы;
* договор на выполнение работ сервисным центром с перечнем выполняемых работ;
* акт выполненных работ;
* отчёты по выполненным работам.

Для разработки модели базы данных в АИС по составлению заявок в сервисной мастерской, использовалась СУБД Access 2010. Данная база данных состоит из 5 связанных между собой таблиц и 6 таблиц, которые не взаимодействуют друг с другом (рисунок 2): заказы, пользователи, вид работ, тип ремонта и цены работ; статус, заказы с клиентами, цены работ для клиента, заказы для клиентов, пользователи для заказов, вид работ Запрос.

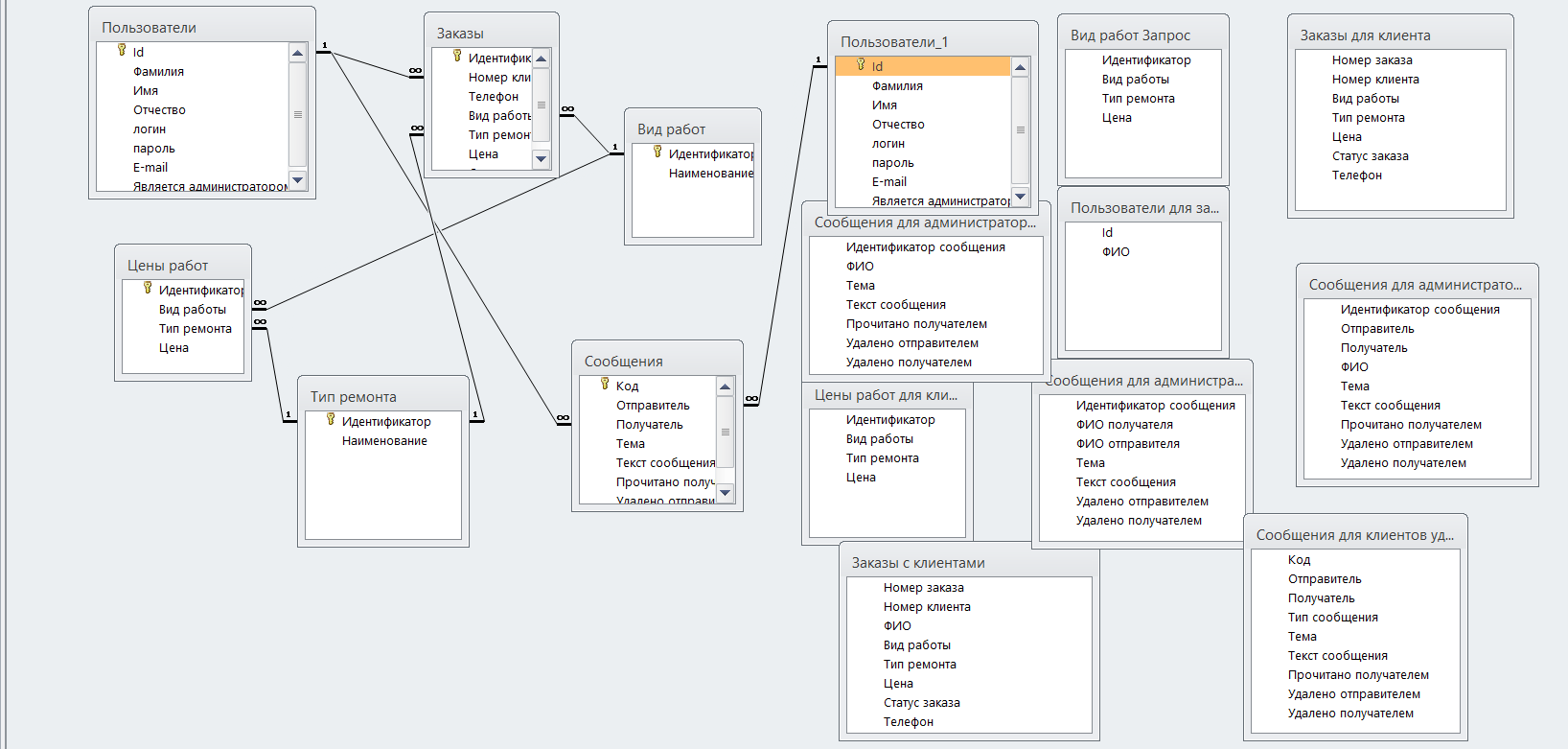


Рисунок 1 – ERD-диаграмма

Данная физическая модель данных АИС по составлению заявок для мастерской состоит их имени поля, уникального идентификационного ключа и типов данных, которые представлены в таблицах

Таблица 5 – Вид работ

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Размер поля |
| Идентификатор | Счётчик | Длинное целое |
| Наименование | Текстовой | 20 |

Таблица 6 – Заказы

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Размер поля |
| Идентификатор | Счётчик | Длинное целое |
| Номер клиента | Числовой | Длинное целое |
| Вид работы | Числовой | Длинное целое |
| Тип ремонта | Числовой | Длинное целое |
| Цена | Денежный | - |
| Статус заказа | Логический | Истина/Ложь |

Таблица 7 – Пользователи

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Размер поля |
| Id | Счётчик | Длинное целое |
| Фамилия | Текстовый | 20 |
| Имя | Текстовый | 20 |
| Отчество | Текстовый | 20 |
| Телефон | Текстовый | 20 |
| логин | Поле MEMO | - |
| пароль | Поле MEMO | - |
| E-mail | Поле MEMO | - |
| Является администратором | Логический | - |

Таблица 8 – Статус

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Размер поля |
| Id | Счётчик | Длинное целое |
| Готов | Логический | - |
| В работе | Логический | - |
| Отложен | Логический | - |

Таблица 9 – Тип ремонта

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Размер поля |
| Идентификатор | Счётчик | Длинное целое |
| Наименование | Текстовой | 20 |

Таблица 10 – Цены работ

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Размер поля |
| Идентификатор | Счётчик | Длинное целое |
| Вид работы | Числовой | Длинное целое |
| Тип ремонта | Числовой | Длинное целое |
| Цена | Денежный | - |

Министерство образования и науки Республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
|  | Утверждаю  Руководитель работы  Дмитриева Л.И.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**2Текст программы**

Лист утверждения

**046.БМТ.09.02.05.747.008.12.01-1-ЛУ**

|  |  |
| --- | --- |
| Нормоконтролер  Зайнутдинова М.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. | Исполнитель  Казаков Д.И.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

2021 г.

Министерство образования и науки республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
| **УтверждеН**  **046. БМТ.09.02.05.747.008.12.01-1-ЛУ** |  |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Текст программы**

**046. БМТ.09.02.05.747.008.12**

**Листов 12**

2021 г.

Авторизация

public partial class Авторизация : Form

{

private string password = "ntvHA2s1y+ryOZizPQ1WHQ==";

int counter = 0;

public Авторизация()

{

if (!CheckDBConnect())

{

this.Close();

}

this.InitializeComponent();

}

/// <summary>

/// Проверяет возможность соединения с БД

/// при необходимости, заменяет путь к резервной копии по указанию пользователя

/// </summary>

private bool CheckDBConnect()

{

while (true)

{

try

{

using (var connection = new OleDbConnection(Settings.Default.k\_systemsConnectionString))

{

connection.Open();

return true;

}

}

catch

{

var reservDBFileQuestionResult = MessageBox.Show(

"Не найдена база данных k-system!\r\nХотите указать файл из резервной копии?",

"Информация",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (reservDBFileQuestionResult != DialogResult.Yes)

{

return false;

}

var openDialog = new OpenFileDialog

{

Title = "Укажите файл базы данных k-system",

Filter = "База данных k-system|\*.mdb",

InitialDirectory = Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location)

};

if (openDialog.ShowDialog() == DialogResult.OK)

{

Settings.Default["k\_systemsConnectionString"] = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + openDialog.FileName;

}

else

{

return false;

}

}

}

}

/// <summary>

/// Открытие формы регистрации

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void pictureBoxRegistration\_Click(object sender, EventArgs e)

{

var reg = new Регистрация(this.password);

reg.ShowDialog();

}

private void Авторизация\_FormClosing(object sender, FormClosingEventArgs e)

{

if (e.CloseReason == CloseReason.UserClosing)

{

DialogResult dialogResult = MessageBox.Show("Вы действиетльно хотите выйти?",

"Закрытие программы", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (dialogResult == DialogResult.Yes)

{

Application.Exit();

}

else e.Cancel = true;

}

else Application.Exit();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

if (this.tbLogin.Text == "" && this.tbPassword.Text == "")

{

MessageBox.Show("Вы не ввели логин или пароль! Пожалуйста, заполните поля!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

this.counter++;

if (string.IsNullOrEmpty(this.tbLogin.Text))

{

this.tbLogin.Focus();

return;

}

if (string.IsNullOrEmpty(this.tbPassword.Text))

{

this.tbPassword.Focus();

return;

}

else if (this.counter >= 3)

{

this.tbLogin.Enabled = false;

this.tbPassword.Enabled = false;

}

var user = this.FindUser(this.tbLogin.Text, this.tbPassword.Text);

if (user != null)

{

MessageBox.Show($"Привет, {user.Имя}!");

if (user.Является\_администратором)

{

var формаАдминистратора = new Администратор();

this.Hide();

формаАдминистратора.Show();

}

else

{

var открыть = new Клиент(user.Id);

this.Hide();

открыть.Show();

}

}

else

{

MessageBox.Show("Вы ввели неверный логин или пароль!Пользователь в базе не найден!", "Сообщение",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void pictureBoxRestart\_Click(object sender, EventArgs e)

{

Application.Restart();

}

/// <summary>

/// Выполняет поиск пользователя по логину и паролю

/// </summary>

/// <param name="login"></param>

/// <param name="password"></param>

/// <returns></returns>

private \_k\_systemsDataSet.ПользователиRow FindUser(string login, string password)

{

var loginCondition = $"логин='{login}' AND пароль='{Регистрация.Encrypt(password, this.password)}'";

var user = EntityManager.FilterUsers(loginCondition).FirstOrDefault();

return user;

}

private \_k\_systemsDataSet.ПользователиRow User(string login)

{

var loginCondition = $"Логин='{login}'";

var user = EntityManager.FilterUsers(loginCondition).FirstOrDefault();

return user;

}

private void PasswordChangeLabel\_Click(object sender, EventArgs e)

{

var открыть = new Смена\_пароля();

открыть.OnValidateUser += (login, newPassword) =>

{

var user = this.User(открыть.GetLogin());

if (user == null)

{

MessageBox.Show("Не найден пользователь с таким логином и паролем");

return false;

}

user.пароль = Регистрация.Encrypt(открыть.GetNewPassword(), this.password);

EntityManager.UpdateUsers();

MessageBox.Show("Пароль изменён!");

return true;

};

открыть.ShowDialog();

}

}

}

Регистрация

public partial class Регистрация : Form

{

private string password;

public Регистрация(string password)

{

this.password = password;

this.InitializeComponent();

}

/// <summary>

/// Метод шифрования пароля

/// </summary>

/// <param name="plainText"></param>

/// <param name="password"></param>

/// <param name="salt"></param>

/// <param name="hashAlgorithm"></param>

/// <param name="passwordInterations"></param>

/// <param name="initialVector"></param>

/// <param name="keySize"></param>

/// <returns></returns>

public static string Encrypt(string plainText, string password, string salt = "Kosher", string hashAlgorithm = "SHA1",

int passwordInterations = 2, string initialVector = "PFRna73\*aze01xY1", int keySize = 256)

{

if (string.IsNullOrEmpty(plainText))

return "";

byte[] initialVectorBytes = Encoding.ASCII.GetBytes(initialVector);

byte[] saltValueBytes = Encoding.ASCII.GetBytes(salt);

byte[] plainTextBytes = Encoding.UTF8.GetBytes(plainText);

var derivedPassword = new PasswordDeriveBytes(password, saltValueBytes, hashAlgorithm, passwordInterations);

byte[] keyBytes = derivedPassword.GetBytes(keySize / 8);

var symmetricKey = new RijndaelManaged();

symmetricKey.Mode = CipherMode.CBC;

byte[] cipherTextBytes = null;

using (ICryptoTransform encryptor = symmetricKey.CreateEncryptor(keyBytes, initialVectorBytes))

{

using (var memStream = new MemoryStream())

{

using (var cryptoStream = new CryptoStream(memStream, encryptor, CryptoStreamMode.Write))

{

cryptoStream.Write(plainTextBytes, 0, plainTextBytes.Length);

cryptoStream.FlushFinalBlock();

cipherTextBytes = memStream.ToArray();

memStream.Close();

cryptoStream.Close();

}

}

}

symmetricKey.Clear();

return Convert.ToBase64String(cipherTextBytes);

}

/// <summary>

/// Проверка заполненности полей

/// </summary>

/// <returns></returns>

private bool CheckTextBoxes()

{

if (this.textBoxSername.Text == "" || this.textBoxName.Text == "" ||

this.textBoxPatronomic.Text == "" ||this.textBoxEmail.Text == "" ||

this.textBoxLogin.Text == "" || this.textBoxPassword.Text == "")

{

return false;

}

return true;

}

private void buttonRegister\_Click(object sender, EventArgs e)

{

if (!this.CheckTextBoxes())

{

MessageBox.Show("Заполните все поля","Информация", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

var isEmailValid = ValidEmail.IsValidEmail(this.textBoxEmail.Text);

if (!isEmailValid)

{

MessageBox.Show("Введите корректный Email!","Информация",MessageBoxButtons.OK,MessageBoxIcon.Information);

return;

}

var newUser = EntityManager.UserDataTable.NewПользователиRow();

newUser.Фамилия = this.textBoxSername.Text;

newUser.Имя = this.textBoxName.Text;

newUser.Отчество = this.textBoxPatronomic.Text;

newUser.\_E\_mail = this.textBoxEmail.Text;

newUser.логин = this.textBoxLogin.Text;

newUser.пароль = Encrypt(this.textBoxPassword.Text, this.password);

EntityManager.UserDataTable.AddПользователиRow(newUser);

EntityManager.UpdateUsers();

MessageBox.Show("Регистрация успешна","Информация",MessageBoxButtons.OK,MessageBoxIcon.Information);

this.Close();

}

private void textBoxPatronomic\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

}

Клиентское окно программы

public partial class Клиент : Form

{

private long idClient;

private \_k\_systemsDataSet.ПользователиRow user;

public Клиент(Int64 userId)

{

this.idClient = userId;

this.InitializeComponent();

}

public Клиент(\_k\_systemsDataSet.ПользователиRow user)

{

this.user = user;

}

private void Клиент\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Цены\_работ\_для\_клиента". При необходимости она может быть перемещена или удалена.

this.цены\_работ\_для\_клиентаTableAdapter.Fill(this.\_k\_systemsDataSet.Цены\_работ\_для\_клиента);

}

/// <summary>

/// Закрытие формы

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Клиент\_FormClosing(object sender, FormClosingEventArgs e)

{

if (e.CloseReason == CloseReason.UserClosing)

{

DialogResult dialogResult = MessageBox.Show("Вы действиетльно хотите выйти?",

"Закрытие программы", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (dialogResult == DialogResult.Yes)

{

Application.Exit();

}

else e.Cancel = true;

}

else Application.Exit();

}

private void priceListFilterTextBox\_TextChanged(object sender, EventArgs e)

{

var findFields = new[] { "[Вид работы]", "[Тип ремонта]" };

var filterString =

EntityManager.GetFilterStringByFields(findFields, this.priceListFilterTextBox.Text).Trim();

this.ценыРаботДляКлиентаBindingSource.Filter = filterString;

}

private void заказыToolStripMenuItem\_Click(object sender, EventArgs e)

{

var открыть = new Заказы(this.idClient);

открыть.ShowDialog();

}

private void сменаПользователяToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

var открыть = new Авторизация();

открыть.ShowDialog();

}

Окно администратора

public partial class Администратор : Form

{

private string password = "ntvHA2s1y+ryOZizPQ1WHQ==";

const string FilterNinAdministrator = "([Является администратором]=False)";

public Администратор()

{

this.InitializeComponent();

this.Администратор\_Load();

}

private void Администратор\_Load()

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter.Fill(this.\_k\_systemsDataSet.Пользователи);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter.Fill(this.\_k\_systemsDataSet.Пользователи);

}

/// <summary>

/// Фильтрация пользователей

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void masterFilterTextBox\_TextChanged(object sender, EventArgs e)

{

var findFields = new[] {"Фамилия", "Имя", "Отчество" };

var filterString = EntityManager.GetFilterStringByFields(findFields, this.masterFilterTextBox.Text).Trim();

this.пользователиBindingSource.Filter = FilterNinAdministrator;

if (!string.IsNullOrEmpty(filterString))

{

this.пользователиBindingSource.Filter += $"And ({filterString})";

}

}

private void Администратор\_FormClosing(object sender, FormClosingEventArgs e)

{

if (e.CloseReason == CloseReason.UserClosing)

{

DialogResult dialogResult = MessageBox.Show("Вы действиетльно хотите выйти?",

"Закрытие программы", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (dialogResult == DialogResult.Yes)

{

Application.Exit();

}

else e.Cancel = true;

}

else Application.Exit();

}

private void AddOrder\_Click(object sender, EventArgs e)

{

var открыть = new Добавить\_заказ();

открыть.ShowDialog();

}

private void dataGridView\_CellEndEdit(object sender, DataGridViewCellEventArgs e)

{

var changedRow = (ПользователиRow)((DataRowView)((DataGridView)sender).CurrentRow.DataBoundItem).Row;

var userTable = EntityManager.FilterUsers($"Id = {changedRow.Id}");

userTable[0].Id = changedRow.Id;

userTable[0].Фамилия = changedRow.Фамилия;

userTable[0].Имя = changedRow.Имя;

userTable[0].Отчество = changedRow.Отчество;

userTable[0].\_E\_mail = changedRow.\_E\_mail;

EntityManager.UpdateUsers();

}

private void СhangeUser\_Click(object sender, EventArgs e)

{

this.Hide();

var открыть = new Авторизация();

открыть.ShowDialog();

}

private void NewClient\_Click(object sender, EventArgs e)

{

var открыть = new Регистрация(this.password);

открыть.ShowDialog();

}

private void AddService\_Click(object sender, EventArgs e)

{

var открыть = new Добавить\_услугу();

открыть.ShowDialog();

}

private void btn\_cancel\_Click(object sender, EventArgs e)

{

this.пользователиBindingSource.Filter = null;

this.masterFilterTextBox.Text = "";

}

private void buttonCancel\_Click(object sender, EventArgs e)

{

//int ind = this.dataGridView.SelectedCells[0].RowIndex;

//this.dataGridView.Rows.RemoveAt(ind);

var idUser= ((ПользователиRow)((DataRowView)this.dataGridView.CurrentRow?.DataBoundItem)?.Row)?.Id;

if (!idUser.HasValue)

{

return;

}

var deleteUser = MessageBox.Show("Вы действительно хотите удалить данного пользователя? ", "Информация",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (deleteUser != DialogResult.Yes)

{

return;

}

using(var connection=new OleDbConnection(Settings.Default.k\_systemsConnectionString))

{

connection.Open();

using(var sqlCommand = connection.CreateCommand())

{

sqlCommand.CommandText = $"DELETE FROM Пользователи WHERE Id={idUser.Value}";

sqlCommand.ExecuteNonQuery();

}

}

this.Администратор\_Load();

}

private void SaveDB\_Click(object sender, EventArgs e)

{

var dataBasePath = Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location) + "\\k-systems.mdb";

var saveDialog = new SaveFileDialog

{

Filter="AccessDB files|\*.mdb"

};

try

{

if (saveDialog.ShowDialog() == DialogResult.OK)

{

File.Copy(dataBasePath, saveDialog.FileName, true);

MessageBox.Show("Резервное копирование прошло успешно!", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch(Exception ex)

{

MessageBox.Show("Не удаётся скопировать файл из-за исключения: " + ex.Message);

}

}

private void RestoreDB\_Click(object sender, EventArgs e)

{

var dataBasePath = Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location) + "\\Библиотека.mdb";

var openDialog = new OpenFileDialog

{

Filter = "AccessDB files|\*.mdb"

};

try

{

if (openDialog.ShowDialog() == DialogResult.OK)

{

File.Copy(openDialog.FileName, dataBasePath, true);

MessageBox.Show("Восстановление прошло успешно!", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

this.пользователиTableAdapter.Adapter.Fill(this.\_k\_systemsDataSet.Пользователи);

this.пользователиTableAdapter.Adapter.Update(this.\_k\_systemsDataSet.Пользователи);

}

catch (Exception exception)

{

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Email\_Click(object sender, EventArgs e)

{

var открыть = new Почта();

открыть.ShowDialog();

}

private void Reporting\_Click(object sender, EventArgs e)

{

var открыть = new Отчётность\_по\_заказам();

открыть.ShowDialog();

}

Форма для отображения заказов

public partial class Добавить\_заказ : Form

{

private string textBoxOrdersFilter;

private string readyOrNotReadyOrdersFilter;

public Добавить\_заказ()

{

this.InitializeComponent();

}

private void Добавить\_заказ\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Заказы\_с\_клиентами". При необходимости она может быть перемещена или удалена.

this.заказы\_с\_клиентамиTableAdapter.Fill(this.\_k\_systemsDataSet.Заказы\_с\_клиентами);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Заказы\_с\_клиентами". При необходимости она может быть перемещена или удалена.

this.заказы\_с\_клиентамиTableAdapter.Fill(this.\_k\_systemsDataSet.Заказы\_с\_клиентами);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Заказы\_с\_клиентами". При необходимости она может быть перемещена или удалена.

this.заказы\_с\_клиентамиTableAdapter.Fill(this.\_k\_systemsDataSet.Заказы\_с\_клиентами);

}

/// <summary>

/// Фильтрует заказы по строке поиска, учитывая фильтрацию по готовности

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void orderFilterTextBox\_TextChanged(object sender, EventArgs e)

{

this.заказыСКлиентамиBindingSource.Filter=this.BuildWorkerCardFilter();

}

private string BuildWorkerCardFilter()

{

var filterExpressionList = new List<string>();

var fieldFilter = this.orderFilterTextBox.Text;

if (!string.IsNullOrEmpty(fieldFilter))

{

filterExpressionList.Add(string.Format("(([ФИО] Like '%{0}%') OR ([Вид работы] Like '%{0}%'))", fieldFilter));

}

return string.Join(" AND ", filterExpressionList);

}

/// <summary>

/// Добавление заказов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void addOrderButton\_Click(object sender, EventArgs e)

{

(new Заказ()).ShowDialog();

this.заказы\_с\_клиентамиTableAdapter.Fill(

this.\_k\_systemsDataSet.Заказы\_с\_клиентами);

}

/// <summary>

/// Обновляет информацию в бд об отредактированном заказе

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void dataGridViewOrder\_CellEndEdit(object sender, DataGridViewCellEventArgs e)

{

var changedRow = (Заказы\_с\_клиентамиRow)((DataRowView)((DataGridView)sender).CurrentRow.DataBoundItem).Row;

var orderTable = EntityManager.FilterOrders($"Id={changedRow.Номер\_заказа}");

orderTable[0].Статус\_заказа = changedRow.Статус\_заказа;

orderTable[0].Цена = changedRow.Цена;

EntityManager.UpdateOrders();

this.заказы\_с\_клиентамиTableAdapter.Fill(this.\_k\_systemsDataSet.Заказы\_с\_клиентами);

}

/// <summary>

/// Переключает фильтрацию заказов по полю "Заказ готов": true, false, all

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void readyOrdersFilterCheckBox\_CheckStateChanged(object sender, EventArgs e)

{

switch (this.readyOrdersFilterCheckBox.CheckState)

{

case CheckState.Checked:

this.readyOrNotReadyOrdersFilter = $"[Статус заказа] = '{WorkStates.Ready}'";

this.readyOrdersFilterCheckBox.Text = "Отображаются готовые заказы";

break;

case CheckState.Indeterminate:

this.readyOrNotReadyOrdersFilter = string.Empty;

this.readyOrdersFilterCheckBox.Text = "Отображаются все заказы";

break;

case CheckState.Unchecked:

this.readyOrNotReadyOrdersFilter = $"[Статус заказа] = '{WorkStates.Working}'";

this.readyOrdersFilterCheckBox.Text = "Отображаются неготовые заказы";

break;

}

this.заказыСКлиентамиBindingSource.Filter = EntityManager.UnionFilter(

this.textBoxOrdersFilter,

this.readyOrNotReadyOrdersFilter);

}

private void dataGridViewOrder\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

var открыть = new Акт\_приёмки\_заказа(this.\_k\_systemsDataSet);

открыть.LoadЗаказы(

(Заказы\_с\_клиентамиRow)((DataRowView)this.dataGridViewOrder.CurrentRow.DataBoundItem).Row);

открыть.ShowDialog();

this.заказы\_с\_клиентамиTableAdapter.Fill(this.\_k\_systemsDataSet.Заказы\_с\_клиентами);

}

Составление заказа

private void Заказ\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Вид\_работ". При необходимости она может быть перемещена или удалена.

this.вид\_работTableAdapter.Fill(this.\_k\_systemsDataSet.Вид\_работ);

// TODO: данная строка кода позволяет загрузить данные в таблицу "\_k\_systemsDataSet.Пользователи\_для\_заказов". При необходимости она может быть перемещена или удалена.

this.пользователи\_для\_заказовTableAdapter.Fill(this.\_k\_systemsDataSet.Пользователи\_для\_заказов);

this.Phone.Mask = "+7(000)-000-0000";

}

/// <summary>

/// Добавляет заказ в таблицу

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void addOrderButton\_Click(object sender, EventArgs e)

{

var orderState = this.GetOrderState();

if (string.IsNullOrEmpty(orderState))

{

MessageBox.Show("Выберите статус заказа");

return;

}

var newOrder = EntityManager.OrderDataTable.NewЗаказыRow();

newOrder.Номер\_клиента = (int)this.clientComboBox.SelectedValue;

newOrder.Вид\_работы = (int)this.workKindComboBox.SelectedValue;

newOrder.Тип\_ремонта = (int)this.dressTypeComboBox.SelectedValue;

newOrder.Цена = this.priceNumericUpDown.Value;

newOrder.Статус\_заказа = orderState;

newOrder.Телефон = this.Phone.Text;

EntityManager.OrderDataTable.AddЗаказыRow(newOrder);

EntityManager.UpdateOrders();

MessageBox.Show("Заказ успешно добавлен","Информация",

MessageBoxButtons.OK,MessageBoxIcon.Information);

this.Close();

}

private string GetOrderState()

{

if (this.Work.Checked)

{

return WorkStates.Working;

}

return WorkStates.Waiting;

}

/// <summary>

/// Загружает цену из таблицы "Цены работ", если выбранная пара значений "Тип ремонта" и "Вид работы"

/// найдена в соответствующих полях

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void orderPriceComboBox\_SelectedValueChanged(object sender, EventArgs e)

{

if (this.workKindComboBox.SelectedItem == null || this.dressTypeComboBox.SelectedItem == null)

{

return;

}

var filterCondition = $"[Вид работы] = {this.workKindComboBox.SelectedValue} AND [Тип ремонта] = {this.dressTypeComboBox.SelectedValue}";

var foundRowWorkPrices = EntityManager.FilterWorkPrices(filterCondition).FirstOrDefault();

this.priceNumericUpDown.Value = foundRowWorkPrices?.Цена as decimal? ?? 0m;

}

private void workKindComboBox\_SelectedValueChanged(object sender, EventArgs e)

{

if (this.workKindComboBox.SelectedItem == null)

{

return;

}

var typeServiceIds = EntityManager.FilterWorkPrices($"[Цены работ].[Вид работы] = {this.workKindComboBox.SelectedValue}")

.Select(x => x.Field<int>("Тип ремонта"))

.ToArray();

if(typeServiceIds.Length == 0)

{

this.типРемонтаBindingSource.Filter = $"False";

}

else

{

this.типРемонтаBindingSource.Filter = $"[Идентификатор] in ({string.Join(", ", typeServiceIds)})";

}

this.тип\_ремонтаTableAdapter.Fill(this.\_k\_systemsDataSet.Тип\_ремонта);

}

private void Phone\_KeyPress(object sender, KeyPressEventArgs e)

{

char c = e.KeyChar;e.Handled = !(char.IsDigit(c) || c == '.' || c == ',' || c == '\b');}

Министерство образования и науки Республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
|  | Утверждаю  Руководитель работы  Дмитриева Л.И.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Руководство программиста**

Лист утверждения

**046.БМТ.09.02.05.747.008.33.01-1-ЛУ**

|  |  |
| --- | --- |
| Нормоконтролер  Зайнутдинова М.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. | Исполнитель  Казаков Д.И.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

2021 г.

Министерство образования и науки республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
| **УтверждеН**  **046. БМТ.09.02.05.747.008.33.01-1-ЛУ** |  |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Руководство программиста**

**046. БМТ.09.02.05.747.008.33**

**Листов 3**

2021 г.

Аннотация

В данном разделе описывается, как установить программный продукт на компьютер для ознакомления и работы с программой. Наглядно показываются все этапы установки.

Для установки программы по заведению заявок по ремонту бытовой техники в сервисном центре, используется инсталлятор, предварительно кликнув по нему, будет предложено выбрать ознакомиться с лицензией данного программного продукта и принять его.

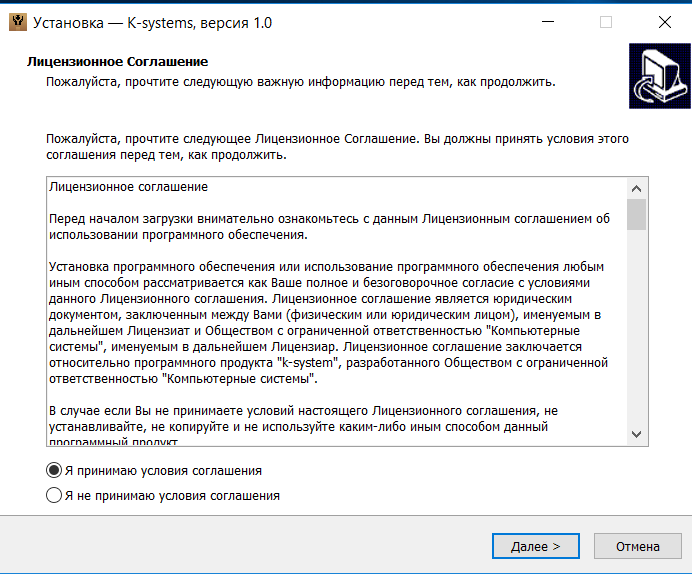


Рисунок 2 – Лицензия программы

Далее будет предложено выбрать каталог, где будет установлена программа, путь по умолчанию C:\Program Files (x86)\K-systems (рисунок 3).

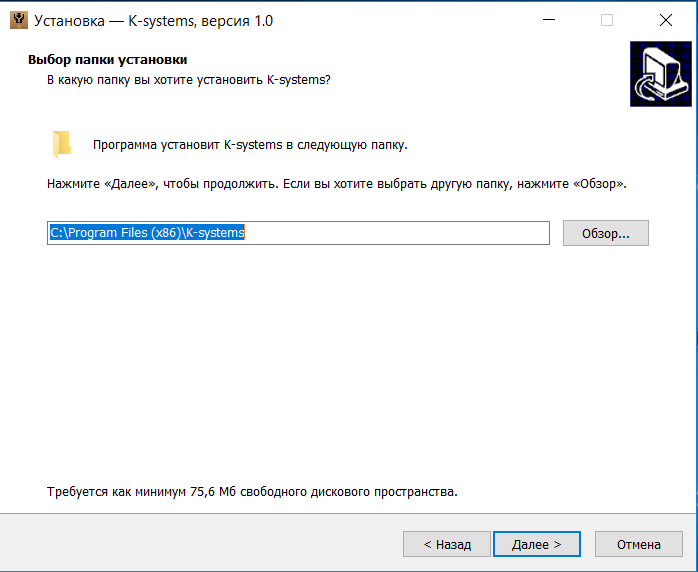


Рисунок 3 – Выбор каталога

Далее идёт создание ярлыка программы на рабочем столе (рисунок 3).

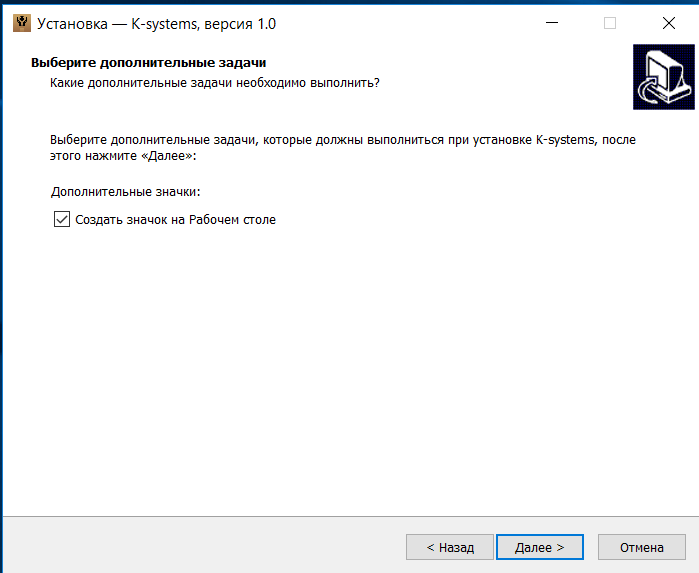


Рисунок 4 – Создание ярлыка

Нажав на «Далее» программа установки покажет параметры, выбранные в ходе предварительной установки (рисунок 5).

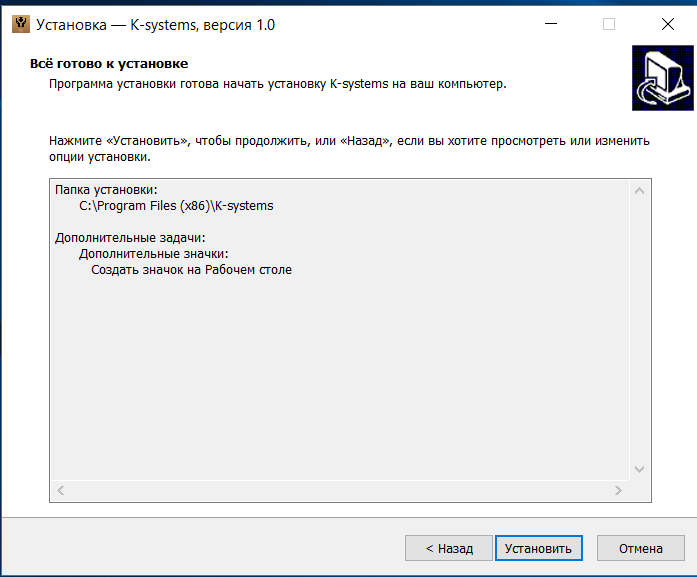


Рисунок 5 – Параметры

После установки программы, установщик предложит запустить программу заведения заявок по ремонту бытовой техники в сервисном центре (рисунок 6).

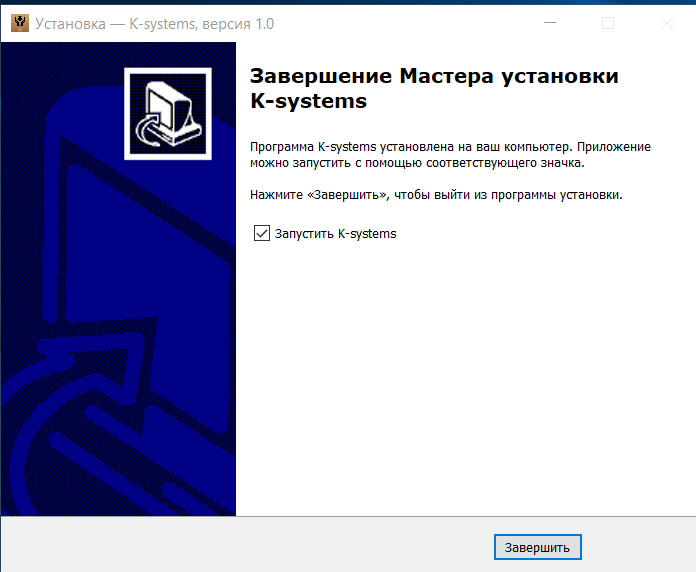


Рисунок 6 – Завершение установки

Министерство образования и науки Республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
|  | Утверждаю  Руководитель работы  Дмитриева Л.И.\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Руководство оператора**

Лист утверждения

**046.БМТ.09.02.05.747.008.34.01-1-ЛУ**

|  |  |
| --- | --- |
| Нормоконтролер  Зайнутдинова М.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_2021 г. | Исполнитель  Казаков Д.И.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

2021 г.

Министерство образования и науки республики Татарстан

ГАПОУ «Бугульминский машиностроительный техникум»

|  |  |
| --- | --- |
| **УтверждеН**  **046. БМТ.09.02.05.747.008.34.01-1-ЛУ** |  |

РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

**Руководство оператора**

**046. БМТ.09.02.05.747.008.34**

**Листов 8**

2021 г.

Аннотация

В подразделе «Руководство оператора» описывается, как работать с программой, описывается графический интерфейс программы и функции приложения по заведению заявок в сервисном центре.

Для запуска программы на рабочем столе находим ярлык установленной программы (рисунок 7)



Рисунок 7 – Ярлык программы

После запуска программы, пользователю необходимо ввести свой логин и пароль для входа в систему (рисунок 8).

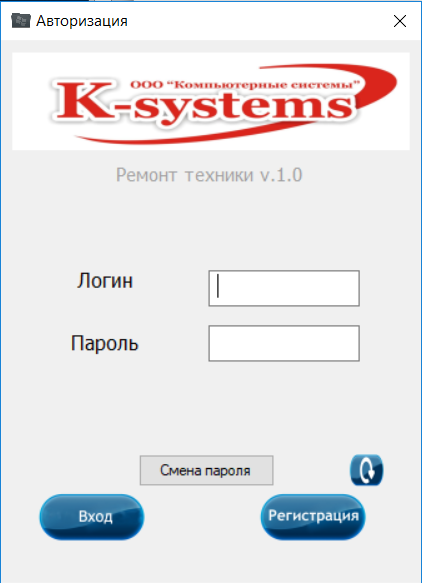


Рисунок 8 – Авторизация

Для закрытия программы, пользователь увидит окно с предупреждением (рисунок 9).

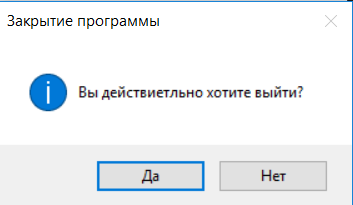
****

Рисунок 9 – Выход из программы

Перед тем как пользователь авторизуется, ему нужно зарегистрироваться в системе, введя данные при регистрации (рисунок 10).

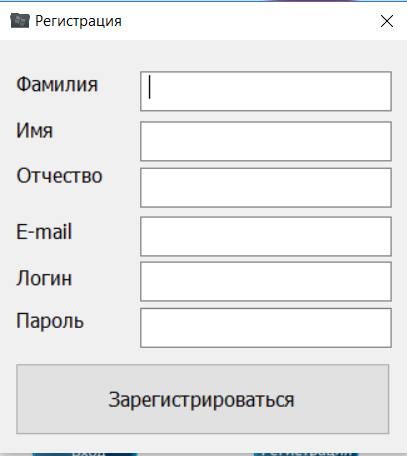


Рисунок 10 – Регистрация пользователя

Если при регистрации не все поля были заполнены, выйдет окно с предупреждением (рисунок 11).

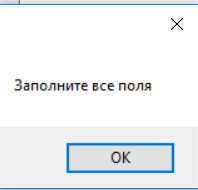


Рисунок 11 – Предупреждение о заполнении полей

При вводе логина и пароля, в окне авторизации, если пользователь не найден в базе данных, то выйдет окно с предупреждением (рисунок 12).

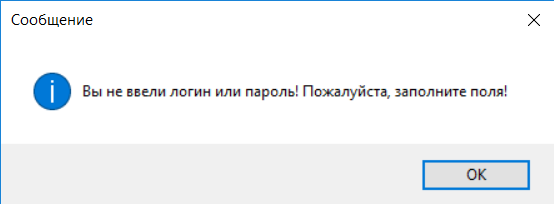


Рисунок 12 – Предупреждением о ненайденном пользователе

После того как пользователь войдёт под учётной записью с привилегией «Администратор», то появится окно с полномочиями администратора. В данной форме, администратор может просматривать всех зарегистрированных пользователей, а также заводить новых клиентов, добавлять новые заказы для клиентов, формировать отчётность, а также отвечать на сообщения клиента (рисунок 13).

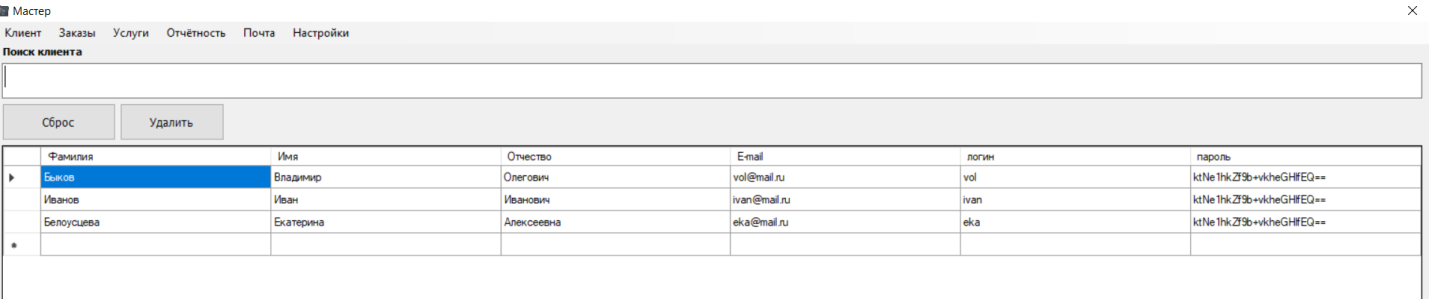


Рисунок 13 – Форма администратора

При входе в программу обычным пользователем, попадаем на форму клиента (рисунок 14).

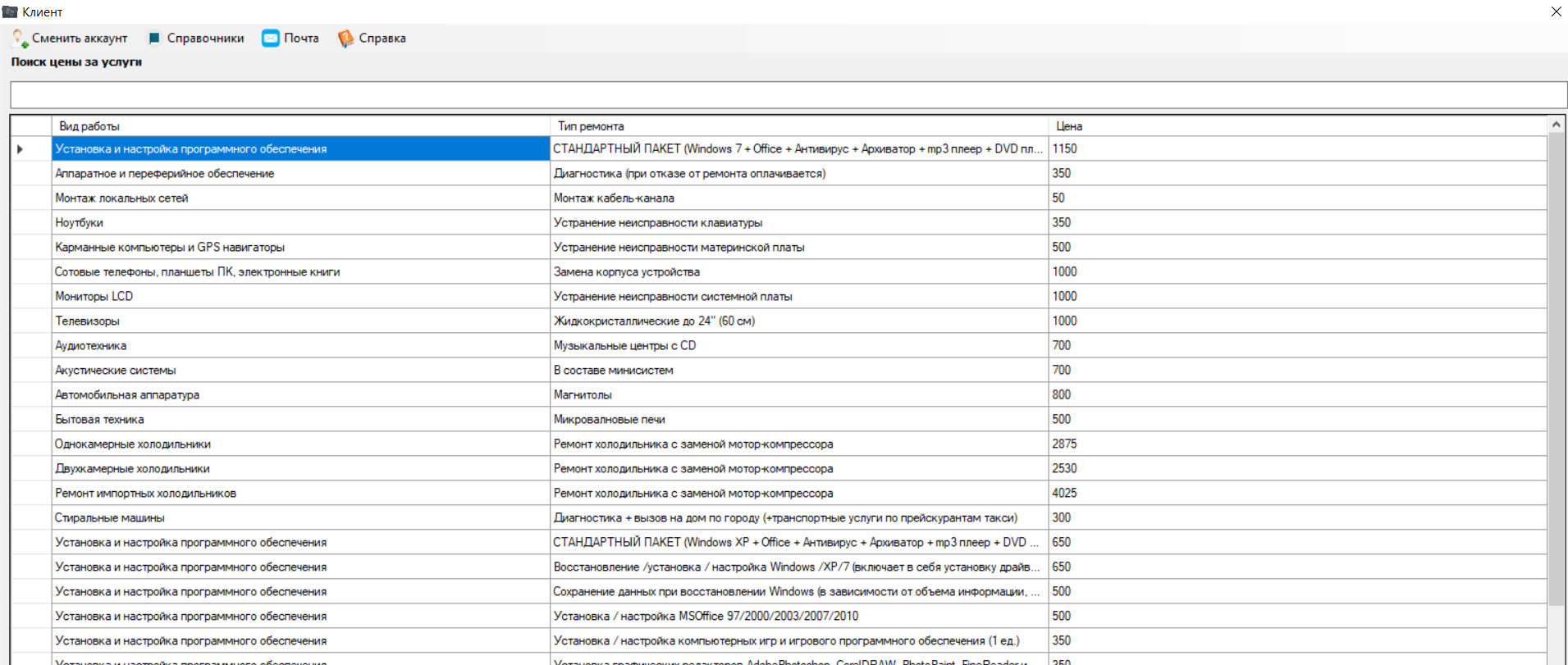


Рисунок 14 – Форма клиента

По нажатию на справочник «Заказы», клиент может увидеть свой заказ и в какой стадии его услуга находится (рисунок 15).

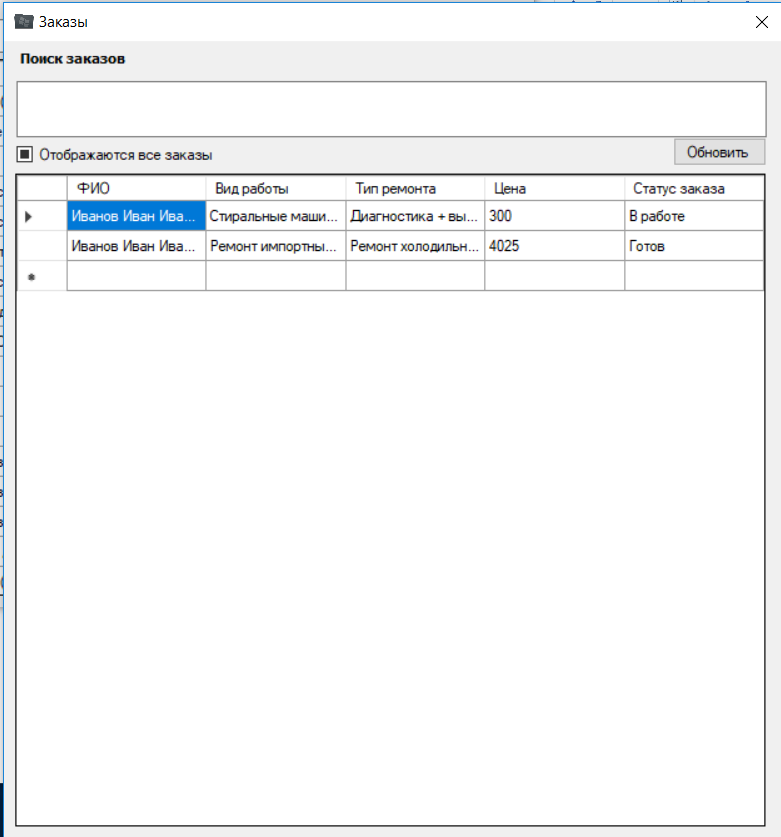


Рисунок 15 – Заказы клиента

Дважды нажав на ячейку заказа, откроется форма «Акта выдачи заказа». С этим актом клиент может прийти в салон, предварительно его скачав со своего аккаунта, и получить своё отремонтированное оборудование (рисунок 16).

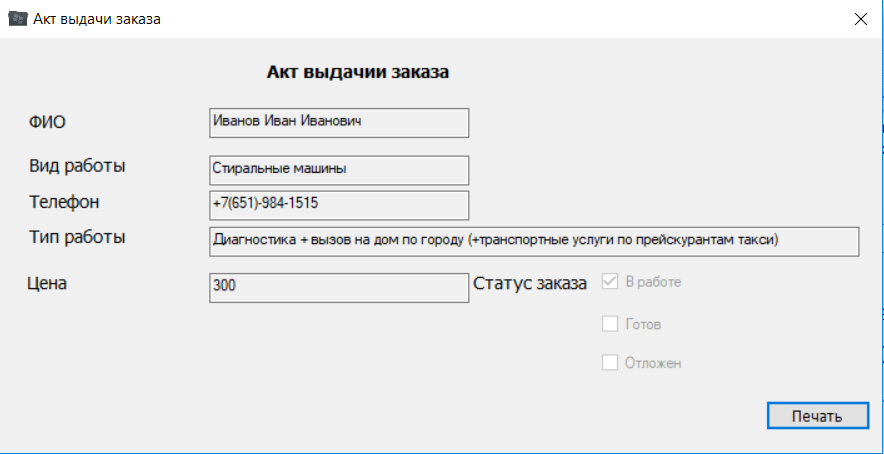


Рисунок 16 – Печать акта выдачи

В разделе «Почта», Клиент может общаться с администратором и узнавать информацию о своём заказе (рисунок 17).

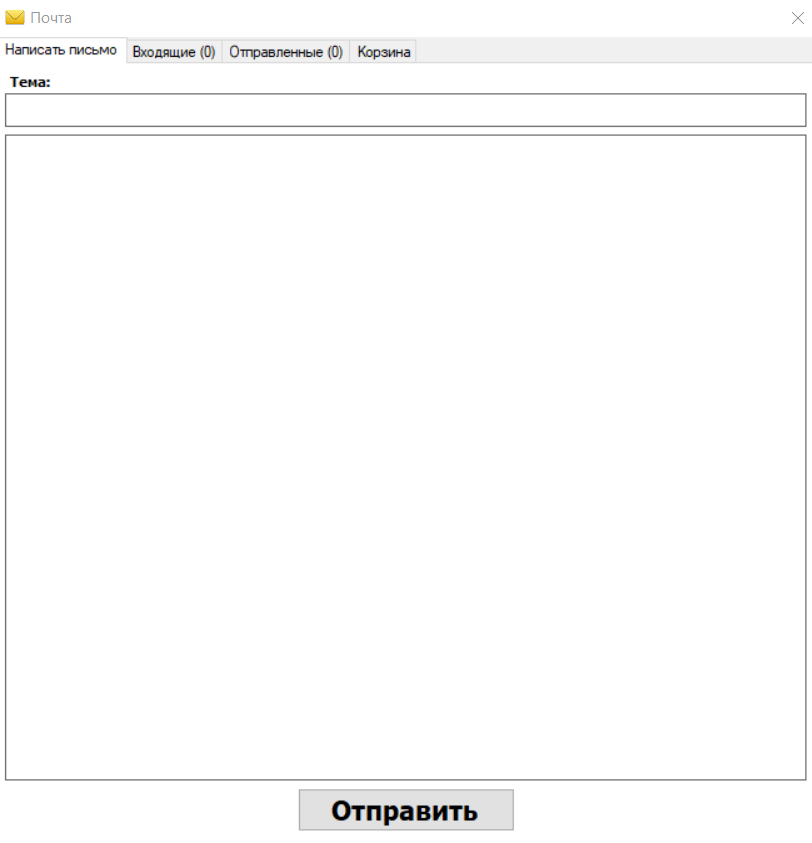


Рисунок 17 – Почта

Когда требуется удалить сообщение из входящих, нажимаем «Удалить сообщение» (рисунок 18).

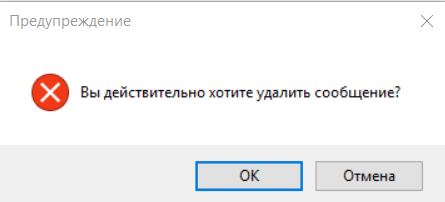


Рисунок 18 –Удаление входящего сообщения

Для того чтобы добавить заказ переходим в учётную запись администратора и открываем окно «Добавить заказ». Далее нажимаем «Добавить новый заказ» (рисунок 19).

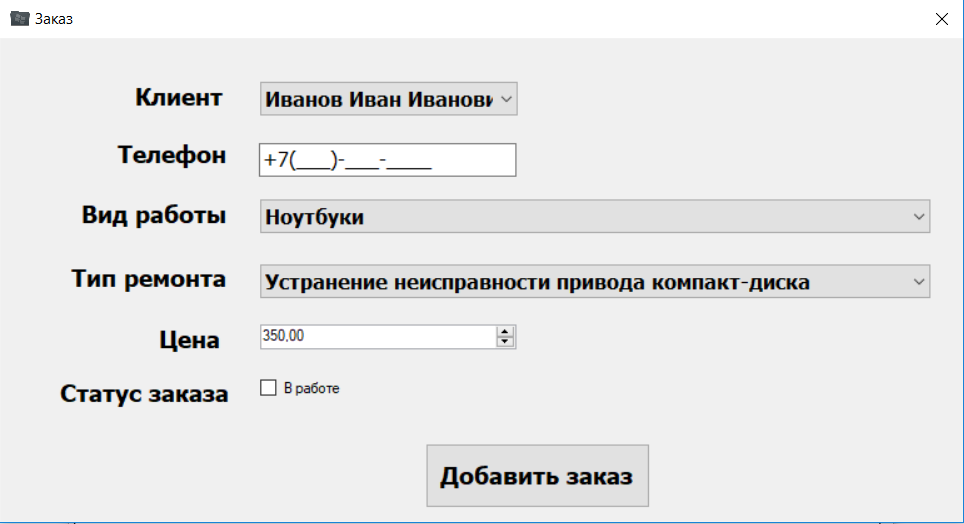


Рисунок 19 – Добавление заказа

После того, как был сформирован заказ, он попадает в список заказов, где специалист видит все заказы от клиентов (рисунок 20).

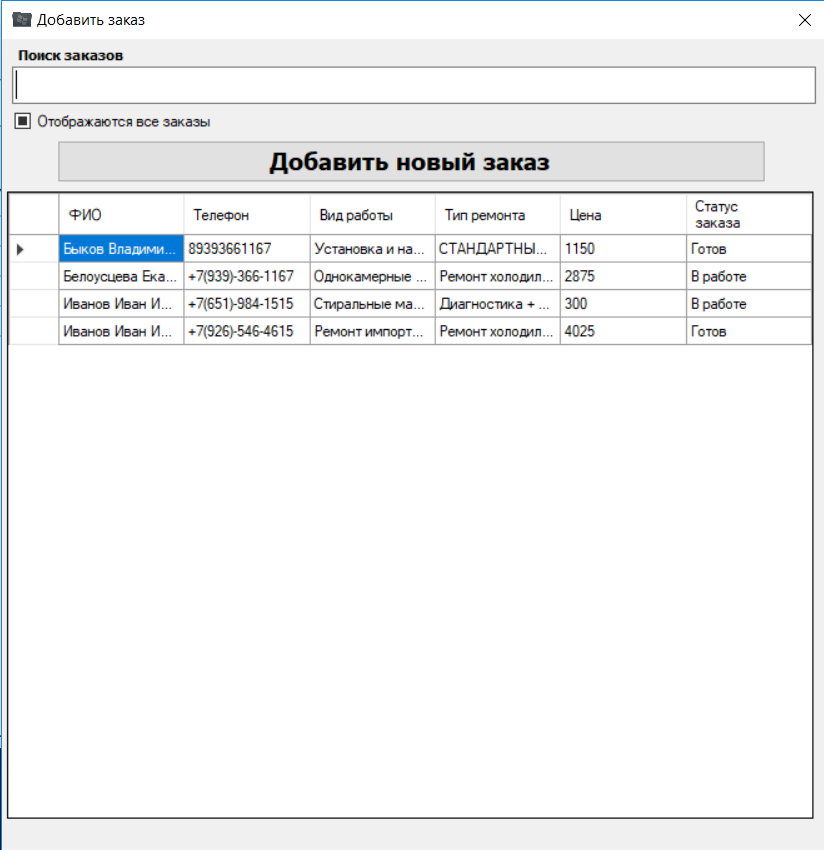


Рисунок 20 – Список всех заказов

Для того чтобы изменить в заказе статус, администратор двойным кликом открывает ячейку и в форме меняет статус заказа (рисунок 21).

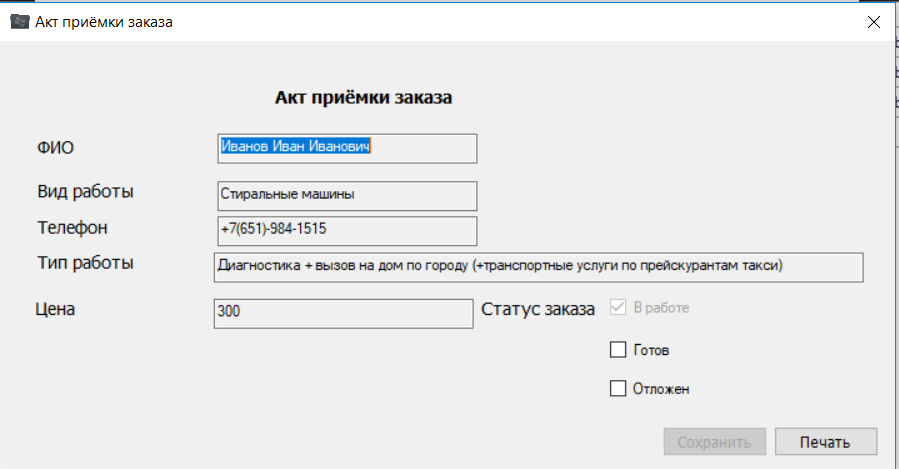


Рисунок 20 – Акт приёмки заказа

Выбрав вкладку «Отчётность» откроется окно «Отчётности по заказам». Определив клиента, формируется отчёт по заказу клиента (рисунок 21).

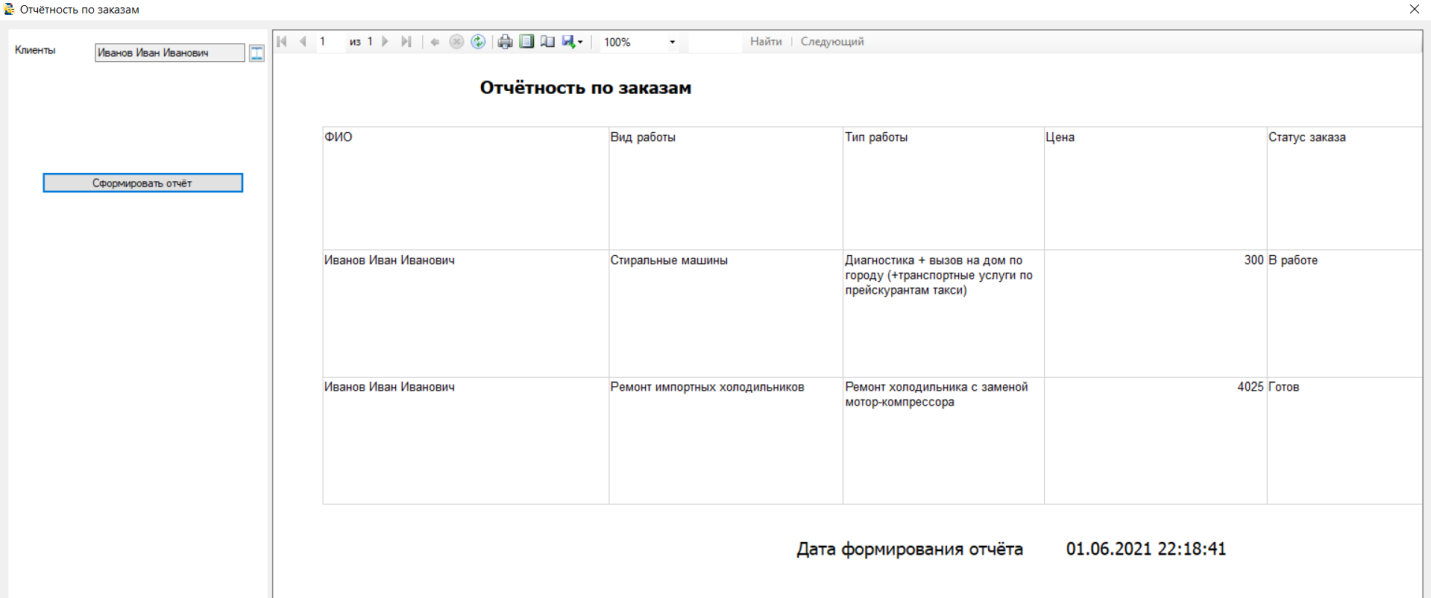


Рисунок 22 – Отчётность по заказам

# 3 Расчётно-экономический раздел

## 3.1 Расчёт амортизационных отчислений

В данной работе используются следующие технические средства:

* компьютер, характеристика которого указана в таблице 14.
* монитор LG;
* принтер HP.

Таблица 14 – Характеристика компьютера

|  |  |
| --- | --- |
| Наименование | Тип |
| Тип компьютера | DEXP Aquilon O175 |
| Оперативная память | 4 Гб |
| Объём жесткого диска | 1000 Гб |
| Тип ЦП | Intel(R) Core(TM) Celeron J#060 CPU @ 2.3GHz |
| Дисковые накопители | Hitachi HDS721010DLE630 ATA Device (931 Гб, IDE) |
| Оптический накопитель | TSSTcorp CDDVDW SH-222BB ATA Device |
| Клавиатура | Клавиатура HID IBM enhanced keyboard |
| Мышь | HID-совместимая мышь |

Закупочная цена:

* компьютера – 18000 руб.,
* монитора – 7500 руб.,
* принтера – 8550 руб.

Примем норму амортизации на технические средства 10 %.

Найдем общую стоимость технических средств, руб.:

|  |  |
| --- | --- |
| Цтс=Цк+Цм+Цп, | (1) |

где Цк – цена компьютера;

    Цм – цена монитора;

Цп – цена принтера.

Отсюда:

Цтс=18000+7500+8550=34050.

Для создания пакета программ, применялось следующее программное обеспечение:

* платформа Windows 10 Home– 4350 руб.;
* Microsoft Visual Studio Community – бесплатная;
* Microsoft Office Excel 2010 – 3950 руб.
* Microsoft Office Word 2010 – 3950 руб.
* Microsoft Office Access 2010 – 3950 руб.

Общая стоимость программного обеспечения (Цпо) составляет 16200 руб.

Общая стоимость технических средств и программного обеспечения, руб.:

|  |  |
| --- | --- |
| Цо=Цтс+Цпо, | (2) |

Цо=34050+16200=50250 руб.

Годовые амортизационные отчисления на полное восстановление технических средств и программного обеспечения рассчитываются по формуле, руб.:

|  |  |
| --- | --- |
| Aо=Цо\*10%, | (3) |

Ао=50250\*0,1=5025 руб.

Амортизационные отчисления за период создания программного продукта, руб.:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Ап = | Ао \* Кдн |  | , | | Крг | | | (4) |

где       Кдн =21 дн. – количество отработанных дней;

Крг= 254 дн. – количество рабочих дней в году.

Ап=5025\*21/254=415,4 руб.

## 3.2 Расчет себестоимости разработки

В себестоимость разработки автоматизированной информационной системы входят следующие статьи затрат:

* основная заработная плата;
* дополнительная заработная плата;
* отчисления на социальное страхование;
* прочие расходы.

Разработку системы проводят два специалиста: начальник отдела (руководитель) и программист.

Зарплата начальника составляет 200 руб/час, программиста – 82 руб/час.

При этом продолжительность рабочего дня каждого из них составляет 7 часов.

Расчет основной заработной платы приведен в таблице 15.

Таблица 15 – Расчет основной заработной платы

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Этапы | Виды работ | | Исполнитель | | | | Часовая ставка | | Длительность выполнения, в часах | | Размер з/п, руб | |
| Кол-во | | Должность | |
| Начальный | Постановка цели и задачи создания программы | | 1 | | Библиотекарь | | 200 | | 7 | | 1400 | |
| Обоснование необходимости разработки программы. | | 1 | | Библиотекарь | | 200 | | 7 | | 1400 | |
| Определение структуры входных и выходных данных. | | 1 | | Программист | | 82 | | 14 | | 1148 | |
| Разработка и утверждение технического задания | | 2 | | Библиотекарь | | 200 | | 7 | | 1400 | |
| Программист | | 82 | | 7 | | 574 | |
|  |  | |  | |  | |  | |  | |  | |
| Продолжение таблицы 15 | | | |  |  |  | |  | |  | |  |
| Основной | | Разработка алгоритма и интерфейса программы | 1 | | Программист | | 82 | | 7 | | 574 | |
| Проектирование модулей | 1 | | Программист | | 82 | | 7 | | 574 | |
| Кодирование | 1 | | Программист | | 82 | | 28 | | 2296 | |
| Комплексное тестирование | 1 | | Программист | | 82 | | 14 | | 1148 | |
| Отладка | 1 | | Программист | | 82 | | 7 | | 574 | |
| Оформление программной документации | 1 | | Программист | | 82 | | 7 | | 574 | |
| Оформление документации | 1 | | Программист | | 82 | | 7 | | 574 | |
| Заключительный этап | | Редактирование программной документации | 1 | | Библиотекарь | | 200 | | 14 | | 2800 | |
| 1 | | Программист | | 82 | | 14 | | 1148 | |
| Итого | |  |  | |  | |  | | 147 | | 16184 | |

К дополнительной заработной плате относятся: оплата отпусков, выплата вознаграждения за выслугу лет и так далее. Дополнительная заработная плата составляет 20% от основной:

16184\*0,2=3237 руб.

К отчислениям на социальное страхование относятся отчисления на оплату перерывов в работе в связи с временной нетрудоспособностью и отчисления в пенсионный фонд. Норматив отчислений на социальное страхование составляет 30% от величины основной заработной платы:

1. Пенсионный фонд – 22% (3560,48 руб.)
2. Фонд социального страхования – 2,9% (469,336 руб.)
3. Федеральный фонд обязательного медицинского страхования – 5,1% (825,384 руб)

19421\*0,30=5826 руб.

## 3.3 Затраты на электроэнергию

К прочим расходам следует отнести расходы на обслуживание ЭВМ и плату за электроэнергию.

Затраты на электроэнергию рассчитываются исходя из потребляемой мощности устройства и тарифа на электроэнергию. В нашем случае предполагается использование компьютера с мощностью 0,8 кВт час. Стоимость одного кВт часа электроэнергии равна 3,69 руб. Время использования электроэнергии в процессе разработки:

21\*6=126 часов

Следовательно, плата за электроэнергию составит:

0,8\*3,69\*126=371,9 руб.

Расходы на обслуживание ЭВМ определяются из стоимости ЭВМ и времени ее эксплуатации, по истечении которого, она подлежит замене (обычно это время не превышает 3-х лет), в течении года ЭВМ использует 254 рабочих дня. Тогда:

|  |  |
| --- | --- |
| 34050 | \*126=938,3 руб. |
| 3\*6\*254 |

Необходимо учесть накладные расходы, отображенные в таблице 16.

Таблица 16 – Прочие расходы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Материалы | Единица измерения | Количество | Цена за единицу | Сумма, руб |
| Бумага А4 (500 листов) | пачка | 1 | 250 | 250 |
| CD-RW | шт. | 1 | 50 | 50 |
| Шариковая ручка | шт. | 1 | 30 | 30 |
| Папка для диплома | шт. | 1 | 200 | 200 |
| Ватман | шт. | 1 | 20 | 20 |
| Краска для принтера | шт. | 1 | 250 | 250 |
| Итого: |  |  |  | 800 |

Расчет себестоимости разработки системы представлен в таблице 17.

Таблица 17 – Расчет себестоимости разработки системы

|  |  |
| --- | --- |
| Статьи затрат | Сумма, руб. |
| Основная заработная плата | 16184 |
| Отчисления | 5826 |
| Дополнительная заработная плата | 3122 |
| Продолжение таблицы 17 |  |
| Расходы на обслуживание ЭВМ | 938,3 |
| Плата за электроэнергию | 371,9 |
| Накладные расходы | 800 |
| Итого | 27357,2 |

## 3.5 Расчет цены программы

Цена программного продукта определяется по стандартной формуле:

|  |  |
| --- | --- |
| Цп=С+П, | (5) |

где Цп – цена программного продукта;

С – себестоимость программного продукта;

П – нормативная прибыль.

Нормативная прибыль составляет 30% от себестоимости:

|  |  |
| --- | --- |
| П=С\*30% , | (6) |

27357,2\*0,3 = 8507,16 рубль.

Таким образом, цена программного продукта равна:

Цп = 27357,2+8507,16 =35564,36 рублей.

Рассчитаем налог на добавленную стоимость:

|  |  |
| --- | --- |
| НДС=Цп \*20% , | (7) |

НДС = 35564,36 \*0,2 = 7112,87 рублей.

С учетом НДС цена программного продукта составляет 42677 рублей.

## 3.6 Оценка экономической эффективности

Экономия от внедрения программы складывается из разности затрат на оформление заказов до внедрения рассматриваемого продукта и их оформление после внедрения.

Данный расчет делается по формуле:

|  |  |
| --- | --- |
| Э=Зстар–Знов , | (8) |

Затраты на оформление старым методом в единицу времени определяются по формуле:

|  |  |
| --- | --- |
| Зстар =Кч\*Сч\*Гд , | (9) |

где Кч – количество часов работы в единицу времени

Сч – стоимость одного часа работы

Гд – коэффициент учитывающий дополнительные затраты (Гд=2)

До применения рассматриваемой программы эту работу выполняли два человека, ответственные за работу с клиентами и оформление заказов.

При периоде в один день:

Кч = 2 \* 7 = 14;

Сч = 82 рубля;

Зстар = 14\*82\*2=2296 рублей.

Затраты на оформление заказов с внедрением программы рассчитывается:

|  |  |
| --- | --- |
| Знов=Тм\*(Цмаш+Сч), | (10) |

Тм – время машинной обработки (6 часов);

Цмаш – цена машинного времени (3,89 руб./час);

Сч – стоимость одного часа работы оператора (82 рубля);

Знов = 6 \* (3,89 + 82)=515,34 рублей.

Вычислим ежегодную экономию:

|  |  |
| --- | --- |
| Эд=Зстар-Знов , | (11) |

Эд = 2296 – 515,34 =1780,66рубля.

Ожидаемая годовая эффективность:

|  |  |
| --- | --- |
| Эгод=Эг-Ен\*Цп , | (12) |

где Ен – коэффициент эффективности капиталовложений (Ен = 0,5 на 2 года)

|  |  |
| --- | --- |
| Эг=254\*Эд , | (13) |

Эг = 254\*1780,66= 452287,64 руб.

Эгод = 452287,64 – 0,5\*5564,36 =434505,64 руб.

Экономическая эффективность разработки определяется по формуле:

|  |  |
| --- | --- |
| Эр=(Эг/Эгод)\*0,5, | (14) |

Эр=(452287,64 /434505,64) \* 0,5 = 0,52.

Так как, Эр больше Ен(0,5) можно сделать вывод, что разработка является экономически целесообразной.

# 4 Организация техники безопасности и охраны труда

## 4.1 Общие положения

К работе программистом допускаются:

* лица не моложе 18 лет, прошедшие обязательные при приеме на работу медицинские освидетельствования на предмет пригодности для работы;
* прошедшие вводный инструктаж по охране труда;
* прошедшие обучение безопасным приемам и методам труда по программе, утвержденной руководителем предприятия (работодателем), разработанной на основе Типовой программы, и прошедшие проверку знаний, в том числе по электробезопасности;
* прошедшие курс обучения на персональном компьютере с использованием конкретного программного обеспечения;
* прошедшие инструктаж по охране труда на конкретном рабочем месте по данной инструкции.

Пользователи персональных компьютеров (ПК) должны проходить периодические медосмотры. К работе с компьютером допускаются только лица, не имеющие противопоказаний.

Рациональный режим труда и отдыха пользователей ПК, установленный с учетом психофизиологической напряженности их труда, динамики функционального состояния систем организма и работоспособности, предусматривает строгое соблюдение регламентированных перерывов.

Основным перерывом является перерыв на обед. При вводе данных, редактировании программ, чтении информации с экрана и т.п. непрерывная продолжительность работы не должна превышать 4-х часов при 8 часовом рабочем дне. Через каждый час работы необходимо вводить регламентированный перерыв на 10 минут, а через 2 часа – 15 минут.

Основным источником вредного воздействия на пользователя является ЖК - монитор компьютера, производящий:

* ЖК-монитор – прямой источник света - неестественные условия для зрения;
* недостаточные диапазоны регулировки яркости – повышенная утомляемость зрения;
* малая глубина черного цвета – сниженная контрастность;
* высокая четкость и структурированность изображения – повышенное зрительное напряжение;
* люминесцентная подсветка – источник ультрафиолетового излучения;
* мерцание изображения – перегрузка зрительного тракта, влияние на биоритмы мозга.

## Требования к помещениям для эксплуатации ПК

* они должны иметь естественное и искусственное освещение;
* естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север и северо-восток, и обеспечивать коэффициент естественной освещенности (КЕО) не ниже 1,5%;
* оконные проемы в помещениях с ВДТ, ЭВМ и ПЭВМ должны быть оборудованы регулируемыми светозащитными устройствами типа: жалюзи, занавеси, внешние козырьки и др.;
* искусственное освещение в помещениях эксплуатации ВДТ, ЭВМ и ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, допускается применение системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). 5 Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Следует ограничить прямую (окна, светильники и др.) и отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения. Необходимо ограничивать неравномерность распределения яркости в поле зрения пользователя, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1 — 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы. При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп мощностью до 250 Вт. Допускается применение ламп накаливания в светильниках местного освещения.

Для освещения помещений следует применять светильники серии ЛПО36 с зеркализованными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Коэффициент запаса для осветительных установок общего освещения должен приниматься равным 1,4. Коэффициент пульсации не должен превышать 5%. Применение светильников без рассеивателей и экранирующих решеток не допускается. Чистку стекол оконных рам и светильников следует проводить не реже двух раз в год, а также заменять перегоревшие лампы.

При выполнении основной работы на ПЭВМ (диспетчерские, операторские, расчетные кабины и посты управления и др. — категория I) уровень шума на рабочем месте не должен превышать 50 дБА.

В помещениях, где работают инженерно-технические работники, осуществляющие лабораторный, аналитический или измерительный контроль (категория II), уровень шума не должен превышать 60 дБА.

В помещениях операторов ЭВМ (без дисплеев) уровень шума не должен превышать 65 дБА (категория III).

На рабочих местах в помещениях для размещения шумных агрегатов вычислительных машин (АЦПУ, принтеры и т. п.) уровень шума не должен превышать 75 дБА (категория IV).

Поверхность пола в помещениях эксплуатации компьютеров должна быть ровной, без выбоин, нескользкой, удобной для очистки и влажной уборки, обладать антистатическими свойствами.

В помещении должны находиться аптечка первой медицинской помощи, углекислотный или порошковый огнетушитель для тушения пожара.

## Требования к организации и оборудованию рабочих мест с ПК

Для обеспечения сохранности и надежной работы ПК необходимо соблюдать следующие правила:

* содержать в чистоте и хорошем состоянии оборудование, магнитные носители;
* при работе на компьютере следует соблюдать аккуратность при использовании клавиатуры, не подвергать ее ударным воздействиям;
* соединительные кабели устройств не следует гнуть, скручивать и придавливать тяжелыми предметами. Это может вызвать нарушение внутренних проводников и привести к неисправностям в работе компьютера.

Для включения компьютера необходимо:

* включить ИБП;
* включить внешние дополнительные устройства (принтер, сканер и др.), если они необходимы в работе;
* включить компьютер.

Запрещается:

* трогать разъемы соединительных кабелей;
* работать при снятом кожухе ПК и любого из устройств, подключенных к ПК;
* прикасаться к тыльной стороне монитора;
* размещать рабочие места у задней поверхности монитора;
* загораживать вентиляционные отверстия монитора;
* работать во влажной одежде и влажными руками;
* вытирать пыль с ПК и монитора при его включенном состоянии;
* оказывать механические воздействия на ПК (падения, удары, поливание жидкостями и т.д.)

## Требования к организации режима работы оператора ПК

Организация работы с ПЭВМ осуществляется в зависимости от вида и категории трудовой деятельности. Виды трудовой деятельности разделяются на 3 группы:

* А – считывание информации с экрана монитора с предварительным запросом;
* Б – работа по вводу информации;
* В – творческая работа в режиме диалога с ПЭВМ.

Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с ПЭВМ, которые определяются: для группы А - по суммарному числу считываемых знаков за рабочую смену, но не более 60 000 знаков за смену; для группы Б - по суммарному числу считываемых или вводимых знаков за рабочую смену, но не более 40 000 знаков за смену; для группы В - по суммарному времени непосредственной работы с ПЭВМ за рабочую смену, но не более 6 часов за смену.

В зависимости от категории трудовой деятельности и уровня нагрузки за рабочую смену при работе с ПЭВМ устанавливается суммарное время регламентированных перерывов. Суммарное время регламентированных перерывов в зависимости от продолжительности работы, вида и категории трудовой деятельности с ПЭВМ.

Соблюдение режима труда и отдыха - одно из требований безопасности при работе с ПЭВМ.

Оценка тяжести и напряженности трудового процесса пользователей ПЭВМ проводится по методикам, утвержденным в установленном порядке.

Оценка тяжести и напряженности работы операторов пультов управления, профессиональная деятельность которых связана с высокой ответственностью, принятием решений в условиях дефицита времени (авиадиспетчеры, железнодорожные диспетчеры, операторы электроустановок и т.д.) должна осуществляться на основе как изучения условий, так и функционального состояния работников с последующей разработкой предложений по рациональной организации труда. Эта работа выполняется научно-исследовательскими организациями, аккредитованными в установленном порядке.

# Заключение

В ходе выполнения выпускной квалификационной работы были решены все поставленные задачи:

* ознакомиться с объектом исследования ООО «Компьютерные системы»;
* основательно изучить предмет исследования по составлению заказов на ремонт техники, а также выявить недостатки ведения сервисного документооборота;
* внести предложения по совершенствованию заведения заявок в сервисной компании ООО «Компьютерные системы».

Результатом проделанной работы является разработанный программный продукт по заведению заявок на ремонт бытовой техники в сервисном центре, который выполняет функцию создания клиентской заявки на ремонт и составления отчётности по заказам.

При выполнении выпускной квалификационной работы был выполнен весь необходимый перечень и объём работ. Приобретены навыки практического решения информационных задач в качестве разработчика программного обеспечения.

Разработанное программное средство имеет удобный и интуитивно понятный интерфейс взаимодействия с пользователем, позволяет повысить качество обработки информации, её достоверность и надёжность. В программе предусмотрена возможность формирования отчётов и использования справочников по получению/сдаче книг читателям. Разработанное приложение соответствует требованиям, предъявляемым к современным программным продуктам.

# Приложение А

Схема потоков данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Отделы  Виды  работы | Клиент | База данных | Администратор |
| 1 | Доступ к базе данных |  | БД |  |
| 2 | Добавление клиента |  |  |
| 3 | Создание заказа |  |  |
| 4 | Просмотр прайс-листа |  |  |
| 5 | Мониторинг готовности заказа |  |  |
| 4 | Вывод акта приёма заказа |  |  |
| 6 | Вывод акта сдачи заказа |  |  |
| 5 | Вывод отчётов по заказам |  |  |
| 6 | Почта |  |  |

# Приложение Б

Алгоритм решения задачи

Главная форма

Форма регистрации

Форма авторизации

Заполнение полей

Проверка

Регистрация успешна

Заполнение полей

Проверка

Приветствие

Прайс-лист

Заказы

Печать акта

+

-

Error

Error

-

+

+

-

Регистрация и авторизация пользователя

Мониторинг цен и просмотр готовности заказа

Вывод

акта

# Список использованных источников

1. Бекаревич Ю.Б. Создание реляционной базы данных и запросов. MS ACCESS 2010/Бекаревич Ю.Б., Пушкина Н.В.//Создание таблиц базы данных: СП.: СПбГУЭФ, 2017. - С.9-42.
2. Васильев, А. C#. Объектно-ориентированное программирование. Учебный курс / А. Васильев. - СПб.: Питер, 2017. - 320 c.
3. Вендров, А.М. Современные методы и средства проектирования информационных систем / А.М. Вендров - М.: Финансы и статистика, 2018. – 65 с.
4. Гагарина Л.Г. Разработка и эксплуатация автоматизированных информационных систем/Киселев Д.В., Федотова Е.Л., Гагарина Л.Г.//Автоматизированные информационные системы: М.: ФОРУМ: ИНФРА-М, 2017. - С.67-80.
5. Гвоздева В.А. Основы построения автоматизированных информационных систем/Гвоздева В.А., Лаврентьева И.Ю.//АИС: основные понятия и определения: М.: ИД «ФОРУМ» - ИНФРА-М, 2017. - С.14 .
6. Гуриков, С.Р. Введение в программирование на языке Visual C#: Учебное пособие / С.Р. Гуриков. - М.: Форум, 2018. - 128 c.
7. Емельянова Н.З. Основы построения автоматизированных информационных систем/Партыка Т.Л., Попов И.И., Емельянова Н.З.//Информация и информационные системы. Основные понятия и классификация: М.: Инфра - М, 2017. - С.9-14.
8. Стив Макконнелл, Совершенный код: учебное пособие / С. Макконнелл. – П.: BHV, 2017
9. Эндрю Троелсен, Язык программирования C# 5.0 и платформа .NET 4.5: учебник по языку C# / Э. Троелсен. – М.: Вильямс, 2015
10. Cyberforum [Электронный ресурс], - http://www.cyberforum.ru/

# Ссылочные нормативные документы

* 1. ГОСТ 19.201-78. Техническое задание. Требования к содержанию и оформлению. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 3 с.
  2. ГОСТ 19.202-78. Спецификация – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 3 с.
  3. ГОСТ 19.301-79. Программа и методика испытаний. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 2 с.
  4. ГОСТ 19.401-78. Текст программы. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 1 с.
  5. ГОСТ 19.402-78. Описание программы. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 2 с.
  6. ГОСТ 19.404-79. Пояснительная записка. Требования к содержанию и оформлению. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 2 с.
  7. ГОСТ 19.502-78. Описание применения. Требования к содержанию и оформлению – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 2 с.
  8. ГОСТ 19.503-79. Руководство системного программиста. Требования к содержанию и оформлению. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 2 с.
  9. ГОСТ 19.504-79. Руководство программиста. Требования к содержанию и оформлению. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 2 с.
  10. ГОСТ 19.505-79. Руководство оператора. Требования к содержанию и оформлению. – Введ. 01.01.80. – М.: Изд-во стандартов, 1987 . – 2 с.
  11. ГОСТ 2.105-95. Общие требования к текстовым документам. Введ. 01.07.1996.

1. Искусственный интеллект [↑](#footnote-ref-1)
2. Система управления базами данных [↑](#footnote-ref-2)