МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Филиал федерального государственного бюджетного образовательного учреждения высшего образования

«Национальный исследовательский университет «МЭИ» в г. Смоленске

Кафедра вычислительной техники

УДК: 004.3(079.2)

М–60

**Милославский Сергей Александрович**

**Автоматизированное рабочее место сервисного центра**

Выпускная квалификационная работа на соискание квалификации

**Бакалавр**

Направление подготовки

**09.03.01 – Информатика и вычислительная техника**

Профиль подготовки

**Программное обеспечение средств вычислительной**

**техники и автоматизированных систем**

Студент группы ПО1-19 С.А. Милославский

Руководитель

к.т.н., доцент В.А. Тихонов

*Допускается к защите*

Заведующий кафедрой

д.т.н., профессор А.С. Федулов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2023 г.

Смоленск – 2023

АННОТАЦИЯ

Выпускная квалификационная работа на соискание квалификации «бакалавр» по направлению 09.03.01 – Информатика и вычислительная техника. Работу выполнил студент группы ПО1-19 Милославский Сергей Александрович. Тема выпускной квалификационной работы: «**Автоматизированное рабочее место сервисного центра**». Руководитель: к.т.н., доцент. Тихонов Владимир Александрович.

Выпускная квалификационная работа изложена на 132 страницах, из них 48 страниц основного текста, состоит из трех глав, заключения, списка литературы из 16 наименований и двух приложений.

Данная выпускная квалификационная работа посвящена разработке автоматизированного рабочего места операторов и мастеров в сервисном центре по ремонту электронной техники.

Целью этой работы является улучшение рабочего процесса и производительности сервисного центра, обеспечение быстрого доступа к информации, эффективное управление запросами клиентов и создание отчетов.

В работе описана предметная область, обоснована актуальность разработки программы. Выполнено проектирование, разработка и тестирование программы.

Ключевые слова: автоматизация, оператор, мастер.

ABSTRACT

Final qualifying work for the qualification «Bachelor» in the direction of 09.03.01 - Informatics and Computer Engineering. The work was performed by a student of the group PO1-19 Miloslavsky Sergey Alexandrovich. The theme of the final qualifying work: «Automated workplace of the service center». Head: Ph.D., Associate Professor. Tikhonov Vladimir Alexandrovich

The final qualifying work is presented on 132 pages, of which 48 pages of the main text, consists of three chapters, a conclusion, a bibliography of 16 titles and two appendices.

This final qualifying work is devoted to the development of an automated workplace for operators and craftsmen in a service center for the repair of electronic equipment.

The purpose of this work is to improve the workflow and productivity of the service center, provide quick access to information, effectively manage customer requests and create reports.

The paper describes the subject area, substantiates the relevance of the development of the program. The design, development and testing of the program has been completed.

Key words: automation, operator, master.

**ОГЛАВЛЕНИЕ**

[Введение 6](#_Toc137400629)

[1. АНАЛИЗ ЗАДАЧ, ПРОГРАММНЫХ СРЕДСТВ И ОБОСНОВАНИЕ РАЗРАБОТКИ ПО 8](#_Toc137400630)

[1.1 Анализ технического задания 8](#_Toc137400631)

[1.2 Описание предметной области 9](#_Toc137400632)

[1.3 Функционально проектирование 11](#_Toc137400633)

[1.4 Обзор аналогов 16](#_Toc137400634)

[1.4.1 СК24 16](#_Toc137400635)

[1.4.2 ProMaster 17](#_Toc137400636)

[1.4.3 ServiceMP 19](#_Toc137400637)

[1.4.3 Краткий вывод по аналогам 21](#_Toc137400638)

[1.5 Выбор средств разработки 21](#_Toc137400639)

[1.6 Постановка задачи 23](#_Toc137400640)

[1.7 Выводы по главе 25](#_Toc137400641)

[2. Проектирование ПО 27](#_Toc137400642)

[2.1 Определение вариантов использования разрабатываемых программных средств 27](#_Toc137400643)

[2.2 Проектирование БД 32](#_Toc137400644)

[2.3 Интерфейс пользователя 35](#_Toc137400645)

[2.4 Выводы по главе 36](#_Toc137400646)

[3 РЕАЛИЗАЦИЯ АРМ 37](#_Toc137400647)

[3.1 Реализация ПО 37](#_Toc137400648)

[3.2 Алгоритмы 38](#_Toc137400649)

[3.3 Тестирование ПО 42](#_Toc137400650)

[3.4 Выводы по главе 44](#_Toc137400651)

[Заключение 45](#_Toc137400652)

[перечень сокращений и условных обозначений 46](#_Toc137400653)

[список использованных источников 47](#_Toc137400654)

[Приложение А 49](#_Toc137400655)

[Приложение б 55](#_Toc137400656)

[Приложение В 128](#_Toc137400657)

Введение

В постоянно развивающейся электронной промышленности сервисные центры играют жизненно важную роль в предоставлении услуг по ремонту и техническому обслуживанию широкого спектра электронного оборудования. Эффективная и точная работа этих сервисных центров имеет решающее значение для удовлетворения потребностей клиентов и поддержания конкурентных преимуществ. С развитием технологий интеграция автоматизации стала многообещающим решением для повышения производительности и результативности таких ремонтных предприятий.

Выпускная квалификационная работа посвящена разработке автоматизированного рабочего места для операторов и мастеров в сервисном центре, специализирующемся на ремонте электронного оборудования.

Цель состоит в том, чтобы создать ПО, которое оптимизирует рабочий процесс, улучшает обслуживание клиентов и, в конечном счете, повышает прибыльность сервисного центра. Используя передовые технологии, включая автоматизированные системы, робототехнику и цифровые инструменты, это автоматизированное рабочее место призвано упростить операции, уменьшить количество человеческих ошибок и повысить общую эффективность.

Выбор этой темы обусловлен признанием важности сервисных центров в электронной промышленности и потенциальных преимуществ, которые автоматизация может привнести в их работу. Автоматизированное рабочее место способно революционизировать традиционные процессы ремонта, обеспечивая более быструю диагностику, эффективное устранение неполадок, точный ремонт и надежный контроль качества. Эти улучшения не только повышают способность сервисного центра соответствовать ожиданиям клиентов, но и способствуют повышению удовлетворенности и лояльности клиентов.

В соответствии с государственным общеобразовательным стандартом высшего образования в области информатики и вычислительной техники, этот проект направлен на применение теоретических знаний и практических навыков для разработки ПО, которое преобразует работу ремонтно-сервисного центра. Используя удобный интерфейс и нормализованную структуру базы данных, система призвана обеспечить быстрый доступ к информации, облегчить управление запросами клиентов и генерировать исчерпывающие отчеты.

Для обеспечения успешной работы автоматизированного рабочего места были определены конкретные технические требования. Эти требования охватывают производительность, структуру и функции, надежность, информационную безопасность, сохранение информации в случае аварий и соответствующую документацию по программному обеспечению. Программа будет разработана с использованием языка C# и будет включать в себя как серверный, так и клиентский компоненты.

Тестирование и валидация разработанной системы будут проводиться в соответствии с установленными стандартами для обеспечения ее функциональности и надежности. Документация будет включать техническое задание, UML-диаграммы, экранные формы и результаты тестирования, обеспечивая всестороннюю поддержку при внедрении и дальнейшем обслуживании автоматизированного рабочего места.

В целом, успешное внедрение автоматизированного рабочего места в сервисном центре по ремонту электронного оборудования потенциально может произвести революцию в способах проведения ремонтных работ. Используя автоматизацию, операторы и мастера могут работать более эффективно, улучшать качество предоставляемых услуг и вносить свой вклад в прибыльность и конкурентоспособность сервисного центра в быстро развивающейся отрасли.

1. АНАЛИЗ ЗАДАЧ, ПРОГРАММНЫХ СРЕДСТВ И ОБОСНОВАНИЕ РАЗРАБОТКИ ПО

1.1 Анализ технического задания

При анализе технического задания становится очевидным, что разработка программы АРМ сервисного центра крайне необходима. Текущая операционная схема неэффективна и страдает от ручных процессов, что приводит к задержкам, ошибкам и снижению производительности. Ограничения программы включают неадекватную доступность информации, неэффективное управление ресурсами и уязвимости в безопасности данных.

Для решения этих проблем ПО АРМ направлено на:

* оптимизацию рабочего процесса;
* повышение доступности информации;
* оптимизацию распределения ресурсов;
* укрепление безопасности данных.

Архитектура системы будет следовать подходу «толстого клиента» с упором на обработку данных и управление ими на стороне клиента.

В технических требованиях особое внимание уделяется:

* удобному интерфейсу;
* нормализованной структуре базы данных;
* многопользовательскому доступу;
* конкретным характеристикам программного и аппаратного обеспечения.

В целом анализ технического задания подчеркивает потребность в автоматизированной системе рабочих мест, которая:

* устраняет текущие ограничения
* оптимизирует операции
* улучшает обслуживание клиентов и повышает прибыльность сервисного центра.

1.2 Описание предметной области

Предметная область АРМ сервисного центра по ремонту электронной техники охватывает управление и оптимизацию процессов ремонта, обслуживание клиентов и общую эффективность работы. В данном контексте сервисный центр выступает в роли хаба по ремонту различных электронных устройств, включая компьютеры, смартфоны, бытовую технику и другое электронное оборудование.

ПО направлено на оптимизацию и автоматизацию действий, выполняемых в сервисном центре, обеспечивая бесперебойный поток операций с момента, когда клиент запрашивает ремонт, до момента возврата устройства. Он включает в себя такие задачи, как регистрация клиентов, отслеживание ремонта, управление запасами деталей, назначение технических специалистов и создание документов о приемке и выдаче.

Эффективность и точность являются критическими аспектами предметной области. Программа автоматизированного рабочего места направлена на повышение производительности сервисного центра за счет сокращения времени ремонта, сведения к минимуму ошибок при вводе и отслеживании данных, оптимизации распределения ресурсов и повышения общей удовлетворенности клиентов.

Кроме того, предметная область фокусируется на обеспечении безопасного и надежного управления данными. Программа должна включать механизмы аутентификации для предотвращения несанкционированного доступа, реализовывать процедуры резервного копирования и восстановления данных для защиты от потери информации и придерживаться строгих протоколов защиты данных для защиты конфиденциальной информации клиентов.

В целом предметная область системы автоматизированных рабочих мест для сервисного центра по ремонту электронного оборудования охватывает бесшовную интеграцию технологий, управления процессами и обслуживания клиентов для обеспечения эффективного ремонта, исключительного качества обслуживания клиентов и повышения прибыльности в отрасли ремонта электронного оборудования.

Разрабатываемое ПО предназначено для трех категорий пользователей:

* инженер;
* администратор/менеджер;
* директор.

Готовое ПО должно выполнять следующие функции:

Директор:

* назначение сотрудников;
* назначение прав сотруднику;
* формирование и печать статистической отчетности;
* регистрация заявок клиентов;
* добавление новых клиентов;
* добавление информации об устройстве;
* заказ комплектующих;
* добавление информации о комплектующих;
* просмотр остатков комплектующих на складе;
* оформление накладных актов приема и выдачи устройства;
* добавление информации о ремонте оборудования;
* изменение статуса заявки.

Администратор/Менеджер:

* регистрация заявок клиентов;
* добавление новых клиентов;
* добавление информации об устройстве;
* заказ комплектующих;
* добавление информации о комплектующих;
* просмотр остатков комплектующих на складе;
* накладных актов приема и выдачи устройства;
* изменение статуса заявки.

Инженер:

* добавление информации о диагностике устройства;
* добавление информации о ремонте оборудования
* изменение статуса заявки.

На основе описания предметной области, можно выделить следующие бизнес-процессы:

1. Регистрация заявки:

* регистрация клиента;
* внесение информации о неисправности.

1. Диагностика:

* определение типа неисправности;
* подробный анализ неисправности;
* составление заявки на комплектующие.

1. Заказ комплектующих:

* проверка наличия комплектующих на складе;
* заказ комплектующих у поставщика;
* оплата комплектующих;
* поступление на склад;
* поступление комплектующих к инженеру со склада.

1. Ремонт оборудования:

* устранение неисправности оборудования;
* тестирование работоспособности отремонтированного оборудования;
* подготовка отчетной документации по ремонту.

1.3 Функционально проектирование

Для проектирования бизнес-процессов используется контентная диаграмма, которая позволяет быстро, кратко и емко описать назначение и границы программы, выявить и устранить коллективные расхождения в их понимании. Кроме этого, контекстная диаграмма служит источником для быстрой генерации первичного набора системных функциональных требований при необходимости проектирования программы от бизнес-модели, бизнес-требований, модели деятельности организации, требований заинтересованных лиц, модели использования.

На рисунке 1.1 представлена контекстная диаграмма, описывающая деятельность сервисного центра.

В своей деятельности сервисный центр руководствуется уставом и законодательными актами РФ (Законодательство):

1. Конституция РФ;
2. Федеральные кодексы РФ (Гражданский Уголовный и т. д.);
3. Федеральные законы;
4. Подзаконные акты;
5. Нормативно-правовые акты субъектов РФ;
6. Муниципальные правовые акты;
7. Налоговый кодекс РФ.



Рисунок 1.1 – Контекстная диаграмма деятельности сервисного центра по ремонту электронного оборудования

Декомпозируем контекстную диаграмму деятельности сервисного центра по ремонту электронного оборудования. Под декомпозицией диаграммы понимается более детальный процесс представления. Каждая функция, представленная на диаграмме отдельным блоком, может быть описана на другой диаграмме, расположенный на один уровень ниже иерархии.

В декомпозиции контекстной диаграммы декомпозируемый модуль всегда делится не более чем на семь модулей. Каждый из этих модулей может быть также декомпозирован подобным образом для более детального представления.

На рисунке 1.2 представлена диаграмма декомпозиции контекстной диаграммы.

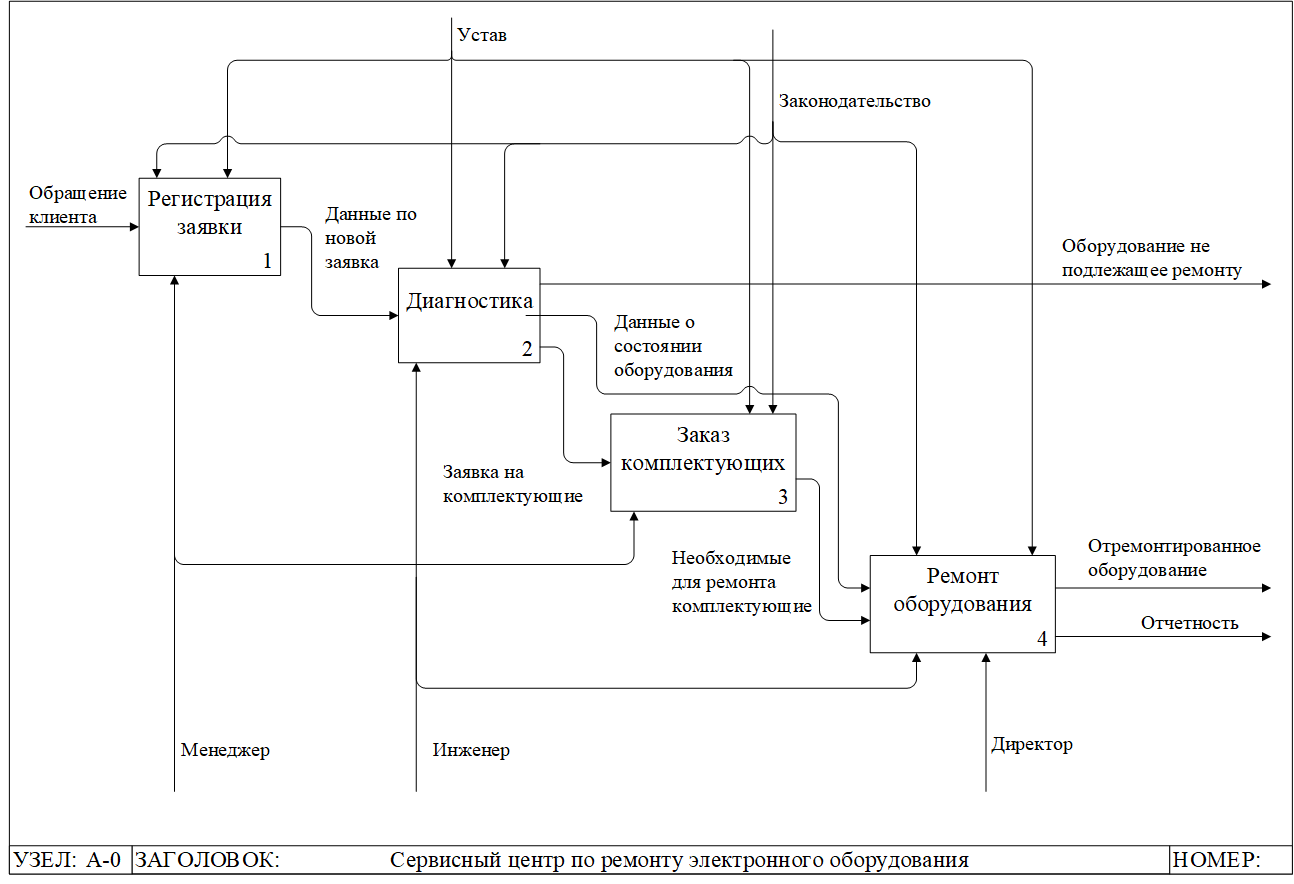


Рисунок .2 – Диаграмма декомпозиции контекстной диаграммы

Ниже представлены диаграммы декомпозиции для каждого бизнес-процесса.

На рисунке 1.3 представлена диаграмма декомпозиции бизнес-процесса «Регистрация заявки».

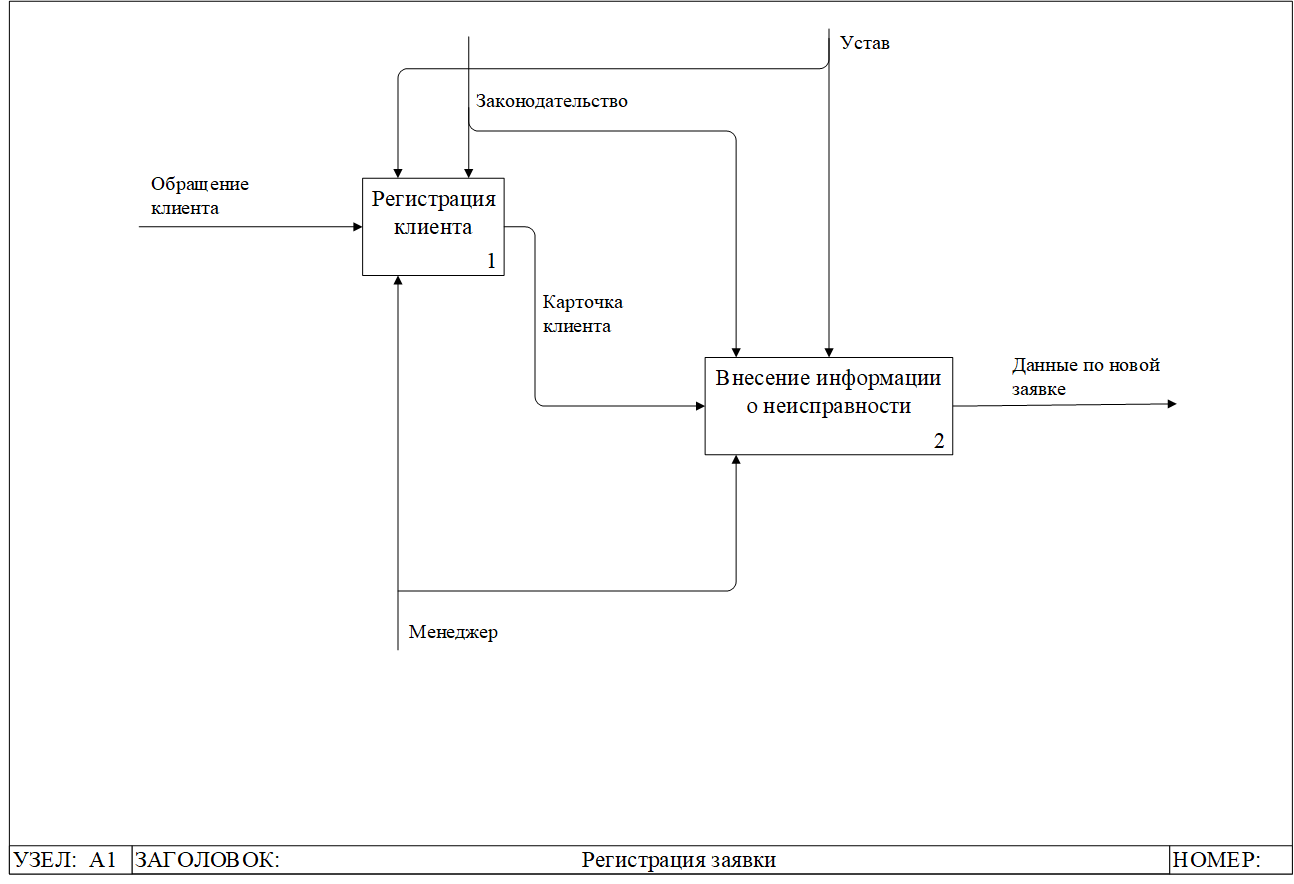


Рисунок 1.3 – Диаграмма декомпозиции бизнес-процесса «Регистрация заявки»

На рисунке 1.4 представлена диаграмма декомпозиции бизнес-процесса «Диагностика».

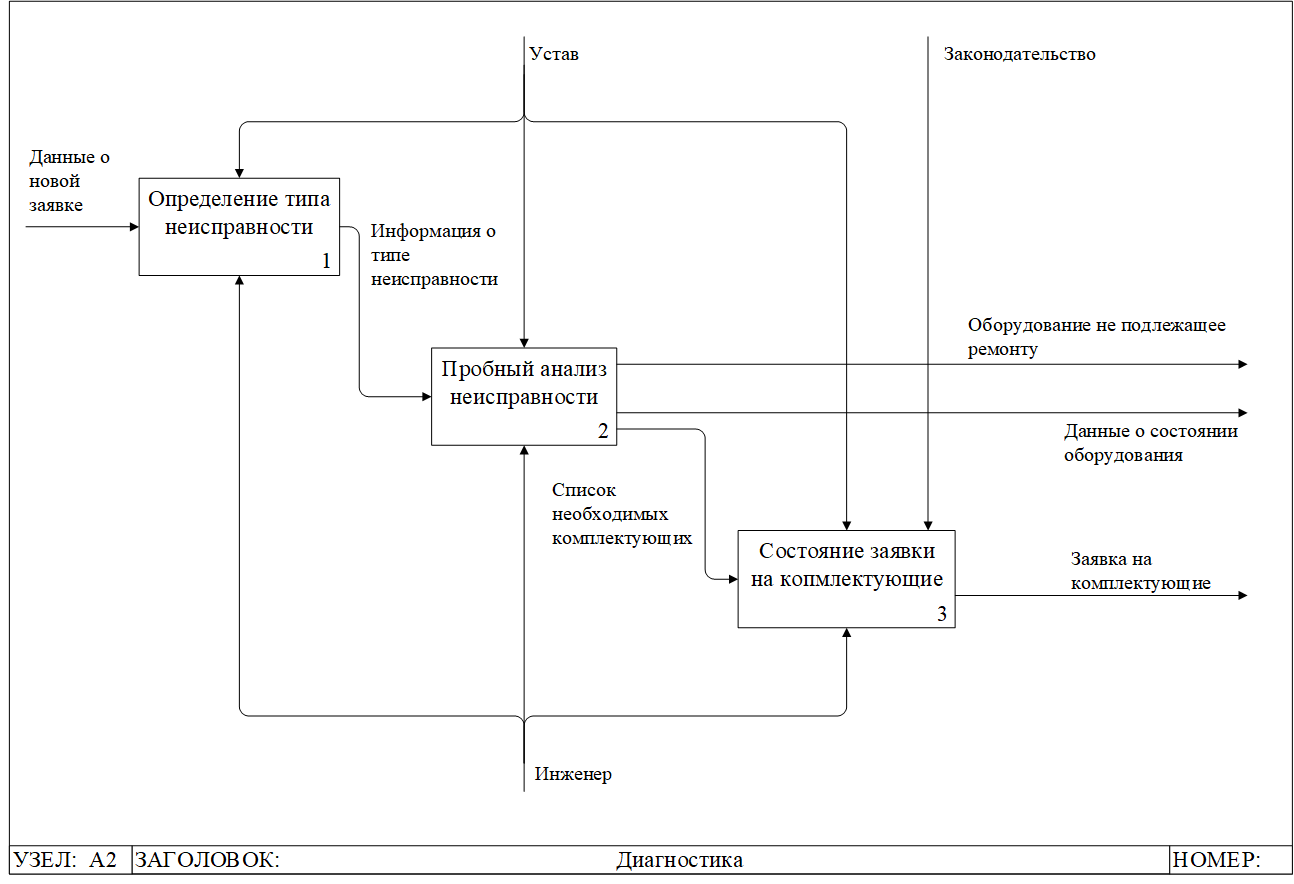


Рисунок 1.4 – Диаграмма декомпозиции бизнес-процесса «Диагностика»

На рисунке 1.5 наглядно изображена диаграмма декомпозиции бизнес-процесса «Заказ комплектующих».

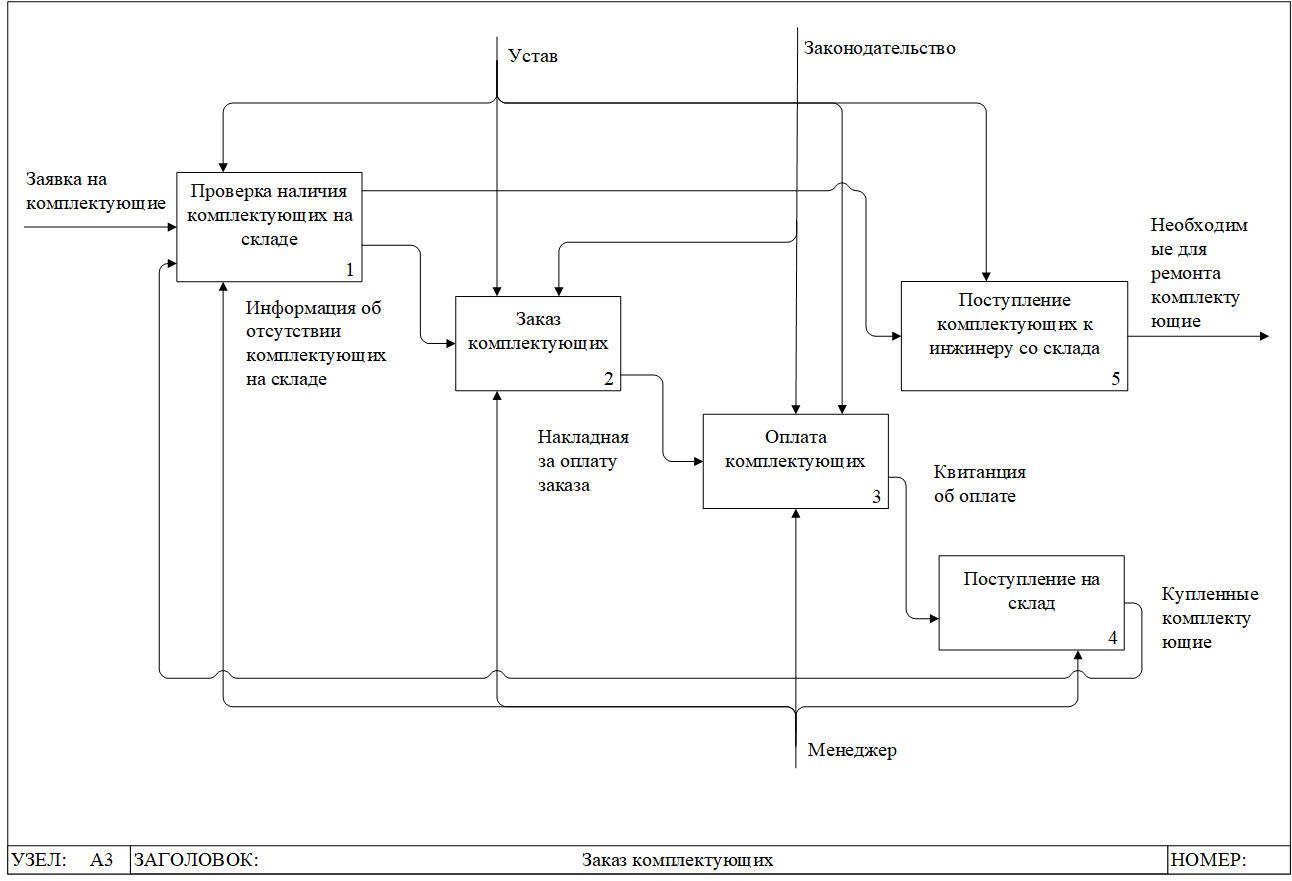


Рисунок 1.5 – Диаграмма декомпозиции бизнес-процесса «Заказ комплектующих»

На рисунке 6 изображена диаграмма декомпозиции бизнес-процесса «Ремонт оборудования».

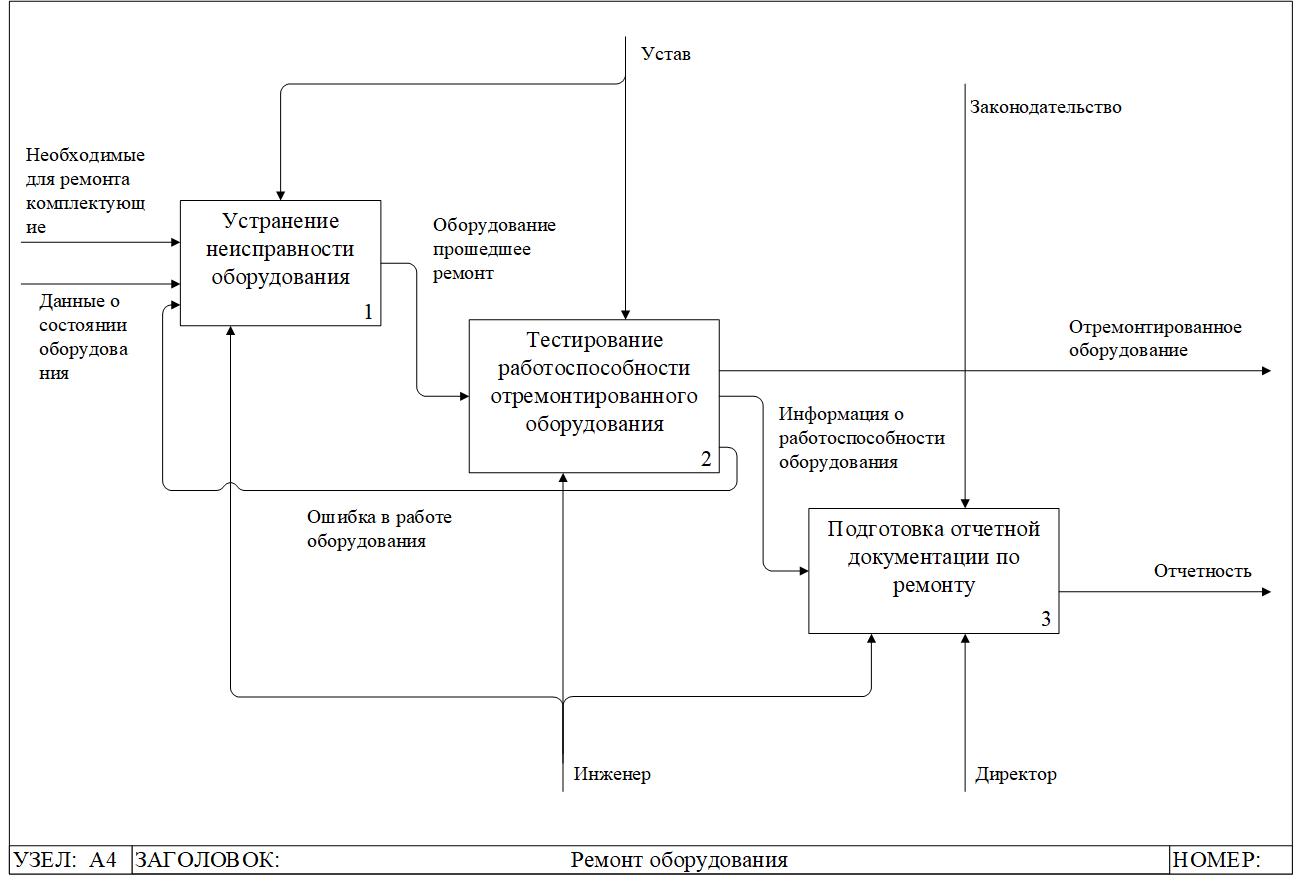


Рисунок 1.6 – Диаграмма декомпозиции бизнес-процесса «Ремонт оборудования».

1.4 Обзор аналогов

Сервисным центрам и ремонтным мастерским требуется эффективное и надежное программное обеспечение для оптимизации их операций, управления запросами клиентов, отслеживания запасов и создания отчетов.

1.4.1 СК24

SC24 — комплексное программное решение, разработанное специально для сервисных центров и ремонтных мастерских [12]. Он предлагает ряд функций для оптимизации операций, повышения эффективности и улучшения обслуживания клиентов. Программа предоставляет модули для управления информацией о клиентах, отслеживания ремонта, планирования встреч, создания счетов и мониторинга запасов (рисунок 1.7).

SC24 также включает возможности отчетности для отслеживания производительности и анализа бизнес-показателей. Благодаря удобному интерфейсу и настраиваемым параметрам SC24 удовлетворяет специфические потребности сервисных центров и ремонтных мастерских, помогая им оптимизировать рабочий процесс и предоставлять исключительные услуги.

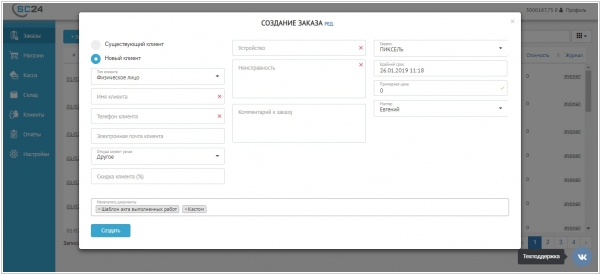


Рисунок 1.7 – Форма программы SC24

Преимущества:

* SC24 предоставляет простой и интуитивно понятный интерфейс, в котором может легко ориентироваться нетехнический персонал;
* предлагает широкий спектр функций, таких как управление клиентами, отслеживание ремонта, инвентаризация запасных частей и отчетность, охватывающие основные аспекты работы сервисного центра;
* SC24 завоевал признание и доверие в отрасли, что делает его популярным выбором среди сервисных центров.

Недостатки:

* программе может не хватать гибкости в плане настройки, что ограничивает ее способность адаптироваться к конкретным требованиям сервисного центра;
* в зависимости от цикла разработки SC24 может не включать в себя новейшие технологические достижения, что может повлиять на эффективность и функциональность;
* доступность своевременной поддержки и регулярных обновлений может варьироваться, что потенциально может привести к проблемам, связанным с исправлением ошибок и совместимостью с развивающимися технологиями.

1.4.2 ProMaster

ProMaster — еще одна популярная программа для сервисных центров и ремонтных мастерских [13]. Он предлагает набор инструментов для управления всеми аспектами процесса ремонта, от приема клиентов до завершения обслуживания. ProMaster включает модули для управления клиентами, создания рабочих заданий, назначения технических специалистов, отслеживания запасных частей и выставления счетов (рисунок 1.8).

Программа также интегрируется с популярным программным обеспечением для бухгалтерского учета, что позволяет легко управлять финансами. ProMaster предоставляет обновления состояния ремонта в режиме реального времени, обеспечивая эффективную связь с клиентами. Кроме того, ProMaster предлагает функции отчетности и аналитики, чтобы получить представление об эффективности бизнеса и определить области для улучшения.

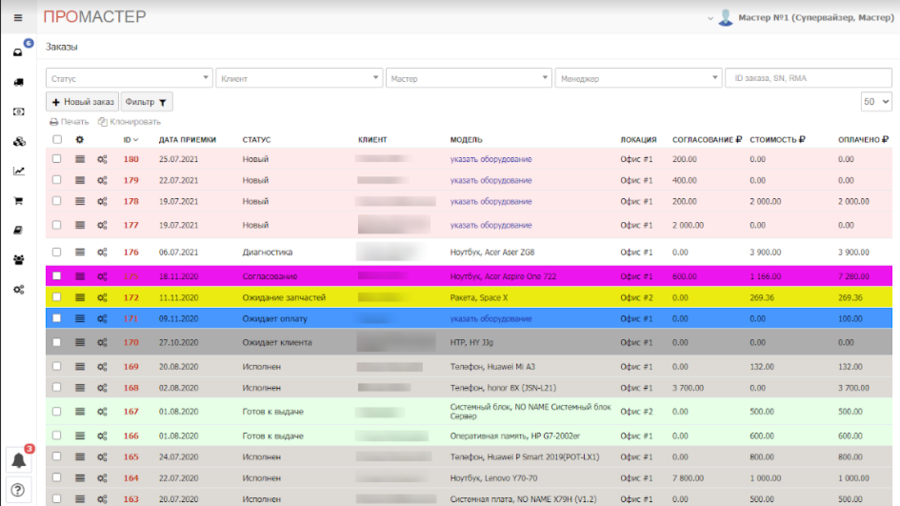


Рисунок 1.8 – Форма программы ProMaster

Преимущества:

* ProMaster предлагает широкий спектр функций, включая управление клиентами, отслеживание ремонта, выставление счетов и отчетность, удовлетворяя разнообразные потребности сервисных центров;
* Программа разработана с учетом роста и расширения сервисных центров, что позволяет управлять несколькими филиалами и увеличивать объем ремонтных работ;
* ProMaster может интегрироваться с различными внешними системами и инструментами, обеспечивая беспрепятственный обмен данными и повышая общую эффективность работы.

Недостатки:

* Обширные возможности ProMaster могут привести к трудностям в обучении пользователей, особенно нетехнического персонала;
* Лицензионные сборы и текущие расходы на техническое обслуживание, связанные с ProMaster, могут быть относительно высокими, что потенциально влияет на доступность для небольших сервисных центров;
* сервисные центры, полагающиеся на ProMaster, могут столкнуться с проблемами, если поставщик прекратит поддержку или не сможет своевременно предоставлять обновления, что потенциально может привести к сбоям в работе.

1.4.3 ServiceMP

ServiceMP — это программа управления обслуживанием, разработанная специально для малых и средних ремонтных мастерских и сервисных центров [14]. Программа предлагает ряд функций для оптимизации операций и повышения удовлетворенности клиентов. ServiceMP включает модули для управления информацией о клиентах, создания рабочих заданий, отслеживания ремонта и планирования встреч (рисунок 1.9).

Программа также предоставляет возможности управления запасами для отслеживания запасных частей и расходных материалов. ServiceMP фокусируется на простоте и удобстве использования, позволяя владельцам ремонтных мастерских и техническим специалистам эффективно управлять своими повседневными операциями без ненужных сложностей. Благодаря доступной цене и интуитивно понятному интерфейсу, ServiceMP является популярным выбором для небольших ремонтных предприятий.

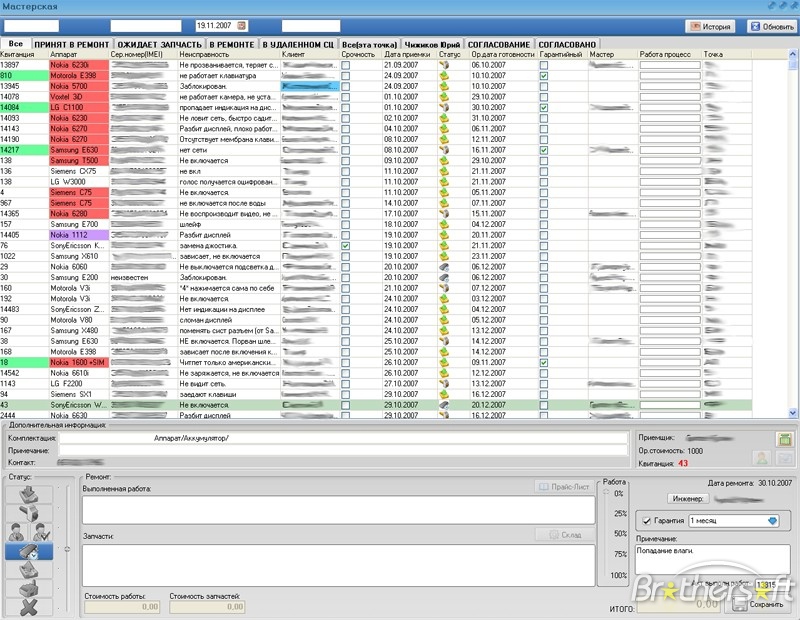


Рисунок 1.9 – Форма программы ServiceMP

Преимущества:

* ServiceMP предлагает целостный подход к управлению сервисным центром, включая управление клиентами, планирование заданий, контроль запасов и отчетность;
* Программа обеспечивает мобильный доступ, позволяя техническим специалистам получать доступ к информации и обновлять ее на ходу, повышая оперативность и эффективность;
* ServiceMP интегрируется с бухгалтерским программным обеспечением, упрощая процессы выставления счетов, выставления счетов-фактур и финансового отслеживания.

Недостатки:

* ServiceMP может потребовать значительных затрат времени и усилий от пользователей, чтобы полностью освоить его функциональные возможности и эффективно использовать его возможности;
* возможности настройки программы могут быть ограничены, что не позволяет сервисным центрам адаптировать ее к своим конкретным потребностям или отраслевым требованиям;
* ServiceMP может иметь ограничения в масштабировании, чтобы приспособиться к росту и расширению сервисных центров с несколькими филиалами или большими объемами ремонта.

1.4.3 Краткий вывод по аналогам

Учитывая преимущества и недостатки существующих аналогов, разработка АРМ, специально адаптированной к потребностям сервисного центра, весьма актуальна. Возможности настройки позволят программе точно соответствовать уникальным рабочим процессам и требованиям сервисного центра, оптимизируя операционную эффективность и обслуживание клиентов.

Используя новейшие технологии и внедряя лучшие отраслевые практики, разработанная программа может преодолеть ограничения аналогов, обеспечивая гибкость, масштабируемость и адаптируемость к будущим потребностям. Кроме того, собственная разработка программы обеспечивает больший контроль, позволяя своевременно обновлять ее, постоянно совершенствовать и лучше интегрировать с существующей инфраструктурой сервисного центра.

1.5 Выбор средств разработки

В соответствии с техническим заданием на разработку автоматизированного рабочего места для операторов и мастеров в сервисном центре по ремонту электронного оборудования важно оценить существующие средства разработки, соответствующие требованиям проекта.

Ниже освещаются некоторые соответствующие инструменты разработки и их пригодность для выполнения указанных технических требований:

IDE:

IDE предоставляют комплексную среду разработки, которая включает в себя возможности кодирования, отладки и тестирования. Популярные IDE, такие как Microsoft Visual Studio, Eclipse и JetBrains Rider, поддерживают язык программирования C#, что делает их подходящими для внедрения системы автоматизированного рабочего места [6].

СУБД:

Для выполнения требований по работе с базами данных и обеспечению быстрого доступа к данным сервер MySQL является надежной и широко используемой СУБД [4]. Его стабильность, производительность и поддержка многопользовательских сред делают его подходящим для обработки данных, требуемых сервисным центром.

Фреймворки UI:

Для создания удобного интерфейса можно рассмотреть такие фреймворки, как Windows Presentation Foundation (WPF) или Windows Forms (WinForms) [3]. Эти фреймворки предлагают богатый набор элементов управления и вариантов дизайна, облегчая разработку интуитивно понятных и визуально привлекательных интерфейсов для нетехнического персонала.

Системы контроля версий:

Системы контроля версий, такие как Git, имеют решающее значение для поддержания целостности кода, облегчения совместной работы и управления изменениями в проекте. Git с его распределенной природой и поддержкой ветвлений и слияний предоставляет эффективное решение для управления исходным кодом, гарантируя, что процесс разработки остается организованным и оптимизированным.

Крайне важно учитывать совместимость выбранных средств разработки с рекомендуемыми спецификациями программного и аппаратного обеспечения. Обеспечение совместимости инструментов с требуемыми операционными системами, фреймворками и библиотеками имеет важное значение для бесперебойного процесса разработки.

Кроме того, в соответствии с техническим заданием рекомендуется придерживаться отраслевых стандартов, таких как ISO и ГОСТ, на протяжении всего процесса разработки. Выбранные средства разработки должны поддерживать эти стандарты и обеспечивать надлежащее соответствие с точки зрения тестирования, документации и обеспечения качества.

Тщательно оценив и выбрав подходящие инструменты разработки, проект может быть выполнен эффективно, в соответствии с техническими требованиями и обеспечить успешное внедрение системы автоматизированного рабочего места для сервисного центра.

1.6 Постановка задачи

Рассматриваемая проблема в ВКР заключается в неэффективности и отсутствии автоматизации на рабочих местах оператора и мастера в сервисном центре по ремонту электронного оборудования. Существующие ручные процессы и разрозненные системы препятствуют способности сервисного центра эффективно управлять своей деятельностью, обеспечивать качественное обслуживание клиентов и максимизировать прибыльность.

Операторы и мастера сталкиваются с проблемами быстрого доступа к соответствующей информации, управления запросами клиентов и эффективного формирования отчетов. Отсутствие отлаженной и автоматизированной системы приводит к задержкам, ошибкам и неоптимальному использованию ресурсов. Отсутствие удобного интерфейса еще больше усугубляет проблему, затрудняя нетехническому персоналу эффективную навигацию по системе и ее использование.

Кроме того, избыточность данных и несогласованные методы управления данными препятствуют способности сервисного центра вести точные записи и анализировать производительность. Отсутствие многопользовательского доступа и различных уровней разрешений ограничивает совместную работу и препятствует бесперебойному обмену информацией в сервисном центре.

Безопасность и защита данных создают дополнительные проблемы, поскольку несанкционированный доступ к информации о клиентах и потенциальная потеря данных могут привести к юридическим последствиям и репутации сервисного центра.

Таким образом, ручное и неэффективное рабочее место оператора и мастера в сервисном центре снижает производительность, затрудняет обслуживание клиентов и ограничивает прибыльность центра. Автоматизированная программа необходима для решения этих проблем, оптимизации операций, улучшения управления данными, предоставления удобного интерфейса, обеспечения безопасности данных и повышения общей эффективности работы сервисного центра.

Исходя из всего вышесказанного, формируется следующая задача: разработать ПО, предоставляющую вышеописанный функционал.

Разработка ПО будет выполнена на языке программирования C# с использованием:

* IDE: Microsoft Visual Studio 2022. Эта IDE предоставляет полный набор функций для кодирования, отладки и тестирования приложений на C#. Visual Studio 2022 предлагает интуитивно понятный интерфейс, расширенные возможности отладки и обширные библиотеки и фреймворки, что делает ее идеальным выбором для разработки системы автоматизированного рабочего места;
* СУБД: MySQL. Эта СУБД будет использоваться для обработки хранения и извлечения данных внутри системы. MySQL — это надежная и широко распространенная СУБД, известная своей производительностью, масштабируемостью и надежностью, что делает ее подходящим выбором для управления базой данных сервисного центра [5];
* UI: Windows Forms (Net Framwork). UI системы автоматизированного рабочего места будет разработан с использованием Windows Forms, которая представляет собой графический интерфейс пользователя, предоставляемый .NET Framework. Windows Forms предлагает богатый набор элементов управления и вариантов оформления для создания интуитивно понятных и удобных в использовании интерфейсов, позволяющих нетехническому персоналу легко ориентироваться в системе и взаимодействовать с ней [2].

1.7 Выводы по главе

В данной главе был выполнен анализ предметной области, где было рассмотрено:

* анализ технического задания, позволивший получить четкое представление о целях, объеме и технических требованиях проекта;
* описание предметной области проливает свет на сферу применения и значимость автоматизированной системы рабочего места в контексте ремонта электронного оборудования;
* обзор аналогов, давший представление о существующих решениях в отрасли, позволяя выявить лучшие практики и потенциальные области для улучшения;
* обзор существующих инструментов разработки, давший оценку доступного программного обеспечения и технологий, которые соответствуют требованиям проекта;
* в постановке задачи была определена главная проблема, с которой столкнулся проект, а именно отсутствие программы автоматизированного рабочего места, которая оптимизировала бы работу ремонтного центра, улучшала обслуживание клиентов и повышала прибыльность.

Принимая во внимание все эти моменты, разработка индивидуальной программы автоматизированного рабочего места на языке программирования C# с использованием Microsoft Visual Studio 2022, MySQL в качестве СУБД, Windows Forms для пользовательского интерфейса и следование клиентской архитектуре «толстый клиент» имеет решающее значение для решения выявленной проблемы и удовлетворения конкретных требований сервисного центра.

Эта программа позволит эффективно управлять рабочим процессом, повысит безопасность данных, оптимизирует процессы ремонта и, в конечном счете, повысит общую производительность и прибыльность сервисного центра.

2. Проектирование ПО

2.1 Определение вариантов использования разрабатываемых программных средств

Перечислим основные варианты использования разрабатываемых программных средств с помощью диаграммы вариантов использования (рисунок 2.1) [15].



Рисунок 2.1 – Диаграмма вариантов использования

Краткое описание варианта использования «Вход в систему» представлено в таблице 2.1.

Таблица 2.1 – Краткое описание варианта использования «Вход в систему»

|  |  |
| --- | --- |
| Название | Вход в систему |
| Действующие лица | Оператор, мастер, кладовщик |
| Цель | Войти в систему |
| Предусловие | - |

*Продолжение таблицы 2.1*

|  |  |
| --- | --- |
| Успешный сценарий | 1. Пользователь запускает exe файл 2. Открывается окно авторизации 3. Пользователь вводит логин и пароль 4. Пользователь нажимает на кнопку войти 5. Проверка введённых данных 6. Открывается основное окно |
| Результат | Осуществлен вход в систему |

Краткое описание варианта использования «Введение сведений о клиентах» представлено в таблице 2.2.

Таблица 2.2 – Краткое описание варианта использования «Введение сведений о клиентах»

|  |  |
| --- | --- |
| Название | Введение сведений о клиентах |
| Действующие лица | Менеджер |
| Цель | Просмотр и обновление данных клиента |
| Предусловие | Открыто окно клиенты |
| Успешный сценарий | 1. Пользователь дважды наживает на выбранного клиента из списка 2. Открывается окно редактирования клиента 3. Пользователь обновляет информацию о клиенте и нажимает на кнопку «Сохранить» 4. Открывается окно клиенты |
| Результат | Обновление данных клиента |

Краткое описание варианта использования «Добавление заявки на ремонт» представлено в таблице 2.3.

Таблица 2.3 – Краткое описание варианта использования «Добавление заявки на ремонт»

|  |  |
| --- | --- |
| Название | Добавление заявки на ремонт |
| Действующие лица | Менеджер |
| Цель | Добавление заказа |
| Предусловие | Открыто окно добавить заказ |

*Продолжение таблицы 2.3*

|  |  |
| --- | --- |
| Успешный сценарий | 1. Пользователь заполняет контактную информацию клиента, сведения об устройстве, неисправность устройства, состояние устройства при приеме. 2. Пользователь наживает кнопку «Добавить запись» 3. Открывается окно «Печать акта приема» |
| Результат | Заказ клиента оформлен и принят |

Краткое описание варианта использования «Просмотр заявки на ремонт» представлено в таблице 2.4.

Таблица 2.4 – Краткое описание варианта использования «Просмотр заявки на ремонт»

|  |  |
| --- | --- |
| Название | Просмотр заявки на ремонт |
| Действующие лица | Менеджер, мастер |
| Цель | Просмотр заявки клиента |
| Предусловие | Открыто основное окно |
| Успешный сценарий | 1. Пользователь выбирает заявку из списка заявок. 2. Пользователь дважды наживает на выбранную заявку 3. Открывается окно просмотра заявки |
| Результат | Заявка клиента просмотрена |

Краткое описание варианта использования «Редактирование заявки на ремонт» представлено в таблице 2.5.

Таблица 2.5 – Краткое описание варианта использования «Редактирование заявки на ремонт»

|  |  |
| --- | --- |
| Название | Редактирование заявки на ремонт |
| Действующие лица | Менеджер, мастер |
| Цель | Изменение сведений в заявке клиента |
| Предусловие | Открыто основное окно |
| Успешный сценарий | 1. Пользователь выбирает заявку из списка заявок. 2. Пользователь дважды наживает на выбранную заявку 3. Открывается окно редактирования заявки 4. Пользователь изменяет или заполняет необходимые поля 5. Пользователь нажимает на кнопку «Сохранить и выйти» 6. Открывается основное окно |

*Продолжение таблицы 2.5*

|  |  |
| --- | --- |
| Результат | Заявка клиента изменена |

Краткое описание варианта использования «Просмотр комплектующих» представлено в таблице 2.6

Таблица 2.6 – Краткое описание варианта использования «Просмотр комплектующих»

|  |  |
| --- | --- |
| Название | Просмотр комплектующих |
| Действующие лица | Мастер |
| Цель | Просмотр и выбор необходимых комплектующих для ремонта |
| Предусловие | Открыто окно редактирования заявки |
| Успешный сценарий | 1. Пользователь нажимает на кнопку «Перейти на склад» 2. Пользователь дважды наживает на выбранную комплектующею 3. Открывается окно просмотра комплектующих 4. Пользователь выбирает необходимое число комплектующих и нажимает на кнопку «Использовать запчасть» 5. Открывается окно редактирования заявки 6. Пользователь нажимает на кнопку «Сохранить и выйти» 7. Открывается основное окно |
| Результат | Изменение заявки клиента |

Краткое описание варианта использования «Добавление комплектующих» представлено в таблице 2.7.

Таблица 2.6 – Краткое описание варианта использования «Добавление комплектующих»

|  |  |
| --- | --- |
| Название | Добавление комплектующих |
| Действующие лица | Кладовщик |
| Цель | Добавление запчастей на склад |
| Предусловие | Открыто окно склад |
| Успешный сценарий | 1. Пользователь нажимает на кнопку «Добавить ЗИП» 2. Открывается окно «Добавить запчасть» 3. Пользователь заполняет необходимые сведения о запчасти 4. Пользователь нажимает на кнопку «Добавить в склад» 5. Открывается окно склад |

*Продолжение таблицы 2.7*

|  |  |
| --- | --- |
| Результат | Запчасть добавлена на склад |

Краткое описание варианта использования «Редактирование комплектующей» представлено в таблице 2.8.

Таблица 2.8 – Краткое описание варианта использования «Редактирование комплектующей»

|  |  |
| --- | --- |
| Название | Редактирование комплектующей |
| Действующие лица | Кладовщик |
| Цель | Изменение информации запчасти на складе |
| Предусловие | Открыто окно склад |
| Успешный сценарий | 1. Пользователь дважды наживает на выбранную комплектующею в списке 2. Открывается окно «Редактировать запчасть» 3. Пользователь изменяет необходимые сведения о запчасти 4. Пользователь нажимает на кнопку «Сохранить» 5. Открывается окно склад |
| Результат | Запчасть измена на склад |

Краткое описание варианта использования «Удаление комплектующей» представлено в таблице 2.9.

Таблица 2.9 – Краткое описание варианта использования «Удаление комплектующей»

|  |  |
| --- | --- |
| Название | Удаление комплектующей |
| Действующие лица | Кладовщик |
| Цель | Удаление информации об комплектующей на складе |
| Предусловие | Открыто окно склад |
| Успешный сценарий | 1. Пользователь дважды наживает на выбранную комплектующею в списке 2. Открывается окно «Редактировать запчасть» 3. Пользователь нажимает на кнопку «Удалить» 4. Открывается окно подтверждения удаления 5. Пользователь нажимает на кнопку «Ок» 6. Открывается окно склад |
| Результат | Запчасть удалена из списка на склад |

2.2 Проектирование БД

Проектирование базы данных является важнейшим аспектом разработанной программы для автоматизированного рабочего места сервисного центра. Это включает в себя создание хорошо структурированной и эффективной базы данных, которая может эффективно хранить, систематизировать и извлекать данные, необходимые для бесперебойной работы программы. Конструкция базы данных обеспечивает целостность данных, устраняет избыточность и облегчает бесперебойную обработку данных и составление отчетов [9].

Схема БД программы представлена на рисунке 2.2.



Рисунок 2.2 – ER-диаграмма

Описание отношений и их атрибутов представлено в таблице 2.10.

Таблица 2.10 – Описание отношений и их атрибутов

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| **catalog (Журнал заказов)** | |

*Продолжение таблицы 2.10*

|  |  |
| --- | --- |
| Id | Идентификатор |
| data\_priema | Дата приема |
| data\_vidachi | Дата выдачи |
| data\_predoplaty | Дата предоплаты |
| fio | ФИО |
| phone | Номер телефона |
| what\_remont | Поломка |
| brand | Бренд устройства |
| model | Модель устройства |
| serial\_number | Серийный номер устройства |
| sostoyanie | Состояние устройства |
| komplecktonst | Комплектация заказа |
| predvaritelnaya\_stoimost | Предварительная стоимость |
| predoplata | Предоплата |
| zatrati | Затраты |
| okonchatelnaya\_stoimost\_remonta | Окончательная стоимость |
| skidka | Скидка |
| status\_remonta | Статус ремонта |
| garanty | гарантия |
| master\_id | Идентификатор мастера |
| client\_id | Идентификатор клиента |
| **clientsmap (Карта клиентов)** | |
| id | Идентификатор |
| fio | ФИО |
| phone | Номер телефона |
| adress | Адрес |
| primechanie | Примечание |
| date | Дата создания |
| **groupdostup (Группы доступа)** | |
| id | Идентификатор |
| group\_name | Имя группы |
| add\_zapis | Флаг добавления записи |
| del\_zapis | Флаг удаления записи |

*Продолжение таблицы 2.10*

|  |  |
| --- | --- |
| save\_zapis | Флаг редактирования записи |
| stock | Флаг склада |
| clients | Флаг клиента |
| add\_stock | Флаг добавления на складе запчасти |
| del\_stock | Флаг удаления на складе запчасти |
| save\_stock | Флаг редактирования на складе запчасти |
| add\_client | Флаг добавления клиента |
| del\_client | Флаг удаления клиента |
| save\_client | Флаг редактирования клиента |
| settings | Флаг настроек |
| **historybd (История действий)** | |
| id | Идентификатор |
| who | Кто сделал |
| what | Что сделал |
| data | Дата создания |
| catalog\_id | Идентификатор журнала |
| **statemap (Состояние заказа)** | |
| id | Идентификатор |
| state | Состояние |
| **users (Пользователи)** | |
| id | Идентификатор |
| username | Логин пользователя |
| group\_dostupa\_id | Идентификатор группы |
| **stock (Склад)** | |
| id | Идентификатор |
| naimenivanie | Наименование |
| kategory | Категория |
| colour | Цвет |
| brand | Бренд |
| model | Модель |
| count | Количество |
| price | Цена |

2.3 Интерфейс пользователя

Пользовательский интерфейс – это набор инструментов и методов, позволяющих пользователям взаимодействовать с различными устройствами и аппаратурой.

Другими словами, это тот набор кнопок, ссылок, форм, диалоговых окон, иконки и так далее, с помощью которого пользователи управляют программой. Интерфейс – это только часть взаимодействия с программой, другая часть – люди.

Для хорошей работы интерфейса нужно точно знать, что именно в любой конкретный момент пользователь воспринимает в интерфейсе, о чем думает, чего хочет добиться.

Опишем основные сценарии взаимодействия через пользовательский интерфейс с помощью диаграмм состояний.

Диаграмма состояний, описывающая добавление заявки на ремонт, представлена на рисунке 2.3.



Рисунок 2.3 – Добавление заявки на ремонт

Диаграмма состояний, описывающая просмотр заявки представлена на рисунке 2.4.



Рисунок 2.4 – Просмотр заявки

Диаграмма состояний, описывающая редактирование заявки на ремонт, представлена на рисунке 2.5.



Рисунок 2.5 – Редактирование заявки

2.4 Выводы по главе

В данной главе было выполнено проектирование ПО, где было рассмотрено:

* визуализация ролей пользователей программы и то, как эти роли взаимодействуют с программой в виде диаграммы прецедентов и описанием и кратного описание каждого варианта использования;
* структура БД в виде ER- диаграммы и описаны отношения между таблицами;
* пользовательский интерфейс ПО, который описан с помощью диаграмм состояний.

3 РЕАЛИЗАЦИЯ АРМ

3.1 Реализация ПО

Этап реализации ПО является решающим шагом в разработке программы АРМ для сервисного центра. Он включает в себя фактическое кодирование, настройку и развертывание ПО.

Первым шагом в реализации программы является настройка среды разработки. Это включает в себя установку необходимых программных средств, таких как Microsoft Visual Studio 2022, сервер баз данных MySQL и любые необходимые библиотеки или фреймворки. Среда разработки должна быть должным образом настроена для обеспечения совместимости и бесперебойного процесса разработки.

Этап реализации включает в себя написание программного кода на основе спроектированной части, которая описана во второй главе ВКР. Разработка обычно делится на модули или компоненты, что обеспечивает модульную разработку и упрощает обслуживание. Код реализуемой программы соответствует лучшим практикам и стандартам кодирования для ясности понимания и использования в бедующем.

В рамках реализации программы БД настраивается для поддержки функциональности программы. Это включает в себя создание необходимых таблиц, определение взаимосвязей и настройку правил проверки данных и ограничений. Конфигурации БД гарантирует, что программа может хранить и извлекать данные точно и эффективно.

На этом этапе разрабатывается UI в соответствии со спецификациями дизайна и с учетом принципов, ориентированных на пользователя, которые были определены во втором разделе ВКР. Разработка UI включает в себя создание форм, экранов, меню и элементов управления, позволяющих пользователям беспрепятственно взаимодействовать с программой [16].

После успешного тестирования программа развертывается в производственной среде. Это включает в себя настройку необходимой инфраструктуры, такой как серверы, сеть и параметры безопасности, чтобы обеспечить доступ к программе сотрудников сервисного центра. Учетные записи пользователей и разрешения на доступ настраиваются для обеспечения безопасности и контроля уровней доступа пользователей. На рисунке 3.1 представлена диаграмма развёртывания.

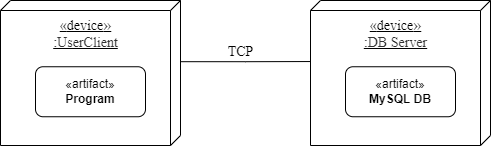


Рисунок 3.1–Диаграмма развертывания

Этап реализации программы имеет решающее значение для превращения проекта и спецификаций в функциональное программное решение.

3.2 Алгоритмы

Алгоритмы являются основополагающим аспектом разработанной программы АРМ сервисного центра. Они содержат пошаговые инструкции для решения конкретных задач и позволяют программе эффективно выполнять различные операции. Опишем основные алгоритмы с помощью диаграмм деятельности.

Диаграмма деятельности, описывающая алгоритм добавление заявки, представлена на рисунке 3.2.



Рисунок 3.2 – Алгоритм добавления заявки

Алгоритм добавления заявки начинает с проверки у пользователя права на добавление записи. Если у пользователя нет этого права, то ему не доступен компонент на форме, который отвечает за это действие, иначе пользователь получает доступ к форме «Добавления заказа». На этой форме пользователь заполняет информацию о клиенте и неисправности его оборудования. После ввода необходимой информации пользователь нажимает на кнопку «Добавить запись», при нажатии которой происходит валидация полей и формирование SQL запроса. При не успешной валидации полей на форме появляется всплывающее окно, указывающая на ошибки при заполнении полей для добавления заявки. Формирование SQL запроса включает в себя:

* 1. Выполнение SQL запроса на добавление заявки;
  2. Выполнение SQL запроса на отслеживание действий при работе с заявкой (иными словами введение истории действий);
  3. Проверка существования данных клиента в БД, если клиент новый и его данных нет в БД, то добавляется запись с необходимой информацией о новом клиенте.

После выполнения запроса на добавление заявки оформляется и печатается накладной акт о приеме устройства и происходит переход на основную форму.

Диаграмма деятельности, описывающая алгоритм изменения данных заявки, представлена на рисунке 3.3.



Рисунок 3.3 – Алгоритм изменения заявки

Алгоритм изменения заявки начинает с проверки у пользователя права на изменение записи. Если у пользователя нет этого права, то ему не доступно изменение заявки, иначе пользователь получает доступ к форме «Редактирование заявки». На этой форме пользователь изменяет необходимую информацию. После ввода необходимой информации пользователь нажимает на кнопку «Сохранить и выйти», при нажатии которой происходит подтверждение действия и формирование SQL запроса. Подтверждение действия представляет собой всплывающее окно, где пользователь подтверждает сохранение изменений. Формирование SQL запроса включает в себя:

* 1. Выполнение SQL запроса на добавление заявки;
  2. Выполнение SQL запроса на отслеживание действий при работе с заявкой (иными словами введение истории действий).

После выполнения запроса происходит переход на основную форму.

Диаграмма деятельности, описывающая алгоритм авторизации, представлена на рисунке 3.4.



Рисунок 3.4 – Алгоритм авторизации

Алгоритм авторизации начинается с ввода пользователем логин и пароля на форме. После чего начинает процесс проверки подлинности пользователя путем сравнения введённого им пароля с паролем, сохраненным в БД (иными словами аутентификация). Этот процесс включает в себя следующие этапы:

* 1. Хеширование введённого пароля по алгоритму md5;
  2. Получение по логину пароля из БД;
  3. Сравнение введённого и полученного пароля;
  4. Вход в программу.

После выполнения успешной аутентификации происходит переход на основную форму.

3.3 Тестирование ПО

Осуществим тестирование определенных ранее вариантов использования. При таком тестировании для каждого варианта использования необходимо разработать тестовые наборы [11]. Для этого выделим различные сценарии для каждого варианта использования.

1. Сценарий №1

Успешный сценарий варианта использования «Авторизация»

1. Сценарий №2

Неверный ввод пароля в варианте использования «Авторизация»

1. Сценарий №3

Успешный сценарий варианта использования «Формирование заявки на ремонт»

1. Сценарий №4

Незаполненное поле «Состояние приема» в варианте использования «Формирование заявки на ремонт»

1. Сценарий №5

Успешный сценарий варианта использования «Печать акта приема»

1. Сценарий №6

Успешный сценарий варианта использования «Просмотр выбранной заявки на ремонт»

1. Сценарий №7

Успешный сценарий варианта использования «Редактирование заявки на ремонт»

Результаты тестирования сценариев представлены в таблице 3.1.

Таблица 3.1 – Результаты тестирования сценариев

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  сценария | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Логин: «Victor»  Пароль: «12345678» | Открытие основной формы | Успешное завершение теста (рисунки В.1 – В.2) | Ожидаемый результат совпал с полученным |
| 2 | Логин: «Victor»  Пароль: «test1234» | Всплывающие окно с ошибкой | Успешное завершение теста (рисунок В.3) | Ожидаемый результат совпал с полученным |
| 3 | ФИО: «Милославский Сергей Александрович»  Телефон: «89082814793»  Название бренда: «ноутбук»  Модель: «thinkbook»  Серийный номер: «7777777777»  Состояние приема: «новый»  Комплектность: «Аппарат, зарядное устройство»  Неисправность: «Сильно греется» | Добавление заявки | Успешное завершение теста (рисунки В.4 – В.5) | Ожидаемый результат совпал с полученным |
| 4 | ФИО: «Милославский Сергей Александрович»  Телефон: «89082814793»  Название бренда: «ноутбук»  Модель: «thinkbook»  Серийный номер: «7777777777»  Комплектность: «Аппарат, зарядное устройство»  Неисправность: «Сильно греется» | Всплывающее окно с ошибкой | Успешное завершение теста (рисунки В.6 – В.7) | Ожидаемый результат совпал с полученным |

*Продолжение таблицы 3.1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | - | Открытие формы для печатания акта приема | Успешное завершение теста (рисунок В.8) | Ожидаемый результат совпал с полученным |
| 6 | - | Открытие формы просмотра | Успешное завершение теста (рисунок В.9) | Ожидаемый результат совпал с полученным |
| 7 | Предв. Стоим. Ремонта: «1000»  Выполненные работы: «Замена термопасты»  Мастер: «Виктор»  Гарантия: «6 месяцев»  Окончательная стоимость: «1000»  Статус заказа: «Готов» | Изменение записи | Успешное завершение теста (рисунки В.10 – В.11) | Ожидаемый результат совпал с полученным |

3.4 Выводы по главе

В данной главе была описана реализация АРМ, где было рассмотрено:

* выполненные этапы реализации АРМ;
* описаны основные алгоритмы в виде диаграмм деятельности;
* выполнено тестирование ПО.

Заключение

В результате проделанной работы была разработана программа, которая автоматизирует работу менеджера и мастера сервисного центра по ремонту электронного оборудования.

Целью работы было создать ПО, которое оптимизирует рабочий процесс, улучшает обслуживание клиентов и, в конечном счете, повышает прибыльность сервисного центра.

В ходе работы были выполнены следующие задачи:

* изучение предметной области;
* анализ аналогов;
* определение технических и программных средств для достижения цели;
* проектирование и разработка структуры программы;
* разработка алгоритмов;
* написание кода;
* тестирование программы.

Разработанная программа позволяет решать проблемы, с которыми сталкиваются менеджеры и мастера сервисного центра, такие как:

* проблемами быстрого доступа к соответствующей информации;
* управления запросами клиентов;
* эффективного формирования отчетов;
* отсутствие удобного интерфейса.

В целом, успешное внедрение программы автоматизированного рабочего места в сервисном центре по ремонту электронного оборудования имеет невероятную возможность изменить процесс ремонта. Благодаря применению автоматизации технические специалисты и руководители могут повысить свою эффективность, повысить качество обслуживания и прибыльность, и конкурентоспособность сервисного центра.

перечень сокращений и условных обозначений

АРМ – автоматизированное рабочее место

ПО – программное обеспечение

СУБД – система управления базой данных

CRUD – create, read, update, delete

IDE – integrated development environment

UI – пользовательский интерфейс

список использованных источников

1. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. — СПб.: Питер, 2013. — 896 с.
2. C# и .NET [электронный ресурс] - metanit.com/sharp/tutorial/1.1.php
3. Microsoft Visual C#. Подробное руководство. 8-е издание / Шарп Д. - СПб.: Питер, 2017. — 848 с.
4. MySQL [электронный ресурс] - dev.mysql.com/doc/
5. MySQL: особенности и сферы применения [Электронный ресурс]. URL: bytemag.ru/articles/detail.php?ID=6547
6. Документация по Visual Studio [Электронный ресурс] https://docs.microsoft.com/ru-ru/visualstudio/?view=vs-2019
7. Изучаем SQL. — СПб.: Питер, 2012. — 592 с.
8. Клиент - сервер [Электронный ресурс] https://ru.wikipedia.org/wiki/Клиент\_—\_сервер
9. Проектирование базы данных [Электронный ресурс] https://studme.org/62415/menedzhment/proektirovanie\_bazy\_dannyh
10. Справочник по языку MySQL [электронный ресурс] www.codenet.ru/db/mysql5/manual.ru\_Reference.php
11. Тестовый сценарий [Электронный ресурс] https://myalm.ru/glossary/%D0%A2%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9-%D1%81%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D0%B8%D0%B9
12. SC24 - обзор, отзывы, аналоги, альтернативы [Электронный ресурс] https://www.livebusiness.ru/tool/2624/
13. ПроМастер - обзор, отзывы, аналоги, альтернативы [Электронный ресурс] https://www.livebusiness.ru/tool/3260/
14. ServiceMP- обзор, отзывы, аналоги, альтернативы [Электронный ресурс] https://www.livebusiness.ru/tool/2110/
15. Диаграммы UML для моделирования процессов и архитектуры проекта [Электронный ресурс] https://evergreens.com.ua/ru/articles/uml-diagrams.html
16. Проектирование интерфейсов пользователя: пособие для студентов специальности 1-47 01 02 «Дизайн электронных и веб-изданий» / Т. П. Брусенцова, Т. В. Кишкурно. – Минск: БГТУ, 2019. – 172 с.

Приложение А

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Филиал федерального государственного бюджетного образовательного учреждения высшего образования

«Национальный исследовательский университет «МЭИ» в г. Смоленске

Кафедра вычислительной техники

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

НА СОИСКАНИЕ КВАЛИФИКАЦИИ

**Бакалавр**

Тема: **Автоматизированное рабочее место сервисного центра**

Направление подготовки

**09.03.01 – Информатика и вычислительная техника**

Профиль подготовки

**Программное обеспечение средств вычислительной**

**техники и автоматизированных систем**



Студент группы ПО1-19 С.А. Милославский

Руководитель

к.т.н., доцент В.А. Тихонов



УТВЕРЖДАЮ

Заведующий кафедрой

д.т.н., профессор А.С. Федулов

«\_\_\_\_» \_\_\_\_\_\_\_\_\_2023 г.

Место выполнения выпускной квалификационной работы: филиал НИУ «МЭИ» в г. Смоленске, кафедра вычислительной техники

Смоленск – 2023

1. Обоснование выбора темы выпускной квалификационной работы

Ремонтный сервисный центр является важным аспектом электронной промышленности, и автоматизированная система может повысить его эффективность и производительность. Поэтому выбранная тема направлена на создание автоматизированного рабочего места, которое поможет ремонтному сервисному центру управлять своей деятельностью, улучшать обслуживание клиентов и повышать прибыльность.

1. Основание для разработки

Государственный общеобразовательный стандарт высшего образования. Направление 09.03.01 – «Информатика и вычислительная техника».

1. Назначение и цель разработки

Целью разработки информационной системы является улучшение рабочего процесса и эффективности работы сервисного центра. Программа позволит сотрудникам сервисного центра быстро получать доступ к информации, управлять запросами клиентов и формировать отчеты.

1. Технические требования

4.1 Требования к функциональным характеристикам

Разрабатываемая программа должна:

* быть способна обслуживать нескольких пользователей, и не должно быть задержек или сбоев системы;
* уметь работать с базой данных и обеспечивать быстрый доступ к данным;
* обрабатывать данные без значительных замедлений или сбоев.
  1. Требования к структуре и функционированию
* Система должна иметь удобный интерфейс, которым может легко пользоваться нетехнический персонал;
* Система должна иметь структуру базы данных, нормализованную для минимизации избыточности данных;
* Система должна поддерживать создание, редактирование, удаление, поиск, сортировку данных, и печатать акты приема и выдачи;
* Система должна обеспечивать многопользовательский доступ с разными уровнями разрешений;
* Система должна позволять пользователям работать в автономном режиме с локальным хранилищем и синхронизироваться с центральной базой данных в режиме онлайн;
* Архитектура системы основана на подходе «толстого клиента», при котором большая часть обработки и управления данными происходит на стороне клиента, а не на стороне сервера.

4.3 Требования к надёжности

* Система должна быть высоконадежной, с минимальным временем простоя и потерей данных;
* Система также должна обеспечивать механизм обработки ошибок, чтобы помочь быстро выявлять и устранять проблемы.

4.4 Требования к защите информации от несанкционированного доступа

Система должна иметь механизм аутентификации, чтобы гарантировать, что только авторизованные пользователи могут получить доступ к системе.

4.5 Требования по сохранению информации при авариях

Защита от сбоев в сети питания осуществляется путём использования устройства бесперебойного питания.

4.6 Требования к видам обеспечения

* + 1. Математическое обеспечение

Применяется стандартный математический аппарат, специальное математическое обеспечение не требуется, формулы для вычисления показателей разрабатываются (при необходимости) в ходе выполнения проекта.

* + 1. Информационное обеспечение

Диалог осуществляется через пользовательский интерфейс системы.

* + 1. Лингвистическое обеспечение

Интерфейс программы и логика обработки данных осуществляется с использованием языка C#.

* + 1. Программное обеспечение

Для серверной части программного обеспечения необходима любая ОС с установленным MySql Server 8.0.30.

Для клиентской части необходима ОС Windows 10 с установленным .NET Framework 4.8.0.

* + 1. Техническое обеспечение

Для работы с программным обеспечением рекомендуется компьютер, оснащенный процессором с тактовой частотой не ниже 1,8 ГГц, двухъядерный или более; 4 ГБ ОЗУ; до 100 Гб свободного места на жестком диске; операционная система Windows 10.

* + 1. Организационное обеспечение

Программа не требует дополнительного организационного обеспечения.

* + 1. Требования к тестированию

Тестирование должно показать полную работоспособность программы, правильное выполнение поставленных перед программой задач. Тестирование разработанной системы следует проводить в соответствии с ГОСТ Р ИСО МЭК 107-99.

1. Требования к программной документации

Программная документация будет оформлена в виде пояснительной записки. Пояснительная записка должна быть выполнена согласно ГОСТ 2.105-95.

6 План работы над выпускной квалификационной работой

| Планируемая работа | Трудоёмкость  % | Срок выполнения | Отметка о выполнении |
| --- | --- | --- | --- |
| Анализ технического задания и обзор технической литературы | 10 | 1.03.23 |  |
| Разработка алгоритмов | 30 | 15.04.23 |  |
| Реализация программного обеспечения | 30 | 15.05.23 |  |
| Тестирование | 10 | 20.05.23 |  |
| Подготовка демонстрационного примера работы программного обеспечения | 5 | 25.05.23 |  |
| Оформление пояснительной записки | 15 | 02.06.23 |  |
| Сдача проекта на проверку | - | 13.06.23 |  |

1. Перечень графического материала

* Техническое задание
* UML диаграммы
* Экранные формы
* Результаты тестирования

8 Рекомендуемая литература

1. Эрнандес, Майкл Дж. Проектирование баз данных для простых смертных: практическое руководство по проектированию реляционных баз данных. 3-е изд., Addison-Wesley Professional, 2013 г. − 672 с.
2. Альбахари, Джозеф и Йоханнсен, Эрик. C# 9.0 в двух словах: полный справочник. 1-е изд., O'Reilly Media, 2021 г. – 1058 с.
3. Прессман, Роджер С. Разработка программного обеспечения: подход практика. 9-е изд., McGraw-Hill Education, 2021. – 704 с.

**Сведения об авторе выпускной квалификационной работы:**

Домашний адрес: г. Смоленск, ул. Куйбышева, д. 9, кв. 106.

Телефон: 8-908-281-37-93.

Электронная почта: MiloslavskiySergey@yandex.ru.

Приложение б

Код программы

Листинг Authorisation.cs

// Authorisation

using SeviceCenter.src;

using System;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Windows.Forms;

public class Authorisation : Form

{

private IniFile INIF = new IniFile("Config.ini");

private Form1 mainForm;

private IContainer components = null;

private ComboBox LoginComboBox;

public TextBox PasswordBox;

public PictureBox pictureBox1;

private Label label1;

private Label label2;

private Button btnSetting;

private Button EnterButton;

public Authorisation(Form1 fm)

{

InitializeComponent();

mainForm = fm;

}

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

PasswordBox.UseSystemPasswordChar = false;

}

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

PasswordBox.UseSystemPasswordChar = true;

}

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

{

comboboxUsersMaker(LoginComboBox);

PasswordBox.UseSystemPasswordChar = true;

if (!INIF.KeyExists(TemporaryBase.UserKey, "LastUser"))

{

return;

}

for (int i = 0; i < LoginComboBox.Items.Count; i++)

{

string a = LoginComboBox.Items[i].ToString();

if (a == INIF.ReadINI(TemporaryBase.UserKey, "LastUser"))

{

LoginComboBox.SelectedIndex = i;

}

}

}

private void comboboxUsersMaker(ComboBox cmbox)

{

cmbox.Items.Clear();

DataTable dataTable = mainForm.basa.UsersBdRead();

if (dataTable.Rows.Count > 0)

{

for (int i = 0; i < dataTable.Rows.Count; i++)

{

cmbox.Items.Add(dataTable.Rows[i].ItemArray[2].ToString());

}

}

}

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

{

VhodVMy();

}

private void VhodVMy()

{

if (LoginComboBox.Text != "")

{

if (Registration.sha1(PasswordBox.Text) == mainForm.basa.UsersGetPass(LoginComboBox.Text))

{

RulesMaker(mainForm.basa.GroupDostupGetgrNameByIdBdRead(mainForm.basa.UsersGetGroupIdByUserName(LoginComboBox.Text)));

mainForm.RulesMackerMainWindow();

mainForm.Enabled = true;

TemporaryBase.USER\_SESSION = LoginComboBox.Text + " " + mainForm.basa.GroupDostupGetgrNameByIdBdRead(mainForm.basa.UsersGetGroupIdByUserName(LoginComboBox.Text));

INIF.WriteINI(TemporaryBase.UserKey, "LastUser", LoginComboBox.Text);

Close();

}

else

{

MessageBox.Show("Неверный логин-пароль");

}

}

else

{

MessageBox.Show("Выберите пользователя");

}

}

private void Authorisation\_FormClosed(object sender, FormClosedEventArgs e)

{

}

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

{

if (!mainForm.Enabled)

{

Application.Exit();

}

}

private void RulesMaker(string grName)

{

DataTable dataTable = mainForm.basa.GroupDostupBdRead(grName);

if (dataTable.Rows.Count > 0)

{

int index = 0;

TemporaryBase.delZapis = ((dataTable.Rows[index].ItemArray[2].ToString() == "1") ? "1" : "0");

TemporaryBase.addZapis = ((dataTable.Rows[index].ItemArray[3].ToString() == "1") ? "1" : "0");

TemporaryBase.saveZapis = ((dataTable.Rows[index].ItemArray[4].ToString() == "1") ? "1" : "0");

TemporaryBase.graf = ((dataTable.Rows[index].ItemArray[5].ToString() == "1") ? "1" : "0");

TemporaryBase.sms = ((dataTable.Rows[index].ItemArray[6].ToString() == "1") ? "1" : "0");

TemporaryBase.stock = ((dataTable.Rows[index].ItemArray[7].ToString() == "1") ? "1" : "0");

TemporaryBase.clients = ((dataTable.Rows[index].ItemArray[8].ToString() == "1") ? "1" : "0");

TemporaryBase.stockAdd = ((dataTable.Rows[index].ItemArray[9].ToString() == "1") ? "1" : "0");

TemporaryBase.stockDel = ((dataTable.Rows[index].ItemArray[10].ToString() == "1") ? "1" : "0");

TemporaryBase.stockEdit = ((dataTable.Rows[index].ItemArray[11].ToString() == "1") ? "1" : "0");

TemporaryBase.clientAdd = ((dataTable.Rows[index].ItemArray[12].ToString() == "1") ? "1" : "0");

TemporaryBase.clientDel = ((dataTable.Rows[index].ItemArray[13].ToString() == "1") ? "1" : "0");

TemporaryBase.clientConcat = ((dataTable.Rows[index].ItemArray[14].ToString() == "1") ? "1" : "0");

TemporaryBase.settings = ((dataTable.Rows[index].ItemArray[15].ToString() == "1") ? "1" : "0");

TemporaryBase.dates = ((dataTable.Rows[index].ItemArray[16].ToString() == "1") ? "1" : "0");

TemporaryBase.editDates = ((dataTable.Rows[index].ItemArray[17].ToString() == "1") ? "1" : "0");

}

}

private void LoginComboBox\_SelectedIndexChanged(object sender, EventArgs e)

{

Text = "Авторизация " + mainForm.basa.GroupDostupGetgrNameByIdBdRead(mainForm.basa.UsersGetGroupIdByUserName(LoginComboBox.Text));

}

private void PasswordBox\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Return)

{

VhodVMy();

}

}

protected override void Dispose(bool disposing)

{

if (disposing && components != null)

{

components.Dispose();

}

base.Dispose(disposing);

}

private void InitializeComponent()

{

this.LoginComboBox = new System.Windows.Forms.ComboBox();

this.PasswordBox = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.EnterButton = new System.Windows.Forms.Button();

this.btnSetting = new System.Windows.Forms.Button();

this.pictureBox1 = new System.Windows.Forms.PictureBox();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

this.SuspendLayout();

//

// LoginComboBox

//

this.LoginComboBox.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

this.LoginComboBox.FormattingEnabled = true;

this.LoginComboBox.Location = new System.Drawing.Point(87, 7);

this.LoginComboBox.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);

this.LoginComboBox.Name = "LoginComboBox";

this.LoginComboBox.Size = new System.Drawing.Size(363, 24);

this.LoginComboBox.TabIndex = 0;

this.LoginComboBox.SelectedIndexChanged += new System.EventHandler(this.LoginComboBox\_SelectedIndexChanged);

//

// PasswordBox

//

this.PasswordBox.Location = new System.Drawing.Point(87, 41);

this.PasswordBox.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);

this.PasswordBox.Name = "PasswordBox";

this.PasswordBox.Size = new System.Drawing.Size(328, 22);

this.PasswordBox.TabIndex = 13;

this.PasswordBox.KeyDown += new System.Windows.Forms.KeyEventHandler(this.PasswordBox\_KeyDown);

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label1.Location = new System.Drawing.Point(5, 11);

this.label1.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(57, 17);

this.label1.TabIndex = 15;

this.label1.Text = "Логин:";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label2.Location = new System.Drawing.Point(5, 44);

this.label2.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(68, 17);

this.label2.TabIndex = 16;

this.label2.Text = "Пароль:";

//

// EnterButton

//

this.EnterButton.Location = new System.Drawing.Point(112, 89);

this.EnterButton.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);

this.EnterButton.Name = "EnterButton";

this.EnterButton.Size = new System.Drawing.Size(116, 28);

this.EnterButton.TabIndex = 17;

this.EnterButton.Text = "Войти";

this.EnterButton.UseVisualStyleBackColor = true;

this.EnterButton.Click += new System.EventHandler(this.EnterButton\_Click);

//

// btnSetting

//

this.btnSetting.AutoSize = true;

this.btnSetting.Location = new System.Drawing.Point(260, 89);

this.btnSetting.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);

this.btnSetting.Name = "btnSetting";

this.btnSetting.Size = new System.Drawing.Size(156, 28);

this.btnSetting.TabIndex = 18;

this.btnSetting.Text = "Настройка сервера";

this.btnSetting.UseVisualStyleBackColor = true;

this.btnSetting.Click += new System.EventHandler(this.btnSetting\_Click);

//

// pictureBox1

//

this.pictureBox1.Location = new System.Drawing.Point(424, 41);

this.pictureBox1.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(20, 20);

this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.AutoSize;

this.pictureBox1.TabIndex = 14;

this.pictureBox1.TabStop = false;

this.pictureBox1.MouseDown += new System.Windows.Forms.MouseEventHandler(this.pictureBox1\_MouseDown);

this.pictureBox1.MouseUp += new System.Windows.Forms.MouseEventHandler(this.pictureBox1\_MouseUp);

//

// Authorisation

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(479, 132);

this.Controls.Add(this.btnSetting);

this.Controls.Add(this.EnterButton);

this.Controls.Add(this.label2);

this.Controls.Add(this.label1);

this.Controls.Add(this.PasswordBox);

this.Controls.Add(this.pictureBox1);

this.Controls.Add(this.LoginComboBox);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedToolWindow;

this.Margin = new System.Windows.Forms.Padding(4, 4, 4, 4);

this.Name = "Authorisation";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

this.Text = "Авторизация";

this.TopMost = true;

this.FormClosing += new System.Windows.Forms.FormClosingEventHandler(this.Authorisation\_FormClosing);

this.FormClosed += new System.Windows.Forms.FormClosedEventHandler(this.Authorisation\_FormClosed);

this.Load += new System.EventHandler(this.Authorisation\_Load);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

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

{

new SettingServer().ShowDialog();

}

}

Листинг

// Form1

using SeviceCenter.DB;

using SeviceCenter.Properties;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

using System.Windows.Forms;

using System.Xml;

public class Form1 : Form

{

public List<VirtualClient> VCList = new List<VirtualClient>();

private IniFile INIF = new IniFile("Config.ini");

private ImageList imageListSmall = new ImageList();

public BDWorker basa;

private ItemComparer itemComparer;

private int daysDiagnostik = -4;

private Color backOfColour;

public bool adPos = false;

public bool setBool = false;

private IContainer components = null;

public ListView MainListView;

public Button NewClientButton;

public Button button1;

public StatusStrip statusStrip;

public ToolStrip toolStrip1;

public ToolStripButton AddPositionButton;

public ToolStripSeparator toolStripSeparator1;

public ToolStripButton SearchFullButton;

public ColumnHeader Number;

public ColumnHeader DataPriema;

public ColumnHeader surname;

public ToolStripButton SettingsButton;

public ToolStripSeparator toolStripSeparator2;

public ToolStripStatusLabel StatusStripLabel;

private ColumnHeader DataVidachi;

private ColumnHeader DataPredoplati;

private ColumnHeader phonee;

private ColumnHeader AboutUS;

private ColumnHeader WhatRemont;

private ColumnHeader Brand;

private ColumnHeader Model;

private ColumnHeader SerialNumber;

private ColumnHeader Sostoyanie;

private ColumnHeader komplektnost;

private ColumnHeader polomka;

private ColumnHeader komment;

private ColumnHeader PredvCoast;

private ColumnHeader Predoplata;

private ColumnHeader Zatrati;

private ColumnHeader Price;

private ColumnHeader Skidka;

private ColumnHeader Status;

private ColumnHeader master;

private ColumnHeader VipolnRaboti;

private ColumnHeader garanty;

private ColumnHeader WaitZakaz;

private ColumnHeader Adress;

private ToolStripStatusLabel toolStripStatusLabel2;

public ToolStripStatusLabel CountListViewLabel;

private ToolStripButton SearchFIOButton;

private ToolStripButton toolStripButton1;

private ToolStripSeparator toolStripSeparator4;

private ToolStripButton toolStripButton2;

private ToolStripSeparator toolStripSeparator5;

private ToolStripSeparator toolStripSeparator7;

private TextBox SearchFIOTextBox;

private ToolStripLabel toolStripLabel2;

private ColumnHeader AdressSC;

private ColumnHeader DeviceColour;

public ToolStripButton ShowPhoneWaitingButton;

public ToolStripButton WaitZakazButton;

private ToolStripButton StockButton;

public ToolStripComboBox ServiceAdressComboBox;

public ToolStripButton ReadyFilterCheckBox;

private ToolStripSeparator toolStripSeparator3;

private ToolStripButton toolStripButton3;

private ToolStripSeparator toolStripSeparator6;

private ToolStrip toolStrip2;

private ToolStripButton AllOrdersButton;

private ToolStripButton DiagnosticksButton;

private ToolStripButton SoglasovanieSKlientomButton;

private ToolStripButton SoglasovanoButton1;

private ToolStripButton InWorkButton;

private ToolStripButton PartWaitingButton;

private ToolStripButton PrinyatPoGarantiiButton;

private ToolStripButton ReadyStatButton;

private ToolStripButton OutOfSCButton;

private ToolStripSeparator toolStripSeparator8;

private ToolStripSeparator toolStripSeparator9;

private ToolStripSeparator toolStripSeparator10;

private ToolStripSeparator toolStripSeparator11;

private ToolStripSeparator toolStripSeparator12;

private ToolStripSeparator toolStripSeparator13;

private ToolStripSeparator toolStripSeparator14;

private ToolStripSeparator toolStripSeparator15;

private ToolStrip toolStrip3;

private ToolStripLabel toolStripLabel1;

private ToolStripTextBox FullSearchPhone;

private ToolStripLabel toolStripLabel3;

private ToolStripComboBox FullSearchType;

private ToolStripLabel toolStripLabel4;

private ToolStripComboBox FullSearchBrand;

private ToolStripLabel toolStripLabel5;

private ToolStripTextBox FullSearchModel;

private ToolStripLabel toolStripLabel6;

private ToolStripTextBox FullSearchSerial;

private ToolStripLabel toolStripLabel7;

private ToolStripComboBox FullSearchMaster;

public Form1()

{

DbContext.Instance.Connect();

TemporaryBase.UserKey = Registration.getHDD();

basa = new BDWorker(this);

InitializeComponent();

MainListView.KeyDown += Program\_KeyDown;

if (INIF.KeyExists("PROGRAMM\_SETTINGS", "colorDiagnostik"))

{

backOfColour = Color.FromArgb(int.Parse(INIF.ReadINI("PROGRAMM\_SETTINGS", "colorDiagnostik")));

}

for (int i = 0; i < MainListView.Columns.Count; i++)

{

if (INIF.KeyExists(TemporaryBase.UserKey, i.ToString()))

{

try

{

MainListView.Columns[i].Width = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, i.ToString()));

}

catch (Exception ex)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex.ToString() + Environment.NewLine);

}

}

}

if (INIF.KeyExists("PROGRAMM\_SETTINGS", "daysDiagnostik"))

{

daysDiagnostik = int.Parse(INIF.ReadINI("PROGRAMM\_SETTINGS", "daysDiagnostik"));

}

MainListViewColumnIndexWriter();

try

{

imageListSmall.Images.Add(Resources.phone);

MainListView.SmallImageList = imageListSmall;

itemComparer = new ItemComparer(this);

MainListView.ListViewItemSorter = itemComparer;

MainListView.ColumnClick += OnColumnClick;

}

catch (Exception ex2)

{

MessageBox.Show(ex2.ToString());

}

}

private void Program\_KeyDown(object sender, KeyEventArgs e)

{

Keys keyData = e.KeyData;

if (keyData == Keys.F12)

{

AddPosition addPosition = new AddPosition(this);

addPosition.Show(this);

}

}

private void ServiceAdressComboBox\_MouseWheel(object sender, MouseEventArgs e)

{

((HandledMouseEventArgs)e).Handled = true;

}

private void OnColumnClick(object sender, ColumnClickEventArgs e)

{

try

{

itemComparer.ColumnIndex = e.Column;

MainListView.VirtualListSize = VCList.Count;

MainListView.VirtualListSize -= 1;

MainListView.VirtualListSize += 1;

}

catch

{

MessageBox.Show("Нечего сортировать");

}

}

public void TempBaseUpdateSearch(string FIO, bool SearchInOld, string Phone = "", string TypeOf = "", string Brand = "", string Model = "", string Status = "", string Master = "", string NeedZakaz = "")

{

TemporaryBase.FIO = FIO;

TemporaryBase.SearchInOld = SearchInOld;

TemporaryBase.Phone = Phone;

TemporaryBase.TypeOf = TypeOf;

TemporaryBase.Brand = Brand;

TemporaryBase.Model = Model;

TemporaryBase.Status = Status;

TemporaryBase.Master = Master;

TemporaryBase.NeedZakaz = NeedZakaz;

}

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

{

if (!adPos)

{

AddPosition addPosition = new AddPosition(this);

addPosition.Show(this);

}

}

private void MainListView\_ColumnClick(object sender, ColumnClickEventArgs e)

{

MainListView.Sorting = SortOrder.Descending;

}

private void MainListView\_ColumnWidthChanged(object sender, ColumnWidthChangedEventArgs e)

{

int width = MainListView.Columns[e.ColumnIndex].Width;

INIF.WriteINI(TemporaryBase.UserKey, e.ColumnIndex.ToString(), width.ToString());

}

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

{

ReadyFilterCheckBox.BackColor = Color.FromArgb(179, 215, 243);

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

FullSearchBrand.Text = "";

FullSearchMaster.Text = "";

FullSearchModel.Text = "";

FullSearchPhone.Text = "";

FullSearchSerial.Text = "";

FullSearchType.Text = "";

if (!toolStrip3.Visible)

{

toolStrip3.Visible = true;

MainListView.Location = new Point(0, 79);

}

else

{

MainListView.Location = new Point(0, 54);

toolStrip3.Visible = false;

}

}

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

{

if (MainListView.SelectedItems.Count > 0)

{

string text = MainListView.SelectedItems[0].Text;

StatusStripLabel.Text = "Редактирование записи номер: " + text;

Editor editor = new Editor(this, text);

editor.Show();

}

else

{

MessageBox.Show("Не выбрана запись для редактирования");

}

}

public void ComboboxMaker(string location, ToolStripComboBox cmb)

{

StreamReader streamReader = new StreamReader(location, Encoding.Default);

for (string text = streamReader.ReadLine(); text != null; text = streamReader.ReadLine())

{

cmb.Items.Add(text);

}

streamReader.Close();

}

private void Form1\_Load(object sender, EventArgs e)

{

TemporaryBase.UserKey = Registration.getHDD();

basa.CreateBd();

basa.UsersTable\_Create();

DataTable dataTable = basa.UsersBdRead();

if (dataTable.Rows.Count > 0)

{

base.Enabled = false;

Authorisation authorisation = new Authorisation(this);

authorisation.Show(this);

}

if (INIF.KeyExists(TemporaryBase.UserKey, "backupPath"))

{

string text = INIF.ReadINI(TemporaryBase.UserKey, "backupPath");

if (Directory.Exists(text))

{

TemporaryBase.pathtoSaveBD = text;

}

}

ReadyFilterCheckBox.BackColor = Color.FromArgb(179, 215, 243);

AllOrdersButton.BackColor = Color.FromArgb(179, 215, 243);

if (INIF.KeyExists("ACTS", "BarcodeH") && INIF.KeyExists("ACTS", "BarcodeW"))

{

TemporaryBase.barcodeH = int.Parse(INIF.ReadINI("ACTS", "BarcodeH"));

TemporaryBase.barcodeW = int.Parse(INIF.ReadINI("ACTS", "BarcodeW"));

}

if (INIF.KeyExists("CHECKBOX", "EveryDayBackup"))

{

TemporaryBase.everyDayBackup = INIF.ReadINI("CHECKBOX", "EveryDayBackup");

}

if (INIF.KeyExists(TemporaryBase.UserKey, "BlistColor"))

{

TemporaryBase.BlistColor = INIF.ReadINI(TemporaryBase.UserKey, "BlistColor");

}

if (INIF.KeyExists(TemporaryBase.UserKey, "AdressSCDefault"))

{

TemporaryBase.AdressSCDefault = INIF.ReadINI(TemporaryBase.UserKey, "AdressSCDefault");

}

if (INIF.KeyExists(Registration.getHDD(), "MasterDefault"))

{

TemporaryBase.MasterDefault = INIF.ReadINI(Registration.getHDD(), "MasterDefault");

}

TemporaryBase.mainForm = this;

FilesExistsOrNot();

ToolTip toolTip = new ToolTip();

toolTip.SetToolTip(SearchFIOTextBox, "Нажмите Enter для поиска");

ComboboxMaker("settings/AdresSC.txt", ServiceAdressComboBox);

ComboboxMaker("settings/ustrojstvo.txt", FullSearchType);

ComboboxMaker("settings/brands.txt", FullSearchBrand);

ComboboxMaker("settings/masters.txt", FullSearchMaster);

if (TemporaryBase.AdressSCDefault.ToString() != "" && ServiceAdressComboBox.Items.Count > int.Parse(TemporaryBase.AdressSCDefault.ToString()))

{

ServiceAdressComboBox.SelectedIndex = int.Parse(TemporaryBase.AdressSCDefault.ToString());

}

if (INIF.KeyExists("PROGRAMM\_SETTINGS", "valuta"))

{

TemporaryBase.valuta = INIF.ReadINI("PROGRAMM\_SETTINGS", "valuta");

}

else

{

TemporaryBase.valuta = "Рублей";

INIF.WriteINI("PROGRAMM\_SETTINGS", "valuta", "Рублей");

}

if (INIF.KeyExists(TemporaryBase.UserKey, "MainFormPosition"))

{

try

{

base.Width = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "MfWidth"));

base.Height = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "MfHeight"));

base.Left = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "MfLeft"));

base.Top = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "MfTop"));

if (base.Left < -10000)

{

base.Left = 0;

base.Top = 0;

base.Width = 600;

base.Height = 600;

}

}

catch

{

}

}

if (INIF.KeyExists("PROGRAMM\_SETTINGS", "colorCheckBox") && INIF.ReadINI("PROGRAMM\_SETTINGS", "colorCheckBox") == "Checked" && INIF.KeyExists("PROGRAMM\_SETTINGS", "colorDiagnostik"))

{

TemporaryBase.diagnostika = true;

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Poloski"))

{

if (INIF.ReadINI(TemporaryBase.UserKey, "Poloski") == "Unchecked")

{

TemporaryBase.Poloski = false;

}

else

{

TemporaryBase.Poloski = true;

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "openClientFolder"))

{

TemporaryBase.openClientFolder = ((!(INIF.ReadINI(TemporaryBase.UserKey, "openClientFolder") == "Unchecked")) ? true : false);

}

TemporaryBase.SearchFULLBegin();

basa.StatesMapTable\_Create();

Text += File.ReadAllText("Settings/Akts/FirmName.txt").Replace("<br>", "");

if (!Directory.Exists("ClientFiles"))

{

Directory.CreateDirectory("ClientFiles");

}

}

public void RulesMackerMainWindow()

{

AddPositionButton.Enabled = ((TemporaryBase.addZapis == "1") ? true : false);

SettingsButton.Enabled = ((TemporaryBase.settings == "1") ? true : false);

toolStripButton2.Enabled = ((TemporaryBase.graf == "1") ? true : false);

StockButton.Enabled = ((TemporaryBase.stock == "1") ? true : false);

toolStripButton3.Enabled = ((TemporaryBase.clients == "1") ? true : false);

}

private static void FilesExistsOrNot()

{

FileInfo fileInfo = new FileInfo("settings/aboutUs.txt");

FileInfo fileInfo2 = new FileInfo("settings/AdresSC.txt");

FileInfo fileInfo3 = new FileInfo("settings/brands.txt");

FileInfo fileInfo4 = new FileInfo("settings/DeviceColour.txt");

FileInfo fileInfo5 = new FileInfo("settings/komplektonst.txt");

FileInfo fileInfo6 = new FileInfo("settings/masters.txt");

FileInfo fileInfo7 = new FileInfo("settings/neispravnost.txt");

FileInfo fileInfo8 = new FileInfo("settings/sostoyaniePriema.txt");

FileInfo fileInfo9 = new FileInfo("settings/ustrojstvo.txt");

FileInfo fileInfo10 = new FileInfo("settings/vipolnRaboti.txt");

FileInfo fileInfo11 = new FileInfo("settings/Akts/DannieOFirme.txt");

FileInfo fileInfo12 = new FileInfo("settings/Akts/DogovorTextPriem.txt");

FileInfo fileInfo13 = new FileInfo("settings/Akts/DogovorTextVidacha.txt");

FileInfo fileInfo14 = new FileInfo("settings/Akts/FirmName.txt");

FileInfo fileInfo15 = new FileInfo("settings/Akts/Phone.txt");

FileInfo fileInfo16 = new FileInfo("settings/Akts/URDannie.txt");

if (!Directory.Exists("settings"))

{

Directory.CreateDirectory("settings");

}

if (!Directory.Exists("settings/Akts"))

{

Directory.CreateDirectory("settings/Akts");

}

if (!Directory.Exists("reports"))

{

Directory.CreateDirectory("reports");

}

if (!Directory.Exists("settings/Stock"))

{

Directory.CreateDirectory("settings/Stock");

}

if (!Directory.Exists("settings/Stock/Photos"))

{

Directory.CreateDirectory("settings/Stock/Photos");

}

if (!Directory.Exists("settings/backup"))

{

Directory.CreateDirectory("settings/backup");

}

if (!fileInfo.Exists)

{

FileStream stream = new FileStream(fileInfo.ToString(), FileMode.Create);

StreamWriter streamWriter = new StreamWriter(stream, Encoding.GetEncoding(1251));

streamWriter.WriteLine("Наружняя реклама");

streamWriter.WriteLine("Интернет");

streamWriter.WriteLine("Этот текст можно");

streamWriter.WriteLine("поменять в файле");

streamWriter.WriteLine("settings/aboutUs.txt");

streamWriter.Close();

}

if (!fileInfo2.Exists)

{

FileStream stream2 = new FileStream(fileInfo2.ToString(), FileMode.Create);

StreamWriter streamWriter2 = new StreamWriter(stream2, Encoding.GetEncoding(1251));

streamWriter2.WriteLine("Улица 3й ноги");

streamWriter2.WriteLine("Переулок 2го уха");

streamWriter2.WriteLine("Этот текст можно");

streamWriter2.WriteLine("поменять в файле");

streamWriter2.WriteLine("settings/AdresSC.txt");

streamWriter2.Close();

}

if (!fileInfo3.Exists)

{

FileStream stream3 = new FileStream(fileInfo3.ToString(), FileMode.Create);

StreamWriter streamWriter3 = new StreamWriter(stream3, Encoding.GetEncoding(1251));

streamWriter3.WriteLine("ACER");

streamWriter3.WriteLine("ASUS");

streamWriter3.WriteLine("APPLE");

streamWriter3.WriteLine("LENOVO");

streamWriter3.WriteLine("SAMSUNG");

streamWriter3.WriteLine("HEWLETT PACKARD");

streamWriter3.WriteLine("DELL");

streamWriter3.WriteLine("MSI");

streamWriter3.WriteLine("DIGMA");

streamWriter3.WriteLine("BENQ");

streamWriter3.WriteLine("BBK");

streamWriter3.Close();

}

if (!fileInfo4.Exists)

{

FileStream stream4 = new FileStream(fileInfo4.ToString(), FileMode.Create);

StreamWriter streamWriter4 = new StreamWriter(stream4, Encoding.GetEncoding(1251));

streamWriter4.WriteLine("Белый");

streamWriter4.WriteLine("Чёрный");

streamWriter4.WriteLine("Серебристый");

streamWriter4.WriteLine("Золотоистый");

streamWriter4.WriteLine("Синий");

streamWriter4.Close();

}

if (!fileInfo5.Exists)

{

FileStream stream5 = new FileStream(fileInfo5.ToString(), FileMode.Create);

StreamWriter streamWriter5 = new StreamWriter(stream5, Encoding.GetEncoding(1251));

streamWriter5.WriteLine("Аппарат");

streamWriter5.WriteLine("АКБ");

streamWriter5.WriteLine("Зарядное устройство");

streamWriter5.WriteLine("Чехол");

streamWriter5.WriteLine("Блок питания");

streamWriter5.WriteLine("Этот текст можно");

streamWriter5.WriteLine("поменять в файле");

streamWriter5.WriteLine("settings/komplektonst.txt");

streamWriter5.Close();

}

if (!fileInfo6.Exists)

{

FileStream stream6 = new FileStream(fileInfo6.ToString(), FileMode.Create);

StreamWriter streamWriter6 = new StreamWriter(stream6, Encoding.GetEncoding(1251));

streamWriter6.WriteLine("Мастер1");

streamWriter6.WriteLine("Этот текст можно");

streamWriter6.WriteLine("поменять в файле");

streamWriter6.WriteLine("settings/masters.txt");

streamWriter6.Close();

}

if (!fileInfo7.Exists)

{

FileStream stream7 = new FileStream(fileInfo7.ToString(), FileMode.Create);

StreamWriter streamWriter7 = new StreamWriter(stream7, Encoding.GetEncoding(1251));

streamWriter7.WriteLine("Что-то сломано");

streamWriter7.WriteLine("Этот текст можно");

streamWriter7.WriteLine("поменять в файле");

streamWriter7.WriteLine("settings/neispravnost.txt");

streamWriter7.Close();

}

if (!fileInfo8.Exists)

{

FileStream stream8 = new FileStream(fileInfo8.ToString(), FileMode.Create);

StreamWriter streamWriter8 = new StreamWriter(stream8, Encoding.GetEncoding(1251));

streamWriter8.WriteLine("Не бит, не крашен");

streamWriter8.WriteLine("Этот текст можно");

streamWriter8.WriteLine("поменять в файле");

streamWriter8.WriteLine("settings/sostoyaniePriema.txt");

streamWriter8.Close();

}

if (!fileInfo9.Exists)

{

FileStream stream9 = new FileStream(fileInfo9.ToString(), FileMode.Create);

StreamWriter streamWriter9 = new StreamWriter(stream9, Encoding.GetEncoding(1251));

streamWriter9.WriteLine("Ноутбук");

streamWriter9.WriteLine("Телефон");

streamWriter9.WriteLine("Патифон");

streamWriter9.WriteLine("Этот текст можно");

streamWriter9.WriteLine("поменять в файле");

streamWriter9.WriteLine("settings/ustrojstvo.txt");

streamWriter9.Close();

}

if (!fileInfo10.Exists)

{

FileStream stream10 = new FileStream(fileInfo10.ToString(), FileMode.Create);

StreamWriter streamWriter10 = new StreamWriter(stream10, Encoding.GetEncoding(1251));

streamWriter10.WriteLine("Замена чего-нибудь");

streamWriter10.WriteLine("Диагностика");

streamWriter10.WriteLine("Этот текст можно");

streamWriter10.WriteLine("поменять в файле");

streamWriter10.WriteLine("settings/vipolnRaboti.txt");

streamWriter10.Close();

}

if (!fileInfo11.Exists)

{

FileStream stream11 = new FileStream(fileInfo11.ToString(), FileMode.Create);

StreamWriter streamWriter11 = new StreamWriter(stream11);

streamWriter11.WriteLine("Режим работы: пн-пт: 10-19, сб: 10-16, вс: выходной <br>г.Петрозаводск, уд.Древлянка д.18, 2 этаж(ТЦ Находка)");

streamWriter11.Close();

}

if (!fileInfo12.Exists)

{

FileStream stream12 = new FileStream(fileInfo12.ToString(), FileMode.Create);

StreamWriter streamWriter12 = new StreamWriter(stream12);

streamWriter12.WriteLine("");

streamWriter12.Close();

}

if (!fileInfo13.Exists)

{

FileStream stream13 = new FileStream(fileInfo13.ToString(), FileMode.Create);

StreamWriter streamWriter13 = new StreamWriter(stream13);

streamWriter13.WriteLine("");

streamWriter13.Close();

}

if (!fileInfo14.Exists)

{

FileStream stream14 = new FileStream(fileInfo14.ToString(), FileMode.Create);

StreamWriter streamWriter14 = new StreamWriter(stream14);

streamWriter14.WriteLine("Название Вашей Фирмы");

streamWriter14.Close();

}

if (!fileInfo15.Exists)

{

FileStream stream15 = new FileStream(fileInfo15.ToString(), FileMode.Create);

StreamWriter streamWriter15 = new StreamWriter(stream15);

streamWriter15.WriteLine("тел.: Вашей фирмы");

streamWriter15.Close();

}

if (!fileInfo16.Exists)

{

FileStream stream16 = new FileStream(fileInfo16.ToString(), FileMode.Create);

StreamWriter streamWriter16 = new StreamWriter(stream16);

streamWriter16.WriteLine("ИП Какойктото В.Е., ОГРНИП 315100234567334 от 19.05.2015 г., ИНН 112301774509");

streamWriter16.Close();

}

}

public void CheckUpdates()

{

try

{

XmlDocument xmlDocument = new XmlDocument();

xmlDocument.Load("http://mywork2.ru/version.xml");

Version v = new Version(xmlDocument.GetElementsByTagName("version")[0].InnerText);

Version v2 = new Version(Application.ProductVersion);

if (v2 < v && MessageBox.Show("Обнаружено обновление, скачать?", "Вы уверены?", MessageBoxButtons.OKCancel) == DialogResult.OK)

{

Process.Start("http://mywork2.ru/update.html");

}

}

catch (Exception)

{

}

}

private string FirstLetterToUpper(string krolik)

{

string source = " \r\n\t";

StringBuilder stringBuilder = new StringBuilder(krolik.ToLower());

if (stringBuilder.Length > 0 && char.IsLetter(stringBuilder[0]))

{

stringBuilder[0] = char.ToUpper(stringBuilder[0]);

}

for (int i = 1; i < stringBuilder.Length; i++)

{

char c = stringBuilder[i];

if (source.Contains(stringBuilder[i - 1]) && char.IsLetter(c))

{

stringBuilder[i] = char.ToUpper(c);

}

}

return stringBuilder.ToString();

}

private void MainListView\_MouseDoubleClick(object sender, MouseEventArgs e)

{

if (VCList.Count > 0)

{

string text = MainListView.Items[MainListView.SelectedIndices[0]].SubItems[0].Text;

StatusStripLabel.Text = "Редактирование записи номер: " + text;

Editor editor = new Editor(this, text);

editor.Show(this);

}

}

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

{

if (!setBool)

{

Settings settings = new Settings(this);

settings.Show(this);

}

}

private void MainListViewColumnIndexReader()

{

INIF.WriteINI(TemporaryBase.UserKey, "id", MainListView.Columns[0].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Data\_priema", MainListView.Columns[1].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Data\_vidachi", MainListView.Columns[2].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Data\_predoplaty", MainListView.Columns[3].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "surname", MainListView.Columns[4].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "phone", MainListView.Columns[5].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "AboutUs", MainListView.Columns[6].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "WhatRemont", MainListView.Columns[7].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "brand", MainListView.Columns[8].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "model", MainListView.Columns[9].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "SerialNumber", MainListView.Columns[10].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "sostoyanie", MainListView.Columns[11].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "komplektonst", MainListView.Columns[12].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "polomka", MainListView.Columns[13].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "kommentarij", MainListView.Columns[14].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "predvaritelnaya\_stoimost", MainListView.Columns[15].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Predoplata", MainListView.Columns[16].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Zatrati", MainListView.Columns[17].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "okonchatelnaya\_stoimost\_remonta", MainListView.Columns[18].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Skidka", MainListView.Columns[19].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Status\_remonta", MainListView.Columns[20].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "master", MainListView.Columns[21].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "vipolnenie\_raboti", MainListView.Columns[22].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Garanty", MainListView.Columns[23].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "wait\_zakaz", MainListView.Columns[24].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "Adress", MainListView.Columns[25].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "AdressSC", MainListView.Columns[26].DisplayIndex.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "DeviceColour", MainListView.Columns[27].DisplayIndex.ToString());

}

private void MainListViewColumnIndexWriter()

{

if (INIF.KeyExists(TemporaryBase.UserKey, "id"))

{

try

{

MainListView.Columns[0].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "id"));

}

catch (Exception ex)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Data\_priema"))

{

try

{

MainListView.Columns[1].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Data\_priema"));

}

catch (Exception ex2)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex2.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Data\_vidachi"))

{

try

{

MainListView.Columns[2].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Data\_vidachi"));

}

catch (Exception ex3)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex3.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Data\_predoplaty"))

{

try

{

MainListView.Columns[3].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Data\_predoplaty"));

}

catch (Exception ex4)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex4.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "surname"))

{

try

{

MainListView.Columns[4].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "surname"));

}

catch (Exception ex5)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex5.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "phone"))

{

try

{

MainListView.Columns[5].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "phone"));

}

catch (Exception ex6)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex6.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "AboutUs"))

{

try

{

MainListView.Columns[6].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "AboutUs"));

}

catch (Exception ex7)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex7.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "WhatRemont"))

{

try

{

MainListView.Columns[7].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "WhatRemont"));

}

catch (Exception ex8)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex8.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "brand"))

{

try

{

MainListView.Columns[8].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "brand"));

}

catch (Exception ex9)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex9.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "model"))

{

try

{

MainListView.Columns[9].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "model"));

}

catch (Exception ex10)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex10.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "SerialNumber"))

{

try

{

MainListView.Columns[10].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "SerialNumber"));

}

catch (Exception ex11)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex11.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "sostoyanie"))

{

try

{

MainListView.Columns[11].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "sostoyanie"));

}

catch (Exception ex12)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex12.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "komplektonst"))

{

try

{

MainListView.Columns[12].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "komplektonst"));

}

catch (Exception ex13)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex13.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "polomka"))

{

try

{

MainListView.Columns[13].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "polomka"));

}

catch (Exception ex14)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex14.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "kommentarij"))

{

try

{

MainListView.Columns[14].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "kommentarij"));

}

catch (Exception ex15)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex15.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "predvaritelnaya\_stoimost"))

{

try

{

MainListView.Columns[15].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "predvaritelnaya\_stoimost"));

}

catch (Exception ex16)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex16.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Predoplata"))

{

try

{

MainListView.Columns[16].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Predoplata"));

}

catch (Exception ex17)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex17.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Zatrati"))

{

try

{

MainListView.Columns[17].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Zatrati"));

}

catch (Exception ex18)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex18.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "okonchatelnaya\_stoimost\_remonta"))

{

try

{

MainListView.Columns[18].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "okonchatelnaya\_stoimost\_remonta"));

}

catch (Exception ex19)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex19.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Skidka"))

{

try

{

MainListView.Columns[19].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Skidka"));

}

catch (Exception ex20)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex20.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Status\_remonta"))

{

try

{

MainListView.Columns[20].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Status\_remonta"));

}

catch (Exception ex21)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex21.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "master"))

{

try

{

MainListView.Columns[21].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "master"));

}

catch (Exception ex22)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex22.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "vipolnenie\_raboti"))

{

try

{

MainListView.Columns[22].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "vipolnenie\_raboti"));

}

catch (Exception ex23)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex23.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Garanty"))

{

try

{

MainListView.Columns[23].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Garanty"));

}

catch (Exception ex24)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex24.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "wait\_zakaz"))

{

try

{

MainListView.Columns[24].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "wait\_zakaz"));

}

catch (Exception ex25)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex25.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "Adress"))

{

try

{

MainListView.Columns[25].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "Adress"));

}

catch (Exception ex26)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex26.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "AdressSC"))

{

try

{

MainListView.Columns[26].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "AdressSC"));

}

catch (Exception ex27)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex27.ToString() + Environment.NewLine);

}

}

if (INIF.KeyExists(TemporaryBase.UserKey, "DeviceColour"))

{

try

{

MainListView.Columns[27].DisplayIndex = int.Parse(INIF.ReadINI(TemporaryBase.UserKey, "DeviceColour"));

}

catch (Exception ex28)

{

MessageBox.Show(DateTime.Now.ToString() + " Что-то с шириной колонок нет так " + ex28.ToString() + Environment.NewLine);

}

}

}

private void MainListView\_ColumnReordered(object sender, ColumnReorderedEventArgs e)

{

MainListViewColumnIndexReader();

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

MainListViewColumnIndexReader();

if (base.Left > -10000 && base.Top > -10000)

{

INIF.WriteINI(TemporaryBase.UserKey, "MainFormPosition", "1");

INIF.WriteINI(TemporaryBase.UserKey, "MfLeft", base.Left.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "MfTop", base.Top.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "MfWidth", base.Width.ToString());

INIF.WriteINI(TemporaryBase.UserKey, "MfHeight", base.Height.ToString());

}

//if (TemporaryBase.everyDayBackup == "Checked" && !File.Exists(TemporaryBase.pathtoSaveBD + "/Backup\_" + DateTime.Now.ToString("dd-MM-yyyy HH") + ".sqlite"))

//{

// File.Copy(basa.dbFileName, TemporaryBase.pathtoSaveBD + "/Backup\_" + DateTime.Now.ToString("dd-MM-yyyy HH") + ".sqlite");

//}

DbContext.Instance.Close();

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

if (SearchFIOTextBox.Text == "QWERTY777")

{

SURPRISE sURPRISE = new SURPRISE(this);

sURPRISE.Show();

}

else

{

TemporaryBase.SearchFULLBegin();

}

}

private void toolStripButton1\_Click\_1(object sender, EventArgs e)

{

StatusButtonColorer();

SearchFIOTextBox.Text = "";

if (SeviceCenter.Properties.Settings.Default["AdressSCDefault"].ToString() != "" && ServiceAdressComboBox.Items.Count > int.Parse(SeviceCenter.Properties.Settings.Default["AdressSCDefault"].ToString()))

{

ServiceAdressComboBox.SelectedIndex = int.Parse(SeviceCenter.Properties.Settings.Default["AdressSCDefault"].ToString());

}

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

FullSearchBrand.Text = "";

FullSearchMaster.Text = "";

FullSearchModel.Text = "";

FullSearchPhone.Text = "";

FullSearchSerial.Text = "";

FullSearchType.Text = "";

ReadyFilterCheckBox.BackColor = Color.FromArgb(179, 215, 243);

ReadyFilterCheckBox.Image = Resources.check\_circle\_outline\_16;

TemporaryBase.SearchCleaner();

TemporaryBase.SearchFULLBegin();

AllOrdersButton.BackColor = Color.FromArgb(179, 215, 243);

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

Graf graf = new Graf(this);

graf.Show(this);

}

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

{

TemporaryBase.IskatVseVidannoe = false;

StatusButtonColorer();

ReadyFilterCheckBox.BackColor = Color.FromArgb(179, 215, 243);

if (!ShowPhoneWaitingButton.Checked)

{

ShowPhoneWaitingButton.Image = Resources.Check1;

ShowPhoneWaitingButton.Checked = true;

TemporaryBase.SearchCleaner();

TemporaryBase.soglasovat = "1";

TemporaryBase.Status = " ";

TemporaryBase.SearchFULLBegin("SearchInOldToo");

}

else

{

if (AllOrdersButton.BackColor != Color.FromArgb(179, 215, 243) && DiagnosticksButton.BackColor != Color.FromArgb(179, 215, 243) && SoglasovanieSKlientomButton.BackColor != Color.FromArgb(179, 215, 243) && SoglasovanoButton1.BackColor != Color.FromArgb(179, 215, 243) && InWorkButton.BackColor != Color.FromArgb(179, 215, 243) && PartWaitingButton.BackColor != Color.FromArgb(179, 215, 243) && ReadyStatButton.BackColor != Color.FromArgb(179, 215, 243) && PrinyatPoGarantiiButton.BackColor != Color.FromArgb(179, 215, 243) && OutOfSCButton.BackColor != Color.FromArgb(179, 215, 243))

{

AllOrdersButton.BackColor = Color.FromArgb(179, 215, 243);

TemporaryBase.ColumnIndex = 0;

TemporaryBase.SortAscending = false;

TemporaryBase.SearchInOld = true;

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.soglasovat = "";

TemporaryBase.NeedZakaz = "";

TemporaryBase.Status = "";

}

ShowPhoneWaitingButton.Image = Resources.phone;

ShowPhoneWaitingButton.Checked = false;

TemporaryBase.SearchCleaner();

TemporaryBase.SearchFULLBegin();

}

WaitZakazButton.Checked = false;

WaitZakazButton.Image = Resources.chip;

}

private void SearchPhone()

{

try

{

MainListView.Items.Clear();

VCList.Clear();

DataTable dataTable = basa.BdSearchPhoneWaiting();

for (int i = 0; i < dataTable.Rows.Count; i++)

{

VirtualClient item = new VirtualClient(dataTable.Rows[i].ItemArray[0].ToString(), dataTable.Rows[i].ItemArray[1].ToString(), dataTable.Rows[i].ItemArray[2].ToString(), dataTable.Rows[i].ItemArray[3].ToString(), dataTable.Rows[i].ItemArray[4].ToString(), dataTable.Rows[i].ItemArray[5].ToString(), dataTable.Rows[i].ItemArray[6].ToString(), dataTable.Rows[i].ItemArray[7].ToString(), dataTable.Rows[i].ItemArray[8].ToString(), dataTable.Rows[i].ItemArray[9].ToString(), dataTable.Rows[i].ItemArray[10].ToString(), dataTable.Rows[i].ItemArray[11].ToString(), dataTable.Rows[i].ItemArray[12].ToString(), dataTable.Rows[i].ItemArray[13].ToString(), dataTable.Rows[i].ItemArray[14].ToString(), dataTable.Rows[i].ItemArray[15].ToString(), dataTable.Rows[i].ItemArray[16].ToString(), dataTable.Rows[i].ItemArray[17].ToString(), dataTable.Rows[i].ItemArray[18].ToString(), dataTable.Rows[i].ItemArray[19].ToString(), dataTable.Rows[i].ItemArray[20].ToString(), dataTable.Rows[i].ItemArray[21].ToString(), dataTable.Rows[i].ItemArray[22].ToString(), dataTable.Rows[i].ItemArray[23].ToString(), dataTable.Rows[i].ItemArray[24].ToString(), dataTable.Rows[i].ItemArray[25].ToString(), dataTable.Rows[i].ItemArray[26].ToString(), TemporaryBase.diagnostika, dataTable.Rows[i].ItemArray[27].ToString(), dataTable.Rows[i].ItemArray[28].ToString(), -1, dataTable.Rows[i].ItemArray[30].ToString());

VCList.Add(item);

}

MainListView.VirtualListSize = VCList.Count;

CountListViewLabel.Text = "Найдено записей: " + dataTable.Rows.Count;

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

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

{

}

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

{

TemporaryBase.IskatVseVidannoe = false;

StatusButtonColorer();

ReadyFilterCheckBox.BackColor = Color.FromArgb(179, 215, 243);

if (!WaitZakazButton.Checked)

{

WaitZakazButton.Image = Resources.Check1;

WaitZakazButton.Checked = true;

TemporaryBase.SearchCleaner();

TemporaryBase.Status = " ";

TemporaryBase.NeedZakaz = "Заказать";

TemporaryBase.SearchFULLBegin();

}

else

{

if (AllOrdersButton.BackColor != Color.FromArgb(179, 215, 243) && DiagnosticksButton.BackColor != Color.FromArgb(179, 215, 243) && SoglasovanieSKlientomButton.BackColor != Color.FromArgb(179, 215, 243) && SoglasovanoButton1.BackColor != Color.FromArgb(179, 215, 243) && InWorkButton.BackColor != Color.FromArgb(179, 215, 243) && PartWaitingButton.BackColor != Color.FromArgb(179, 215, 243) && ReadyStatButton.BackColor != Color.FromArgb(179, 215, 243) && PrinyatPoGarantiiButton.BackColor != Color.FromArgb(179, 215, 243) && OutOfSCButton.BackColor != Color.FromArgb(179, 215, 243))

{

AllOrdersButton.BackColor = Color.FromArgb(179, 215, 243);

TemporaryBase.ColumnIndex = 0;

TemporaryBase.SortAscending = false;

TemporaryBase.SearchInOld = true;

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.soglasovat = "";

TemporaryBase.NeedZakaz = "";

TemporaryBase.Status = "";

}

WaitZakazButton.Image = Resources.chip;

WaitZakazButton.Checked = false;

TemporaryBase.SearchCleaner();

TemporaryBase.SearchFULLBegin();

}

ShowPhoneWaitingButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

}

private void MainListView\_RetrieveVirtualItem(object sender, RetrieveVirtualItemEventArgs e)

{

if (e.ItemIndex < 0 || e.ItemIndex >= VCList.Count)

{

return;

}

e.Item = new ListViewItem(VCList[e.ItemIndex].Id);

if (VCList[e.ItemIndex].Image\_key == "1")

{

e.Item.ImageIndex = 0;

}

Color backOfColour2 = backOfColour;

if (VCList[e.ItemIndex].Data\_priema != "")

{

bool flag = DateTime.Parse(VCList[e.ItemIndex].Data\_priema) < DateTime.Today.AddDays(daysDiagnostik);

if (VCList[e.ItemIndex].Data\_vidachi == "" && flag && VCList[e.ItemIndex].Status\_remonta == "Диагностика")

{

if (VCList[e.ItemIndex].Diagnosik)

{

e.Item.BackColor = backOfColour;

}

}

else if (TemporaryBase.Poloski && e.ItemIndex % 2 == 0)

{

e.Item.BackColor = Color.FromArgb(240, 240, 240);

}

}

e.Item.SubItems.Add(VCList[e.ItemIndex].Data\_priema);

e.Item.SubItems.Add(VCList[e.ItemIndex].Data\_vidachi);

e.Item.SubItems.Add(VCList[e.ItemIndex].Data\_predoplaty);

e.Item.SubItems.Add(FirstLetterToUpper(VCList[e.ItemIndex].Surname));

e.Item.SubItems.Add(VCList[e.ItemIndex].Phone);

e.Item.SubItems.Add(VCList[e.ItemIndex].AboutUs);

e.Item.SubItems.Add(FirstLetterToUpper(VCList[e.ItemIndex].WhatRemont));

e.Item.SubItems.Add(VCList[e.ItemIndex].Brand);

e.Item.SubItems.Add(VCList[e.ItemIndex].Model);

e.Item.SubItems.Add(VCList[e.ItemIndex].SerialNumber);

e.Item.SubItems.Add(VCList[e.ItemIndex].Sostoyanie);

e.Item.SubItems.Add(VCList[e.ItemIndex].Komplektonst);

e.Item.SubItems.Add(VCList[e.ItemIndex].Polomka);

e.Item.SubItems.Add(VCList[e.ItemIndex].Kommentarij);

e.Item.SubItems.Add(VCList[e.ItemIndex].Predvaritelnaya\_stoimost);

e.Item.SubItems.Add(VCList[e.ItemIndex].Predoplata);

e.Item.SubItems.Add(VCList[e.ItemIndex].Zatrati);

e.Item.SubItems.Add(VCList[e.ItemIndex].Okonchatelnaya\_stoimost\_remonta);

e.Item.SubItems.Add(VCList[e.ItemIndex].Skidka);

e.Item.SubItems.Add(VCList[e.ItemIndex].Status\_remonta);

e.Item.SubItems.Add(FirstLetterToUpper(VCList[e.ItemIndex].Master));

e.Item.SubItems.Add(VCList[e.ItemIndex].Vipolnenie\_raboti);

e.Item.SubItems.Add(VCList[e.ItemIndex].Garanty);

e.Item.SubItems.Add(VCList[e.ItemIndex].Wait\_zakaz);

e.Item.SubItems.Add(VCList[e.ItemIndex].Adress);

e.Item.SubItems.Add(VCList[e.ItemIndex].AdressSC);

e.Item.SubItems.Add(VCList[e.ItemIndex].DeviceColour);

}

private void MainListView\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void MainListView\_CacheVirtualItems(object sender, CacheVirtualItemsEventArgs e)

{

TemporaryBase.startIndex = e.StartIndex;

TemporaryBase.endIndex = e.EndIndex;

}

private void SearchFIOTextBox\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode != Keys.Return)

{

return;

}

if (AllOrdersButton.BackColor != Color.FromArgb(179, 215, 243) && DiagnosticksButton.BackColor != Color.FromArgb(179, 215, 243) && SoglasovanieSKlientomButton.BackColor != Color.FromArgb(179, 215, 243) && SoglasovanoButton1.BackColor != Color.FromArgb(179, 215, 243) && InWorkButton.BackColor != Color.FromArgb(179, 215, 243) && PartWaitingButton.BackColor != Color.FromArgb(179, 215, 243) && ReadyStatButton.BackColor != Color.FromArgb(179, 215, 243) && PrinyatPoGarantiiButton.BackColor != Color.FromArgb(179, 215, 243) && OutOfSCButton.BackColor != Color.FromArgb(179, 215, 243))

{

AllOrdersButton.BackColor = Color.FromArgb(179, 215, 243);

TemporaryBase.ColumnIndex = 0;

TemporaryBase.SortAscending = false;

TemporaryBase.SearchInOld = true;

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.soglasovat = "";

TemporaryBase.NeedZakaz = "";

TemporaryBase.Status = "";

}

if (int.TryParse(SearchFIOTextBox.Text, out int result))

{

if (basa.CatlogIDExists(SearchFIOTextBox.Text) != 0)

{

Editor editor = new Editor(this, result.ToString());

editor.Show(this);

}

else

{

MessageBox.Show("Запись с данным номером в базе не найдена");

}

SearchFIOTextBox.Text = "";

}

else

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

}

private void ReadyFilterCheckBox\_CheckedChanged(object sender, EventArgs e)

{

}

private void ServiceAdressComboBox\_SelectedIndexChanged(object sender, EventArgs e)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

private void ReadyFilterCheckBox\_MouseClick(object sender, MouseEventArgs e)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

private void button2\_Click(object sender, EventArgs e)

{

}

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

{

if (SearchFIOTextBox.Text.Length == 12)

{

if (long.TryParse(SearchFIOTextBox.Text, out long \_))

{

string text = basa.BdReadBarcode(SearchFIOTextBox.Text.Trim());

SearchFIOTextBox.Text = "";

if (text != "")

{

Editor editor = new Editor(this, text);

editor.Show(this);

}

}

else

{

TemporaryBase.FIO = SearchFIOTextBox.Text.ToUpper();

}

}

else

{

TemporaryBase.FIO = SearchFIOTextBox.Text.ToUpper();

}

}

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

{

basa.CreateStock();

basa.CreateStockMap();

Stock stock = new Stock(this);

stock.Show(this);

}

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

{

if (ReadyFilterCheckBox.BackColor == Color.FromArgb(179, 215, 243))

{

ReadyFilterCheckBox.BackColor = Color.FromArgb(240, 240, 240);

ReadyFilterCheckBox.Image = Resources.check\_circle\_outline\_blank\_16;

}

else

{

ReadyFilterCheckBox.BackColor = Color.FromArgb(179, 215, 243);

ReadyFilterCheckBox.Image = Resources.check\_circle\_outline\_16;

}

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

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

{

TemporaryBase.SearchFULLBegin();

}

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

{

MessageBox.Show("В разработке");

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

ClientsEditor clientsEditor = new ClientsEditor(this);

clientsEditor.Show(this);

}

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

{

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

private void SoglasovanoButton1\_Click(object sender, EventArgs e)

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

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

{

ToolStripButton toolStripButton = (ToolStripButton)sender;

if (toolStripButton.BackColor == Color.FromArgb(179, 215, 243))

{

ButtonTool(AllOrdersButton);

}

else

{

ButtonTool((ToolStripButton)sender);

}

}

private void ButtonTool(ToolStripButton tsb)

{

StatusButtonColorer();

tsb.BackColor = Color.FromArgb(179, 215, 243);

if (tsb.Tag == null || tsb.Tag.ToString() == "")

{

tsb.Tag = "";

TemporaryBase.ColumnIndex = 0;

TemporaryBase.SortAscending = false;

TemporaryBase.SearchInOld = true;

TemporaryBase.IskatVseVidannoe = false;

}

else if (tsb.Tag.ToString() == "Выдан")

{

TemporaryBase.ColumnIndex = 2;

TemporaryBase.SortAscending = false;

TemporaryBase.SearchInOld = false;

TemporaryBase.IskatVseVidannoe = true;

}

else

{

TemporaryBase.ColumnIndex = 0;

TemporaryBase.SortAscending = false;

TemporaryBase.SearchInOld = false;

TemporaryBase.IskatVseVidannoe = false;

}

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.soglasovat = "";

TemporaryBase.NeedZakaz = "";

TemporaryBase.Status = tsb.Tag.ToString();

TemporaryBase.SearchFULLBegin();

}

private void StatusButtonColorer()

{

AllOrdersButton.BackColor = Color.FromArgb(240, 240, 240);

DiagnosticksButton.BackColor = Color.FromArgb(240, 240, 240);

SoglasovanieSKlientomButton.BackColor = Color.FromArgb(240, 240, 240);

SoglasovanoButton1.BackColor = Color.FromArgb(240, 240, 240);

InWorkButton.BackColor = Color.FromArgb(240, 240, 240);

PartWaitingButton.BackColor = Color.FromArgb(240, 240, 240);

ReadyStatButton.BackColor = Color.FromArgb(240, 240, 240);

OutOfSCButton.BackColor = Color.FromArgb(240, 240, 240);

PrinyatPoGarantiiButton.BackColor = Color.FromArgb(240, 240, 240);

}

private void FullSearchType\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Return)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

}

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

{

TemporaryBase.Brand = FullSearchBrand.Text.ToUpper().Trim();

}

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

{

TemporaryBase.Model = FullSearchModel.Text.ToUpper().Trim();

}

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

{

TemporaryBase.SerialImei = FullSearchSerial.Text.ToUpper().Trim();

}

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

{

TemporaryBase.Master = FullSearchMaster.Text.Trim();

}

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

{

TemporaryBase.TypeOf = FullSearchType.Text.ToUpper().Trim();

}

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

{

TemporaryBase.Phone = FullSearchPhone.Text.Trim().Replace(" ", "");

}

private void FullSearchPhone\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Return)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

}

private void FullSearchBrand\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Return)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

}

private void FullSearchModel\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Return)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

}

private void FullSearchSerial\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Return)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

}

private void FullSearchMaster\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Return)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

}

private void FullSearchType\_SelectedIndexChanged(object sender, EventArgs e)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

private void FullSearchBrand\_SelectedIndexChanged(object sender, EventArgs e)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

private void FullSearchMaster\_SelectedIndexChanged(object sender, EventArgs e)

{

ShowPhoneWaitingButton.Checked = false;

WaitZakazButton.Checked = false;

ShowPhoneWaitingButton.Image = Resources.phone;

WaitZakazButton.Image = Resources.chip;

TemporaryBase.SearchFULLBegin();

}

protected override void Dispose(bool disposing)

{

if (disposing && components != null)

{

components.Dispose();

}

base.Dispose(disposing);

}

private void InitializeComponent()

{

this.MainListView = new System.Windows.Forms.ListView();

this.Number = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.DataPriema = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.DataVidachi = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.DataPredoplati = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.surname = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.phonee = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.AboutUS = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.WhatRemont = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Brand = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Model = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.SerialNumber = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Sostoyanie = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.komplektnost = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.polomka = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.komment = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.PredvCoast = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Predoplata = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Zatrati = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Price = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Skidka = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Status = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.master = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.VipolnRaboti = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.garanty = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.WaitZakaz = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.Adress = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.AdressSC = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.DeviceColour = ((System.Windows.Forms.ColumnHeader)(new System.Windows.Forms.ColumnHeader()));

this.NewClientButton = new System.Windows.Forms.Button();

this.button1 = new System.Windows.Forms.Button();

this.statusStrip = new System.Windows.Forms.StatusStrip();

this.StatusStripLabel = new System.Windows.Forms.ToolStripStatusLabel();

this.toolStripStatusLabel2 = new System.Windows.Forms.ToolStripStatusLabel();

this.CountListViewLabel = new System.Windows.Forms.ToolStripStatusLabel();

this.toolStrip1 = new System.Windows.Forms.ToolStrip();

this.AddPositionButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator1 = new System.Windows.Forms.ToolStripSeparator();

this.SearchFIOButton = new System.Windows.Forms.ToolStripButton();

this.toolStripLabel2 = new System.Windows.Forms.ToolStripLabel();

this.toolStripButton1 = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator2 = new System.Windows.Forms.ToolStripSeparator();

this.SearchFullButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator4 = new System.Windows.Forms.ToolStripSeparator();

this.ShowPhoneWaitingButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator3 = new System.Windows.Forms.ToolStripSeparator();

this.WaitZakazButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator7 = new System.Windows.Forms.ToolStripSeparator();

this.toolStripButton2 = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator5 = new System.Windows.Forms.ToolStripSeparator();

this.SettingsButton = new System.Windows.Forms.ToolStripButton();

this.ReadyFilterCheckBox = new System.Windows.Forms.ToolStripButton();

this.ServiceAdressComboBox = new System.Windows.Forms.ToolStripComboBox();

this.StockButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator6 = new System.Windows.Forms.ToolStripSeparator();

this.toolStripButton3 = new System.Windows.Forms.ToolStripButton();

this.SearchFIOTextBox = new System.Windows.Forms.TextBox();

this.toolStrip2 = new System.Windows.Forms.ToolStrip();

this.AllOrdersButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator8 = new System.Windows.Forms.ToolStripSeparator();

this.DiagnosticksButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator9 = new System.Windows.Forms.ToolStripSeparator();

this.SoglasovanieSKlientomButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator10 = new System.Windows.Forms.ToolStripSeparator();

this.SoglasovanoButton1 = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator11 = new System.Windows.Forms.ToolStripSeparator();

this.InWorkButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator12 = new System.Windows.Forms.ToolStripSeparator();

this.PartWaitingButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator14 = new System.Windows.Forms.ToolStripSeparator();

this.ReadyStatButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator15 = new System.Windows.Forms.ToolStripSeparator();

this.PrinyatPoGarantiiButton = new System.Windows.Forms.ToolStripButton();

this.toolStripSeparator13 = new System.Windows.Forms.ToolStripSeparator();

this.OutOfSCButton = new System.Windows.Forms.ToolStripButton();

this.toolStrip3 = new System.Windows.Forms.ToolStrip();

this.toolStripLabel1 = new System.Windows.Forms.ToolStripLabel();

this.FullSearchPhone = new System.Windows.Forms.ToolStripTextBox();

this.toolStripLabel3 = new System.Windows.Forms.ToolStripLabel();

this.FullSearchType = new System.Windows.Forms.ToolStripComboBox();

this.toolStripLabel4 = new System.Windows.Forms.ToolStripLabel();

this.FullSearchBrand = new System.Windows.Forms.ToolStripComboBox();

this.toolStripLabel5 = new System.Windows.Forms.ToolStripLabel();

this.FullSearchModel = new System.Windows.Forms.ToolStripTextBox();

this.toolStripLabel6 = new System.Windows.Forms.ToolStripLabel();

this.FullSearchSerial = new System.Windows.Forms.ToolStripTextBox();

this.toolStripLabel7 = new System.Windows.Forms.ToolStripLabel();

this.FullSearchMaster = new System.Windows.Forms.ToolStripComboBox();

this.statusStrip.SuspendLayout();

this.toolStrip1.SuspendLayout();

this.toolStrip2.SuspendLayout();

this.toolStrip3.SuspendLayout();

this.SuspendLayout();

//

// MainListView

//

this.MainListView.AllowColumnReorder = true;

this.MainListView.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.MainListView.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

this.MainListView.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] {

this.Number,

this.DataPriema,

this.DataVidachi,

this.DataPredoplati,

this.surname,

this.phonee,

this.AboutUS,

this.WhatRemont,

this.Brand,

this.Model,

this.SerialNumber,

this.Sostoyanie,

this.komplektnost,

this.polomka,

this.komment,

this.PredvCoast,

this.Predoplata,

this.Zatrati,

this.Price,

this.Skidka,

this.Status,

this.master,

this.VipolnRaboti,

this.garanty,

this.WaitZakaz,

this.Adress,

this.AdressSC,

this.DeviceColour});

this.MainListView.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.MainListView.FullRowSelect = true;

this.MainListView.GridLines = true;

this.MainListView.HideSelection = false;

this.MainListView.Location = new System.Drawing.Point(0, 66);

this.MainListView.Margin = new System.Windows.Forms.Padding(4);

this.MainListView.MultiSelect = false;

this.MainListView.Name = "MainListView";

this.MainListView.Size = new System.Drawing.Size(1577, 728);

this.MainListView.TabIndex = 2;

this.MainListView.UseCompatibleStateImageBehavior = false;

this.MainListView.View = System.Windows.Forms.View.Details;

this.MainListView.VirtualMode = true;

this.MainListView.CacheVirtualItems += new System.Windows.Forms.CacheVirtualItemsEventHandler(this.MainListView\_CacheVirtualItems);

this.MainListView.ColumnClick += new System.Windows.Forms.ColumnClickEventHandler(this.MainListView\_ColumnClick);

this.MainListView.ColumnReordered += new System.Windows.Forms.ColumnReorderedEventHandler(this.MainListView\_ColumnReordered);

this.MainListView.ColumnWidthChanged += new System.Windows.Forms.ColumnWidthChangedEventHandler(this.MainListView\_ColumnWidthChanged);

this.MainListView.RetrieveVirtualItem += new System.Windows.Forms.RetrieveVirtualItemEventHandler(this.MainListView\_RetrieveVirtualItem);

this.MainListView.SelectedIndexChanged += new System.EventHandler(this.MainListView\_SelectedIndexChanged);

this.MainListView.MouseDoubleClick += new System.Windows.Forms.MouseEventHandler(this.MainListView\_MouseDoubleClick);

//

// Number

//

this.Number.Text = "№";

//

// DataPriema

//

this.DataPriema.Text = "Дата приёма";

this.DataPriema.Width = 80;

//

// DataVidachi

//

this.DataVidachi.Text = "Дата выдачи";

this.DataVidachi.Width = 80;

//

// DataPredoplati

//

this.DataPredoplati.Text = "Дата предоплаты";

this.DataPredoplati.Width = 80;

//

// surname

//

this.surname.Text = "ФИО";

//

// phonee

//

this.phonee.Text = "Телефон";

this.phonee.Width = 90;

//

// AboutUS

//

this.AboutUS.Text = "Узнали о нас";

this.AboutUS.Width = 80;

//

// WhatRemont

//

this.WhatRemont.Text = "Тип устройства";

this.WhatRemont.Width = 95;

//

// Brand

//

this.Brand.Text = "Бренд";

//

// Model

//

this.Model.Text = "Модель";

//

// SerialNumber

//

this.SerialNumber.Text = "Серийный номер";

this.SerialNumber.Width = 84;

//

// Sostoyanie

//

this.Sostoyanie.Text = "Состояние";

this.Sostoyanie.Width = 70;

//

// komplektnost

//

this.komplektnost.Text = "Комплект";

this.komplektnost.Width = 65;

//

// polomka

//

this.polomka.Text = "Неисправность";

this.polomka.Width = 80;

//

// komment

//

this.komment.Text = "Комментарий";

this.komment.Width = 80;

//

// PredvCoast

//

this.PredvCoast.Text = "Предв. стоимость";

//

// Predoplata

//

this.Predoplata.Text = "Предоплата";

//

// Zatrati

//

this.Zatrati.Text = "Затраты";

//

// Price

//

this.Price.Text = "Цена";

//

// Skidka

//

this.Skidka.Text = "Скидка";

//

// Status

//

this.Status.Text = "Статус";

//

// master

//

this.master.Text = "Мастер";

//

// VipolnRaboti

//

this.VipolnRaboti.Text = "Выполненные работы";

//

// garanty

//

this.garanty.Text = "Гарантия";

//

// WaitZakaz

//

this.WaitZakaz.Text = "Заказать";

//

// Adress

//

this.Adress.Text = "Адрес";

//

// AdressSC

//

this.AdressSC.Text = "Адрес СЦ";

//

// DeviceColour

//

this.DeviceColour.Text = "Цвет";

//

// NewClientButton

//

this.NewClientButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;

this.NewClientButton.Location = new System.Drawing.Point(0, 1);

this.NewClientButton.Margin = new System.Windows.Forms.Padding(4);

this.NewClientButton.Name = "NewClientButton";

this.NewClientButton.Size = new System.Drawing.Size(153, 28);

this.NewClientButton.TabIndex = 1;

this.NewClientButton.Text = "Новый клиент";

this.NewClientButton.UseVisualStyleBackColor = true;

//

// button1

//

this.button1.FlatStyle = System.Windows.Forms.FlatStyle.Popup;

this.button1.Location = new System.Drawing.Point(161, 1);

this.button1.Margin = new System.Windows.Forms.Padding(4);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(152, 28);

this.button1.TabIndex = 2;

this.button1.Text = "Редактировать";

this.button1.UseVisualStyleBackColor = true;

//

// statusStrip

//

this.statusStrip.Anchor = ((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.statusStrip.AutoSize = false;

this.statusStrip.Dock = System.Windows.Forms.DockStyle.None;

this.statusStrip.ImageScalingSize = new System.Drawing.Size(20, 20);

this.statusStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.StatusStripLabel,

this.toolStripStatusLabel2,

this.CountListViewLabel});

this.statusStrip.Location = new System.Drawing.Point(0, 795);

this.statusStrip.Name = "statusStrip";

this.statusStrip.Padding = new System.Windows.Forms.Padding(1, 0, 19, 0);

this.statusStrip.Size = new System.Drawing.Size(1577, 27);

this.statusStrip.TabIndex = 1;

this.statusStrip.Text = "statusStrip1";

//

// StatusStripLabel

//

this.StatusStripLabel.Font = new System.Drawing.Font("Segoe UI", 9F, System.Drawing.FontStyle.Italic);

this.StatusStripLabel.Name = "StatusStripLabel";

this.StatusStripLabel.Size = new System.Drawing.Size(0, 21);

//

// toolStripStatusLabel2

//

this.toolStripStatusLabel2.Name = "toolStripStatusLabel2";

this.toolStripStatusLabel2.Size = new System.Drawing.Size(13, 21);

this.toolStripStatusLabel2.Text = "|";

//

// CountListViewLabel

//

this.CountListViewLabel.Name = "CountListViewLabel";

this.CountListViewLabel.Size = new System.Drawing.Size(0, 21);

this.CountListViewLabel.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

//

// toolStrip1

//

this.toolStrip1.Font = new System.Drawing.Font("Comic Sans MS", 10F);

this.toolStrip1.ImageScalingSize = new System.Drawing.Size(20, 20);

this.toolStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.AddPositionButton,

this.toolStripSeparator1,

this.SearchFIOButton,

this.toolStripLabel2,

this.toolStripButton1,

this.toolStripSeparator2,

this.SearchFullButton,

this.toolStripSeparator4,

this.ShowPhoneWaitingButton,

this.toolStripSeparator3,

this.WaitZakazButton,

this.toolStripSeparator7,

this.toolStripButton2,

this.toolStripSeparator5,

this.SettingsButton,

this.ReadyFilterCheckBox,

this.ServiceAdressComboBox,

this.StockButton,

this.toolStripSeparator6,

this.toolStripButton3});

this.toolStrip1.Location = new System.Drawing.Point(0, 0);

this.toolStrip1.Name = "toolStrip1";

this.toolStrip1.Size = new System.Drawing.Size(1577, 31);

this.toolStrip1.TabIndex = 3;

this.toolStrip1.Text = "toolStrip1";

//

// AddPositionButton

//

this.AddPositionButton.Image = global::SeviceCenter.Properties.Resources.add;

this.AddPositionButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.AddPositionButton.Name = "AddPositionButton";

this.AddPositionButton.Size = new System.Drawing.Size(111, 28);

this.AddPositionButton.Text = "Добавить";

this.AddPositionButton.Click += new System.EventHandler(this.AddPositionButton\_Click);

//

// toolStripSeparator1

//

this.toolStripSeparator1.Name = "toolStripSeparator1";

this.toolStripSeparator1.Size = new System.Drawing.Size(6, 31);

//

// SearchFIOButton

//

this.SearchFIOButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.SearchFIOButton.Name = "SearchFIOButton";

this.SearchFIOButton.Size = new System.Drawing.Size(57, 28);

this.SearchFIOButton.Text = "ФИО:";

this.SearchFIOButton.Click += new System.EventHandler(this.SearchFIOButton\_Click);

//

// toolStripLabel2

//

this.toolStripLabel2.AutoSize = false;

this.toolStripLabel2.Name = "toolStripLabel2";

this.toolStripLabel2.Size = new System.Drawing.Size(146, 23);

this.toolStripLabel2.Text = "toolStripLabel2";

//

// toolStripButton1

//

this.toolStripButton1.ImageTransparentColor = System.Drawing.Color.Magenta;

this.toolStripButton1.Name = "toolStripButton1";

this.toolStripButton1.Size = new System.Drawing.Size(124, 28);

this.toolStripButton1.Text = "Сброс поиска";

this.toolStripButton1.Click += new System.EventHandler(this.toolStripButton1\_Click\_1);

//

// toolStripSeparator2

//

this.toolStripSeparator2.Name = "toolStripSeparator2";

this.toolStripSeparator2.Size = new System.Drawing.Size(6, 31);

//

// SearchFullButton

//

this.SearchFullButton.Image = global::SeviceCenter.Properties.Resources.search;

this.SearchFullButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.SearchFullButton.Name = "SearchFullButton";

this.SearchFullButton.Size = new System.Drawing.Size(148, 28);

this.SearchFullButton.Text = "Расширенный";

this.SearchFullButton.Click += new System.EventHandler(this.SearchFullButton\_Click);

//

// toolStripSeparator4

//

this.toolStripSeparator4.Name = "toolStripSeparator4";

this.toolStripSeparator4.Size = new System.Drawing.Size(6, 31);

//

// ShowPhoneWaitingButton

//

this.ShowPhoneWaitingButton.DisplayStyle = System.Windows.Forms.ToolStripItemDisplayStyle.Image;

this.ShowPhoneWaitingButton.Image = global::SeviceCenter.Properties.Resources.phone;

this.ShowPhoneWaitingButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.ShowPhoneWaitingButton.Name = "ShowPhoneWaitingButton";

this.ShowPhoneWaitingButton.RightToLeftAutoMirrorImage = true;

this.ShowPhoneWaitingButton.Size = new System.Drawing.Size(29, 28);

this.ShowPhoneWaitingButton.Text = "Показать ожидающие звонка";

this.ShowPhoneWaitingButton.Click += new System.EventHandler(this.ShowPhoneWaitingButton\_Click);

//

// toolStripSeparator3

//

this.toolStripSeparator3.Name = "toolStripSeparator3";

this.toolStripSeparator3.Size = new System.Drawing.Size(6, 31);

//

// WaitZakazButton

//

this.WaitZakazButton.DisplayStyle = System.Windows.Forms.ToolStripItemDisplayStyle.Image;

this.WaitZakazButton.Image = global::SeviceCenter.Properties.Resources.chip;

this.WaitZakazButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.WaitZakazButton.Name = "WaitZakazButton";

this.WaitZakazButton.Size = new System.Drawing.Size(29, 28);

this.WaitZakazButton.Text = "Требует заказа";

this.WaitZakazButton.Click += new System.EventHandler(this.WaitZakazButton\_Click);

//

// toolStripSeparator7

//

this.toolStripSeparator7.Name = "toolStripSeparator7";

this.toolStripSeparator7.Size = new System.Drawing.Size(6, 31);

//

// toolStripButton2

//

this.toolStripButton2.DisplayStyle = System.Windows.Forms.ToolStripItemDisplayStyle.Image;

this.toolStripButton2.Image = global::SeviceCenter.Properties.Resources.graph;

this.toolStripButton2.ImageTransparentColor = System.Drawing.Color.Magenta;

this.toolStripButton2.Name = "toolStripButton2";

this.toolStripButton2.Size = new System.Drawing.Size(29, 28);

this.toolStripButton2.Text = "Графики и отчёты";

this.toolStripButton2.Click += new System.EventHandler(this.toolStripButton2\_Click);

//

// toolStripSeparator5

//

this.toolStripSeparator5.Name = "toolStripSeparator5";

this.toolStripSeparator5.Size = new System.Drawing.Size(6, 31);

//

// SettingsButton

//

this.SettingsButton.Image = global::SeviceCenter.Properties.Resources.settings;

this.SettingsButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.SettingsButton.Name = "SettingsButton";

this.SettingsButton.Size = new System.Drawing.Size(122, 28);

this.SettingsButton.Text = "Настройки";

this.SettingsButton.Click += new System.EventHandler(this.SettingsButton\_Click);

//

// ReadyFilterCheckBox

//

this.ReadyFilterCheckBox.BackColor = System.Drawing.SystemColors.Control;

this.ReadyFilterCheckBox.BackgroundImageLayout = System.Windows.Forms.ImageLayout.None;

this.ReadyFilterCheckBox.Image = global::SeviceCenter.Properties.Resources.check\_circle\_outline\_16;

this.ReadyFilterCheckBox.ImageTransparentColor = System.Drawing.Color.Magenta;

this.ReadyFilterCheckBox.Name = "ReadyFilterCheckBox";

this.ReadyFilterCheckBox.Size = new System.Drawing.Size(91, 28);

this.ReadyFilterCheckBox.Text = "Готово";

this.ReadyFilterCheckBox.ToolTipText = "Показывать записи со статусом Готов, вместе с остальными.\r\nТак же при поиске учит" +

"ывается, какой из статусов выбран";

this.ReadyFilterCheckBox.Click += new System.EventHandler(this.ReadyFilterCheckBox\_Click);

//

// ServiceAdressComboBox

//

this.ServiceAdressComboBox.FlatStyle = System.Windows.Forms.FlatStyle.Standard;

this.ServiceAdressComboBox.Items.AddRange(new object[] {

""});

this.ServiceAdressComboBox.Name = "ServiceAdressComboBox";

this.ServiceAdressComboBox.Size = new System.Drawing.Size(172, 31);

this.ServiceAdressComboBox.ToolTipText = "Искать только в этом сервисе, если пусто, то во всех";

this.ServiceAdressComboBox.TextChanged += new System.EventHandler(this.ServiceAdressComboBox\_TextChanged);

//

// StockButton

//

this.StockButton.Image = global::SeviceCenter.Properties.Resources.stock;

this.StockButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.StockButton.Name = "StockButton";

this.StockButton.Size = new System.Drawing.Size(82, 28);

this.StockButton.Text = "Склад";

this.StockButton.ToolTipText = "Склад";

this.StockButton.Click += new System.EventHandler(this.StockButton\_Click);

//

// toolStripSeparator6

//

this.toolStripSeparator6.Name = "toolStripSeparator6";

this.toolStripSeparator6.Size = new System.Drawing.Size(6, 31);

//

// toolStripButton3

//

this.toolStripButton3.Image = global::SeviceCenter.Properties.Resources.clients;

this.toolStripButton3.ImageTransparentColor = System.Drawing.Color.Magenta;

this.toolStripButton3.Name = "toolStripButton3";

this.toolStripButton3.Size = new System.Drawing.Size(108, 28);

this.toolStripButton3.Text = "Клиенты";

this.toolStripButton3.Click += new System.EventHandler(this.toolStripButton3\_Click);

//

// SearchFIOTextBox

//

this.SearchFIOTextBox.BackColor = System.Drawing.SystemColors.Info;

this.SearchFIOTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

this.SearchFIOTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.SearchFIOTextBox.ForeColor = System.Drawing.SystemColors.InfoText;

this.SearchFIOTextBox.Location = new System.Drawing.Point(207, 1);

this.SearchFIOTextBox.Margin = new System.Windows.Forms.Padding(4);

this.SearchFIOTextBox.Name = "SearchFIOTextBox";

this.SearchFIOTextBox.Size = new System.Drawing.Size(189, 26);

this.SearchFIOTextBox.TabIndex = 4;

this.SearchFIOTextBox.TextChanged += new System.EventHandler(this.SearchFIOTextBox\_TextChanged);

this.SearchFIOTextBox.KeyDown += new System.Windows.Forms.KeyEventHandler(this.SearchFIOTextBox\_KeyDown);

//

// toolStrip2

//

this.toolStrip2.AllowItemReorder = true;

this.toolStrip2.BackColor = System.Drawing.SystemColors.Control;

this.toolStrip2.Font = new System.Drawing.Font("Comic Sans MS", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.toolStrip2.ImageScalingSize = new System.Drawing.Size(20, 20);

this.toolStrip2.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.AllOrdersButton,

this.toolStripSeparator8,

this.DiagnosticksButton,

this.toolStripSeparator9,

this.SoglasovanieSKlientomButton,

this.toolStripSeparator10,

this.SoglasovanoButton1,

this.toolStripSeparator11,

this.InWorkButton,

this.toolStripSeparator12,

this.PartWaitingButton,

this.toolStripSeparator14,

this.ReadyStatButton,

this.toolStripSeparator15,

this.PrinyatPoGarantiiButton,

this.toolStripSeparator13,

this.OutOfSCButton});

this.toolStrip2.LayoutStyle = System.Windows.Forms.ToolStripLayoutStyle.HorizontalStackWithOverflow;

this.toolStrip2.Location = new System.Drawing.Point(0, 31);

this.toolStrip2.Name = "toolStrip2";

this.toolStrip2.Size = new System.Drawing.Size(1577, 31);

this.toolStrip2.TabIndex = 5;

this.toolStrip2.Text = "toolStrip2";

//

// AllOrdersButton

//

this.AllOrdersButton.Image = global::SeviceCenter.Properties.Resources.Status\_allOreders;

this.AllOrdersButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.AllOrdersButton.Name = "AllOrdersButton";

this.AllOrdersButton.Size = new System.Drawing.Size(107, 28);

this.AllOrdersButton.Text = "Текущие";

this.AllOrdersButton.ToolTipText = "Показывает все записи, которые в данный момент в сервисе, и не имеют даты выдачи";

this.AllOrdersButton.Click += new System.EventHandler(this.AllOrdersButton\_Click);

//

// toolStripSeparator8

//

this.toolStripSeparator8.Name = "toolStripSeparator8";

this.toolStripSeparator8.Size = new System.Drawing.Size(6, 31);

//

// DiagnosticksButton

//

this.DiagnosticksButton.Image = global::SeviceCenter.Properties.Resources.Status\_diagnostika;

this.DiagnosticksButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.DiagnosticksButton.Name = "DiagnosticksButton";

this.DiagnosticksButton.Size = new System.Drawing.Size(140, 28);

this.DiagnosticksButton.Tag = "Диагностика";

this.DiagnosticksButton.Text = "Диагностика";

this.DiagnosticksButton.Click += new System.EventHandler(this.DiagnosticksButton\_Click);

//

// toolStripSeparator9

//

this.toolStripSeparator9.Name = "toolStripSeparator9";

this.toolStripSeparator9.Size = new System.Drawing.Size(6, 31);

//

// SoglasovanieSKlientomButton

//

this.SoglasovanieSKlientomButton.Image = global::SeviceCenter.Properties.Resources.Status\_SoglasovanieSClientom;

this.SoglasovanieSKlientomButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.SoglasovanieSKlientomButton.Name = "SoglasovanieSKlientomButton";

this.SoglasovanieSKlientomButton.Size = new System.Drawing.Size(249, 28);

this.SoglasovanieSKlientomButton.Tag = "Согласование с клиентом";

this.SoglasovanieSKlientomButton.Text = "Согласование с клиентом";

this.SoglasovanieSKlientomButton.Click += new System.EventHandler(this.SoglasovanieSKlientomButton\_Click);

//

// toolStripSeparator10

//

this.toolStripSeparator10.Name = "toolStripSeparator10";

this.toolStripSeparator10.Size = new System.Drawing.Size(6, 31);

//

// SoglasovanoButton1

//

this.SoglasovanoButton1.Image = global::SeviceCenter.Properties.Resources.Status\_Soglasovano;

this.SoglasovanoButton1.ImageTransparentColor = System.Drawing.Color.Magenta;

this.SoglasovanoButton1.Name = "SoglasovanoButton1";

this.SoglasovanoButton1.Size = new System.Drawing.Size(139, 28);

this.SoglasovanoButton1.Tag = "Согласовано";

this.SoglasovanoButton1.Text = "Согласовано";

this.SoglasovanoButton1.Click += new System.EventHandler(this.SoglasovanoButton1\_Click);

//

// toolStripSeparator11

//

this.toolStripSeparator11.Name = "toolStripSeparator11";

this.toolStripSeparator11.Size = new System.Drawing.Size(6, 31);

//

// InWorkButton

//

this.InWorkButton.Image = global::SeviceCenter.Properties.Resources.Status\_inWork;

this.InWorkButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.InWorkButton.Name = "InWorkButton";

this.InWorkButton.Size = new System.Drawing.Size(169, 28);

this.InWorkButton.Tag = "Принят в работу";

this.InWorkButton.Text = "Принят в работу";

this.InWorkButton.Click += new System.EventHandler(this.InWorkButton\_Click);

//

// toolStripSeparator12

//

this.toolStripSeparator12.Name = "toolStripSeparator12";

this.toolStripSeparator12.Size = new System.Drawing.Size(6, 31);

//

// PartWaitingButton

//

this.PartWaitingButton.Image = global::SeviceCenter.Properties.Resources.Status\_parts;

this.PartWaitingButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.PartWaitingButton.Name = "PartWaitingButton";

this.PartWaitingButton.Size = new System.Drawing.Size(152, 28);

this.PartWaitingButton.Tag = "Ждёт запчасть";

this.PartWaitingButton.Text = "Ждёт запчасть";

this.PartWaitingButton.Click += new System.EventHandler(this.PartWaitingButton\_Click);

//

// toolStripSeparator14

//

this.toolStripSeparator14.Name = "toolStripSeparator14";

this.toolStripSeparator14.Size = new System.Drawing.Size(6, 31);

//

// ReadyStatButton

//

this.ReadyStatButton.Image = global::SeviceCenter.Properties.Resources.StatusReady;

this.ReadyStatButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.ReadyStatButton.Name = "ReadyStatButton";

this.ReadyStatButton.Size = new System.Drawing.Size(81, 28);

this.ReadyStatButton.Tag = "Готов";

this.ReadyStatButton.Text = "Готов";

this.ReadyStatButton.Click += new System.EventHandler(this.ReadyStatButton\_Click);

//

// toolStripSeparator15

//

this.toolStripSeparator15.Name = "toolStripSeparator15";

this.toolStripSeparator15.Size = new System.Drawing.Size(6, 31);

//

// PrinyatPoGarantiiButton

//

this.PrinyatPoGarantiiButton.Image = global::SeviceCenter.Properties.Resources.Status\_PrinyatPoGarantii;

this.PrinyatPoGarantiiButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.PrinyatPoGarantiiButton.Name = "PrinyatPoGarantiiButton";

this.PrinyatPoGarantiiButton.Size = new System.Drawing.Size(202, 28);

this.PrinyatPoGarantiiButton.Tag = "Принят по гарантии";

this.PrinyatPoGarantiiButton.Text = "Принят по гарантии";

this.PrinyatPoGarantiiButton.Click += new System.EventHandler(this.PrinyatPoGarantiiButton\_Click);

//

// toolStripSeparator13

//

this.toolStripSeparator13.Name = "toolStripSeparator13";

this.toolStripSeparator13.Size = new System.Drawing.Size(6, 31);

//

// OutOfSCButton

//

this.OutOfSCButton.Image = global::SeviceCenter.Properties.Resources.Status\_Vidan;

this.OutOfSCButton.ImageTransparentColor = System.Drawing.Color.Magenta;

this.OutOfSCButton.Name = "OutOfSCButton";

this.OutOfSCButton.Size = new System.Drawing.Size(87, 28);

this.OutOfSCButton.Tag = "Выдан";

this.OutOfSCButton.Text = "Выдан";

this.OutOfSCButton.Click += new System.EventHandler(this.OutOfSCButton\_Click);

//

// toolStrip3

//

this.toolStrip3.Font = new System.Drawing.Font("Comic Sans MS", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.toolStrip3.ImageScalingSize = new System.Drawing.Size(20, 20);

this.toolStrip3.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.toolStripLabel1,

this.FullSearchPhone,

this.toolStripLabel3,

this.FullSearchType,

this.toolStripLabel4,

this.FullSearchBrand,

this.toolStripLabel5,

this.FullSearchModel,

this.toolStripLabel6,

this.FullSearchSerial,

this.toolStripLabel7,

this.FullSearchMaster});

this.toolStrip3.Location = new System.Drawing.Point(0, 64);

this.toolStrip3.Name = "toolStrip3";

this.toolStrip3.Size = new System.Drawing.Size(1577, 31);

this.toolStrip3.TabIndex = 6;

this.toolStrip3.Text = "toolStrip3";

this.toolStrip3.Visible = false;

//

// toolStripLabel1

//

this.toolStripLabel1.Name = "toolStripLabel1";

this.toolStripLabel1.Size = new System.Drawing.Size(48, 28);

this.toolStripLabel1.Text = "Тел:";

//

// FullSearchPhone

//

this.FullSearchPhone.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

this.FullSearchPhone.Font = new System.Drawing.Font("Segoe UI", 9F);

this.FullSearchPhone.Name = "FullSearchPhone";

this.FullSearchPhone.Size = new System.Drawing.Size(173, 31);

this.FullSearchPhone.KeyDown += new System.Windows.Forms.KeyEventHandler(this.FullSearchPhone\_KeyDown);

this.FullSearchPhone.TextChanged += new System.EventHandler(this.FullSearchPhone\_TextChanged);

//

// toolStripLabel3

//

this.toolStripLabel3.Name = "toolStripLabel3";

this.toolStripLabel3.Size = new System.Drawing.Size(49, 28);

this.toolStripLabel3.Text = "Тип:";

//

// FullSearchType

//

this.FullSearchType.FlatStyle = System.Windows.Forms.FlatStyle.Standard;

this.FullSearchType.Name = "FullSearchType";

this.FullSearchType.Size = new System.Drawing.Size(199, 31);

this.FullSearchType.SelectedIndexChanged += new System.EventHandler(this.FullSearchType\_SelectedIndexChanged);

this.FullSearchType.KeyDown += new System.Windows.Forms.KeyEventHandler(this.FullSearchType\_KeyDown);

this.FullSearchType.TextChanged += new System.EventHandler(this.FullSearchType\_TextChanged);

//

// toolStripLabel4

//

this.toolStripLabel4.Name = "toolStripLabel4";

this.toolStripLabel4.Size = new System.Drawing.Size(64, 28);

this.toolStripLabel4.Text = "Бренд:";

//

// FullSearchBrand

//

this.FullSearchBrand.FlatStyle = System.Windows.Forms.FlatStyle.Standard;

this.FullSearchBrand.Name = "FullSearchBrand";

this.FullSearchBrand.Size = new System.Drawing.Size(169, 31);

this.FullSearchBrand.SelectedIndexChanged += new System.EventHandler(this.FullSearchBrand\_SelectedIndexChanged);

this.FullSearchBrand.KeyDown += new System.Windows.Forms.KeyEventHandler(this.FullSearchBrand\_KeyDown);

this.FullSearchBrand.TextChanged += new System.EventHandler(this.FullSearchBrand\_TextChanged);

//

// toolStripLabel5

//

this.toolStripLabel5.Name = "toolStripLabel5";

this.toolStripLabel5.Size = new System.Drawing.Size(79, 28);

this.toolStripLabel5.Text = "Модель:";

//

// FullSearchModel

//

this.FullSearchModel.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

this.FullSearchModel.Font = new System.Drawing.Font("Segoe UI", 9F);

this.FullSearchModel.Name = "FullSearchModel";

this.FullSearchModel.Size = new System.Drawing.Size(162, 31);

this.FullSearchModel.KeyDown += new System.Windows.Forms.KeyEventHandler(this.FullSearchModel\_KeyDown);

this.FullSearchModel.TextChanged += new System.EventHandler(this.FullSearchModel\_TextChanged);

//

// toolStripLabel6

//

this.toolStripLabel6.Name = "toolStripLabel6";

this.toolStripLabel6.Size = new System.Drawing.Size(125, 28);

this.toolStripLabel6.Text = "Серийный №:";

//

// FullSearchSerial

//

this.FullSearchSerial.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

this.FullSearchSerial.Font = new System.Drawing.Font("Segoe UI", 9F);

this.FullSearchSerial.Name = "FullSearchSerial";

this.FullSearchSerial.Size = new System.Drawing.Size(146, 31);

this.FullSearchSerial.KeyDown += new System.Windows.Forms.KeyEventHandler(this.FullSearchSerial\_KeyDown);

this.FullSearchSerial.TextChanged += new System.EventHandler(this.FullSearchSerial\_TextChanged);

//

// toolStripLabel7

//

this.toolStripLabel7.Name = "toolStripLabel7";

this.toolStripLabel7.Size = new System.Drawing.Size(75, 28);

this.toolStripLabel7.Text = "Мастер:";

//

// FullSearchMaster

//

this.FullSearchMaster.FlatStyle = System.Windows.Forms.FlatStyle.Standard;

this.FullSearchMaster.Name = "FullSearchMaster";

this.FullSearchMaster.Size = new System.Drawing.Size(121, 31);

this.FullSearchMaster.SelectedIndexChanged += new System.EventHandler(this.FullSearchMaster\_SelectedIndexChanged);

this.FullSearchMaster.KeyDown += new System.Windows.Forms.KeyEventHandler(this.FullSearchMaster\_KeyDown);

this.FullSearchMaster.TextChanged += new System.EventHandler(this.FullSearchMaster\_TextChanged);

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(1577, 822);

this.Controls.Add(this.toolStrip3);

this.Controls.Add(this.toolStrip2);

this.Controls.Add(this.SearchFIOTextBox);

this.Controls.Add(this.toolStrip1);

this.Controls.Add(this.statusStrip);

this.Controls.Add(this.button1);

this.Controls.Add(this.NewClientButton);

this.Controls.Add(this.MainListView);

this.Margin = new System.Windows.Forms.Padding(4);

this.Name = "Form1";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

this.FormClosed += new System.Windows.Forms.FormClosedEventHandler(this.Form1\_FormClosed);

this.Load += new System.EventHandler(this.Form1\_Load);

this.statusStrip.ResumeLayout(false);

this.statusStrip.PerformLayout();

this.toolStrip1.ResumeLayout(false);

this.toolStrip1.PerformLayout();

this.toolStrip2.ResumeLayout(false);

this.toolStrip2.PerformLayout();

this.toolStrip3.ResumeLayout(false);

this.toolStrip3.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

}

Приложение В

Результаты тестирования

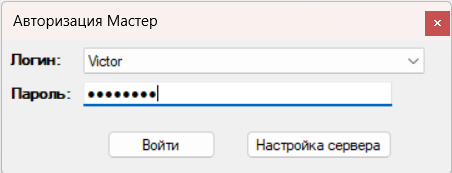


Рисунок В.1 – Тестирование сценария №1 (Вход в программу)

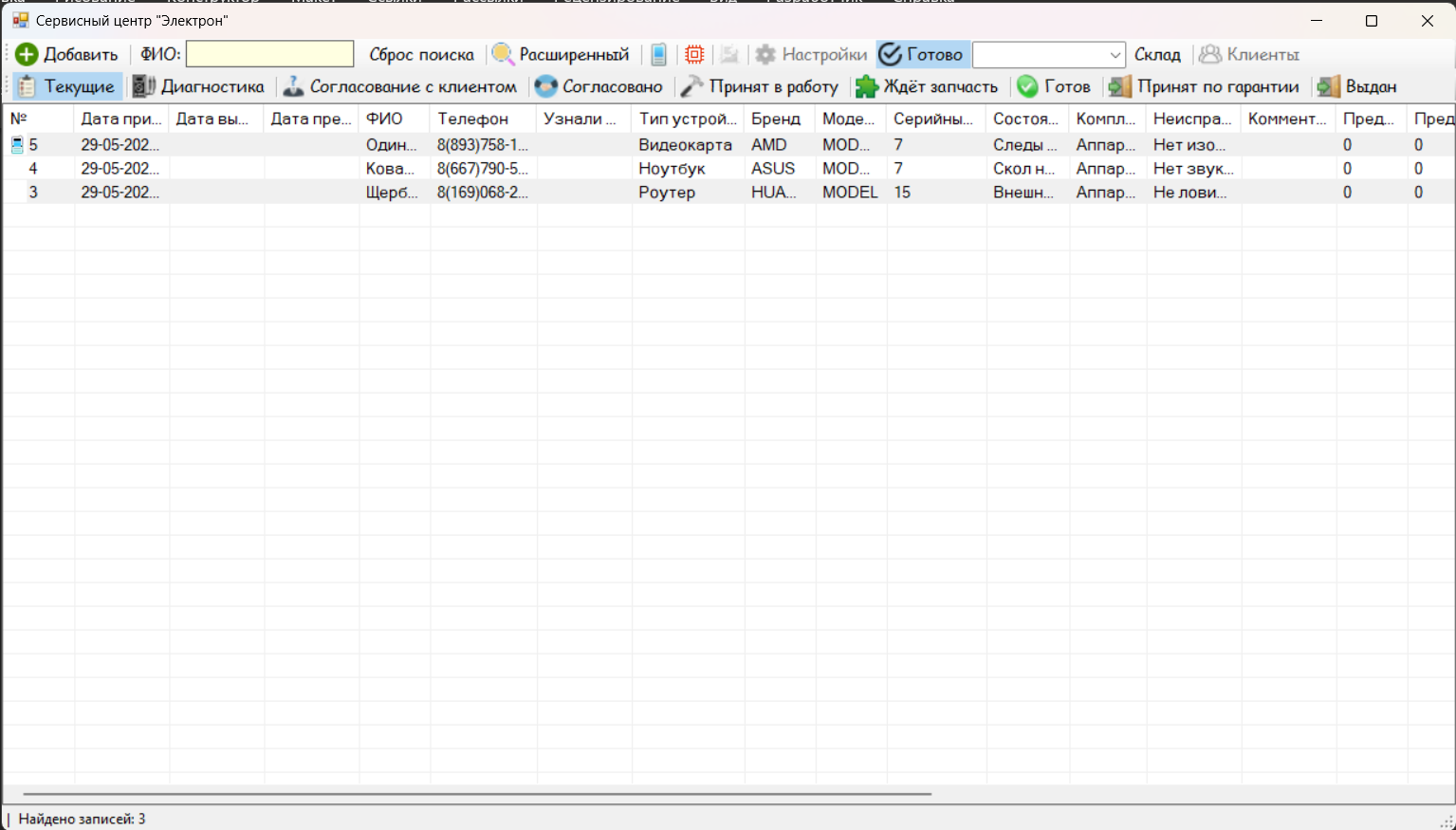


Рисунок В.2 – Тестирование сценария №1 (Главная форма)

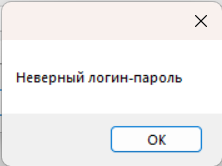


Рисунок В.3– Тестирование сценария №2 (Ошибка авторизации)

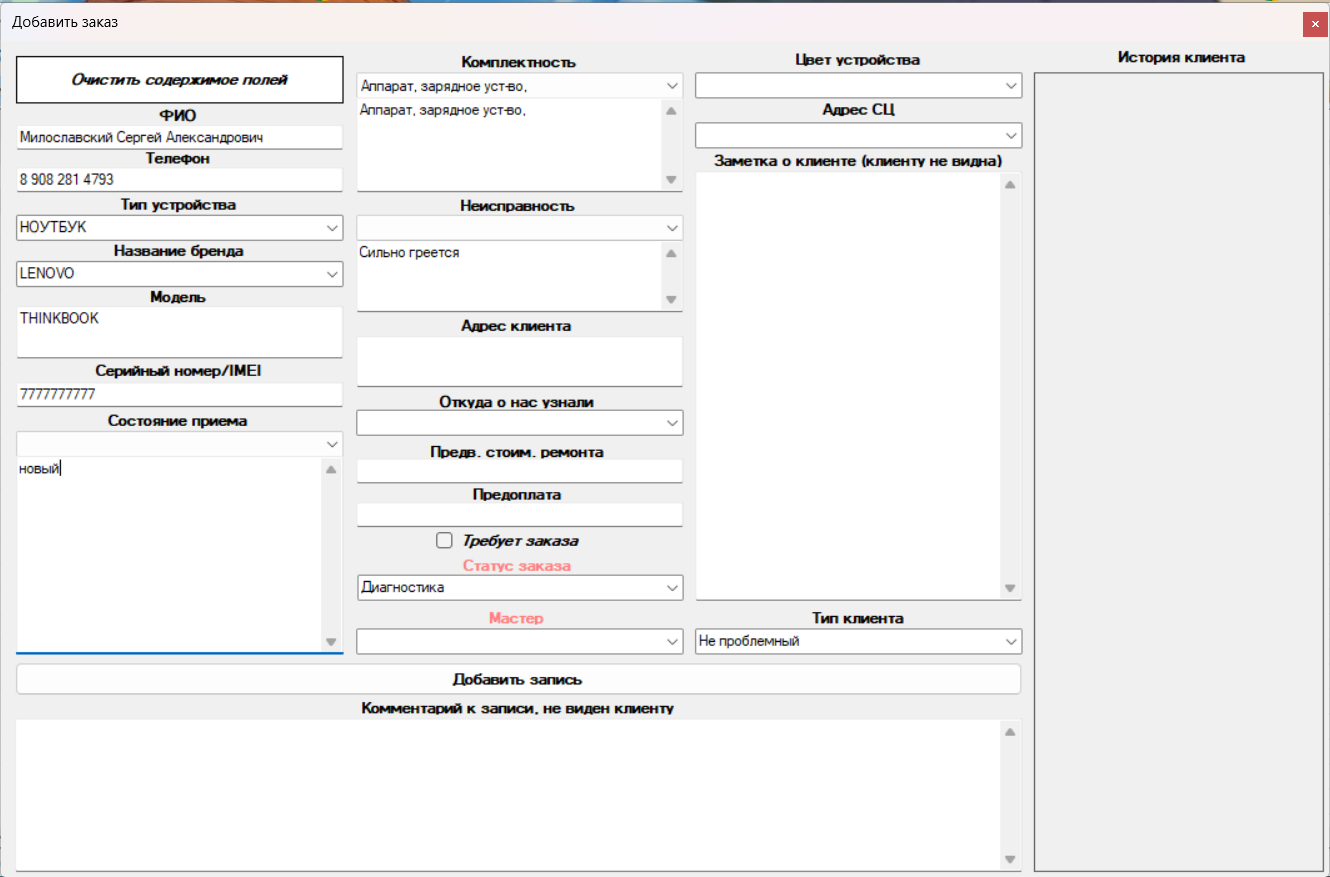


Рисунок В.4 – Тестирование сценария №3 (Ввод данных)

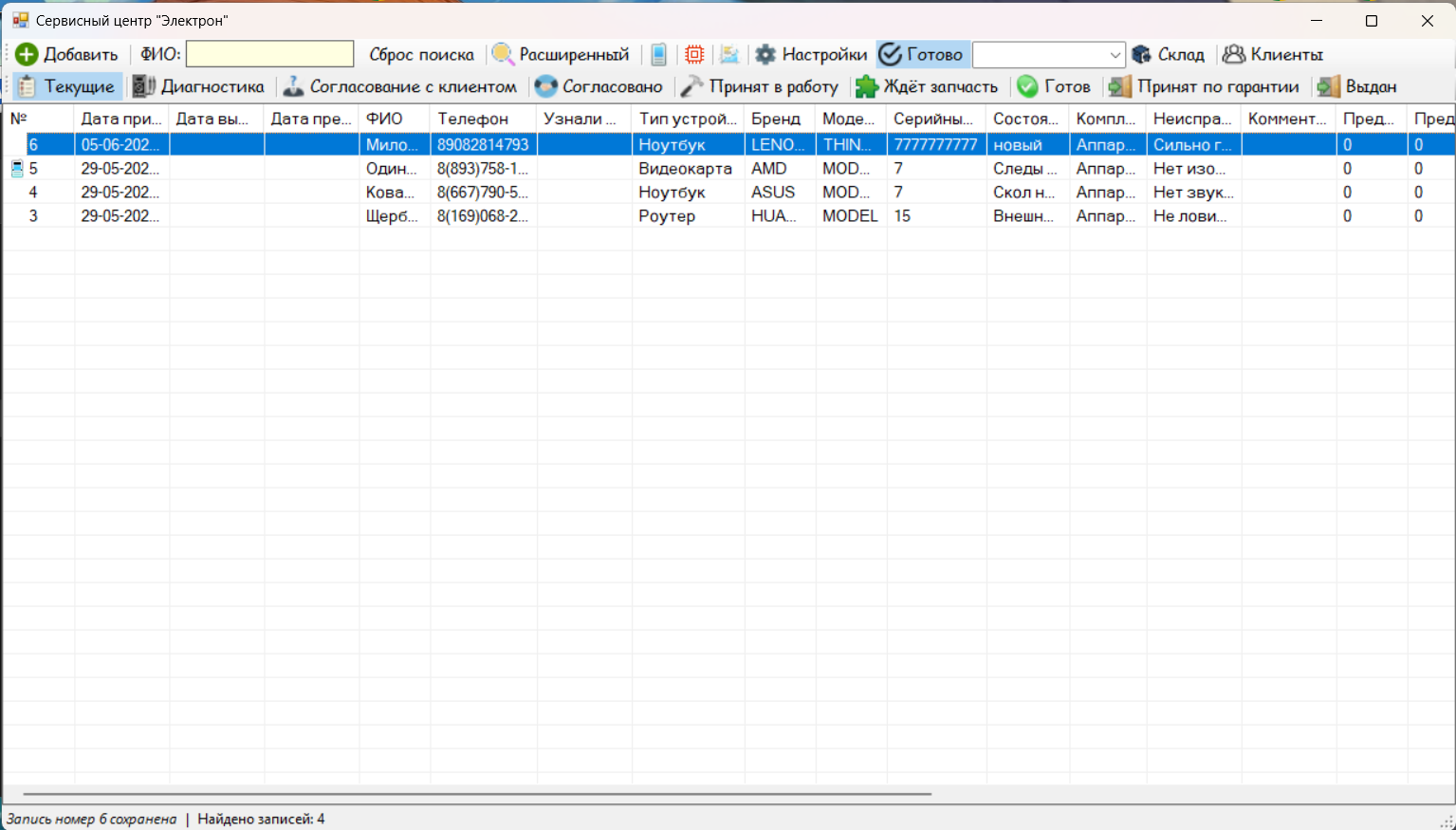


Рисунок В.5 – Тестирование сценария №3 (Запись создана)

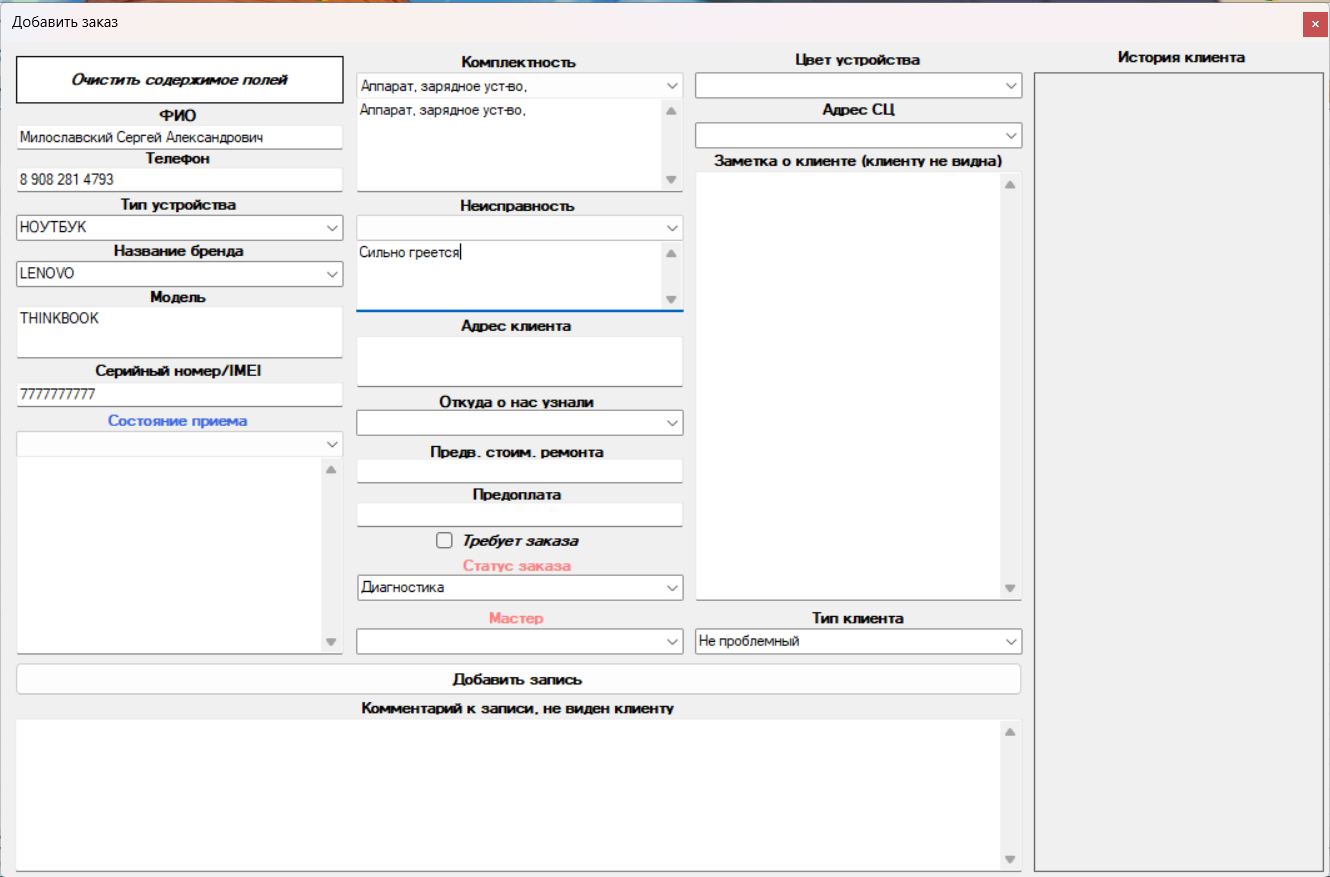


Рисунок В.6 – Тестирование сценария №4 (Ввод данных)

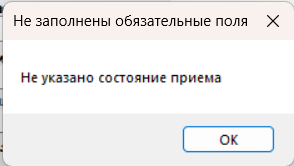


Рисунок В.7 –Тестирование сценария №4 (Окно ошибки)

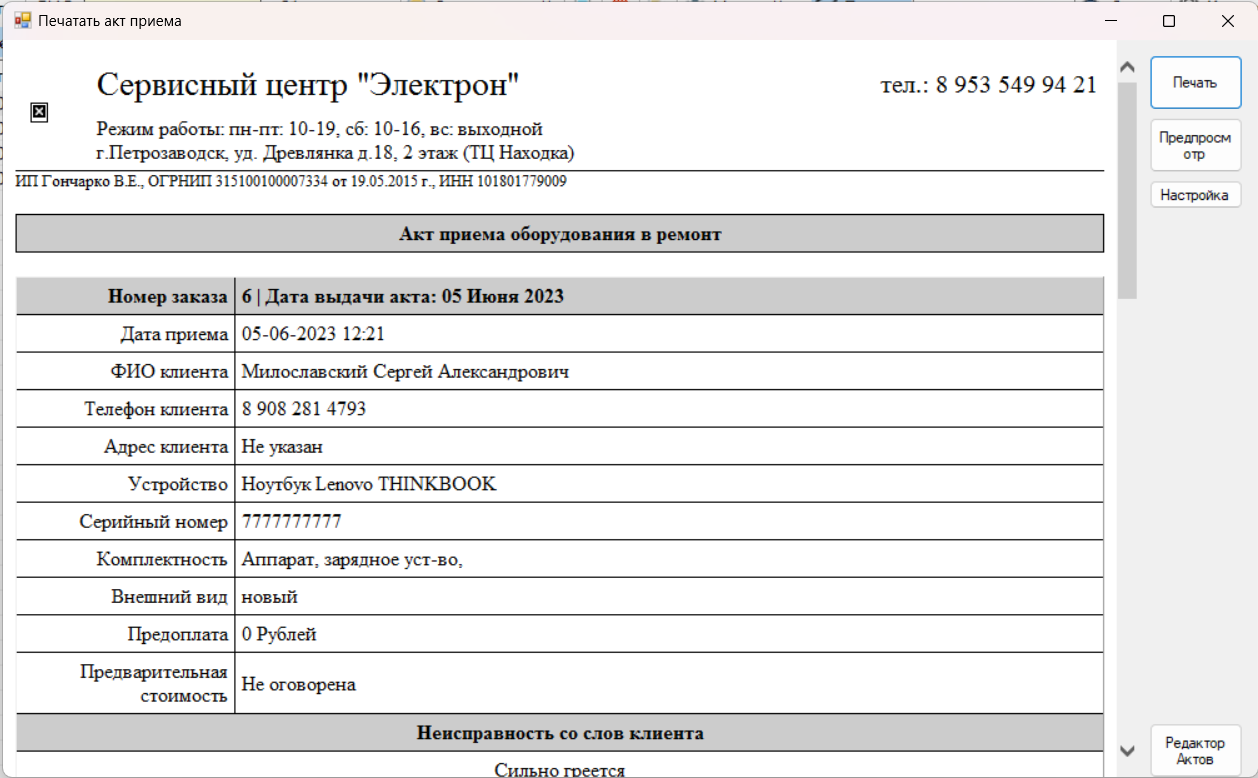


Рисунок В.8– Тестирование сценария №5 (Форма печати акта приема)

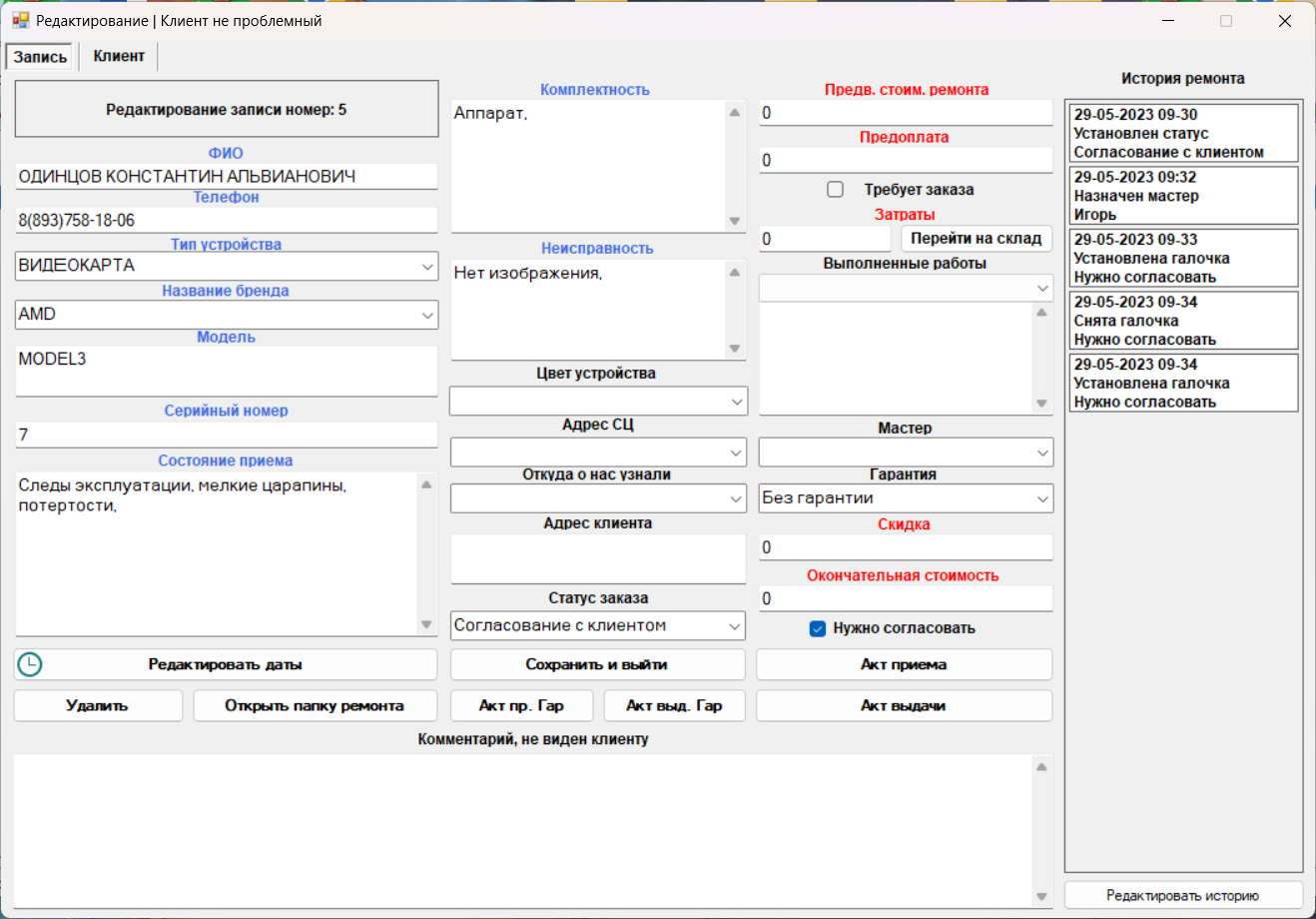


Рисунок В.9 – Тестирование сценария №6 (Просмотр заявки)

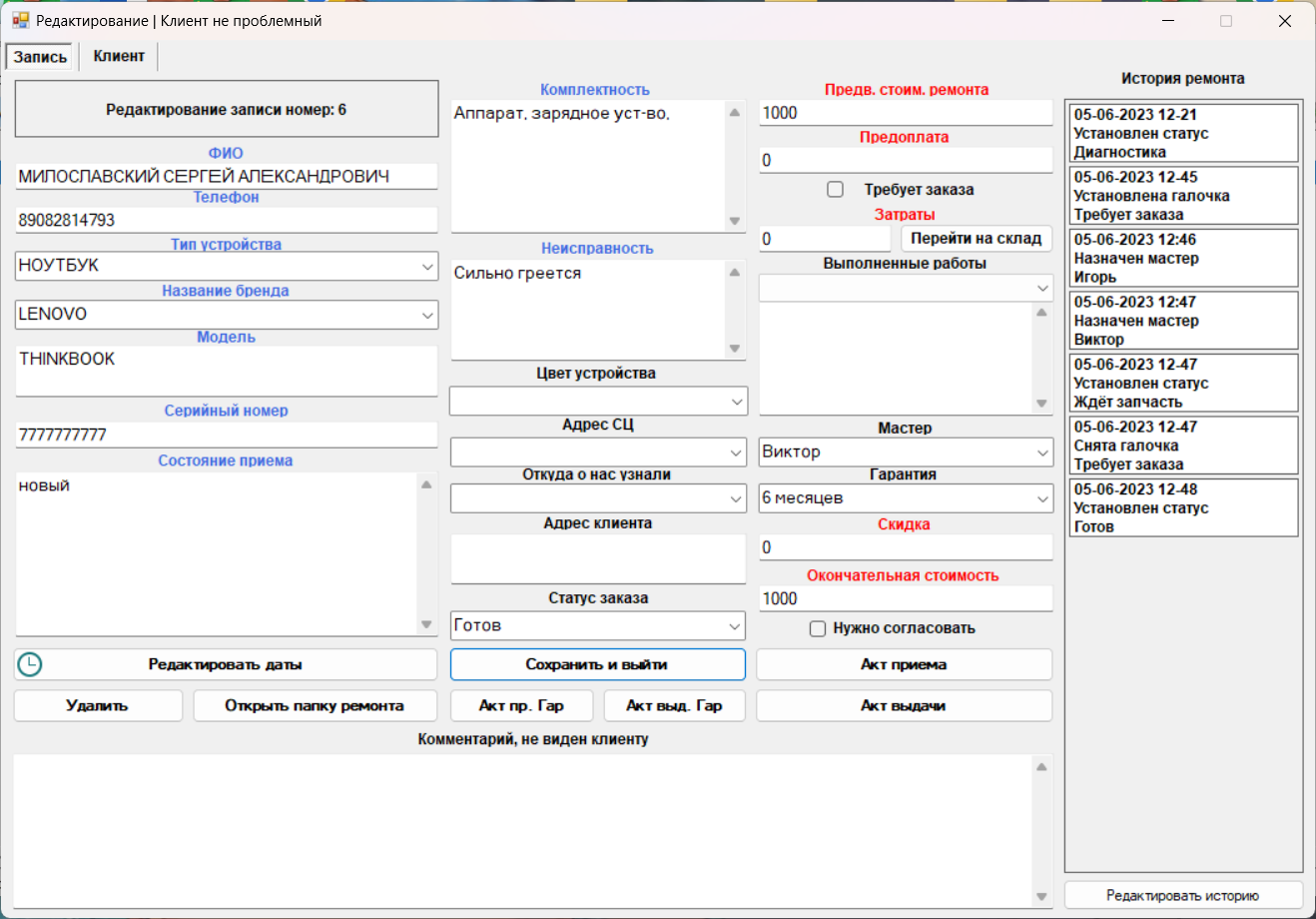


Рисунок В.10– Тестирование сценария №7 (Редактирование заявки)

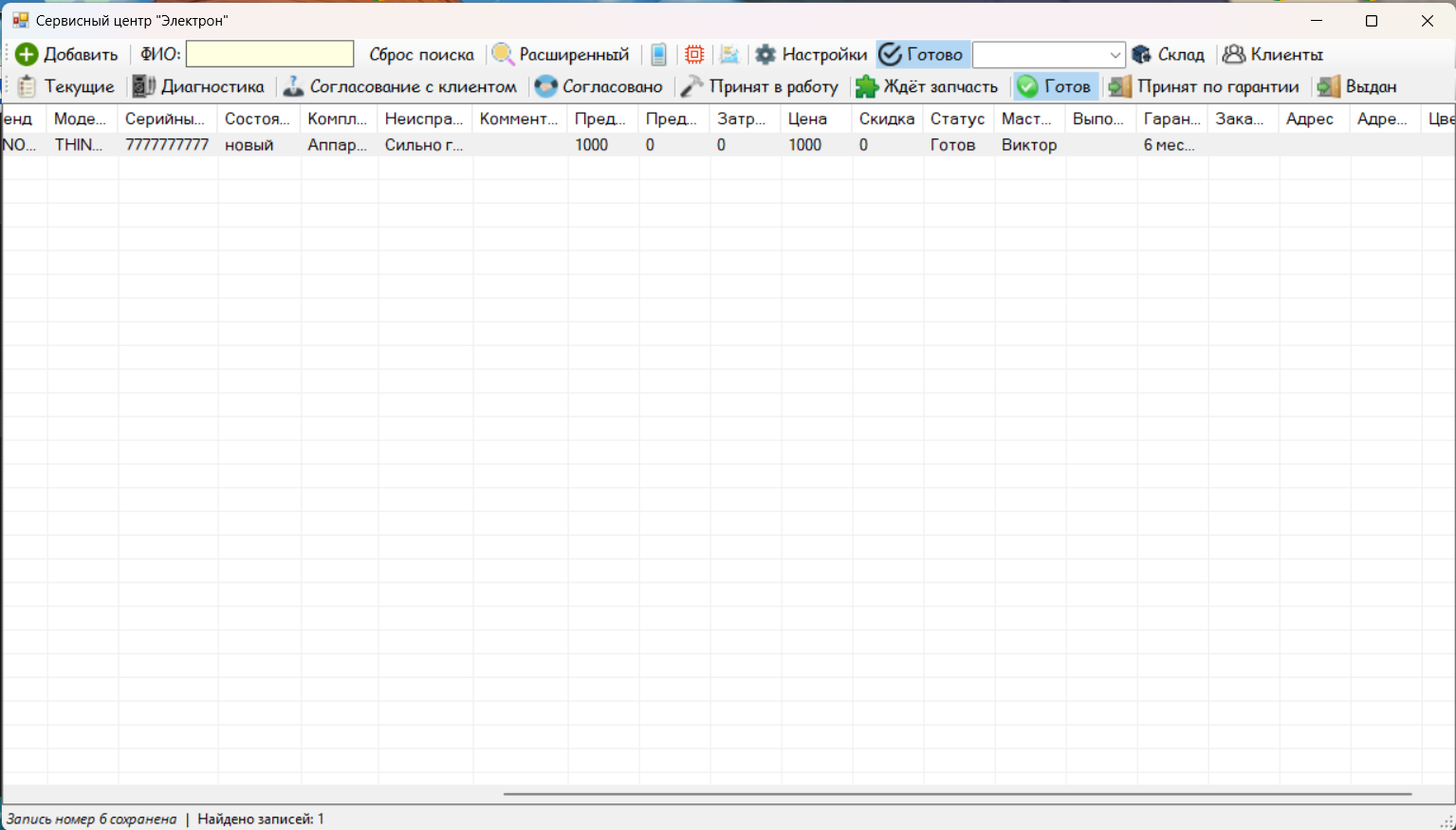


Рисунок В.11– Тестирование сценария №7 (Результат редактирования заявки)