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: 004.4(079.2)

В75

**Милославский Сергей Александрович**

**Автоматизированное рабочее место сервисного центра**

Выпускная квалификационная работа на соискание квалификации

**Бакалавр**

Направление подготовки

**09.03.01 – Информатика и вычислительная техник**

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 80 страницах, из них 55 страниц основного текста, состоит из, трех глав, заключения, списка литературы, из 19 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

Ключевые слова: автоматизация, оператор, мастер.

**ОГЛАВЛЕНИЕ**

[Введение 5](#_Toc136432050)

[1. АНАЛИЗ ЗАДАЧ, ПРОГРАММНЫХ СРЕДСТВ И ОБОСНОВАНИЕ РАЗРАБОТКИ ПО 7](#_Toc136432051)

[1.1 Анализ технического задания 7](#_Toc136432052)

[1.2 Описание предметной области 8](#_Toc136432053)

[1.3 Обзор аналогов 9](#_Toc136432054)

[1.3.1 СК24 9](#_Toc136432055)

[1.3.2 ProMaster 10](#_Toc136432056)

[1.3.3 ServiceMP 12](#_Toc136432057)

[1.3.3 Краткий вывод по аналогам 14](#_Toc136432058)

[1.4 Обзор существующих средств разработки 14](#_Toc136432059)

[1.5 Постановка задачи 16](#_Toc136432060)

[1.6 Выводы по главе 18](#_Toc136432061)

[2. Проектирование ПО 19](#_Toc136432062)

[2.1 Определение вариантов использования разрабатываемых программных средств 19](#_Toc136432063)

[2.2 Проектирование БД 24](#_Toc136432064)

[2.3 Интерфейс пользователя 27](#_Toc136432065)

[2.4 Выводы по главе 28](#_Toc136432066)

[3 разработка ПО 29](#_Toc136432067)

[3.1 Реализация ПО 29](#_Toc136432068)

[3.2 Тестирование ПО 29](#_Toc136432069)

[3.3 Выводы по главе 29](#_Toc136432070)

[Заключение 30](#_Toc136432071)

[перечень сокращений и условных обозначений 31](#_Toc136432072)

[список использованных источников 32](#_Toc136432073)

[Приложение А 33](#_Toc136432074)

[Приложение б 34](#_Toc136432075)

[Приложение В 35](#_Toc136432076)

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.

Эта 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.

1.3 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.

1.3.1 СК24

SC24 — комплексное программное решение, разработанное специально для сервисных центров и ремонтных мастерских. 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.

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. [12]

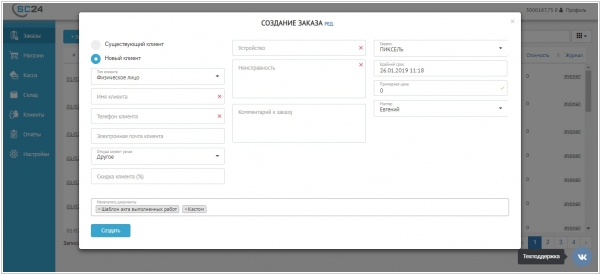


Рисунок 1.3.1.1 – Форма программы SC24

Преимущества:

* SC24 предоставляет простой и интуитивно понятный интерфейс, в котором может легко ориентироваться нетехнический персонал;
* он 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.

1.3.2 ProMaster

ProMaster — еще одна популярная программа для сервисных центров и ремонтных мастерских. 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.

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. [13]

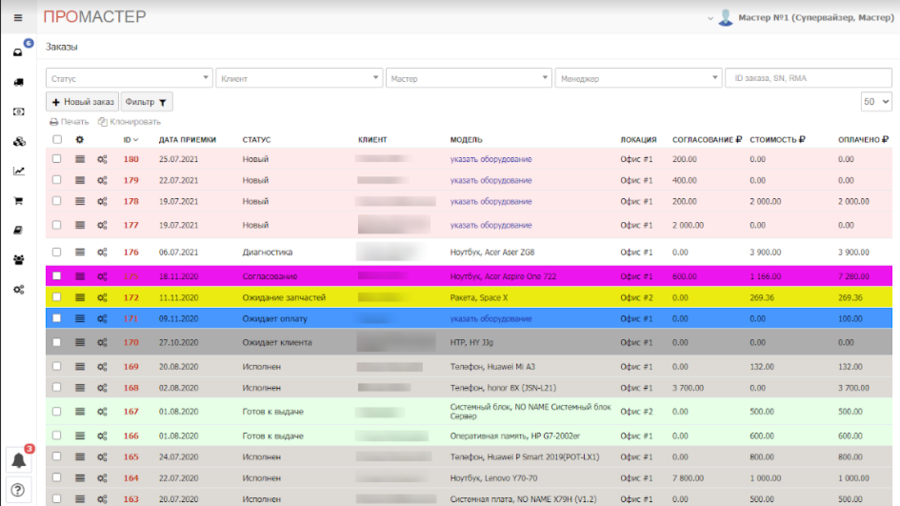


Рисунок 1.3.1.2 – Форма программы ProMaster

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.

1.3.3 ServiceMP

ServiceMP — это программа управления обслуживанием, разработанная специально для малых и средних ремонтных мастерских и сервисных центров. Он 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.

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. [14]

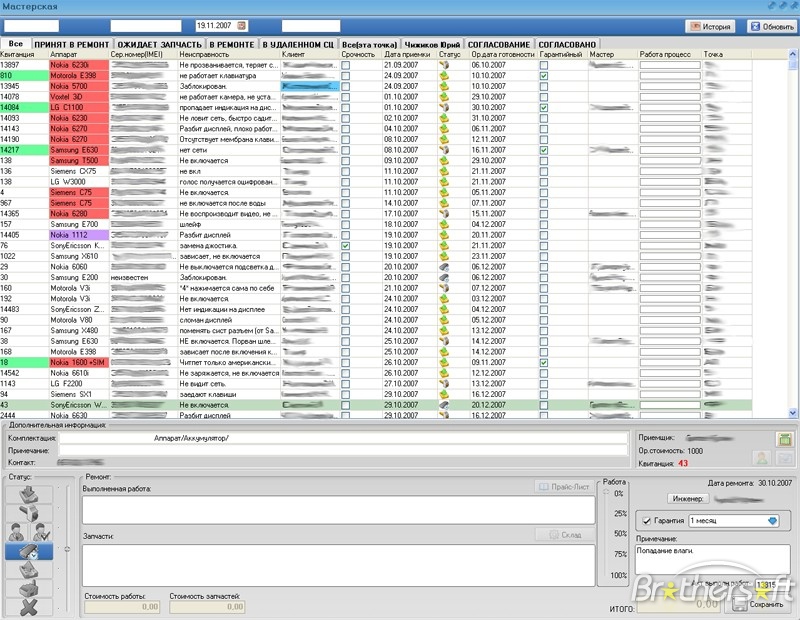


Рисунок 1.3.1.3 – Форма программы ServiceMP

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.

1.3.3 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.

1.4 Обзор существующих 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. Его стабильность, производительность и поддержка многопользовательских сред делают его подходящим для обработки данных, требуемых сервисным центром. [4]

Фреймворки UI:

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

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.

1.5 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 ИС «RepairCenter», предоставляющая вышеописанный функционал.

Разработка ИС 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].

1.6 Выводы по главе

В данной главе был выполнен анализ предметной области, где был рассмотрен:

* 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 2.1.1). [15]



Рисунок 2.1.1 – Диаграмма вариантов использования

Краткое описание варианта использования «Вход в систему» представлено в таблице 2.1.1

Таблица 2.1.1 – Краткое описание варианта использования «Вход в систему»

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

*Продолжение таблицы 2.1.1*

|  |  |
| --- | --- |
| 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 2.1.2

Таблица 2.1.2 – Краткое описание варианта использования «Введение сведений о клиентах»

|  |  |
| --- | --- |
| Название | Введение сведений о клиентах |
| Действующие лица | Оператор |
| 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 2.1.3

Таблица 2.1.3 – Краткое описание варианта использования «Добавление заявки на ремонт»

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

*Продолжение таблицы 2.1.3*

|  |  |
| --- | --- |
| 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 2.1.4

Таблица 2.1.4 – Краткое описание варианта использования «Просмотр заявки на ремонт»

|  |  |
| --- | --- |
| Название | Просмотр заявки на ремонт |
| Действующие лица | Оператор, 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 2.1.5

Таблица 2.1.5 – Краткое описание варианта использования «Редактирование заявки на ремонт»

|  |  |
| --- | --- |
| Название | Редактирование заявки на ремонт |
| Действующие лица | Оператор, 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 2.1.5*

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

Краткое описание варианта использования «Просмотр запчастей» представлено в таблице 2.1.6

Таблица 2.1.6 – Краткое описание варианта использования «Просмотр запчастей»

|  |  |
| --- | --- |
| Название | Просмотр запчастей |
| Действующие лица | Мастер |
| Цель | Просмотр и выбор необходимых запчастей для ремонта |
| Предусловие | Открыто окно редактирования заявки |
| Успешный сценарий | 1. Пользователь нажимает на кнопку «Перейти на склад» 2. Пользователь дважды наживает на выбранную запчасть 3. Открывается окно просмотра запчасти 4. Пользователь выбирает необходимое число запчастей 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 запчастей» представлено в таблице 2.1.7

Таблица 2.1.6 – Краткое описание варианта использования «Добавление запчастей»

|  |  |
| --- | --- |
| Название | Добавление запчастей |
| 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 |
| Результат | Запчасть добавлена на склад |

Краткое описание варианта использования «Редактирование запчасти» представлено в таблице 2.1.8

Таблица 2.1.8 – Краткое описание варианта использования «Редактирование запчасти»

|  |  |
| --- | --- |
| Название | Редактирование запчастей |
| Действующие лица | Кладовщик |
| Цель | Изменение информации запчасти на складе |
| Предусловие | Открыто окно склад |
| Успешный сценарий | 1. Пользователь дважды наживает на выбранную запчасть 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 запчасти» представлено в таблице 2.1.8

Таблица 2.1.8 – Краткое описание варианта использования «Удаление запчасти»

|  |  |
| --- | --- |
| Название | Удаление запчастей |
| Действующие лица | Кладовщик |
| Цель | Удаление информации об запчасти на складе |
| Предусловие | Открыто окно склад |
| Успешный сценарий | 1. Пользователь дважды наживает на выбранную запчасть 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 2.3.1.



Рисунок 2.3.1 – ER-диаграмма

Описание отношений и их атрибутов представлено в таблице 2.3.1.

Таблица 2.3.1 – Описание отношений и их атрибутов

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

*Продолжение таблицы 2.3.1*

|  |  |
| --- | --- |
| 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 2.3.1*

|  |  |
| --- | --- |
| 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 2.3.1.



Рисунок 2.3.1 – Добавление заявки на ремонт

Диаграмма состояний, описывающая добавление просмотр заявки представлена на рисунке 2.3.2.



Рисунок 2.3.2 – Просмотр заявки

Диаграмма состояний, описывающая добавление редактирование заявки на ремонт, представлена на рисунке 2.3.2.



Рисунок 2.3.3 – Редактирование заявки

2.4 Выводы по главе

В данной главе было выполнено проектирование ПО, где были определены:

* основные варианты использования;
* структура БД;
* пользовательский интерфейс ПО.

3 разработка ПО

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 3.1.1 представлена диаграмма развёртывания.

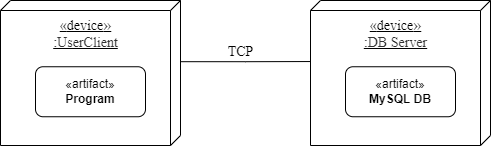


Рисунок 3.1.1 –Диаграмма развертывания

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

3.2 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 3.2.1.

Таблица 3.2.1 – 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» | Всплывающее окно с ошибкой | Успешное завершение теста (рисунки В.6 – В.7) | Ожидаемый результат совпал с полученным |

*Продолжение таблицы 3.2.1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | Неисправность: «Сильно греется» |  |  |  |
| 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 |

3.3 Выводы по главе

В данной главе была описана реализация и выполнено тестирование ПО.

Заключение

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

Приложение б

Код программы

Приложение В

Результаты тестирования

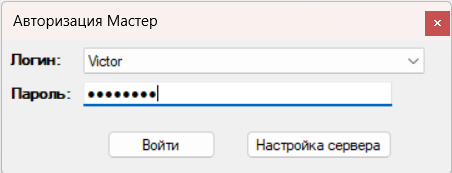


Рисунок В.1 – Тестирование сценария №1(Вход в программу)

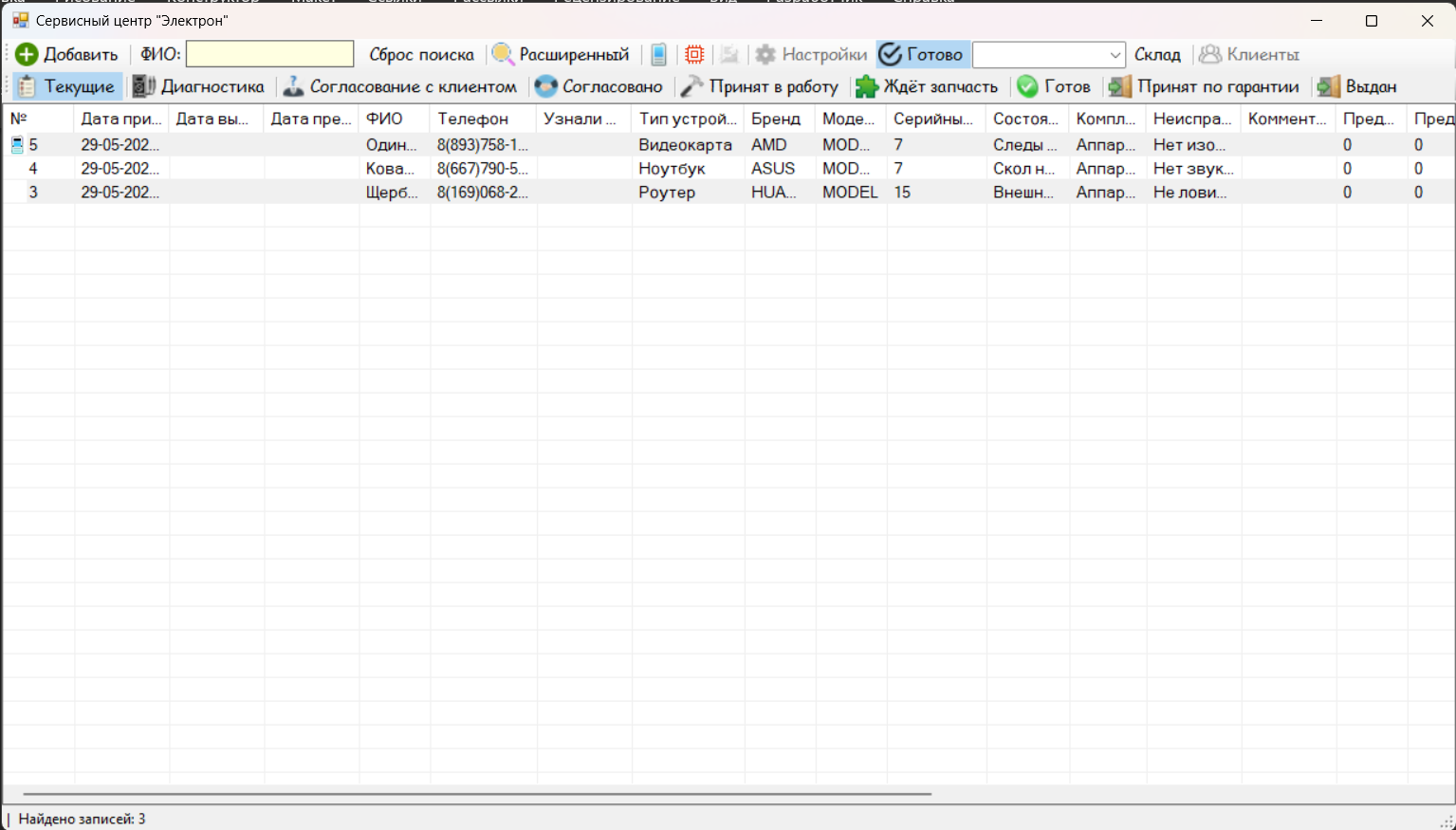


Рисунок В.2 – Тестирование сценария №1(Главная форма)

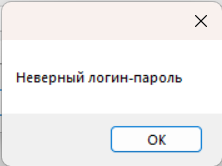


Рисунок В.3– Тестирование сценария №2 (Ошибка авторизации)

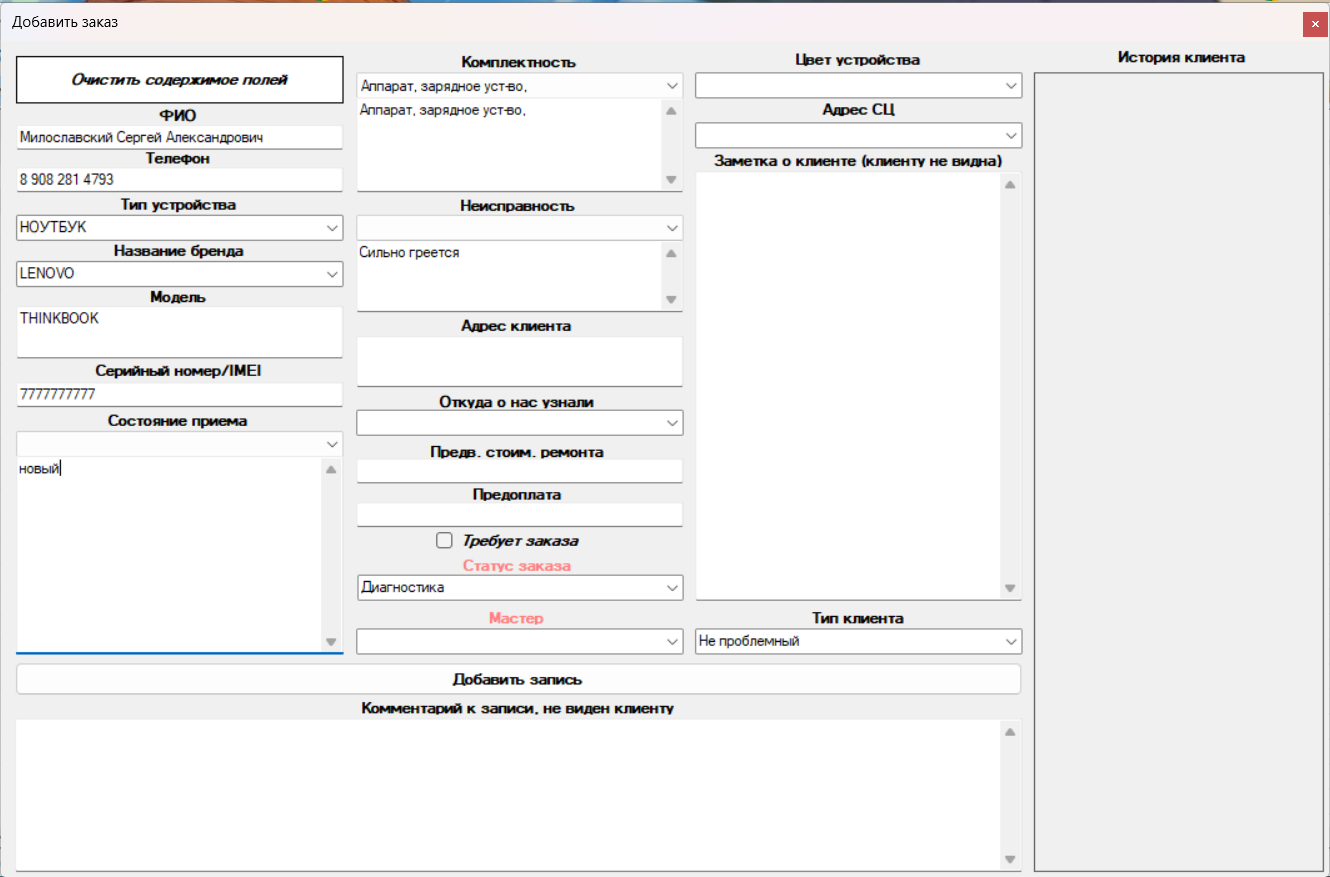


Рисунок В.4 – Тестирование сценария №3 (Ввод данных)

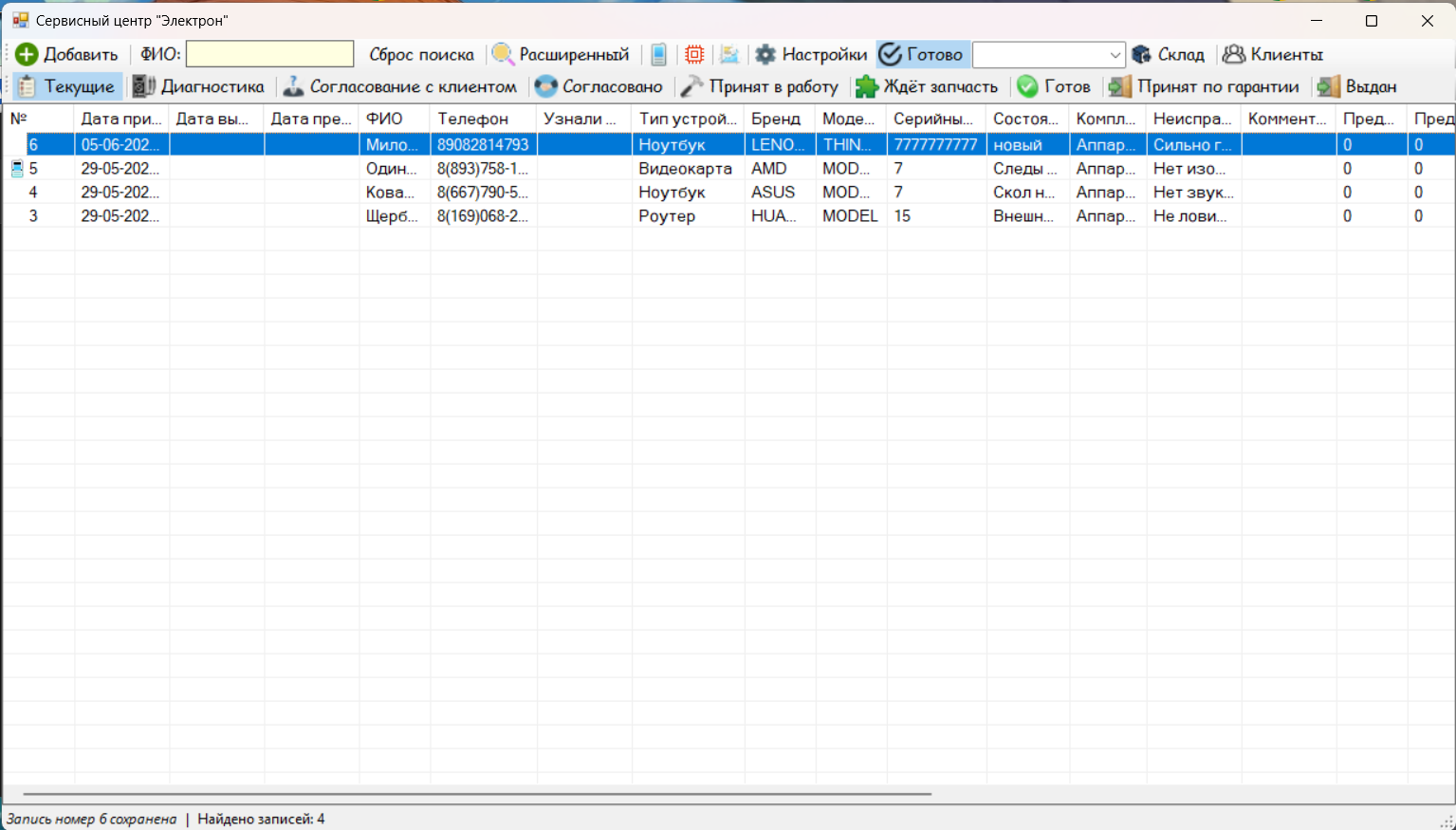


Рисунок В.5 – Тестирование сценария №3 (Запись создана)

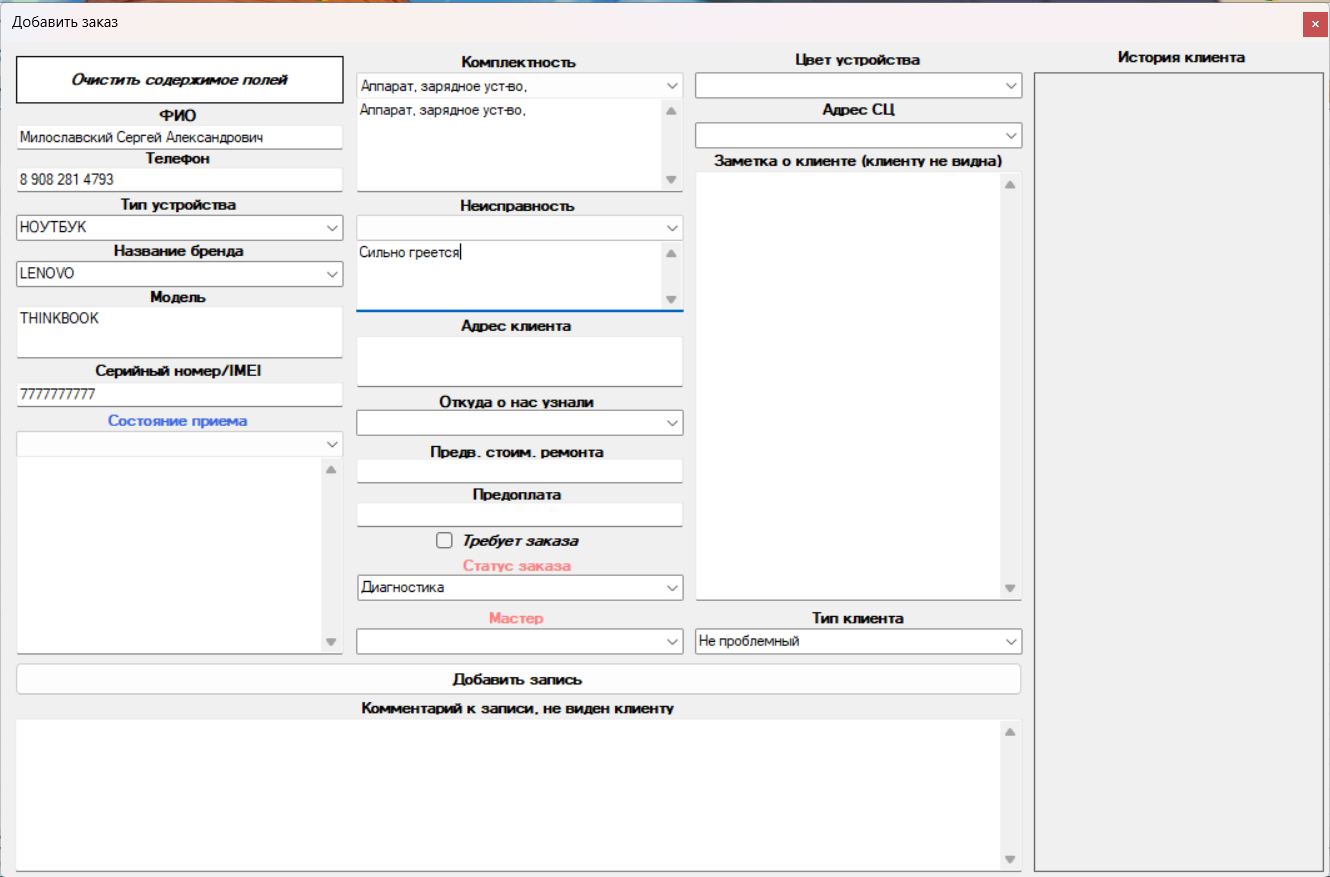


Рисунок В.6 – Тестирование сценария №4 (Ввод данных)

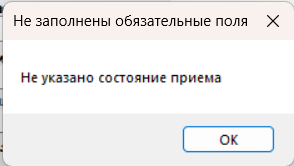


Рисунок В.7 –Тестирование сценария №4 (Окно ошибки)

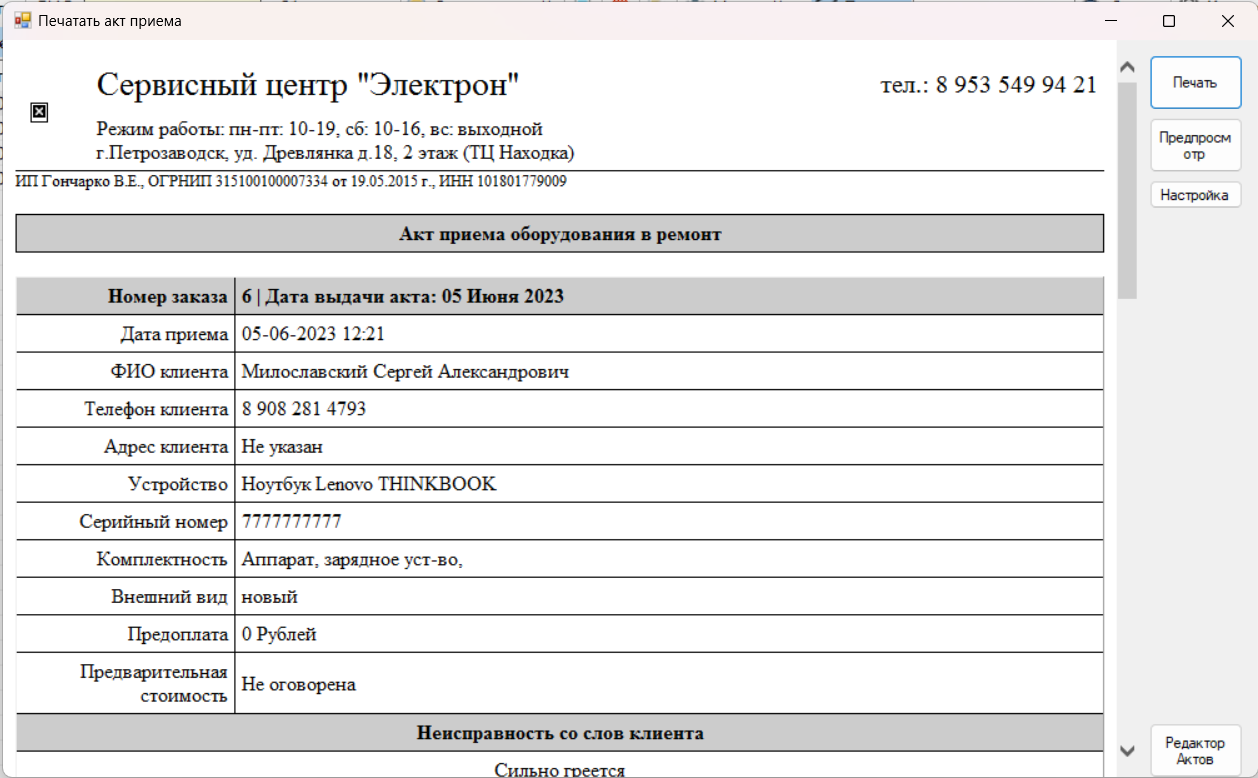


Рисунок В.8– Тестирование сценария №5 (Форма печати акта приема)

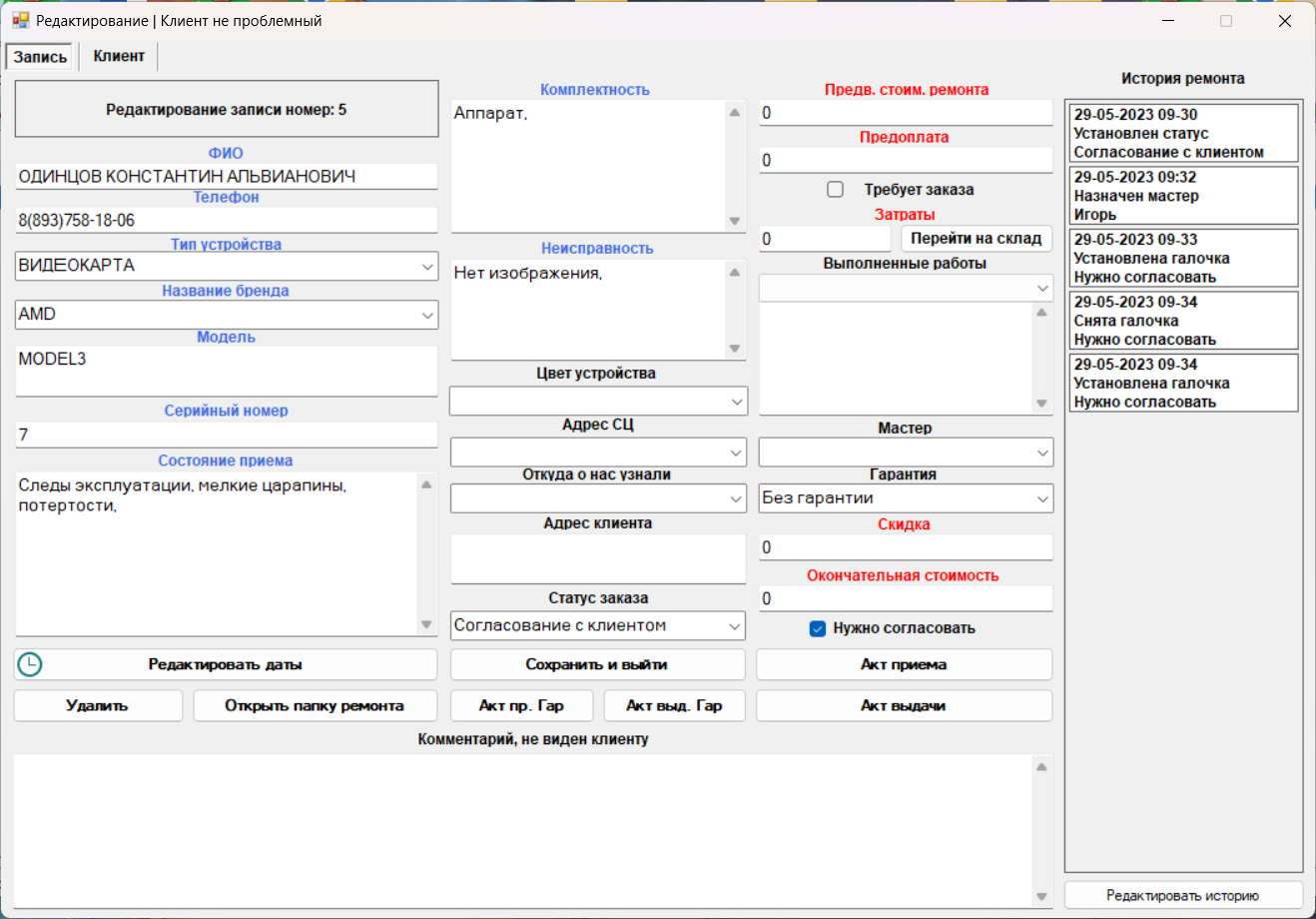


Рисунок В.9 – Тестирование сценария №6 (Просмотр заявки)

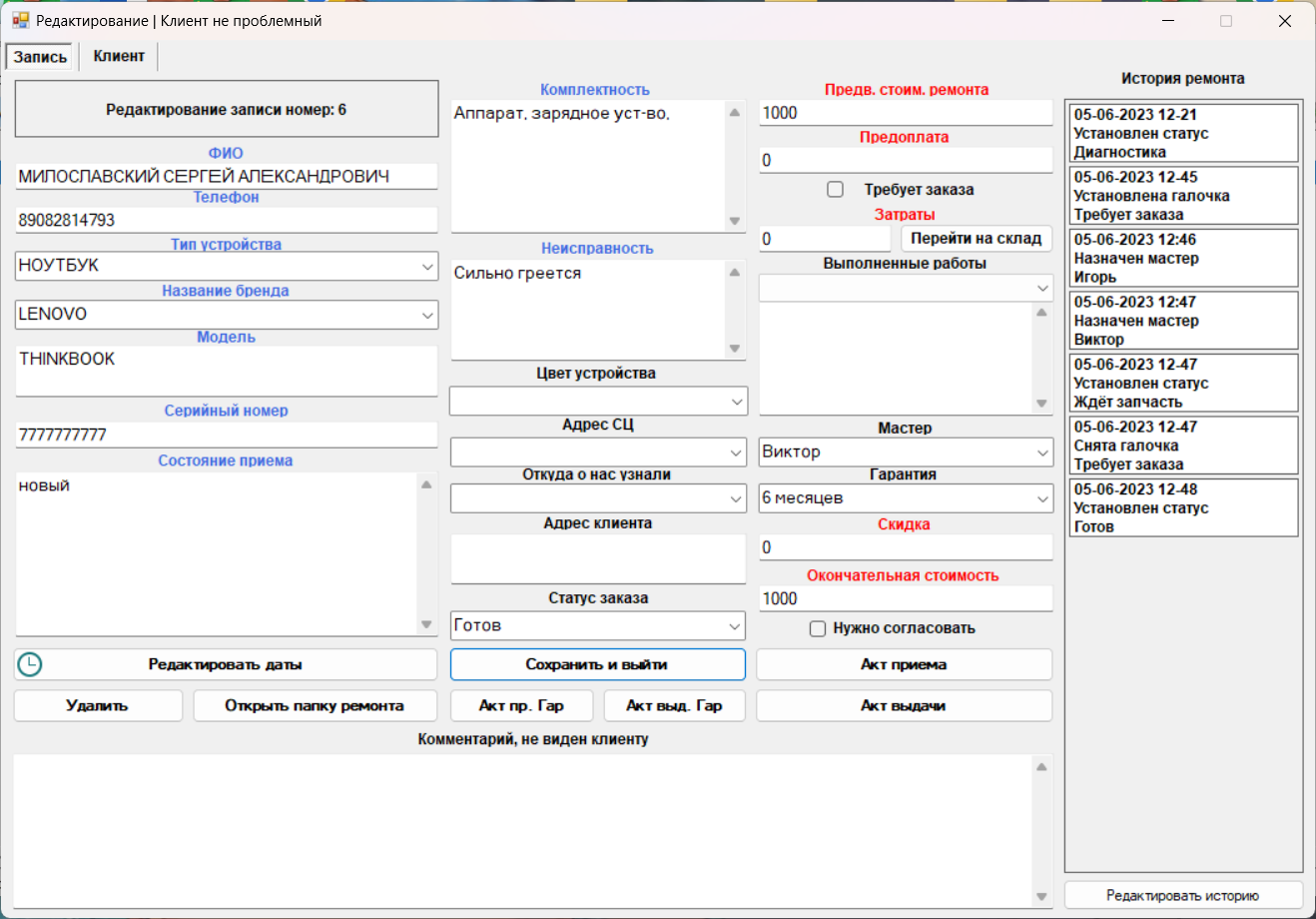


Рисунок В.10– Тестирование сценария №7 (Редактирование заявки)

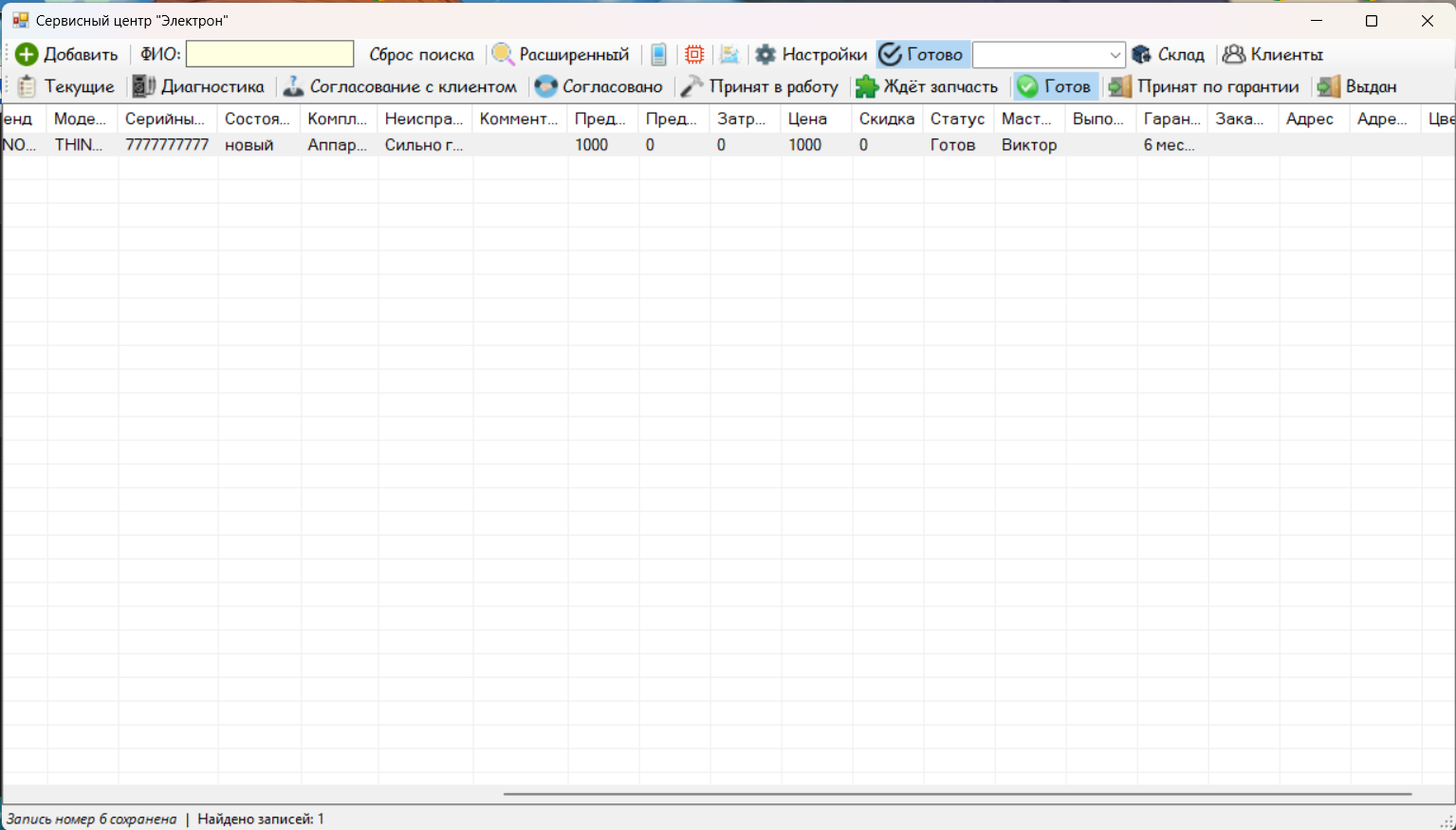


Рисунок В.11– Тестирование сценария №7 (Результат редактирования заявки)