|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Министерство образования Московской области  Государственное образовательное учреждение высшего образования Московской области  «Государственный гуманитарно-технологический университет»  Промышленно-экономический колледж | | | | | |
| **КУРСОВОЙ ПРОЕКТ** | | | | | |
| **РАЗРАБОТКА ИНФОРМАЦИОННОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЁТА ОБОРУДОВАНИЯ ДЛЯ ФАРМАЦЕВТИЧЕСКОЙ КОМПАНИИ** | | | | | |
| дисциплина МДК.01.01. Разработка программных модулей | | | | | |
|  | **Выполнил:**  **Жагрин Александр Денисович**  студента(-ка) группы ИСП.21.1А  по специальности  09.02.07 Информационные системы и программирование  очной формы обучения | | | | |
| **Руководитель:**  **Николаев Александр Александрович** | | | | |
| Оценка | | | | |
|  |  | / |  | / |
|  |  |  | Подпись руководителя | |
| Орехово-Зуево  2025 | | | | | |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc189494378)

[Глава 1. Общие понятия инструментов разработки 8](#_Toc189494379)

[1.1 Общие требования к приложению 8](#_Toc189494380)

[1.2 Обоснование выбора среды разработки 9](#_Toc189494381)

[1.3 Обзор и выбор СУБД 10](#_Toc189494382)

[1.4 Описание области применения программы 12](#_Toc189494383)

[1.5 Требования к функциональным характеристикам 15](#_Toc189494384)

[1.6 Требования к составу и параметрам технических средств 17](#_Toc189494385)

[1.7 Требования к информационной и программной совместимости 17](#_Toc189494386)

[Глава 2. Практическая часть 19](#_Toc189494387)

[2.1 Проектирование и создание базы данных 19](#_Toc189494388)

[2.2 Разработка информационной системы 23](#_Toc189494389)

[2.3 Описание интерфейса программы 25](#_Toc189494390)

[2.4 Тестирование приложения 33](#_Toc189494391)

[ЗАКЛЮЧЕНИЕ 39](#_Toc189494392)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 40](#_Toc189494393)

**ВВЕДЕНИЕ**

Информационная система учета оборудования фармацевтической компании. Основная деятельность технического отдела заключается в сборе, хранении и обработке информации об оборудовании, его состоянии, а также в принятии и выполнении заявок на ремонт и обслуживание. Данный процесс крайне важен для фармацевтической компании, так как техническое состояние оборудования напрямую влияет на бесперебойность производственного процесса, соответствие оборудования санитарным нормам и требованиям, а также общую эффективность работы предприятия.

Роль технического отдела в компании. Технический отдел представляет собой структурное подразделение, отвечающее за техническое состояние оборудования, эксплуатацию инженерных систем, а также обеспечение бесперебойной работы производственных мощностей. Специалисты отдела занимаются не только мониторингом состояния оборудования, но и плановым и внеплановым обслуживанием, ремонтом, обновлением технических средств, что позволяет минимизировать простои в производственном процессе.

Важную роль в работе отдела играет сотрудник поддержки пользователей. Это специалист, который обеспечивает техническую поддержку и помогает клиентам решать возникающие проблемы, связанные с эксплуатацией оборудования, программным обеспечением, интернет-сервисами и прочими цифровыми инструментами. Его основные задачи включают консультирование, диагностику проблем, устранение неполадок, а также обучение пользователей эффективному использованию оборудования.

Помимо сотрудников технического отдела, в работе компании задействованы различные специалисты, каждый из которых выполняет определенные функции. Любой сотрудник предприятия является физическим лицом, заключившим трудовой договор с работодателем, что определяет его права, обязанности и зону ответственности. Дисциплинарные нормы также регулируют работу персонала, обеспечивая соблюдение стандартов безопасности и производственной этики.

Определение и учет оборудования.Оборудование, используемое на фармацевтическом производстве, представляет собой совокупность технических устройств, механизмов и систем, предназначенных для обработки, производства, транспортировки и упаковки медицинских препаратов и их компонентов. Данное оборудование может включать лабораторные установки, производственные линии, упаковочные аппараты, инженерные системы и другие технические средства, требующие регулярного обслуживания и контроля.

Для того чтобы обеспечить эффективное управление техническими средствами, необходимо организовать процесс учета оборудования. В информационной системе фиксируются следующие данные:

* **Инвентаризационный номер** – уникальный идентификатор каждого устройства;
* **Тип оборудования** – классификация в зависимости от функционального назначения;
* **Тип ремонта** – плановое обслуживание или устранение неисправностей;
* **Описание** – детальная информация о состоянии устройства и возможных проблемах;
* **Дата начала и окончания ремонта** – временные рамки выполнения работ;
* **Исполнитель** – сотрудник, ответственный за выполнение задачи.

Автоматизация заявок на ремонт и обслуживание. Важным элементом учета оборудования является процесс обработки заявок. Заявка – это официальный запрос на выполнение работ, связанный с ремонтом, техническим обслуживанием или заменой оборудования. В системе заявки регистрируются менеджерами технического отдела, которые анализируют поступающие запросы, назначают исполнителей и контролируют выполнение ремонтных работ.

Автоматизированная система учета оборудования обеспечивает удобную и эффективную работу с заявками, позволяя отслеживать их статус, выполнять фильтрацию по различным параметрам и формировать отчеты о проделанной работе. Внедрение такой системы позволяет снизить риск ошибок, сократить время обработки заявок и повысить прозрачность работы технического отдела.

Применение информационных технологий в системе учета. Современные информационные технологии позволяют автоматизировать процесс учета оборудования, обеспечивая удобный доступ к данным, их структурированное хранение и возможность оперативного редактирования. В информационной системе реализован интуитивно понятный интерфейс, который предоставляет следующие возможности:

* Работа с базой данных через удобные формы ввода и редактирования;
* Автоматизированное добавление, обновление и удаление информации;
* Контроль за изменениями в базе данных;
* Формирование отчетов и аналитических сводок.

Использование цифровых технологий в управлении техническими средствами компании позволяет не только упростить учет оборудования, но и значительно повысить уровень контроля за его состоянием, минимизировать ошибки, связанные с человеческим фактором, и повысить общую эффективность работы предприятия.

Актуальность проекта. Выбранная тема является актуальной, поскольку эффективный учет и контроль за состоянием оборудования играют ключевую роль в деятельности фармацевтических предприятий. Современные стандарты производства требуют строгого соблюдения норм эксплуатации, планового обслуживания и оперативного реагирования на неисправности. Отсутствие автоматизированной системы учета может привести к задержкам в ремонте, некорректному ведению документации и даже к нарушению требований безопасности, что может повлиять на качество выпускаемой продукции.

Разработка информационной системы учета оборудования позволит фармацевтической компании:

* Систематизировать процесс инвентаризации оборудования;
* Снизить вероятность ошибок в учете;
* Обеспечить своевременный ремонт и обслуживание оборудования;
* Повысить прозрачность работы технического отдела;
* Оптимизировать использование технических ресурсов.

Цель и задачи проекта. Основной целью проекта является разработка автоматизированной информационной системы учета оборудования для фармацевтической компании, обеспечивающей эффективное управление техническими средствами и их своевременное обслуживание.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Провести анализ доступных информационных систем учета оборудования, их функциональных возможностей и требований.
* Исследовать существующие технологии программирования и выбрать наиболее подходящие для реализации проекта.
* Разработать архитектуру базы данных, обеспечивающую удобное хранение и обработку информации.
* Создать пользовательский интерфейс, обеспечивающий интуитивное взаимодействие с системой.
* Провести тестирование системы для выявления и устранения возможных ошибок.
* Оценить возможности дальнейшего масштабирования и адаптации системы под нужды компании.

**Объект исследования** – процессы учета оборудования в фармацевтических компаниях, их организация и автоматизация.  
**Предмет исследования** – автоматизированная система учета оборудования, методы ее реализации и внедрения в производственный процесс.

Методы исследования. В рамках работы были использованы следующие методы:

* **Анализ** – изучение существующих решений в области автоматизированного учета;
* **Наблюдение** – исследование особенностей работы технического отдела фармацевтического предприятия;
* **Индукция** – выявление общих закономерностей на основе анализа частных случаев;
* **Моделирование** – разработка логической структуры базы данных и интерфейса информационной системы.

Структура курсового проекта. Курсовая работа включает в себя три основные части:

* **Введение** – обоснование актуальности темы, постановка цели и задач исследования.
* **Теоретическая часть** – анализ существующих систем учета, обзор языков программирования и систем управления базами данных, выбор технологий для реализации проекта.
* **Практическая часть** – описание разработки системы, проектирование базы данных, создание интерфейса, подключение к СУБД и тестирование функциональности.
* **Заключение** – подведение итогов, оценка достигнутых результатов и перспектив развития системы.

Разработка данной информационной системы обеспечит эффективное управление техническими ресурсами фармацевтической компании, что позволит оптимизировать процесс учета, повысить точность и оперативность обработки информации и улучшить контроль за состоянием оборудования.

**Глава 1. Общие понятия инструментов разработки**

* 1. **Общие требования к приложению**

Информационная автоматизированная система учета оборудования предназначена для эффективного управления данными о наличии, состоянии и перемещении оборудования в фармацевтической компании. Такая система должна обеспечивать автоматизацию учета, контроль исправности, планирование технического обслуживания и удобный доступ к информации для ответственных сотрудников.

В состав системы должны входить три основные части:

* Конструктивная (предусматривает разработку удобного пользовательского интерфейса, обеспечивающего взаимодействие с базой данных и другими модулями системы).
* Модельная (включает адекватную модель структуры хранения информации об оборудовании, его характеристиках, техническом состоянии и истории эксплуатации).
* Функциональная (обеспечивает автоматизацию бизнес-процессов, связанных с учетом оборудования, включая формирование отчетности, контроль сроков эксплуатации и технического обслуживания).

Современные автоматизированные системы учета оборудования появились в ответ на необходимость повышения эффективности управления активами предприятия. Развитие информационных технологий способствовало созданию компьютеризированных решений, позволяющих минимизировать человеческий фактор и повысить точность учета.

Информационная система учета оборудования должна предусматривать:

1. Регистрацию и ведение карточек оборудования с указанием технических характеристик.
2. Контроль состояния и технического обслуживания оборудования.
3. Автоматизацию формирования отчетов о наличии и использовании оборудования.
4. Возможность поиска, фильтрации и сортировки данных.
5. Управление доступом пользователей с разграничением прав.
   1. **Обоснование выбора среды разработки**

Система будет построена по клиент-серверной архитектуре, включающей базу данных и клиентское приложение. Выбор среды разработки основывается на следующих критериях:

* Производительность и масштабируемость.
* Удобство работы с базами данных.
* Поддержка объектно-ориентированного программирования.
* Развитая экосистема инструментов и библиотек.
* Простота развертывания и поддержки.

Популярными языками для разработки клиентских приложений являются Java, C++, C#, Visual Basic.Net и Python:

* Java – кроссплатформенный язык с мощной экосистемой, но требующий установки дополнительного программного обеспечения (JVM) и обладающий более сложным процессом интеграции с Windows-приложениями.
* C++ – высокопроизводительный язык, широко применяемый в системном программировании, но обладающий высокой сложностью управления памятью и сложным синтаксисом.
* C# – современный объектно-ориентированный язык, разработанный Microsoft, который обеспечивает удобную работу с базами данных, мощные средства для разработки пользовательских интерфейсов и простоту интеграции с Windows.
* Visual Basic .NET – удобен для быстрой разработки, но имеет менее мощные возможности по сравнению с C# и ограниченный выбор инструментов.
* Python – гибкий язык с широкой поддержкой различных библиотек, но уступающий C# в плане быстродействия и интеграции с Windows-приложениями.

Для реализации поставленной задачи выбраны C# и среда разработки Microsoft Visual Studio, так как они обеспечивают удобную интеграцию с SQL Server, развитые инструменты для создания пользовательского интерфейса и простоту развертывания.

* 1. **Обзор и выбор СУБД**

Современные технологии хранения данных предлагают множество подходов к организации информации, особенно при работе с большими объёмами данных. Эти подходы варьируются от иерархических структур до объектно-ориентированных решений, но наибольшей популярностью сегодня пользуются реляционные базы данных, основанные на принципах реляционной алгебры.

СУБД подразделяются на:

* Файл-серверные, которые обеспечивают доступ к базе данных через локальную сеть, при этом синхронизация чтения и обработки использует механизм файловых блокировок;
* Клиент-серверные, обеспечивающие монопольный доступ по сети и производящие централизованную обработку запросов клиентов.

Для хранения данных могут использоваться различные СУБД, включая SQL Server, MySQL, Oracle и PostgreSQL.

При выборе системы проводится сравнение нескольких распространённых СУБД, среди которых:

* Oracle Database от Oracle;
* SQL Server от Microsoft;
* MySQL от Oracle;
* PostgreSQL от сообщества Postgre.

Все перечисленные СУБД имеют бесплатные версии, что делает их доступными для пользователей. Результаты сравнения этих свободно распространённых версий представлены ниже (таблица 1).

Одним из главных преимуществ SQL Server является лёгкость в организации работы и удобный интерфейс. Ограничения бесплатной версии не слишком значительны, что позволяет использовать её полностью в микропредприятиях. В то же время, решения от Oracle отличаются высокой функциональностью, хотя могут возникнуть сложности с реализацией проектов.

Таблица 1.

Описание популярных свободно распространяемых СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание характеристики | Oracle  Database 23 ai | SQL Server  2019 Express | MySQL Community 8.0.40 | PostgreSQL 17.2 |
| Работа с операционными системами | Solaris (x86 systems, 64-bit), Windows x64 (64 bit), Linux, IBM AIX, HU-UX ia64, Linux of system Z (64 bit) | Windows, Virtual Machine in Azure, Red Hat Enterprise Linux 7.3, Ubuntu Linux 16.04, SUSE Linux Enterprise Server v12 SP2, Docker | Solaris (x86 systems, 64-bit), Windows х64, (64-bit), macOS Platforms, Linux, FreeBSD, Open BSD, NetBSD, Linux of system Z (64 bit) | Solaris (x86 systems, 64-bit), Windows х64, (64-bit), macOS Platforms, Linux, FreeBSD, Open BSD, NetBSD, Linux of system Z (64 bit) |
| Инструменты для работы с языками | ODBC, JDBC, SQLJ, OLE DB, ODP.NET, SQL/XML, XQuery и WebDAV | ODBC, JDBC, OLE DB, ODP.NET, SQL/XML, XQuery | C API (libmysqlclient),  Connector/C++,  Connector/J,  Connector/NET,  Connector/Node.js,  Connector/ODBC,  Connector/Python,  MySQL Native Driver for PHP | PL/sh, PL/Ruby, pl/R, plPHP, PL/Java |
| Процедурные языки | PL/SQL | Transact-SQL | Язык хранимых процедур MySQL | PL/sh, PL/Ruby, pl/R, plPHP, PL/Java |

На основе сравнительного анализа выбрана SQL Server 2019 Express, так как она обеспечивает:

* 1. Простоту развертывания и администрирования.
  2. Гибкость работы с реляционной моделью данных.
  3. Высокую производительность и безопасность.
  4. Поддержку сложных SQL-запросов и процедур.
  5. **Описание области применения программы**

Традиционно фармацевтические компании используют разнообразные методы и технологии учёта оборудования, позволяющие систематизировать и контролировать процессы эксплуатации, технического обслуживания и модернизации. Эти методы можно классифицировать по различным критериям, что помогает оптимизировать работу с техническими средствами и обеспечивать их эффективное использование.

Классификация технологий учёта оборудования

**По назначению оборудования** учёт ведётся в зависимости от типа используемого оборудования:

* **Производственное оборудование** — включает в себя машины и аппараты, задействованные в процессе производства лекарственных препаратов.
* **Лабораторное оборудование** — охватывает устройства, используемые в исследовательских и контрольных лабораториях.
* **Складское оборудование** — применяется для хранения, транспортировки и учёта фармацевтических товаров.
* **Офисное оборудование** — включает технические средства, используемые для административных задач.

**По видам учёта** разные процессы требуют различных методов регистрации и контроля:

* **Инвентаризация** — проверка наличия и состояния оборудования в компании.
* **Техническое обслуживание** — контроль выполнения профилактических и ремонтных работ.
* **Списание** — учет оборудования, выведенного из эксплуатации.
* **Модернизация** — фиксация всех улучшений и модификаций технических средств.

**По временным характеристикам** системы учета могут различаться по частоте обновления информации:

* **Периодический учет** — выполняется через определённые интервалы времени.
* **Ежемесячный учет** — обновление информации раз в месяц.
* **Ежегодный учет** — глобальная проверка оборудования на основе годового анализа.
* **Внеплановый учет** — проводится по необходимости, например, при поломке или потере оборудования.

Основные виды учета оборудования в фармацевтических компаниях

Современные фармацевтические предприятия применяют различные методы учёта, чтобы обеспечить максимальную точность данных и прозрачность процессов эксплуатации оборудования. К числу наиболее распространённых относятся:

* **Инвентаризация**  
  Данный метод позволяет проверять наличие, техническое состояние и работоспособность оборудования. Он помогает выявлять расхождения между фактическими данными и учётными записями, а также предотвращать потери и несвоевременное обслуживание техники.
* **Регистрация и учёт технического обслуживания**  
  Включает в себя плановые проверки, профилактическое обслуживание, ремонтные работы и внеплановые ремонты. Этот метод помогает предотвратить неожиданные поломки, сократить время простоя оборудования и обеспечить его долговечность.
* **Анализ эксплуатационных характеристик** - Используется для мониторинга параметров работы оборудования, оценки его загруженности, учёта сроков службы и выявления потенциальных проблем. Позволяет оптимизировать использование ресурсов и своевременно планировать обновление оборудования.
* **Документирование перемещений и модификаций** - Включает в себя фиксацию передачи оборудования между подразделениями, проведение модернизации, списание устаревшей техники и ввод в эксплуатацию новых единиц. Это помогает отслеживать полную историю эксплуатации каждого устройства.
* **Методы самоконтроля** - Предполагают, что ответственные сотрудники периодически проверяют состояние закреплённого за ними оборудования, фиксируют выявленные неисправности и принимают меры по их устранению.

Необходимость автоматизированной системы учёта оборудования

Фармацевтические компании владеют большим количеством технологического оборудования, которое требует строгого контроля, своевременного технического обслуживания и точного планирования замены. Ведение учёта вручную приводит к высоким временным затратам, возможным ошибкам и снижению эффективности управления техническими ресурсами.

Автоматизированная система учёта оборудования поможет решить эти проблемы, обеспечивая:

* **Формирование единого реестра оборудования**  
  Система будет содержать полные технические характеристики всех устройств, включая их серийные номера, дату приобретения, срок службы и историю ремонта.
* **Контроль местоположения и технического состояния**  
  Все перемещения оборудования будут фиксироваться, а актуальное состояние техники станет доступным в режиме реального времени.
* **Автоматизированное управление графиками обслуживания**  
  Запланированные работы по техническому обслуживанию и ремонту будут контролироваться системой, что позволит избежать пропусков и несвоевременного проведения профилактических мероприятий.
* **Генерацию отчётности и прогнозирование закупок**  
  Система позволит анализировать эксплуатационные данные, составлять отчёты о текущем состоянии оборудования и прогнозировать потребность в новых закупках.

Преимущества внедрения автоматизированной системы

Использование современной системы управления учётом оборудования значительно повысит прозрачность всех процессов, связанных с эксплуатацией технических средств. Это приведёт к снижению ошибок, уменьшению затрат на обслуживание, оптимизации рабочих процессов и повышению общей эффективности работы фармацевтической компании.

* 1. **Требования к функциональным характеристикам**

Программный модуль, разрабатываемый для данной системы, предназначен для существенного упрощения и автоматизации рабочих процессов сотрудников отдела технической поддержки, занимающихся учётом и обслуживанием оборудования.

Его основная цель — минимизировать ручной труд, снизить вероятность ошибок при обработке данных, ускорить процесс регистрации и отслеживания информации о технических средствах, а также повысить общую эффективность взаимодействия между сотрудниками. Благодаря внедрению данного модуля, управление оборудованием станет более структурированным, прозрачным и удобным, что положительно скажется на рабочем процессе и позволит снизить нагрузку на специалистов.

Основные функциональные компоненты модуля:

* **Централизованная база данных оборудования**  
  Одной из ключевых функций программного модуля является предоставление удобного механизма для хранения и управления всей информацией, связанной с оборудованием. В базе данных будут храниться сведения о каждом устройстве, включая его наименование, серийный номер, модель, производителя, текущее местонахождение, дату приобретения, дату последнего ремонта, текущее состояние и другие важные характеристики. Система обеспечит автоматическое обновление данных в режиме реального времени, что позволит сотрудникам быстро находить необходимую информацию и оперативно реагировать на любые изменения. Кроме того, доступ к базе данных будет реализован с учётом разграничения прав пользователей, что обеспечит защиту конфиденциальной информации и предотвратит несанкционированные изменения.
* **Система управления заявками на ремонт**  
  Ещё одной важной частью программного модуля является автоматизация процесса подачи, обработки и отслеживания заявок на ремонт оборудования. Сотрудники смогут легко регистрировать новые заявки, указывая причину неисправности, предполагаемую проблему и приоритет выполнения работ. После создания заявки система автоматически зафиксирует её в базе данных, присвоит статус и уведомит ответственных специалистов. В дальнейшем информация о ходе выполнения заявки будет обновляться в автоматическом режиме, позволяя пользователям в любое время проверить статус ремонта. Данный подход обеспечит прозрачность работы отдела технической поддержки, уменьшит вероятность потери или игнорирования заявок и сократит время ожидания выполнения ремонтных работ.
* **Интуитивно понятный и удобный пользовательский интерфейс**  
  Для успешного внедрения программного модуля особое внимание будет уделено созданию удобного, логичного и понятного интерфейса, который позволит сотрудникам быстро освоить новую систему без необходимости прохождения длительного обучения. Пользовательский интерфейс будет разработан с учётом современных стандартов удобства использования, что обеспечит лёгкость навигации, минимизацию количества действий, необходимых для выполнения задач, и интуитивное восприятие всех элементов управления. Гибкие настройки интерфейса позволят адаптировать систему под потребности конкретных пользователей, предоставляя возможность настраивать отображаемые данные, изменять приоритеты уведомлений и быстро переходить к часто используемым функциям.

Ожидаемые результаты и преимущества внедрения модуля:

Разработка и внедрение данного программного модуля позволит значительно повысить уровень автоматизации рабочих процессов, связанных с учётом и обслуживанием оборудования, а также улучшить взаимодействие сотрудников отдела технической поддержки.

Автоматизированный учёт технических средств снизит вероятность ошибок, ускорит поиск необходимой информации и повысит точность регистрации данных. Благодаря системе управления заявками на ремонт будет сокращено время обработки заявок, а также исключены задержки, связанные с человеческим фактором. Интуитивно понятный интерфейс позволит сотрудникам без проблем адаптироваться к новой системе, что уменьшит затраты времени на обучение и повысит уровень удовлетворённости пользователей.

В конечном итоге, внедрение данного модуля обеспечит оптимизацию рабочих процессов, повысит эффективность работы отдела технической поддержки и окажет положительное влияние на производительность всей компании.

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

Примерные характеристики компьютера для стабильной работы приложения:

Таблица 2.

Примерные характеристики компьютера

|  |  |
| --- | --- |
| Процессор | Intel(R) Core(TM) i5-9400 |
| Память | 8 Гбайт |
| Разрешение экрана | 1920x1080 |
| Устройства ввода | Клавиатура, мышь |
| Дополнительно | Постоянное подключение к сети, в которой находится база данных |

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

Для правильной работы программы требуется следующее:

* SQL Server Management Studio (SSMS) с установленной базой данных — это бесплатное средство управления базами данных, включающее инструменты для администрирования Microsoft SQL Server.
* Windows 10 — это операционная система для ПК и рабочих станций, разработанная компанией Microsoft в рамках линейки Windows NT.
* Microsoft Excel 2019 — известный и широко используемый продукт от Microsoft для создания и анализа электронных таблиц.
* Microsoft Visual Studio 2022 — это мощная интегрированная среда разработки (IDE) для создания многофункциональных и визуально привлекательных кроссплатформенных приложений, поддерживающих Windows, Mac, Linux, iOS и Android. Разработка клиентских приложений может осуществляться с использованием таких технологий, как WinForms, WPF, WinUI, MAUI или Xamarin.
* MS SQL Server — реляционная система управления базами данных (РСУБД), разработанная Microsoft. Основной язык запросов, используемый в системе, — Transact-SQL, который был создан совместно с Sybase. Transact-SQL представляет собой реализацию стандарта ANSI/ISO SQL с расширениями и применяется для работы с базами данных различного размера, от небольших до крупных.

**Глава 2. Практическая часть**

**2.1 Проектирование и создание базы данных**

При выполнении курсового проекта были разработана концептуальная модель данных в нотации Crow’s Foot.

Учитывая постановку задачи следует хранить следующие данные:

* Информацию о оборудовании, находящимся на производстве
* Информацию о должностях, на которые устроены пользователи
* Информацию о пользователях, которые пользуются программой
* Информацию о действиях пользователей, которые делают пользователи
* Информацию о заявках на ремонт, которые делают пользователи

Оборудование как основной субъект учёта оборудования характеризуется: названием, серийным номером, моделью, производителем, датой покупки, состоянием.

Заявки характеризуются: оборудованием, датой заявки, создателем, статусом, назначенным, описанием.

Пользователи характеризуются: Именем пользователя, паролем, почтой, ролью.

Таким образом концептуальная модель данных с первичными и внешними ключами имеет вид (Рисунок 1).

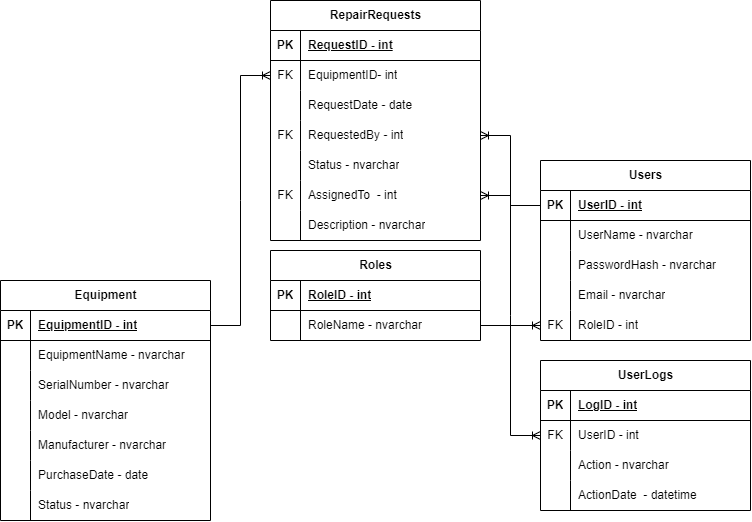


Рисунок 1 – Концептуальная модель данных в виде диаграммы в нотации Crow’s Foot.

На основе концептуальной модели данных была сформирована структура сущностей и их взаимосвязей.

Таблицы 3 – 8 представляют характеристики структуры данных.

Таблица 3

Структура данных Roles (Таблица ролей)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя атрибута | Наименование | Тип поля | Ключи, разрешения |
| roleID | Номер роли в приложении (уникальный) | int, autoincrement | PK, NOT NULL |
| roleName | Наименование роли | nvarchar(50) | NOT NULL |
|  |  |  |  |

Таблица 4

Структура данных Equipment (Таблица оборудования)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя атрибута | Наименование | Тип поля | Ключи, разрешения |
| EquipmentID | Номер оборудования в приложении (уникальный) | int, autoincrement | PK, NOT NULL |
| EquipmentName | Наименование оборудования | nvarchar(100) | NOT NULL |
| SerialNumber | Серийный номер оборудования | nvarchar(50) | NOT NULL |
| Model | Модель оборудования | nvarchar(100) | NOT NULL |
| Manufacturer | Производитель | nvarchar(100) | NOT NULL |
| PurchaseDate | Дата покупки | date | NOT NULL |
| Status | Состояние | nvarchar(50) | NOT NULL |
|  |  |  |  |

Таблица 5

Структура данных RepairRequests (Таблица заявок)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя атрибута | Наименование | Тип поля | Ключи, разрешения |
| RequestID | Номер заявки в приложении (уникальный) | int, autoincrement | PK, NOT NULL |
| EquipmentID | Номер оборудования в приложении | int | FK, NOT NULL |
| RequestDate | Дата заявки | date | NOT NULL |
| RequestedBy | Создатель | int | FK, NOT NULL |
| Status | Статус | nvarchar(50) | NOT NULL |
| AssignedTo | Назначенный на заявку | int | FK, NOT NULL |
| Description | Описание | nvarchar(500) | NOT NULL |

Таблица 6

Структура данных Users (Таблица пользователей)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя атрибута | Наименование | Тип поля | Ключи, разрешения |
| UserID | Номер пользователя в приложении (уникальный) | int, autoincrement | PK, NOT NULL |
| UserName | Имя пользователя | nvarchar(100) | NOT NULL |
| PasswordHash | Пароль | nvarchar(255) | NOT NULL |
| Email | Почта | nvarchar(100) | NOT NULL |
| RoleID | Номер роли | int | FK, NOT NULL |

Таблица 7

Структура данных UserLogs (Таблица логов)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя атрибута | Наименование | Тип поля | Ключи, разрешения |
| LogID | Номер лога в приложении (уникальный) | int, autoincrement | PK, NOT NULL |
| UserID | Номер пользователя в приложении | nvarchar(50) | FK, NOT NULL |
| Action | Действие | nvarchar(255) | NOT NULL |
| ActionDate | Дата действия | datetime | NOT NULL |

На рисунке 2 представлена схема данных, разработанная на основе физической структуры в Microsoft SQL Server 2019.

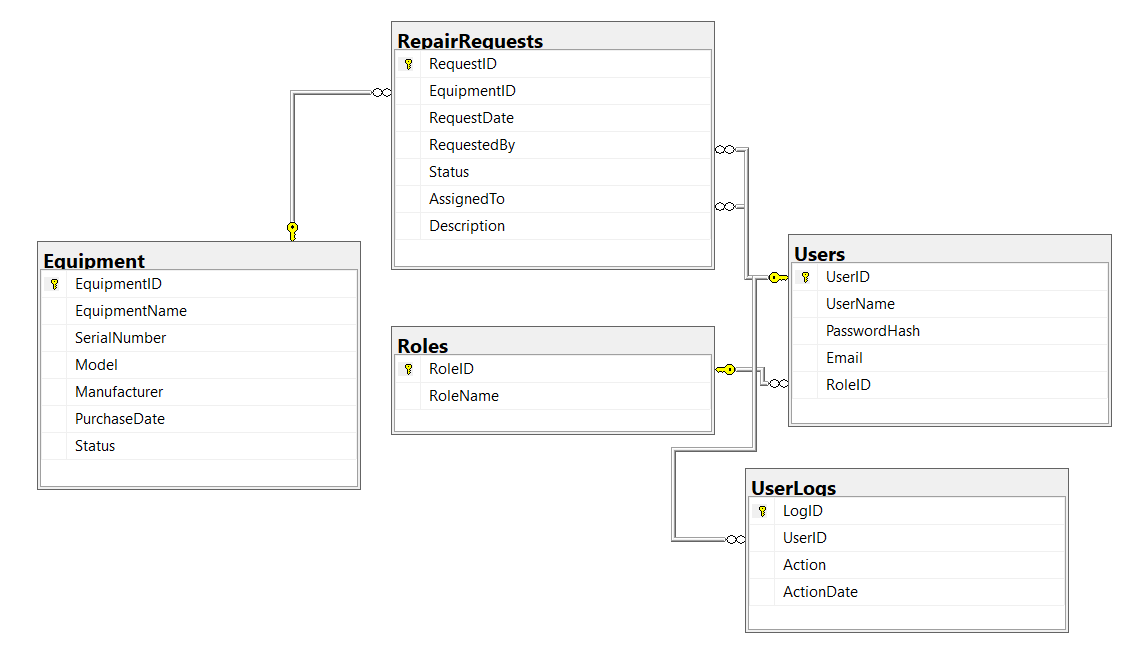


Рисунок 2 – Схема данных в Microsoft SQL Server 2019.

**2.2 Разработка информационной системы**

Таблица 8

Таблица Модулей приложения

| Наименование модуля | Функции модуля |
| --- | --- |
| LoginPage.xaml.cs | Модуль обработки операции по проведению авторизации пользователя. Позволяет перейти на модуль регистрации пользователя. |
| RegistrationWindow.xaml.cs | Модуль обработки операции регистрации пользователя. Позволяет пользователю создать новый аккаунт, введя необходимые данные, и сохраняет информацию в базе данных. |
| MenuPage.xaml.cs | Модуль управления меню пользователя. Предоставляет доступ к основным разделам приложения, позволяя перемещаться между различными модулями и функциональными страницами. |
| EquipmentPage.xaml.cs | Модуль управления оборудованием. Обеспечивает отображение данных об оборудовании, удаление, переход на добавление, редактирование, а также взаимодействие с базой данных для работы с информацией о сущностях оборудования. |
| RepairRequestPage.xaml.cs | Модуль управления заявками на ремонт. Обеспечивает отображение заявок на ремонт, удаление, переход на создание, редактирование, а также взаимодействие с базой данных для обработки информации о заявках и их статусах. |
| LogsPage.xaml.cs | Модуль управления журналом событий. Обеспечивает отображение записей журнала, содержащих информацию о действиях пользователей и системе. |
| EditEquipmentWindow.xaml.cs | Модуль редактирования информации об оборудовании. Обеспечивает добавление, изменение и сохранение данных об оборудовании, а также взаимодействие с базой данных для обновления информации. |
| EditRepairRequestWindow.xaml.cs | Модуль редактирования заявки на ремонт. Обеспечивает добавление, изменение и сохранение данных о заявке на ремонт, а также взаимодействие с базой данных для обновления информации о статусе и деталях заявки. |
| ReportsWindow.xaml.cs | Модуль генерации отчетов. Обеспечивает создание, отображение и экспорт отчетов по оборудованию, заявкам на ремонт, с возможностью взаимодействия с базой данных для формирования и анализа отчетной информации. |

Диаграмма взаимодействия элементов для приложения представлена на Рисунке 3.

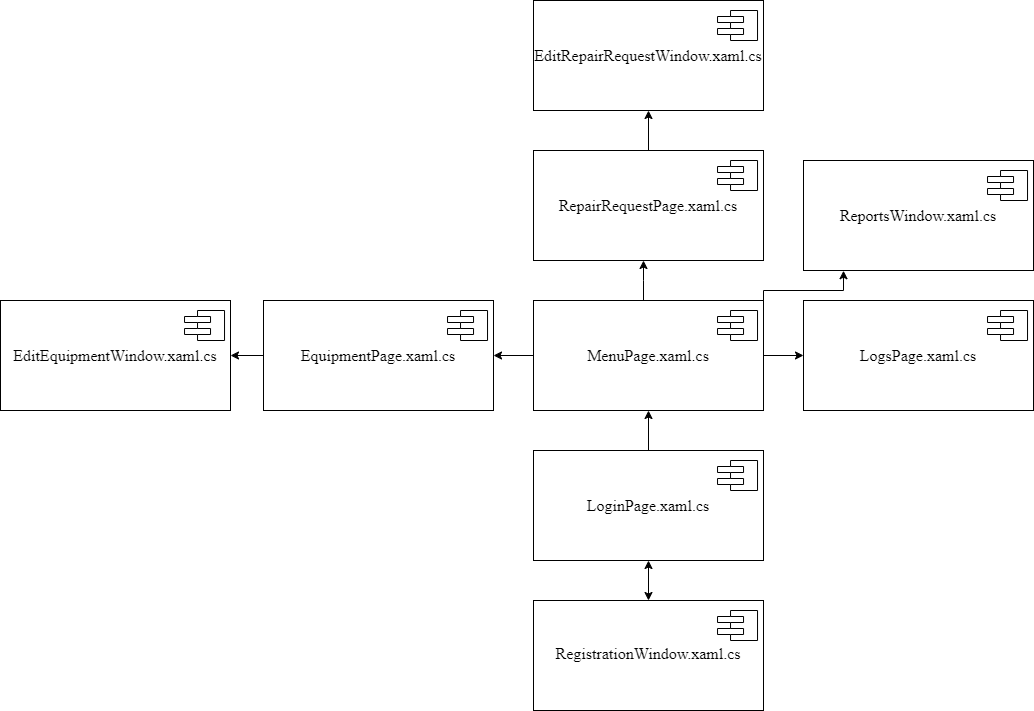


Рисунок 3. – Диаграмма взаимодействия программных модулей приложения.

В ходе разработки информационной системы был спроектирован и реализован ряд модулей, обеспечивающих основные функциональные возможности приложения. Каждый модуль выполняет свою задачу, позволяя пользователям взаимодействовать с системой, управлять данными и автоматизировать процессы учёта оборудования и заявок на ремонт.

Разработанные модули включают в себя авторизацию и регистрацию пользователей, управление меню и переходами между разделами, работу с данными об оборудовании и заявках на ремонт, а также генерацию отчетов. Взаимодействие модулей реализовано таким образом, чтобы обеспечить удобную и интуитивно понятную навигацию, а также эффективную обработку данных в базе.

Использование структуры отдельных модулей позволило достичь гибкости и масштабируемости системы. В будущем это обеспечит возможность расширения функционала, интеграции с другими сервисами и улучшения пользовательского опыта. Диаграмма взаимодействия программных модулей (Рисунок 3). наглядно демонстрирует связи между компонентами приложения и их роль в общей системе.

Таким образом, созданная информационная система соответствует требованиям к автоматизированному учету оборудования, обеспечивая надежность, удобство и эффективность работы пользователей.

**2.3 Описание интерфейса программы**

Проектирование интерфейса связано с выделением групп пользователей, которые получают доступ к различным данным и набору возможностей в системе. Это означает, что необходимо тщательно продумать, каким образом каждая категория пользователей будет взаимодействовать с системой. Поэтому технология входа в систему предусматривает разделение на несколько групп пользователей, а именно: администратор, менеджер, техник и обычный пользователь. Каждая из этих групп обладает своим собственным уровнем прав в системе, что накладывает ограничения на доступ к различным функциональным возможностям. Авторизация осуществляется исключительно после ввода пользователем своих учетных данных.

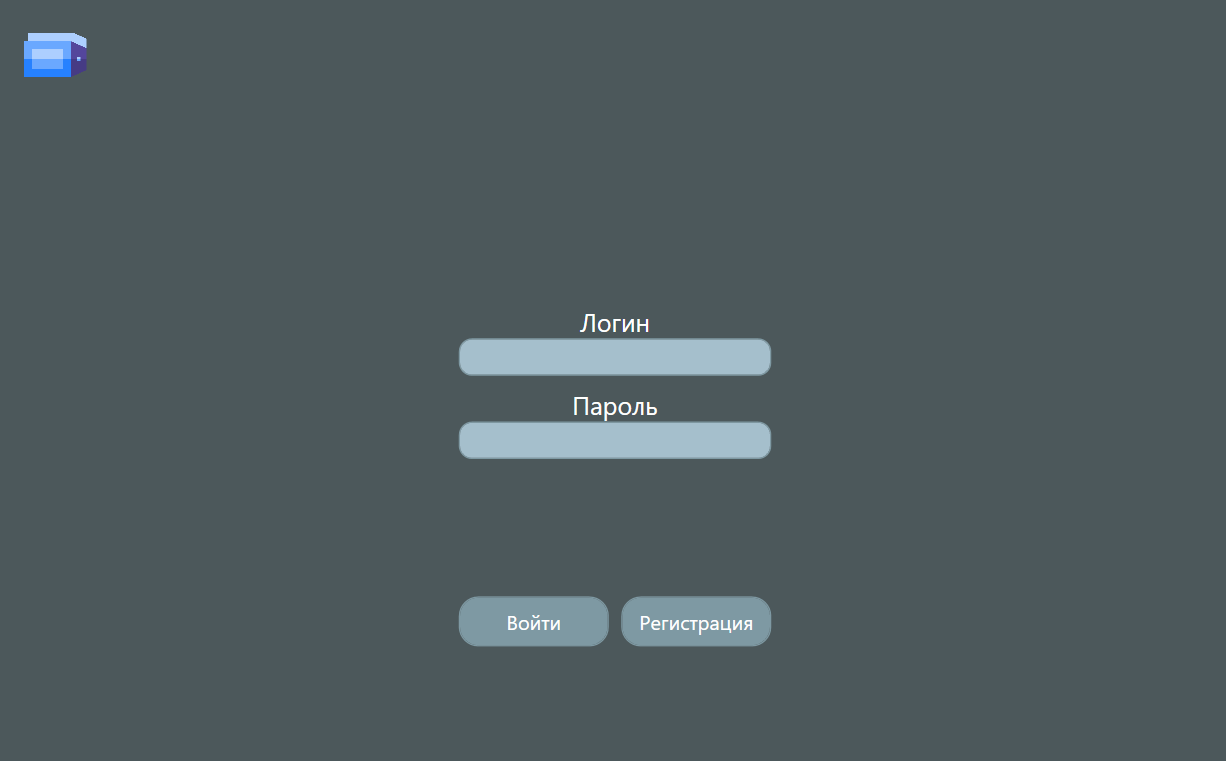
Работа с системой начинается для каждого пользователя со страницы авторизации (Рисунок 4). В случае успешного ввода корректных учетных данных пользователя система автоматически перенаправляет его на страницу главного меню (Рисунок 5). Если же пользователь не имеет учетной записи в системе, он может воспользоваться возможностью регистрации, нажав соответствующую кнопку "Регистрация". После этого система перенаправит его на страницу регистрации (Рисунок 6), где будет предложено ввести необходимые данные для создания нового аккаунта 

Рисунок 4. –Макет страницы Авторизации в системе.

На странице "Меню" пользователь в зависимости от своей должности получает возможность перемещаться по различным разделам системы. Это могут быть страницы учета оборудования, обработки заявок, создания отчетов, а также просмотра журнала действий.

В случае необходимости пользователь может завершить текущий сеанс работы с системой, воспользовавшись кнопкой "Выход", которая перенаправляет его обратно на страницу авторизации.

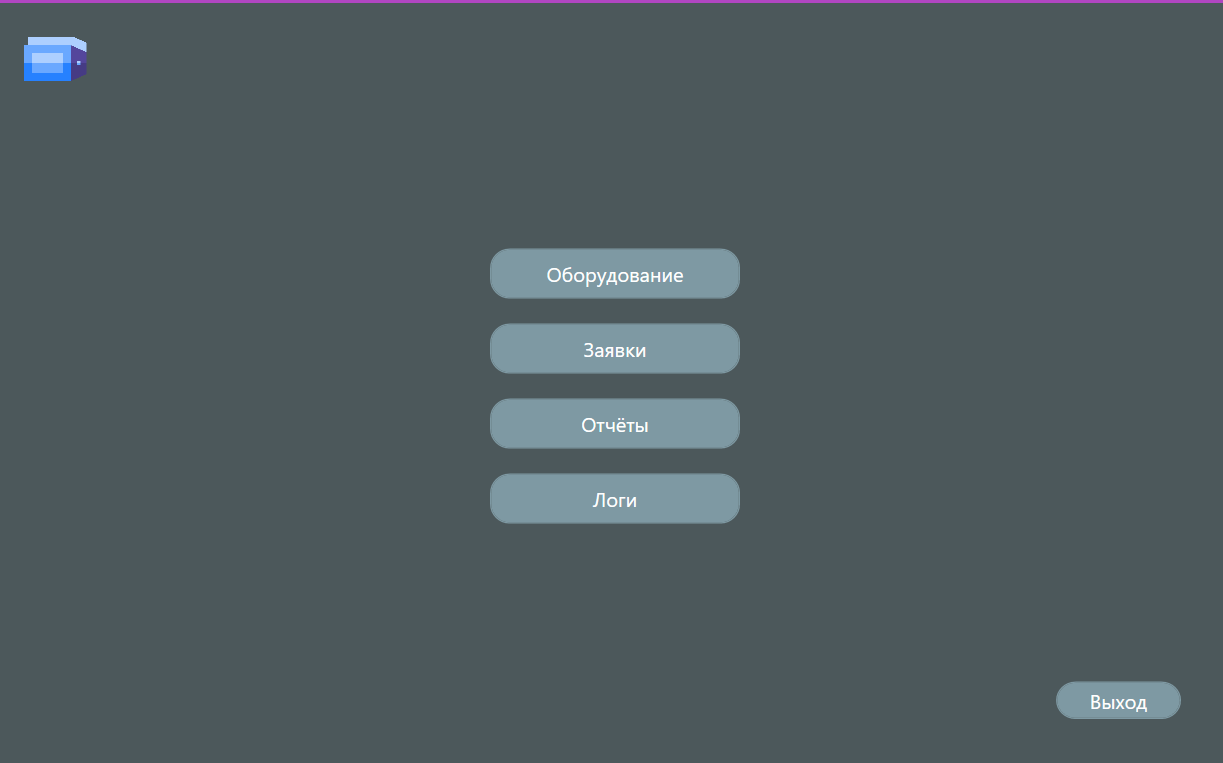


Рисунок 5. – Макет страницы Меню в приложении.

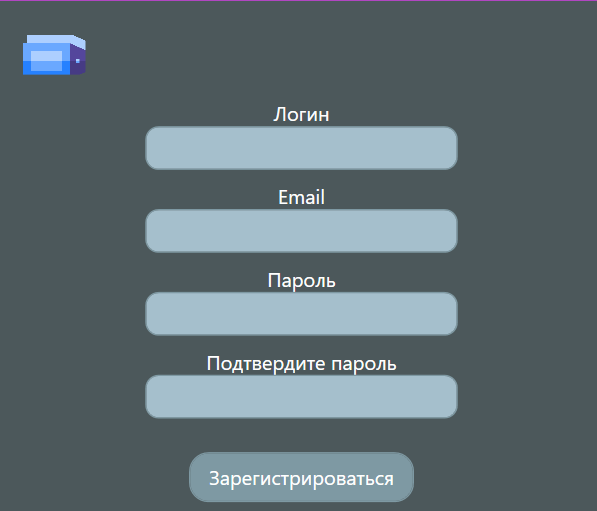


Рисунок 6. –Макет окна Регистрации в системе.

Управление информацией об оборудовании осуществляется через страницу учета оборудования (Рисунок 7). На этой странице, в зависимости от должностных обязанностей, пользователи могут выполнять просмотр записей об оборудовании, осуществлять фильтрацию данных и при необходимости изменять статус оборудования на "Неисправное" с помощью функционала редактирования (Рисунок 8).

Менеджеры и техники обладают расширенными правами, что позволяет им не только редактировать существующие записи, но и добавлять новые или удалять устаревшие данные. Для выполнения редактирования необходимо выбрать соответствующее оборудование в списке и нажать кнопку "Редактировать".

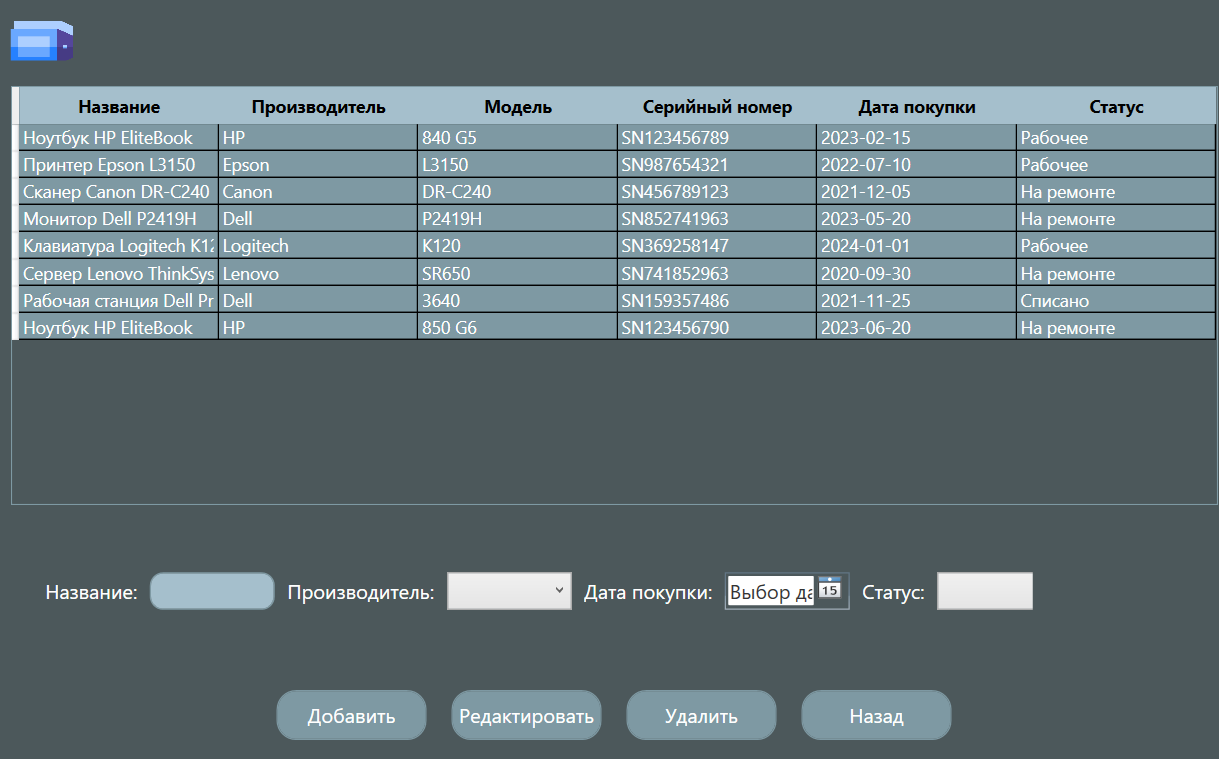


Рисунок 7. – Макет страницы Учёта оборудования.

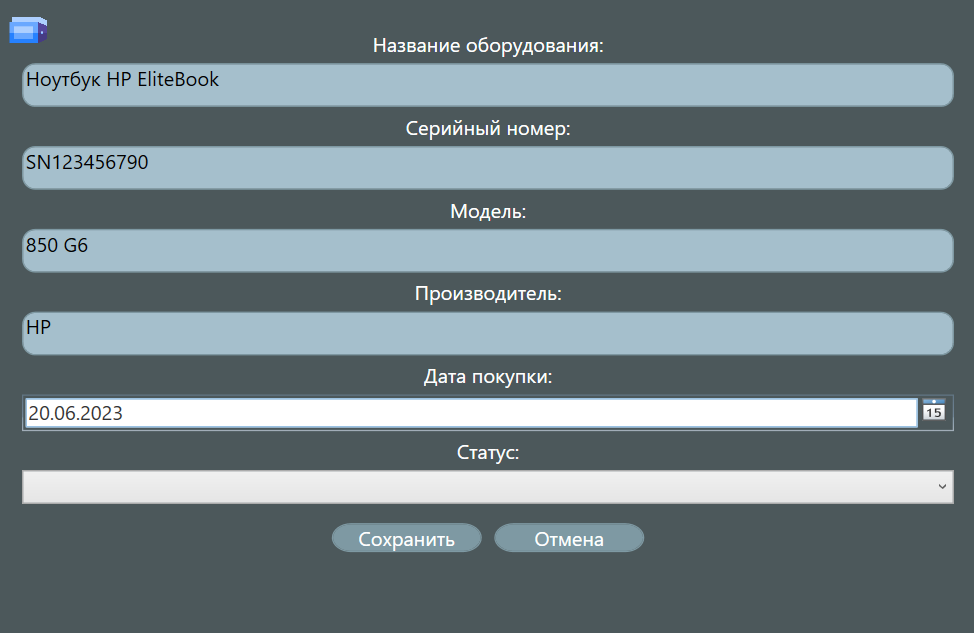


Рисунок 8. – Макет окна Редактирования оборудования.

В окне редактирования оборудования предоставляется возможность как добавления новых записей, так и внесения изменений в существующие. Важно учитывать, что для успешного сохранения данных необходимо корректно заполнить все требуемые поля и затем нажать кнопку "Сохранить". Обычные пользователи не имеют доступа к большинству полей и могут изменять лишь статус оборудования. Для создания новой заявки на ремонт пользователю предлагается перейти в раздел "Создание заявки", который доступен из главного меню посредством кнопки "Заявки". В этом разделе необходимо выбрать оборудование, находящееся в состоянии "Неисправно", и указать описание проблемы (Рисунок 9).

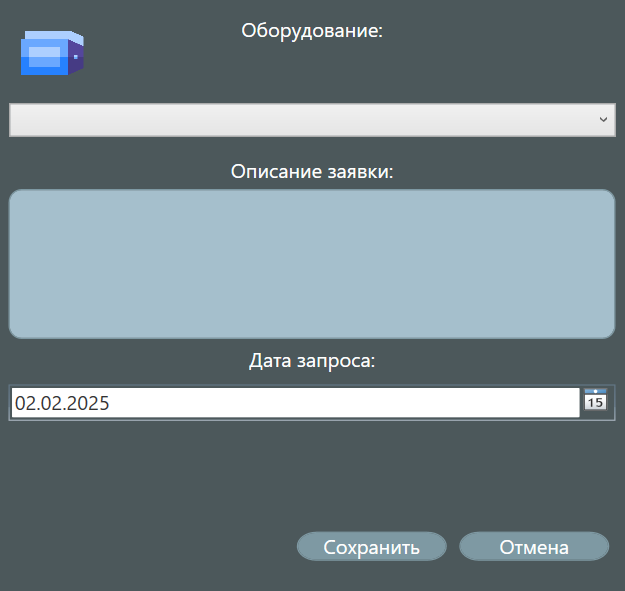


Рисунок 9. – Макет окна Создания заявки.

Если кнопку "Заявки" нажимает менеджер или техник, система перенаправляет его на страницу учета заявок (Рисунок 10). Здесь предоставляется возможность просмотра всех заявок, выполнения их фильтрации и редактирования данных с помощью окна "Редактирования заявки". Кроме того, менеджеры могут назначать техников на заявки, находящиеся в статусе "Ожидает", что автоматически приводит к изменению состояния оборудования на "На ремонте".

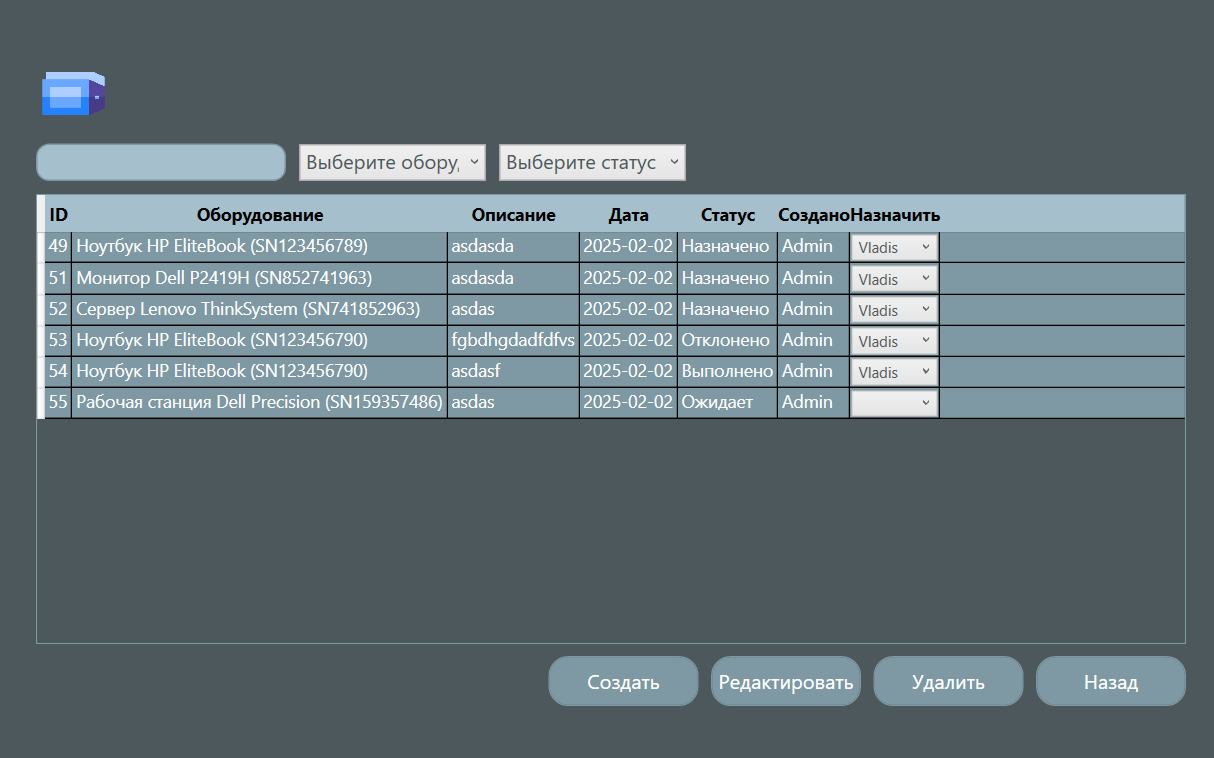


Рисунок 10. – Макет страницы Учёта заявок.

Для формирования отчетов пользователь может воспользоваться кнопкой "Отчеты" в главном меню, после чего откроется окно "Формирование отчетов". В этом разделе можно выбрать конкретное оборудование, задать временной диапазон и сгенерировать отчет.

Отчет содержит сведения о количестве созданных заявок за указанный период, информации о создателе каждой заявки, дате создания и текущем статусе. Сформированный отчет автоматически сохраняется на рабочий стол с именем соответствующего оборудования (Рисунок 11).

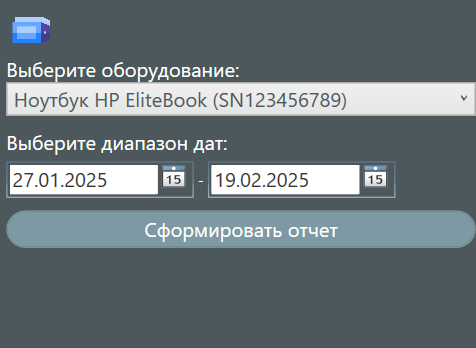


Рисунок 11. – Макет окна Формирования отчётов.

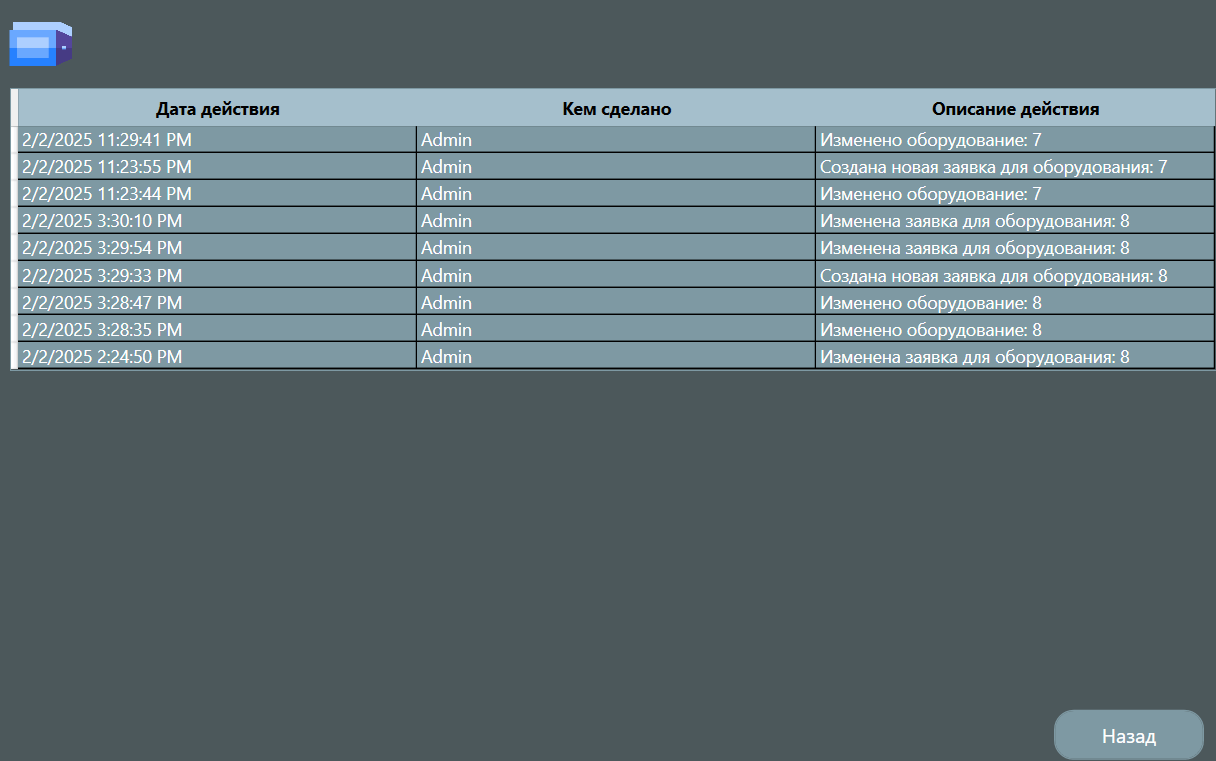
Для администраторов предусмотрена возможность просмотра журнала действий пользователей (Рисунок 12). Данный раздел доступен через кнопку "Логи" в главном меню и позволяет администратору анализировать, какие действия были совершены, кем и в какое время.

Рисунок 12. – Макет страницы Журнала действий.

Разработка интерфейса информационной системы сыграла ключевую роль в обеспечении удобства взаимодействия пользователей с программой. При проектировании учитывались особенности различных категорий пользователей, а также их потребности в доступе к информации и управлении данными. Реализованный подход позволяет разграничить уровни доступа, обеспечивая безопасность данных и удобство работы.

Была разработана понятная система навигации, позволяющая пользователям легко ориентироваться в возможностях приложения. Каждый модуль выполняет строго определенные функции, что способствует логичности и целостности системы. Кроме того, интуитивно понятный дизайн интерфейса минимизирует потребность в дополнительном обучении, что значительно сокращает время на освоение системы.

В результате разработанный интерфейс позволяет эффективно решать задачи управления оборудованием и заявками, обеспечивая пользователей всеми необходимыми инструментами для работы. Автоматизация процесса формирования отчетов и ведения журнала действий значительно повышает прозрачность работы системы и упрощает контроль над операциями. Таким образом, данная информационная система становится надежным и удобным инструментом, способствующим улучшению учета оборудования, повышению эффективности работы сотрудников и оптимизации бизнес-процессов организации.

**2.4 Тестирование приложения**

Для помощи пользователю в работе с системой и для обеспечения корректной работы и безопасности системы реализованы некоторые ограничения, не позволяющие пользователю выполнять действия, которые способны вызвать ошибки.

На странице авторизации при вводе пользователем в поля данных авторизации не существующих в базе данных пользователю будет выведена ошибка авторизации (Рисунок 13).

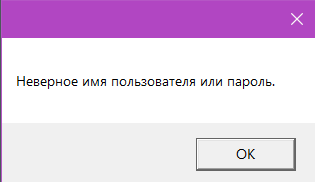


Рисунок 13. – Ошибка авторизации.

На странице регистрации пользователя при не заполнении всех полей будет выведено предупреждение (Рисунок 14).

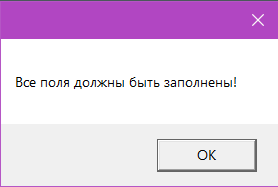


Рисунок 14. – Ошибка при регистрации пустые поля.

На странице регистрации пользователя при вводе пароля меньше 6 символов, одну заглавную букву и одну цифру пользователю будет выведено предупреждение (Рисунок 15).

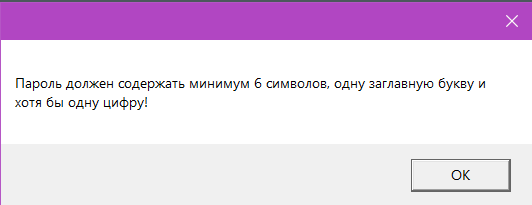


Рисунок 15. – Ошибка при регистрации, неправильное заполнение пароля.

На странице регистрации пользователя при вводе пароля меньше 6 символов, одну заглавную букву и одну цифру пользователю будет выведено предупреждение (Рисунок 16).

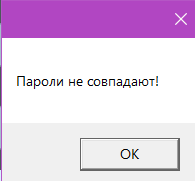


Рисунок 16. – Ошибка при регистрации, пароли не совпадают.

На странице учёта оборудования если пользователь не выберет оборудование из списка будет выведена ошибка (Рисунок 17).

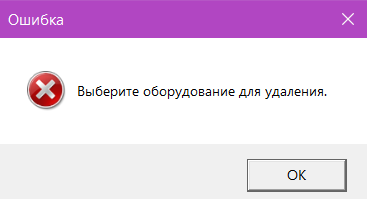


Рисунок 17. – Ошибка при удалении оборудования.

На странице учёта оборудования при удалении оборудование из списка будет выведено предупреждение (Рисунок 18).

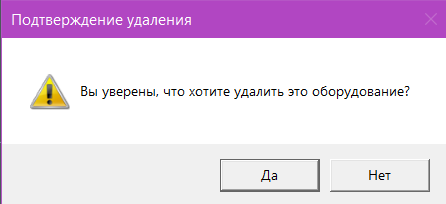


Рисунок 18. – Предупреждение при удалении оборудования.

На странице учёта оборудования если пользователь не выберет оборудование для редактирования из списка будет выведена ошибка (Рисунок 19).

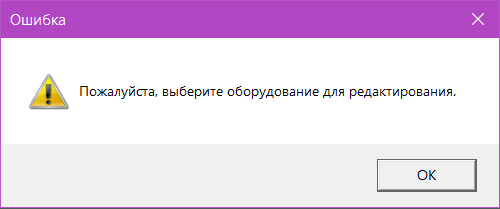


Рисунок 19. – Ошибка при редактировании оборудования.

На окне добавления/редактирования оборудования если пользователь не заполнит все поля для добавления оборудования будет выведена ошибка (Рисунок 20).

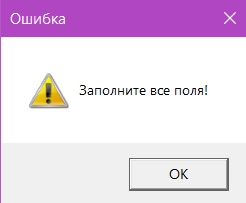


Рисунок 20. – Ошибка при добавлении оборудования.

На странице учёта заявок если пользователь не выберет заявку из списка будет выведена ошибка (Рисунок 21).

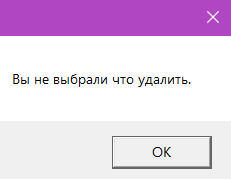


Рисунок 21. – Ошибка при удалении заявки.

На странице учёта заявок при удалении заявки пользователю будет выведено предупреждение (Рисунок 22).

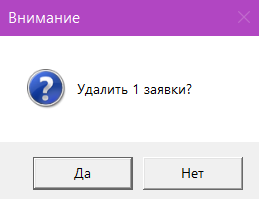


Рисунок 22. – Предупреждение при удалении заявки.

На странице учёта заявок если пользователь не выберет заявку из списка, пользователю будет выведено предупреждение (Рисунок 23).

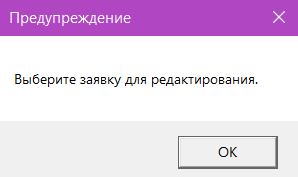


Рисунок 23. – Предупреждение при редактировании заявки.

На окне добавления/редактирования заявки если пользователь не заполнит все поля для добавления заявки будет выведена ошибка (Рисунок 24).

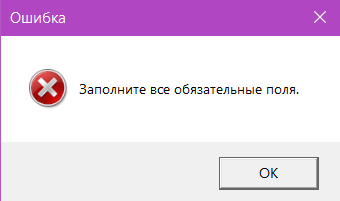


Рисунок 24. – Ошибка при добавлении заявки.

На окне формировании отчёта если пользователь при указании даты укажет, что дата начала не может быть позже даты окончания будет выведена ошибка (Рисунок 25).

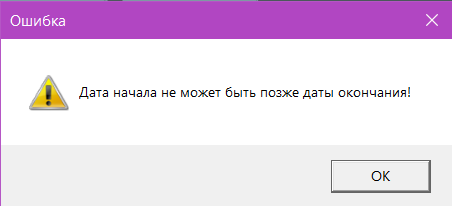


Рисунок 25. – Ошибка при формировании отчёта.

На окне формировании отчёта если у оборудования не будет заявок за указанный период будет выведена информация (Рисунок 26).

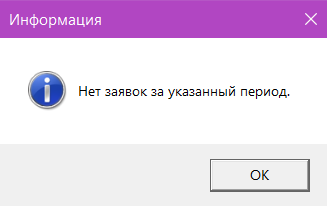


Рисунок 26. – Информация при формировании отчёта.

На окне формировании отчёта если отчёт будет удачно сформирован будет выведена информация о сохранении (Рисунок 27).

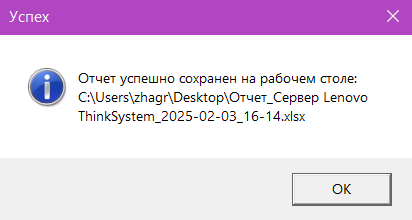


Рисунок 27. – Информация при удачном формировании отчёта.

**ЗАКЛЮЧЕНИЕ**

В ходе работы была разработана информационная система для учета оборудования и управления заявками на ремонт. В рамках исследования были рассмотрены вопросы организации учета технических средств, их инвентаризации, мониторинга состояния и процесса обработки заявок на техническое обслуживание.

В процессе разработки системы были определены ключевые требования к функционалу, проведен анализ существующих подходов и выбраны инструменты реализации. В качестве среды разработки клиентского приложения использовалась платформа WPF на языке C#, а в качестве системы управления базами данных применен SQL Server 2019.

Создана реляционная база данных, обеспечивающая хранение информации об оборудовании, заявках на ремонт и действиях пользователей. Реализована логика работы системы, включая авторизацию, управление пользователями, обработку заявок, учет и инвентаризацию оборудования, а также формирование отчетов.

Разработанное программное обеспечение прошло тестирование, в ходе которого не было выявлено критических ошибок. Система успешно выполняет поставленные задачи, обеспечивая удобный доступ к данным и автоматизацию процессов учета и обслуживания оборудования.

Таким образом, цель курсовой работы достигнута. Разработанная система может быть использована для эффективного управления техническими ресурсами предприятия, а ее гибкая архитектура позволяет адаптировать и расширять функционал в соответствии с требованиями организации.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Златопольский, Д. М. Основы программирования на языке Python / Д.М. Златопольский. - М.: ДМК Пресс, 2022
2. Культин, Н. C/C++ в задачах и примерах / Н. Культин. - М.: БХВ-Петербург, 2022.
3. Пржиялковский, В. В. Введение в Oracle SQL / В.В. Пржиялковский. - М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2020.
4. С. Е. Иванов Прикладные алгоритмы на языке ООП C# [Текст] / С. Е. Иванов — . — : Национальный исследовательский университет ИТМО, 2022 — c.
5. Фленов, Михаил Библия C# / Михаил Фленов. - М.: БХВ-Петербург, 2021.
6. Брайан, Джонсон Основы Microsoft Visual Studio .NET 2003 / Джонсон Брайан. - М.: Русская Редакция, 2022.
7. Браст, Э.Дж. Разработка приложений на основе Microsoft SQL Server 2008 / Э.Дж. Браст. - М.: Русская Редакция, 2020.
8. Гэри, Маклин Холл Адаптивный код на C#. Проектирование классов и интерфейсов, шаблоны и принципы SOLID / Гэри Маклин Холл. - М.: Вильямс, 2021.
9. Джереми Тинли , Сильвия Ботрос MySQL по максимуму. [Текст] / Джереми Тинли , Сильвия Ботрос — 4-e изд.. — : Издательский дом «Питер», 2023 — c.
10. Петцольд, Чарльз Microsoft Windows Presentation Foundation. Базовый курс / Чарльз Петцольд. - М.: Питер, Русская Редакция, 2021.
11. Уорсли, Дж. PostgreSQL. Для профессионалов (+ CD) / Дж. Уорсли, Дж. Дрейк. - М.: СПб: Питер, 2020.
12. Эккель, Брюс Философия Java / Брюс Эккель. - М.: Питер, 2022.
13. Alvin Ashcraft Learn WinUI 3: Leverage WinUI and the Windows App SDK to create modern Windows applications with C# and XAML [Текст] / Alvin Ashcraft — 2 Ed. — : Packt Publishing Limited, 2023 — c.
14. Amit Khandelwal , Sumit Sarabhai Ultimate SQL Server and Azure SQL for Data Management and Modernization [Текст] / Amit Khandelwal , Sumit Sarabhai — . — : Orange Education Pvt Ltd, AVA™, 2024 — c.
15. Berthold, Daum Eclipse 2 for Java Developers / Berthold Daum. - Москва: Мир, 2020.
16. Bob Ward Azure SQL Revealed: The Next-Generation Cloud Database with AI and Microsoft Fabric. [Текст] / Bob Ward — 2 Ed. — : Apress Media, LLC., 2024 — c.
17. Cuantum Technologies Data Analysis Foundations with Python: Master Python and Data Analysis using NumPy, Pandas, Matplotlib, and Seaborn: A Hands-On Guide with Projects. From Basics to Real-World Applications [Текст] / Cuantum Technologies — . — : Independent publishing, 2023 — 551 c.
18. Darl Kuhn , Michelle Malcher Pro Oracle Database 23c Administration: Manage and Safeguard Your Organization’s Data. [Текст] / Darl Kuhn , Michelle Malcher — 4 Ed. — : Apress Media, LLC., 2024 — c.
19. Hans-Jürgen Schönig Mastering PostgreSQL 15: Advanced techniques to build and manage scalable, reliable, and fault-tolerant database applications. [Текст] / Hans-Jürgen Schönig — 5 Ed. — : Packt Publishing Limited, 2023 — 522 c.
20. James L. Weaver , Johan Vos , Stephen Chin The Definitive Guide to Modern Java Clients with JavaFX: Cross-Platform Mobile and Cloud Development. [Текст] / James L. Weaver , Johan Vos , Stephen Chin — 3 Ed. — : Apress Media, LLC, 2024 — 637 c.
21. John Horton Beginning C++ Game Programming: Learn C++ from scratch by building fun games. [Текст] / John Horton — 3 Ed. — : Packt Publishing Limited, 2024 — c.
22. Laurentiu Spilca Troubleshooting Java: Read, debug, and optimize JVM applications [Текст] / Laurentiu Spilca — . — : Manning Publications Co., 2023 — c.
23. Mark J. Price Tools and Skills for .NET 8: Get the career you want with good practices and patterns to design, debug, and test your solutions [Текст] / Mark J. Price — . — : Packt Publishing Limited, 2024 — c.
24. Roger Ye .NET MAUI Cross-Platform Application Development: Build high-performance apps for Android, iOS, macOS, and Windows using XAML and Blazor with .NET 8 [Текст] / Roger Ye — 2 Ed. — : Packt Publishing Limited, 2024 — 497 c.
25. Romain Ottonelli Dabadie Mastering Visual Studio 2022: Develop apps like a pro with advanced Visual Studio techniques using C# and .NET [Текст] / Romain Ottonelli Dabadie — . — : Packt Publishing Limited, 2024 — 324 c.