**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

|  |
| --- |
| **Допускается к защите**  Первый заместитель директора  \_\_\_\_\_\_\_\_\_Романов А.В.  "29" февраля 2024 г. |

**ДИПЛОМНЫЙ ПРОЕКТ**

**Тема «**Разработка информационной системы поставок комплектующих изделий для организации»

**ДП. 09.02.07.24. 041 ПЗ**

Выполнил студент

Шуметов Максим Сергеевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

Руководитель

Чирков Иван Сергеевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

Рецензент

Кунский Ярослав Игоревич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

Воркута, 2024 г.

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

УТВЕРЖДАЮ

Первый заместитель

директора

\_\_\_\_\_\_\_\_\_Романов А.В.

"29" февраля 2024 г.

**ЗАДАНИЕ**

**для дипломного проекта**

Студенту: Шуметову Максиму Сергеевичу

Группа: ИСП-20

Специальность: информационные системы и программирование

Код: 09.02.07

Руководитель: Чирков Иван Сергеевич

**ТЕМА ДИПЛОМНОГО ПРОЕКТА:**

**«Разработка информационной системы поставок комплектующих изделий для организации»**

**СОДЕРЖАНИЕ**

1 Введение

2 Основная часть:

Глава 1. Техническое задание и выбор инструментария

Глава 2. Проектирование базы данных

Глава 3. Разработка информационной системы

3 Заключение

4 Список используемых источников и литературы

5 Рецензия

Дата выдачи задания «29» февраля 2024 г.

УТВЕРЖДАЮ

Первый заместитель

директора

\_\_\_\_\_\_\_\_\_Романов А.В.

"29" февраля 2024 г.

**Специальность:** 09.02.07информационные системы и программирование

**Индивидуальный план задание подготовки и выполнения**

**дипломного проекта**

**Фамилия, имя, отчество студента** Шуметов Максим Сергеевич

**Группа №:** ИСП**-**20

**Тема дипломной проекта:** «Разработка информационной системы поставок комплектующих изделий для организации»

**Фамилия, имя, отчество руководителя:** Чирков Иван Сергеевич

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование действий** | | **Исполнители** | **Сроки выполнения** | **Контроль** | **Отметка о выполнении** |
| Выбор темы дипломной работы (проекта) | | Студент  руководитель | Не позднее 2-х недель после начала производственной практики | Руководитель |  |
| Составление библиографического списка | | Студент Консультация руководителя | 29.03.2024г. | Руководитель |  |
| Сбор материалов, подготовка плана дипломной работы (проекта) | | Студент  Консультация  руководителя | 12.04.2024г. | Руководитель |  |
| Составление перечня заданий по работе на период прохождения практики | | Руководитель Методист | Не позднее 2-х недель после начала производственной практики | Первый зам. директора |  |
| Описания итогов выполнения задания, проведенного во время прохождения производственной практики (отчёт о прохождении практики) | Студент | Во время практики | Руководитель |  |
| Консультирование по написанию теоретической части | Руководитель | По графику | Руководитель |  |
| Написание теоретической части | Студент | До 03.06.2024г. | Руководитель |  |
| Представление руководителю первого варианта теоретической части работы и обсуждение представленного материала | Студент | 17.05.2024г. | Руководитель |  |
| Составление окончательного варианта работы | Студент | 10.05.2024г. | Руководитель |  |
| Заключительное консультирование | Руководитель | 03.06.2024г. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оформление работы в соответствии с требованиями, подготовка отзыва руководителя | Студент Руководитель | За 2 недели до защиты | Первый зам. директора |  |
| Рецензирование дипломной работы (проекта) | Рецензент | 10 дней до защиты | Первый зам. директора |  |
| Ознакомление студента с рецензией на ДР/ДП | Руководитель | Не позднее 2-х дней до защиты | Руководитель |  |
| Защита дипломной работы (проекта) | Студент | По графику | Комиссия |  |
| Сдача дипломной работы (проекта) на хранение |  | После  проведения  защиты | Первый зам. директора |  |

Руководитель

(подпись)

**«Ознакомлен»**

Шуметов Максим Сергеевич «29» февраля 2024г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О. студента) (дата) (подпись)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc169479183)

[ГЛАВА 1. Техническое задание и выбор инструментария 9](#_Toc169479184)

[1.1 Техническое задние 9](#_Toc169479185)

[1.2 Выбор инструментария 12](#_Toc169479186)

[СУБД SQL Server 12](#_Toc169479187)

[Microsoft SQL Server Management Studio 14](#_Toc169479188)

[Entity Framework 15](#_Toc169479189)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 17](#_Toc169479190)

[2.1 Разработка диаграммы ERD 17](#_Toc169479191)

[2.2 Разработка базы данных 19](#_Toc169479192)

[2.3 Словарь данных 24](#_Toc169479193)

[ГЛАВА 3. РАЗРАБОКА ИНФОРМАЦИОННОЙ СИСТЕМЫ 28](#_Toc169479194)

[3.1 Создание дизайна пользовательского интерфейса 28](#_Toc169479195)

[3.2 Написание кода для реализации функциональности системы 37](#_Toc169479196)

[ЗАКЛЮЧЕНИЕ 42](#_Toc169479197)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 44](#_Toc169479198)

ВВЕДЕНИЕ

В современном мире эффективное управление поставками комплектующих изделий играет ключевую роль в успешной деятельности организаций. Растущая конкуренция, изменяющиеся потребности потребителей, а также динамично развивающиеся технологии создают необходимость в разработке информационных систем, способных оптимизировать процессы управления поставками и обеспечить более эффективное взаимодействие между поставщиками и заказчиками.

Данная дипломная работа посвящена разработке информационной системы поставок комплектующих изделий для организаций, что представляет собой актуальную проблему в контексте современной деловой среды. Стремительное развитие мировой экономики и увеличение объемов международной торговли создают повышенный спрос на инновационные подходы к управлению поставками, что делает данную тему особенно значимой.

Основной целью данного исследования является разработка информационной системы, способной оптимизировать процессы управления поставками комплектующих изделий, учитывая специфику потребностей современных организаций. Это позволит улучшить оперативность поставок, уменьшить издержки и риски, а также повысить общую эффективность логистических цепочек.

В данной дипломной работе будет рассмотрена не только техническая сторона разработки информационной системы, но и ее особенности на повышение эффективности управления данными системы управления поставками.

Таким образом, данная дипломная работа имеет как теоретическую, так и практическую значимость, поскольку представляет собой вклад в развитие методов управления поставками через разработку информационной системы.

Объектом дипломной работы является база данных информационной системы поставок комплектующих изделий для организаций. База данных играет ключевую роль в функционировании любой информационной системы, особенно в области управления поставками. Она представляет собой структурированное хранилище данных, которое позволяет эффективно организовывать, хранить и обрабатывать информацию о поставках комплектующих изделий.

Предмет дипломной работы: Разработка информационной системы управления поставками комплектующих изделий для повышения эффективности процессов за счёт повышения эффективность работы, сокращения временных затрат на решения задачи и оптимизации процесса в целом.

Целью данной дипломной работы является: разработка информационной системы управления поставками комплектующих изделий для организаций с целью повышения эффективности процессов, которая обеспечит эффективный учёт поставок, улучшения контроля над запасами и обеспечения бесперебойного производственного процесса, а также повышение качества и скорости обработки информации.

Для достижения цели продукта нужно выполнить следующие задачи.

Составить техническое задание к информационной системе:

* определение функциональных требований, таких как регистрация клиентов, ввод и обработка данных, генерация отчётов.
* описание пользовательских ролей и их прав доступа к функциональности системы.
* формирование документации по проекту, включая описание бизнес-процессов и сценариев использования.

Определиться с инструментарием.

Спроектировать базу данных информационной системы:

* идентификация всех сущностей, атрибутов, необходимых для учёта комплектующих и процесса доставки товара.
* определение связей между сущностями и нормализация базы данных для предотвращения избыточности данных.
* разработка диаграммы базы данных, включая ER-диаграмму для визуализации структуры базы данных.
* учёт к производительности и безопасности при проектировании базы данных.

Разработать информационную систему:

* создание дизайна пользовательского интерфейса с учётом удобства работы пользователей.
* написание кода для реализации функционала системы, используя технологии разработки программного обеспечения.

Оптимизировать временные затраты сотрудников:

* разработка решений для ускорения процессов, таких как автоматическое заполнение форм, генерация отчётов

ГЛАВА 1. Техническое задание и выбор инструментария

1.1 Техническое задние

**1.Наименование работы**: Разработка информационной системы поставок комплектующих изделий для организаций.

**2. Назначение разработки**

Разработка информационной системы поставок комплектующих для организаций представляет собой обобщение информации о возможных поставках и заказов, которые необходимо выполнить. Это позволит значительно упростить процедуры приёма заказов, сократить временные затраты на обработку документов и повысить качество работы организации.

**3. Требования к программе:**

**3.1. Требования к функциональным характеристикам:**

Информационная подсистема обеспечивает следующий функционал:

* Ведется учёт клиентов, поставляемых товаров, поставщиков.
* Возможность создания новых заказов.
* Возможность оформления накладной.

Программа должна иметь возможность входа по логину и пароля. Все списки должны иметь функционал добавления, редактирования, удаления данных. Список с заказами должен иметь статусы такие как не заказано, заказано, доставка до клиента, у клиента которые должны отражать на каком этапе заказ в данный момент. А также возможность оформления накладной к заказу из шаблона.

**Авторизация:**

Разрабатываемая система будет иметь модульную структуру, доступ к модулям будет зависеть от роли пользователя. Доступ возможен только для авторизованных пользователей.

В связи с этим при запуске системы первым окном будет окно входа.

Для реализации авторизации необходимо создать базу данных с необходимыми таблицами и заполнить их тестовыми данными.

Доступ к системе имеют только сотрудники. У каждого сотрудника есть логин, пароль по которому и разграничиваются права доступа.

Алгоритм авторизации:

* сотрудник вводит свой логин и пароль;
* при вводе всех данных сотрудником и нажатии кнопку войти происходит проверка и вход в систему.

Сотрудник вводит свой логин и пароль в специальных полях после чего нажимает кнопку войти происходит проверка логина и пароля сотрудника.

**Разграничение прав доступа:**

Оператор имеет возможность: добавлять, редактировать, удалять данные о заказах, клиентах, поставщиках, доставляемых компонентов, типы компонентов. Так же возможность посмотреть состав заказа.

Работник склада имеет возможность: смотреть состав заказа, оформлять накладную на заказ, и менять статус заказа на «У клиента»

Администратор имеет возможность: добавлять, редактировать, удалять пользователей системы.

**3.2. Специальные требования**

Программное обеспечение должно иметь дружественный интерфейс, рассчитанный на пользователя средней квалификации (с точки зрения компьютерной грамотности).

**4. Экономический эффект от внедрения информационной системы**

При выполнении операций по созданию заказов и поставляемых комплектующих используется ручной труд, а именно ведётся письменный документа оборот. Очевидно, что использование программы значительно сократит время, затрачиваемое на учёт данных.

Разработка информационной системы направлена на оптимизацию процессов создания, реализации и обеспечение более эффективного управления данными о комплектующих, что в конечном итоге способствует повышению качества предприятия.

Экономический эффект от информационной системы ожидается за счет сокращения времени на выполняемые сотрудниками операции, исключения ошибок при формировании отчетов.

1.2 Выбор инструментария

СУБД SQL Server

Microsoft SQL Server — это система управления базами данных (СУБД), разработанная корпорацией Microsoft. Она предназначена для хранения и обработки структурированных данных, обеспечивая надежное и эффективное управление информацией в рамках организаций любого масштаба.

SQL Server предлагает широкий спектр функциональности, включая поддержку транзакций, масштабируемость, возможности репликации данных, аналитической обработки, интеграции с другими приложениями и многие другие возможности. Он позволяет хранить данные в структурированном виде, обеспечивает безопасность и управление доступом, а также предоставляет мощные инструменты для администрирования баз данных.

SQL Server поддерживает язык запросов SQL (Structured Query Language), который используется для извлечения, обновления и управления данными в базе данных. Он также обладает возможностями хранения процедур и триггеров, что позволяет создавать сложную логику обработки данных непосредственно на стороне сервера базы данных.

SQL Server доступен в различных версиях, включая бесплатные и коммерческие редакции, что делает его доступным для широкого круга пользователей, начиная от небольших компаний до крупных корпораций. Благодаря своей распространенности и гибкости, SQL Server является одной из наиболее популярных СУБД в мире и широко применяется в различных отраслях и сферах деятельности. Чтобы упростить работу с такими хранилищами данных и повысить эффективность их применения, создаются специализированные системы управления. Одной из наиболее популярных является разработка от Microsoft – SQL Server.

Преимущества решения:

* тесная интеграция с операционной системой Windows;
* высокая производительность, отказоустойчивость;
* поддержка многопользовательской среды;
* расширенные функции резервирования данных;
* работа с удаленным подключением.

Каждый выпуск включает в себя несколько специализированных редакций. Это снижает сложность внедрения и затраты на процесс разработки собственных решений, адаптированных для «узких» задач. При написании программного кода активно используется интеграция с продуктами Microsoft, например, с платформой Visual Studio.

Появление таких продуктов позволило объединить разное понимание БД (баз данных) со стороны пользователей и системных администраторов. Неискушенные в технических деталях люди «видят» таблицы как некий перечень данных с колонками и строками. Системный подход включает файлы с табличными данными, связанными друг с другом.

Функции базы данных:

* постоянное хранение информации;
* поиск по ключевым критериям;
* чтение и редактирование по запросу.

Клиентами БД являются прикладные программы, их интерфейс, различные интерактивные модули сайтов вроде калькуляторов и онлайн-редакторов. Но есть еще один компонент системы – СУБД. Он предназначен для ручного доступа к информации и позволяет извлекать данные на диск, работать с ними в памяти сервера, в том числе с применением структурированного языка SQL.

Всего различают три типа БД – клиент-серверные, файл-серверные и встраиваемые. MS SQL Server относится к первой категории. Плюс система является реляционной, т.е. адаптированной для хранения данных без избыточности, с минимальными рисками появления аномалий и нарушения целостности внутренних таблиц.

Microsoft SQL Server Management Studio

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server.

Бесплатная графическая среда, включающая набор инструментов для разработки сценариев на T-SQL и управления инфраструктурой Microsoft SQL Server.

Среда SQL Server Management Studio – это основной, стандартный и полнофункциональный инструмент для работы с Microsoft SQL Server, разработанный компанией Microsoft, который предназначен как для разработчиков, так и для администраторов SQL Server.

С помощью SSMS Вы можете разрабатывать базы данных, выполнять инструкции T-SQL, а также администрировать Microsoft SQL Server.

Если в Ваши задачи входит полное сопровождение Microsoft SQL Server, начиная от создания баз данных, написания SQL запросов, создания хранимых процедур и функций, и заканчивая администрированием SQL Server, включая управление безопасностью, то основным Вашим инструментом будет как раз среда SQL Server Management Studio.

Среда SQL Server Management Studio реализована только под Windows, поэтому если Вам нужен инструмент для работы с Microsoft SQL Server, который будет работать на других платформах, например, на Linux или macOS, то Вам следует использовать инструмент Azure Data Studio, который также является официальным инструментом, разработанным компанией Microsoft.

Entity Framework

Entity Framework — это решение для работы с базами данных, которое используется в программировании на языках семейства .NET. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), а не таблиц. Также код с использованием EF пишется гораздо быстрее.

Например, работая с базами данных напрямую, разработчик должен беспокоиться о подключении, подготовке SQL и параметров, отправке запросов и транзакций. На Entity Framework всё это делается автоматически — программист же работает непосредственно с сущностями и только говорит EF, что нужно сохранить изменения.

Набор технологий в ADO.NET, которые поддерживают разработку программных приложений, ориентированных на данные.

Архитекторам и разработчикам приложений, ориентированных на обработку данных, приходится учитывать необходимость достижения двух совершенно различных целей. Они должны моделировать сущности, связи и логику решаемых бизнес-задач, а также работать с ядрами СУБД, используемыми для сохранения и получения данных. Данные могут распределяться по нескольким системам хранения данных, в каждой из которых применяются свои протоколы, но даже в приложениях, работающих с одной системой хранения данных, необходимо поддерживать баланс между требованиями системы хранения данных и требованиями написания эффективного и удобного для обслуживания кода приложения.

Платформа Entity Framework позволяет работать с данными в форме специфических для домена объектов и свойств (например, с клиентами и их адресами) без необходимости учитывать формат базовых таблиц и столбцов базы данных, где хранятся эти данные.

Entity Framework дает разработчикам возможность работать с данными на более высоком уровне абстракции, создавать и сопровождать приложения, ориентированные на работу с данными, одновременно с этим сокращая объем кода по сравнению с традиционными приложениями. поскольку

Entity Framework является компонентом платформа .NET Framework, Entity Framework приложения могут работать на любом компьютере, на котором установлена платформа .NET Framework с пакетом обновления 1 (SP1) версии 3,5.

В итоге, проектирование базы данных в SQL Server Management Studio представляет собой слаженную работу команды разработчиков и аналитиков, цель которой – создание эффективной и надежной базы данных, соответствующей всем требованиям информационной системы.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

2.1 Разработка диаграммы ERD

Диаграмма сущность-связь (ERD) является важным инструментом в проектировании базы данных, который позволяет визуализировать структуру данных и их взаимосвязи. Давайте разберём основные понятия, связанные с ERD, а затем составим поэтапный план разработки диаграммы.

Базовыми понятиями ER-модели данных (ER — Entity— Relati-onship) являются сущность, атрибут и связь.

Сущность — это класс однотипных реальных или абстрактных объектов (людей, событий, состояний, предметов и т.п.), информация о которых имеет существенное значение для рассматриваемой предметной области. Структурой данных называют совокупность правил и ограничений, которые отражают связи, существующие между отдельными частями (элементами) данных.

Каждая сущность должна иметь:

* уникальное имя;
* один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь;
* один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности.

Атрибуты представляют собой конкретные данные, которые хранятся для каждой сущности. Например, для сущности "Пользователь" атрибутами могут быть "имя", "фамилия", "дата рождения" и т.д. Атрибуты описывают характеристики сущностей и помогают определить, какую информацию нужно хранить.

Первичный ключ — это атрибут или совокупность атрибутов и (или) связей, предназначенная для уникальной идентификации каждого экземпляра сущности (совокупность признаков, позволяющих идентифицировать объект). Ключевые атрибуты помещают в начало списка и помечают символом «#».

Обязательные атрибуты для каждой сущности всегда имеют конкретное значение, необязательные могут быть не определены. Обязательные и необязательные описательные атрибуты помечают символами «\*» и «°» соответственно.

Связи отображают отношения между различными сущностями в базе данных.

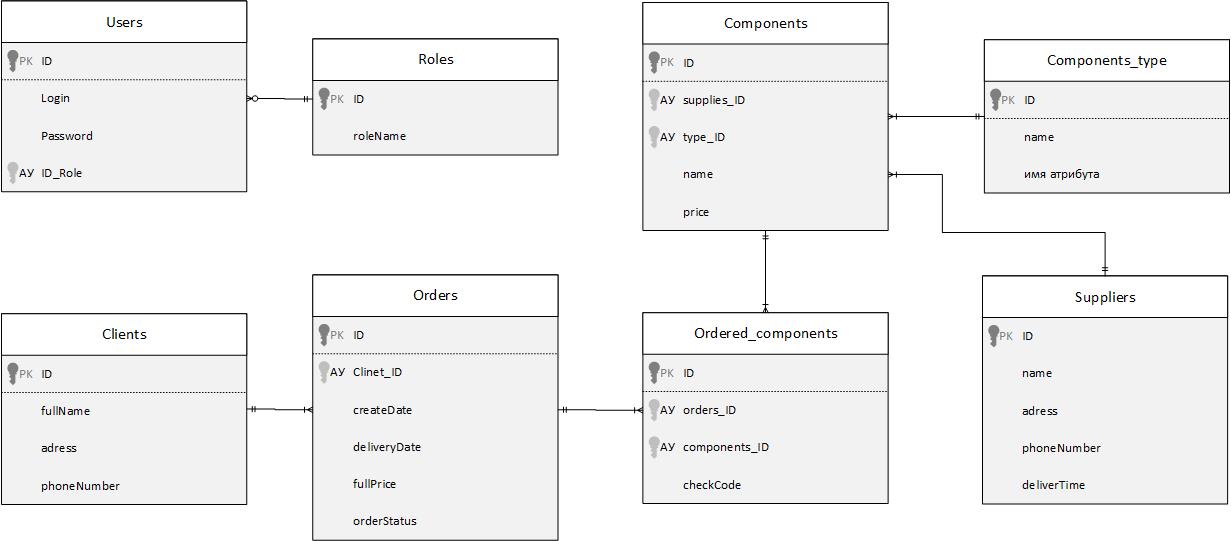


Рис. 1. Диаграмма ERD

2.2 Разработка базы данных

SQL Server Management Studio — это набор административных средств для управления компонентами, относящимися к SQL Server. Эта интегрированная среда позволяет пользователям выполнять разнообразные задачи, например резервное копирование данных, редактирование запросов и автоматизацию общих функций в одном интерфейсе.

Среда SQL Server Management Studio включает в себя следующие средства:

Редактор кода — богатый возможностями редактор скриптов для написания и редактирования скриптов. Среда SQL Server Management Studio предоставляет четыре версии редактора кода: редактор запросов ядра СУБД для скриптов Transact-SQL, редактор запросов многомерных выражений, редактор запросов расширения интеллектуального анализа данных и редактор запросов XML/A.

Обозреватель объектов для размещения, изменения, создания скрипта или выполнения объектов, принадлежащих экземплярам SQL Server.

Обозреватель шаблонов для размещения и написания сценариев шаблонов.

Обозреватель решений для организации и хранения связанных скриптов как части проекта.

Среда SQL Server Management Studio обеспечивает эффективность рабочих процессов, предоставляя:

Отключенный доступ. Можно писать и изменять скрипты, не соединяясь с экземпляром SQL Server.

Создание сценариев из любого диалогового окна. Можно создать скрипт из любого диалогового окна, а также читать, изменять, сохранять и многократно использовать скрипты после создания.

Создание базы данных для информационной системы учёта абитуриентов колледжа в SQL Server Management Studio (SSMS) включает несколько этапов, начиная с проектирования структуры базы данных и заканчивая созданием таблиц, связей и других объектов. Ниже приведён подробный порядок действий для создания базы данных в SQL Server Management Studio.

План проектирования базы данных:

* Планирование и проектирование.
* Открыть SQL Server Management Studio.
* Создание новой базы данных.
* Создание таблиц.
* Определение первичных ключей.

Планирование и проектирование базы данных для информационной системы в SQL Server Management Studio (SSMS) является фундаментальным этапом, определяющим структуру и организацию хранения данных.

На основе бизнес-требований определите основные сущности (например, "абитуриент", "специальность", "экзамен" и т.д.) и их атрибуты. Это поможет определить информацию, которую необходимо хранить.

Это включает разделение информации на отдельные таблицы и определение связей между ними.

На этом этапе создаётся высокоуровневая модель, отображающая основные сущности и связи между ними. Это обычно представляется в виде диаграммы сущность-связь (ER-диаграмма).

На основе концептуальной модели создается более детализированная логическая модель, включающая спецификацию таблиц, атрибутов, первичных и внешних ключей.

Определите связи между таблицами, включая первичные и внешние ключи, чтобы обеспечить целостность данных.

Открытие SQL Server Management Studio:

После запуска SQL Server Management Studio необходимо подключиться к серверу баз данных, на котором будет создаваться новая база данных. Для этого укажите имя сервера, тип аутентификации и учетные данные.

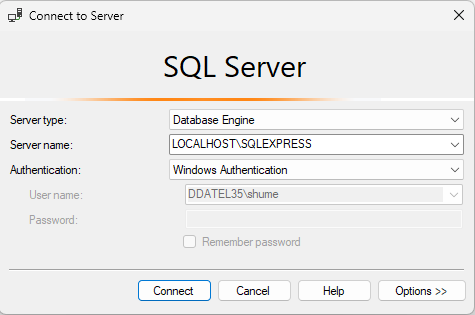


Рис. 2. Окно входа

Создание новой базы данных:

Чтобы создать новую базу данных, щелкните правой кнопкой мыши на папке "Базы данных" в Object Explorer и выберите опцию "Новая база данных". В появившемся окне укажите имя базы данных, параметры хранения данных и логов, а также другие параметры, необходимые для создания базы данных.

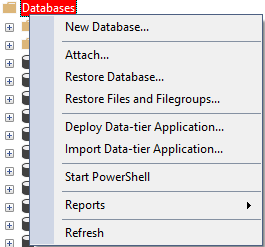


Рис. 3. Object Explorer

Определение таблиц и структуры данных:

После создания базы данных определите необходимые таблицы и структуру данных. Для этого можно использовать встроенный инструмент для проектирования таблиц в SSMS. Создайте таблицы, определите поля и их типы данных, а также ограничения (например, первичные ключи, внешние ключи).

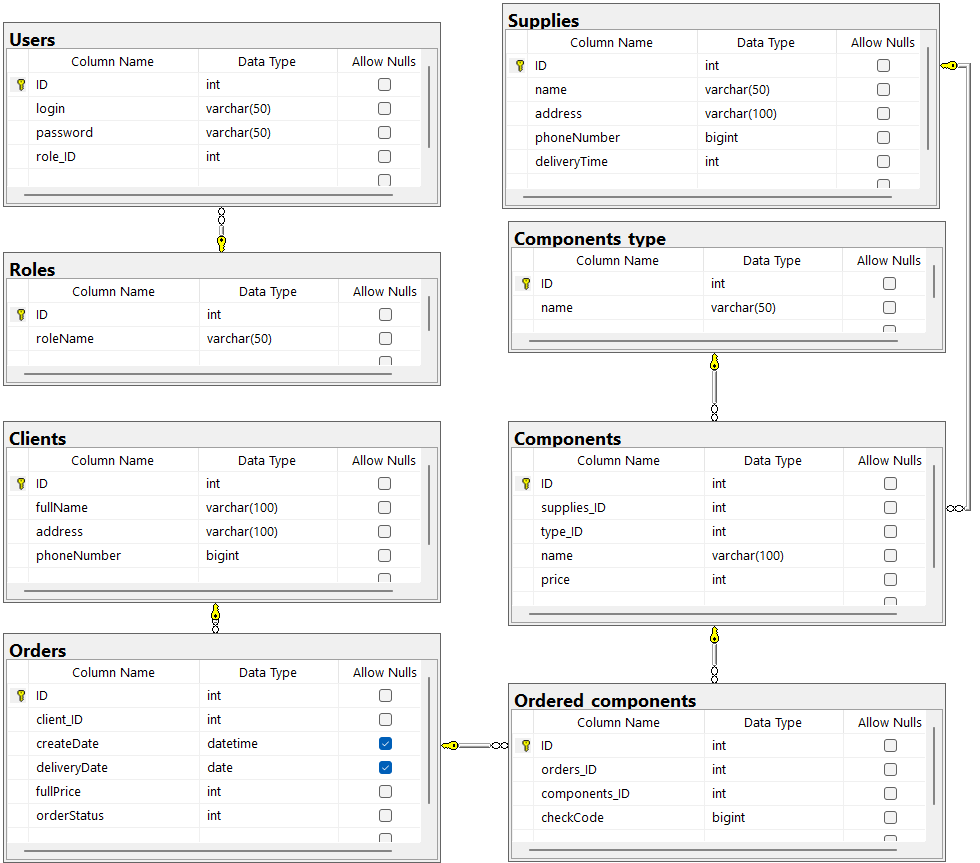


Рис. 4. Таблицы

Для реализации базы данных информационной системы поставки комплектующих за основу используется спроектированная ER диаграмма.

В данной работе используется таблицы User, Roles, Clients, Orders, Ordered\_components, Components, Components\_type, Supplies. Каждая таблица содержит уникальный ключ, атрибуты и связь.

Этот этап создания базы данных для информационной системы в SQL Server Management Studio является ключевым для успешной реализации проекта и обеспечивает эффективную работу информационной системы.

2.3 Словарь данных

Словарь данных содержит информацию об источниках, взаимосвязях между данными, их описания, сведения о характере использования. Словарь данных можно рассматривать как вспомогательную базу данных, в которой хранится информация об основной базе данных.

Пользователи баз данных и разработчики приложений могут получить выгоду от единого стандартизированного документа словаря данных, который перечисляет организацию, содержимое, соглашения по одной или более баз данных. Это обычно включает в себя имена и описания различных таблиц и полей в каждой базе данных, дополнительные детали такие, как тип и длина каждого элемента данных. Не существует универсального стандарта, описывающего уровень детализации в подобном документе, но есть основное описание метаданных о структуре базы данных, а не о самих данных.

Одним из преимуществ хорошо спроектированного словаря данных является то, что он помогает упорядочить структуру базы данных или большого комплекса баз данных.

Далее в таблицах 1-8 в виде словаря данных будет описана база данных для информационной системы поставок комплектующих изделий для организации.

Таблица 1. Пользователи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Users** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | int | Y | ID |
|  | login | varchar(50) | Y | Логин |
|  | password | varchar(50) | Y | Пароль |
|  |  |  |  |  |
| FK | role\_ID | int | Y | ID роли |

Таблица 2. Роли

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Roles** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | int | Y | ID |
|  | roleName | varchar(50) | Y | Название |

Таблица 3. Клиенты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Clients** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | INT | Y | ID |
|  | fullName | varchar(100) | Y | Полное имя |
|  | address | varchar(100) | Y | Адрес |
|  | phoneNumber | bigint | Y | Номер телефона |

Таблица 4. Заказы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Orders** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | int | Y | ID |
| FK | client\_ID | int | Y | ID клиента |
|  | createDate | datetime | N | Дата создания заказа |
|  | deliveryDate | date | N | Дата доставки заказа до организации |
|  | fullPrice | int | Y | Ценна заказа |
|  | orderStatus | int | Y | Статус заказа |

Таблица 5. Заказанные компоненты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Ordered\_components** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | int | Y | ID |
| FK | orders\_ID | int | Y | ID заказа |
| FK | components\_ID | int | Y | ID компонента |
|  | checkCode | bigint | Y | Проверочный код |

Таблица 6. Компоненты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Components** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | INT | Y | ID |
| FK | supplies\_ID | int | Y | ID поставщика |
| FK | type\_ID | int | Y | ID типа компонента |
|  | name | varchar(100) | Y | Название |
|  | price | int | Y | Ценна |

Таблица 7. Тип компонента

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Components\_type** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | int | Y | ID |
|  | name | varchar(50) | Y | Название |

Таблица 8. Поставщики

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Dictionary** | | | | |
| **Supplies** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ID | INT | Y | ID |
|  | name | varchar(100) | Y | Название организации |
|  | address | varchar(100) | Y | Адрес |
|  | phoneNumber | bigint | Y | Номер телефона |
|  | deliveryDate | int | Y | Количество дней за которое компоненты придут до организации |

ГЛАВА 3. РАЗРАБОКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Создание дизайна пользовательского интерфейса

Создание дизайна пользовательского интерфейса в Visual Studio включает в себя несколько шагов, которые позволяют создать удобный и привлекательный интерфейс для вашего приложения.

Первым шагом для создания дизайна пользовательского интерфейса в Visual Studio является создание нового проекта. Выберите тип проекта (например, Windows Forms Application или WPF Application) и укажите имя проекта и место его сохранения.

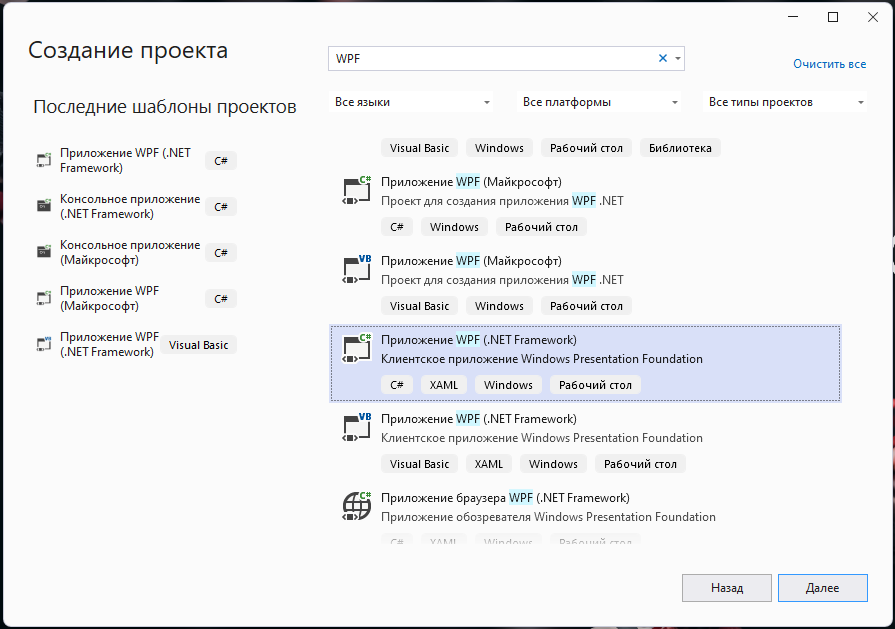


Рис. 5. создание нового проекта

Дизайн формы: Откройте файл формы в режиме дизайнера, где вы сможете создавать дизайн интерфейса визуально. Вы увидите пустую форму, на который можно размещать элементы управления.

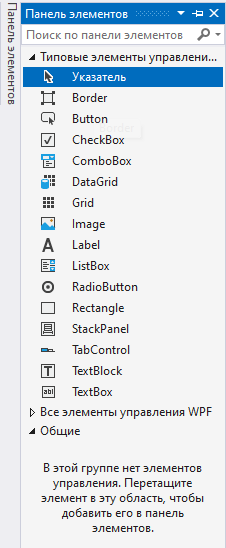


Рис. 6. Пример элементов

Добавление элементов управления: для добавления элементов управления на форму, перейдите к панели инструментов, на которой находятся все необходимые элементы: кнопки, текстовые поля, метки, списки и другие. Выберите элемент, перетащите его на форму и разместите его в нужном месте.

Настройка свойств элементов управления: После добавления элемента управления, откройте свойства этого элемента. Здесь вы можете настроить свойства элемента, такие как текст, цвет, размер, шрифт, выравнивание и другие параметры.

Дизайн макета: Расположите элементы управления на форме таким образом, чтобы дизайн был удобен для пользователя. Используйте выравнивание, сетку и другие инструменты для создания привлекательного макета.

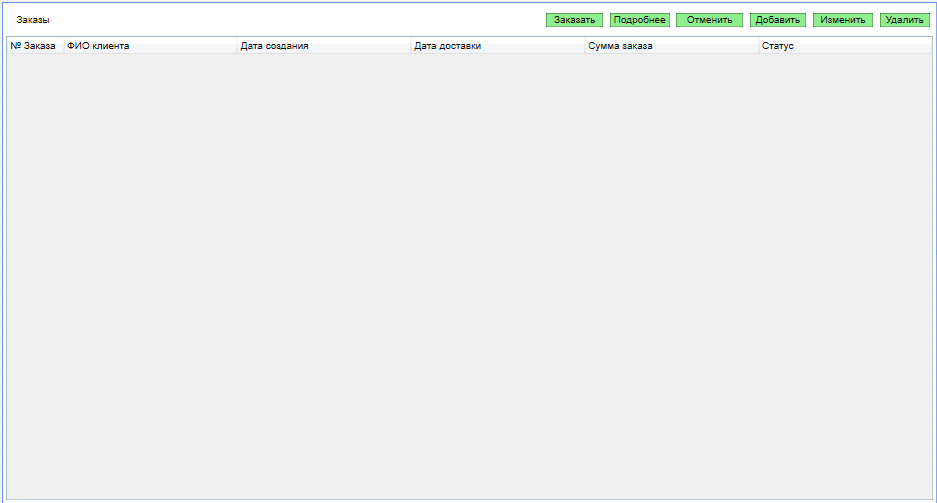


Рис. 7. Расположение элементов

Создание стилей и тем: В Visual Studio можно использовать стили и темы для создания единообразного дизайна вашего приложения. Вы можете создать собственные стили или воспользоваться готовыми темами.

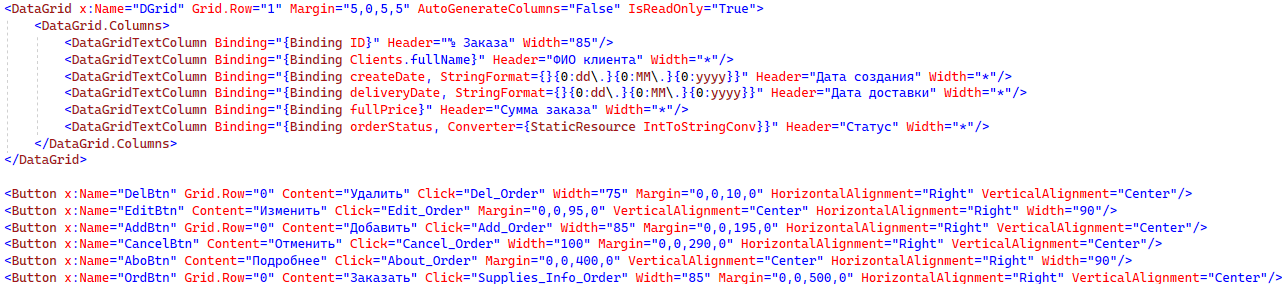


Рис. 8. Пример XAML разметки

Тестирование и отладка: После завершения дизайна интерфейса, запустите приложение для проверки работы элементов и визуального отображения. Проведите тестирование и отладку, чтобы устранить возможные ошибки.

Оптимизация для различных разрешений экрана: Учтите различные разрешения экрана и размеры устройств, чтобы ваш интерфейс выглядел хорошо на разных устройствах. Используйте адаптивный дизайн для того, чтобы ваш интерфейс корректно отображался на разных экранах.

Разработка дизайна в Visual Studio является важным аспектом создания программного обеспечения, который включает в себя не только функциональность приложения, но и его внешний вид и пользовательский интерфейс. Visual Studio предоставляет разработчикам мощные инструменты для работы с дизайном, позволяя создавать красивые и удобные пользовательские интерфейсы.

Одним из ключевых элементов разработки дизайна в Visual Studio является использование различных инструментов для создания графических элементов, таких как кнопки, текстовые поля, таблицы и другие компоненты интерфейса.

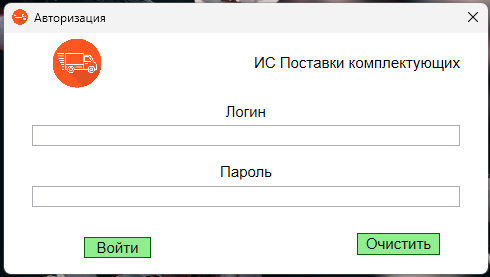
Кроме того, Visual Studio поддерживает работу с различными стилями и темами оформления, что позволяет создавать современные и стильные интерфейсы приложений.

Важным аспектом разработки дизайна в Visual Studio является также возможность предварительного просмотра и тестирования интерфейса приложения непосредственно в среде разработки.

Далее будет описан дизайн разрабатываемой информационной системы.

Окно авторизации.

Так как для использования информационной системы необходимо в ней авторизироваться, чтобы понимать какой пользователь на данный момент за компьютером. Как выглядит окно авторизации можно увидеть на рисунке 9.

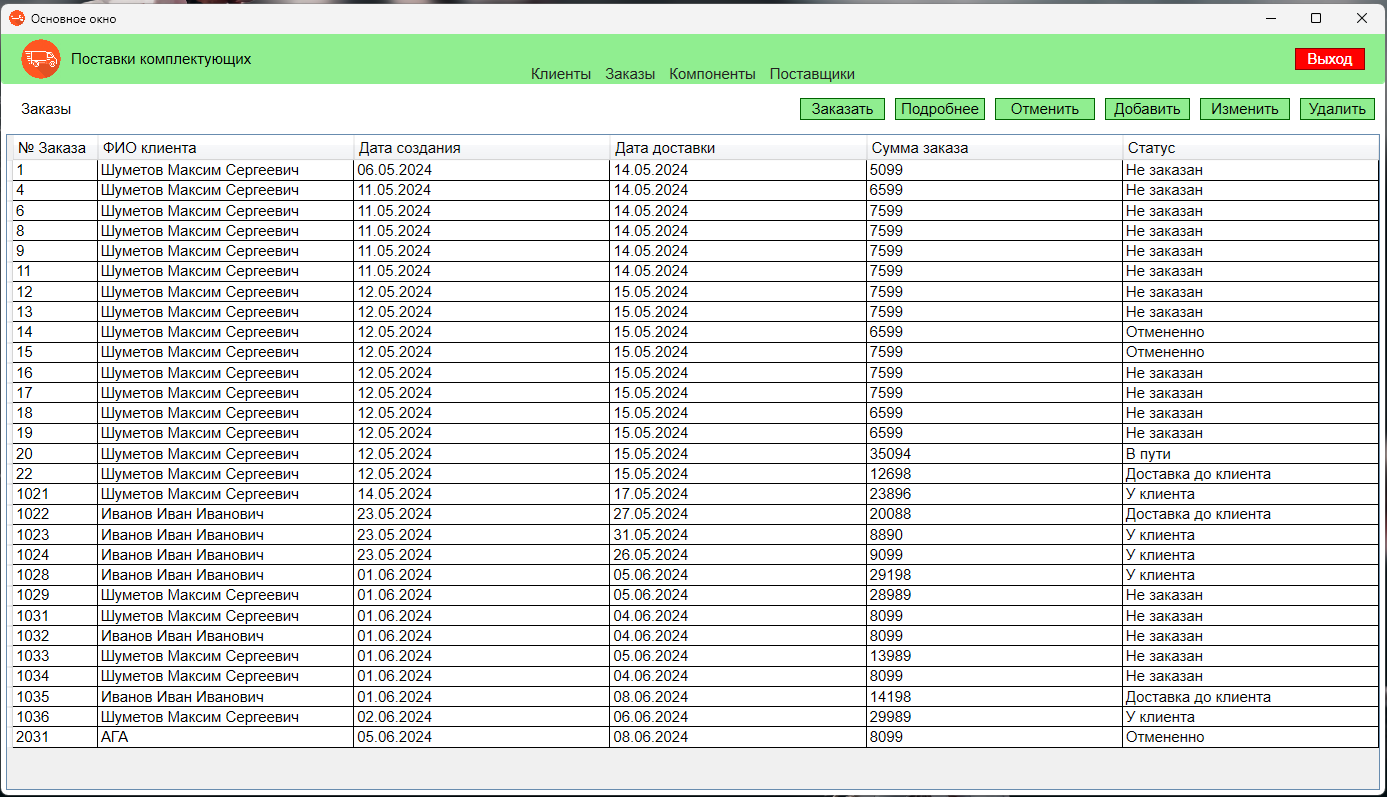


*Рис. 9. Окно авторизации*

На окне авторизации можно увидеть тектовые поля для ввода данных пользователя и несколько функциональных кнопок: вход, очистить.

Основное окно информационной системы для оператора.

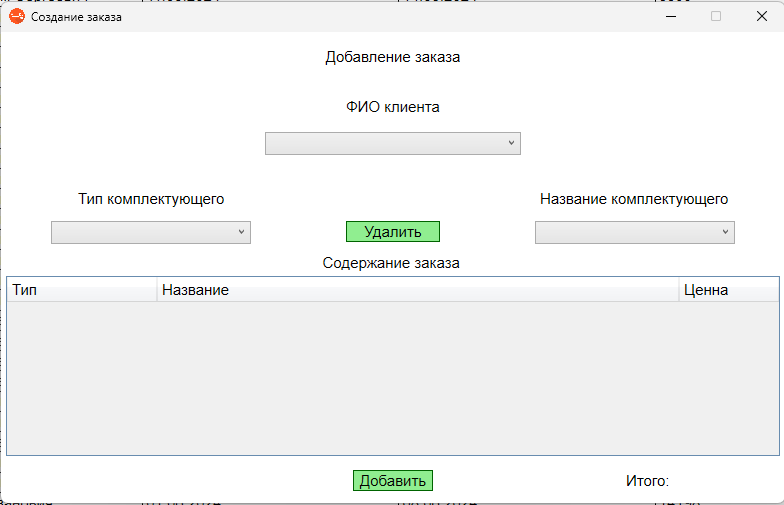
В разрабатываемой информационной системе используется основное окно которое можно увидеть на рисунке 10. На основном окне размещён объект под названием Frame в котором возможна смена страниц со своими элементами.



*Рис. 10. Основное окно с основной страницей*

На основной странице показана основная страница с заказами. На этой странице изображенна таблица с заказами и функциональные кнопки такие как: добавить, изменить, удалить, заказать, подробнее, отменить. Так же сверху посередине можно увидеть меню для перехода по останльным страницам которые похожи на страницы с заказами. Данная страница выглядит так только для пользователя с ролью оператора.

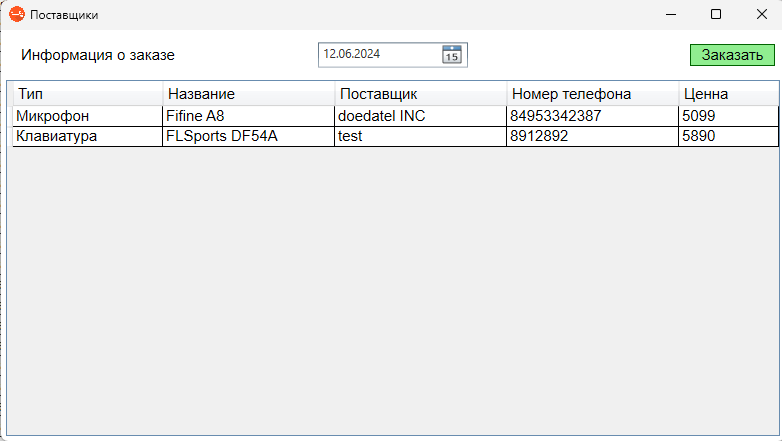
Окно добавления заказа.



*Рис. 11. Окно добавления заказа*

На окне добавления заказа можно увидеть выпадающие списки для выбора клиента и поиска компонента который необходимо добваить к заказу. Ниже можно увидеть таблицу которая отображает предварительный состав заказа, а так же кнопку добавления заказа и предварительной ценны.

Окно для заказа.

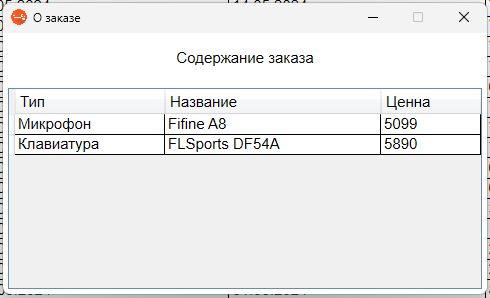


*Рис. 12. Окно для закза*

На рисунок 12 можно увидеть таблицу с составом заказа и так же номерами поставщикам которых нужно обзовнить для заказа компонентов. Так же сверху можно увидеть поле для ввода даты, если поставщик не успеет привести товар до написанной даты, а так же кнопку «Заказать» для смены статуса заказа на «В пути».

Окно вывода состава заказа.

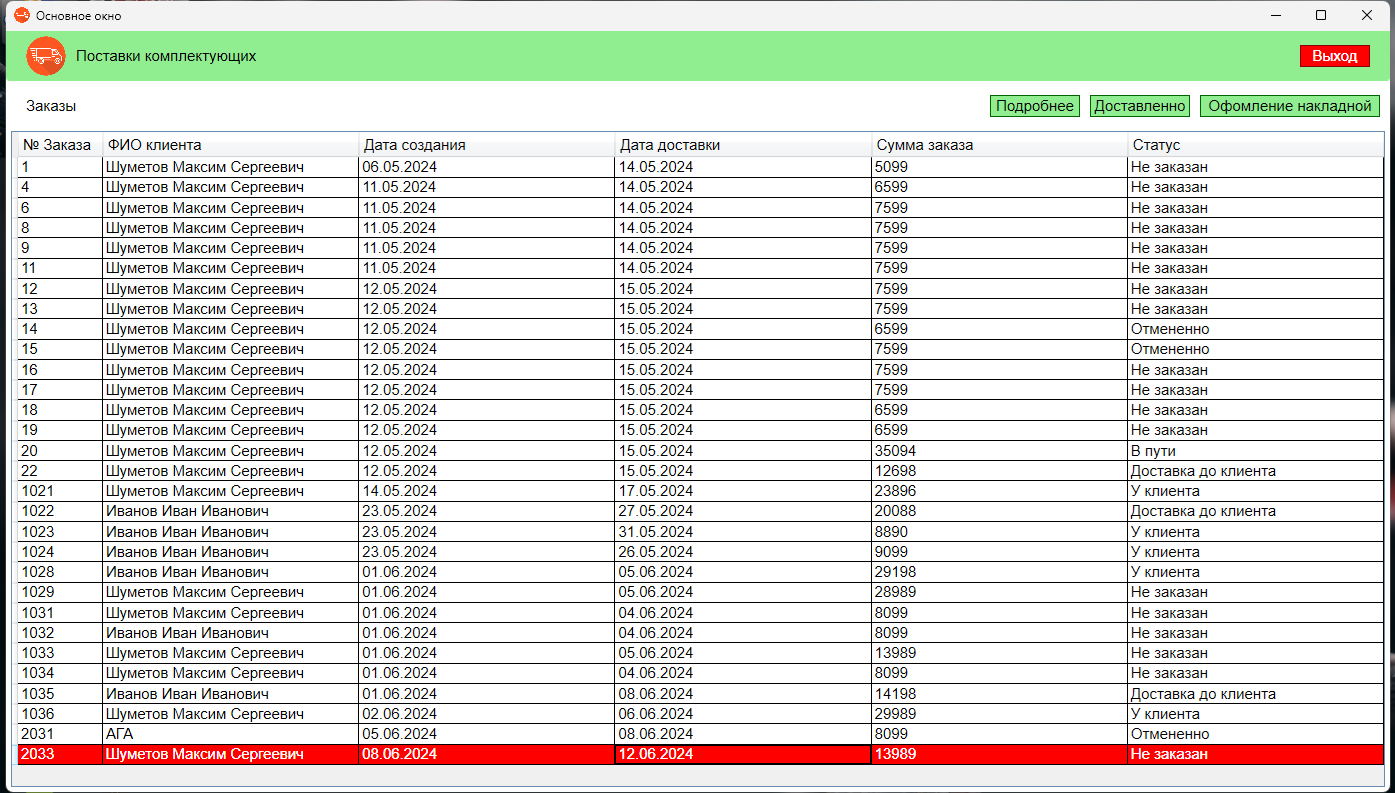
Если необходимо посмотреть состав заказа который добавлен уже в информационную систему, то его можно увидеть в окне состава заказа, данное окно можно увидеть на рисунке 13.



*Рис. 13. Окно вывода состава заказа*

На данном окне можно увидеть всего лишь таблицу с выводом состава выбранного заказа из таблицы заказов.

Основное окно информационной системы для работника склада.



*Рис. 14. Основное окно с основной страницей*

На основной странице показанна основная страница с заказами. На этой странице изображенна таблица с заказами и функциональные кнопки такие как: оформление накладной, доставленно, подробнее. Кнопки, кроме кнопки «Подробнее», меняют статус заказа для понимания на какой стадии находиться заказ. Данная страница выглядит так только для пользователя с ролью работника склада.

Таким образом, разработка дизайна в Visual Studio играет важную роль в создании качественных и удобных программных продуктов, обеспечивая эстетичный внешний вид и удобство использования для конечных пользователей. Благодаря богатому набору инструментов и возможностей Visual Studio, разработчики могут эффективно работать над дизайном своих приложений и достигать высоких результатов в создании современных интерфейсов.

3.2 Написание кода для реализации функциональности системы

Процесс написания кода для реализации функциональности информационной системы в Visual Studio включает в себя несколько этапов, начиная от создания проекта и завершая тестированием и оптимизацией. Для обработки событий, которые происходят в информационной системе (например, нажатие кнопок, отправка форм и т.д.), а также для взаимодействия с пользователем нужен код.

Производство разработки модулей информационной системы в соответствии с техническим заданием.

Работа с базой данных. Если ваша информационная система использует базу данных, напишите код для взаимодействия с базой данных: выполнение запросов, обновление данных, извлечение информации и т.д.

После написания кода проведите тестирование функциональности вашей информационной системы. Это может включать в себя модульное тестирование отдельных компонентов, интеграционное тестирование и тестирование пользовательского интерфейса.

Если необходимо, отладьте код для выявления и исправления ошибок. Оптимизируйте код для повышения производительности и эффективности информационной системы.

Это общий процесс написания кода для реализации функциональности информационной системы в Visual Studio. Конечно, каждая информационная система может иметь свои особенности и требования, но эти шаги представляют общий каркас разработки информационных систем.

Таблица 9 описывает взаимодействие пользователей системы (по ролям) с функциональными модулями системы. Описание отражает действия пользователя в зависимости от его роли и реакцию системы на это действие. Описание строится на основе предметной области.

Таблица 9. Взаимодействие пользователей системы

|  |  |
| --- | --- |
| Пользователь | Система |
| Пользователь заходит в ИС под своим логином и паролем | Проверка логина и пароля из БД если данные существуют, то пользователь попадает в ИС. |
| Пользователь выходит из приложения | Главное окно закрывается, вместо него появляется окно для входа |
| Администратор заходит в ИС | Вывод первого окна администратору на котором выводится список пользователей из БД |
| Администратор добавляет пользователя | Вывод окна для добавления нового работника (логин, пароль и т.п) после нажатия на кнопку «Добавить» данные сохраняются в БД. |
| Администратор редактирует пользователя | Вывод окна редактирования выбранного пользователя и сохранение данных в БД |
| Администратор удаляет пользователя | Удаление выбранного пользователя из БД |
| Оператор заходит в ИС | Вывод главного окна, на котором выводиться список заказов от клиентов из БД |
| Оператор переходит по фреймам | Вывод другого фрейма в главном окне |
| Оператор добавляет клиента, комплектующее, тип комплектующего, поставщика | Вывод окна добавления с полями для ввода данных, после нажатия на кнопку «Добавить» данные сохраняются в БД |
| Оператор редактирует клиента, комплектующее, поставщика | Вывод окна редактирования с полями для изменения данных выбранного элемента и сохранение в БД |
| Оператор удаляет клиента, комплектующее, тип комплектующего, поставщика | Удаление выбранного элемента из БД |
| Оператор добавляет заказ | Вывод окна добавления заказа с выбором клиента и таблицей с предварительным составом заказа |
| Оператор сохраняет заказ | Сохранение выбранных компонентов в БД |
| Оператор удаляет заказ | Удаление выбранного заказа из БД |
| Оператор заказывает заказ | Вывод окна с отображением состава заказа и номеров поставщиков, после нажатия на кнопку «Заказать» статус заказа меняется на «В пути» |
| Оператор отменяет заказ | Смена статуса заказа на «Отменено» |
| Работник склада заходит в ИС | Вывод главного окна, на котором выводится список заказов из БД |
| Работник склада смотрит состав заказа | Вывод окна с составом заказа из БД |
| Работник склада оформляет накладную на заказ | Смена статуса выбранного заказа на «Доставка до клиента» и формирование накладной к заказу с данными из БД |
| Работник склада меняет статус заказа | Смена статус выбранного заказа на «У клиента» и сохранение в БД |

Пользовательские истории (User Stories) — это инструмент описания требований к программному обеспечению, который фокусируется на потребностях и целях конечных пользователей. Они используются в методологии разработки ПО, для определения функциональности и поведения системы из перспективы пользователей.

Каждая пользовательская история описывает конкретный аспект функциональности системы с точки зрения определенной роли пользователя.

* Роль пользователя: определяет, кто именно будет использовать систему или функциональность, и какая у него есть цель.
* Действия пользователя: описывает конкретные действия, которые пользователь будет выполнять в системе.
* Цели действий: объясняет, какие цели у пользователя при выполнении этих действий и какую пользу он от них получит.

Таблица 10 описывает пользовательские истории в зависимости от роли пользователя. Описание включает роль пользователя системы, производимые им действия и цели этих действий. У одного действия, не может быть более одной цели.

Таблица 10. Пользовательские истории

|  |  |  |
| --- | --- | --- |
| Роль | Действие | Цель |
| Администратор | Заходит в ИС | Зайти в ИС для просмотра или изменения данных |
| Администратор | Добавление пользователя | Добавление нового пользователя в БД для возможности войти ему в приложение под своей ролью |
| Администратор | Редактирование пользователя | Изменение данных выбранного пользователя из БД |
| Администратор | Удаление пользователя | Удаление выбранного пользователя из БД |
| Оператор | Заходит в ИС | Зайти в ИС для просмотра или изменения данных |
| Оператор | Переход по фреймам | Просмотр или изменение данных из БД |
| Оператор | Добавление заказа, клиента, комплектующего, типа комплектующего, поставщика | Добавить клиента, комплектующее, типа комплектующего, поставщика |
| Оператор | Редактирование заказа, клиента, комплектующего, поставщика | Редактирование клиента, комплектующего, поставщика |
| Оператор | Удаление заказа, клиента, комплектующего, тип комплектующего, поставщика | Удаление клиента, комплектующего, тип комплектующего, поставщика |
| Оператор | Заказ заказа | Заказ заказа |
| Оператор | Отмена заказа | Отмена заказа |
| Работник склада | Заходит в ИС | Зайти в ИС для просмотра или изменения данных |
| Работник склада | Просмотр состава заказа | Просмотр состава заказа |
| Работник склада | Формирование накладной к заказу | Формирование накладной к заказу |
| Работник склада | Смена статуса заказа | Смена статуса заказа |

Таким образом, пользовательские истории помогают команде разработки ПО лучше понять потребности пользователей и создавать функциональность системы, которая будет соответствовать их ожиданиям и целям.

**ЗАКЛЮЧЕНИЕ**

Информационные системы являются актуальными на сегодняшний день - одна из наиболее перспективных и быстроразвивающихся отраслей.

В ходе выполнения дипломной работы была разработана информационная система поставок комплектующих изделий для организации, что позволит оптимизировать процессы управления поставками и повысить эффективность работы предприятия. Была спроектирована диаграмма: Entity-Relationship Diagram, и разработана база данных, а также разработан интерфейс приложения и произведено программирование информационной системы.

Целью работы было создание системы, способной эффективно обрабатывать информацию управления поставками комплектующих изделий в условиях современного бизнеса. Постоянно меняющиеся рыночные условия требуют более гибкого и точного управления поставками, что делает разработку информационной системы важной задачей для предприятия. В результате выполнения поставленных задач была разработана информационная система, способствующая оптимизации системы поставками комплектующих изделий.

Оптимизация процессов управления поставками привела к сокращению издержек на логистику, улучшению планирования производственных процессов.

В процессе выполнения работы были поставлены и успешно выполнены следующие задачи:

* Проектирование информационной системы, включая разработку структуры базы данных, интерфейсов и функциональности.
* Разработка информационной системы.
* Определение функциональных требований.
* Описание пользовательских ролей.
* Оптимизировать временные затраты сотрудников.
* Составить техническое задание к информационной системе.

Эффект от внедрения информационной системы поставки комплектующих для организаций проявляется через сокращение временных и трудовых затрат на обработку информации, оптимизацию процесса и улучшение качества обслуживания. Это приводит к повышению эффективности работы, улучшению качества данных и процессов в целом.

Таким образом, разработка и внедрение информационной системы поставки комплектующих для организации имеет высокую актуальность, обладает значительным потенциалом для повышения эффективности управления данными о клиентах, заказах, поставщиках, компонентах.

Ссылка на GitHub репозиторий дипломного проекта: <https://github.com/Ddatel35/ISSCO>

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Герасимова Е. К. Цифровизация образования: от теории к практике: учебное пособие/ Герасимова Елена, 2022. – 157 с.
2. Гриффитс И. Программируем на C# 8.0. Разработка приложений: учебное пособие/ Иэн Гриффитс, 2020. – 944 с.
3. Литвак Е. SQL с нуля и быстро: учебное пособие/ Елена Литвак, 2023. – 190 с.
4. Молинаро Э. де Граф Р. SQL. Сборник рецептов: учебное пособие/ Энтони Молинаро, Роберт де Граф, 2021. – 591 с.
5. Натан А. WPF 4. Подробное руководство: учебное пособие/ Адам Натан, 2021. – 880 с.
6. Ошероув Р. Искусство автономного тестирования с примерами на C#: учебное пособие/ Рой Ошероув, 2022. – 362 с.
7. Рафалович В. Data mining, или Интеллектуальный анализ данных для занятых. Практический курс: учебное пособие/ Владимир Рафалович, 2020. – 110 с.
8. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#: учебное пособие/ Джеффри Рихтер, 2022. – 893 с.
9. Стиллман Э. Грин Д. Изучаем C#: учебное пособие/ Эндрю Стиллман, Дженнифер Грин, 2021. – 769 с.
10. Ульстайн С. Шепот питона: учебное пособие/ Cилье Ульстайн, 2020. – 350 с.

**Интернет-ресурсы**

1. C Sharp – Режим доступа: <https://ru.wikipedia.org/wiki/C_Sharp>
2. Entity Framework: как быстрее написать код для работы с базой данных – Режим доступа: https://skillbox.ru/media/code/entity\_framework/
3. Geek Brains Изучение C# // ИТ-инженер. Передовые технологии [Электронный ресурс] – Режим доступа: <https://gb.ru/geek_university/engineer/main>
4. Geek Brains Изучение C# // Мета вселенные [Электронный ресурс] – Режим доступа: <https://gb.ru/geek_university/engineer/metaverse>
5. Geek Brains Изучение C# // Разработчик: старт в ИТ с нуля до Junior [Электронный ресурс] – Режим доступа: <https://gb.ru/geek_university/developer>
6. Geek Brains Изучение C# // Разработчик: старт в ИТ с нуля до Middle [Электронный ресурс] – Режим доступа: <https://gb.ru/geek_university/developer/engineer/it-master>
7. Geek Brains Изучение C# // Разработчик: старт в ИТ с нуля до Pro [Электронный ресурс] – Режим доступа: <https://gb.ru/geek_university/developer/engineer/it-pro>
8. Meta nit Изучение C# // Начало работы. Visual Studio [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/tutorial/1.2.php>
9. Meta nit Изучение C# // Первое приложение в WSL [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/tutorial/1.7.php>
10. Meta nit Изучение C# // Язык C# и платформа .NET [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/tutorial/1.1.php>
11. Microsoft SQL Server – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_SQL\_Server
12. NET Framework – Режим доступа: https://ru.Wikipedia.org/wiki/.NET\_Framework
13. Skill Box Изучение C# // Профессия Frontend-разработчик [Электронный ресурс] – Режим доступа: <https://skillbox.ru/course/frontend-developer/>
14. Skill Box Изучение C# // Профессия Специалист по кибер­безопас­нос­ти с нуля [Электронный ресурс] – Режим доступа: <https://skillbox.ru/course/profession-cybersecurity/>
15. SQL Server Management Studio – Режим доступа: https://ru.wikipedia.org/wiki/SQL\_Server\_Management\_Studio.