**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc180622744)

[1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ 5](#_Toc180622745)

[1.1 Анализ предметной области 5](#_Toc180622746)

[1.2 Выбор средств разработки 6](#_Toc180622747)

[1.2.1 Язык программирования 6](#_Toc180622748)

[1.2.2 Система управления базами данных 10](#_Toc180622749)

[1.2.3 Среда разработки 12](#_Toc180622750)

[1.3 Описание предметной области 15](#_Toc180622751)

# ВВЕДЕНИЕ

Актуальность разработки программного обеспечения для предоставления строительных услуг обусловлена ростом спроса на автоматизацию бизнес-процессов в строительной отрасли и необходимостью эффективного управления проектами, ресурсами и заказами. Строительные проекты часто требуют координации большого количества участников и ресурсов, что делает важным внедрение современных технологий для учета и контроля процессов строительства.

Предметная область данной работы – информационные системы для управления процессами в компаниях, предоставляющих строительные услуги.

Объект исследования – процесс автоматизации работы строительных компаний.

Предмет исследования – разработка и внедрение автоматизированной информационной системы для управления строительными проектами и предоставлением строительных услуг.

Цель курсовой работы – разработка программного обеспечения, которое позволит автоматизировать управление строительными услугами, обеспечивая эффективное хранение информации и упрощение работы с клиентами и проектами.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать предметную область и существующие программные решения для управления строительными проектами и услугами.
2. Определить функциональные и нефункциональные требования к программному обеспечению.
3. Спроектировать архитектуру системы и структуру базы данных.
4. Реализовать основные модули приложения, включая управление проектами, клиентами, сотрудниками и ресурсами.

# 1. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

## 1.1 Анализ предметной области

Разработка программного обеспечения по предоставлению строительных услуг представляет собой организацию, которая специализируется на выполнении широкого спектра строительных работ для частных лиц и компаний. Основной задачей предприятия является обеспечение эффективного управления проектами строительства, своевременное выполнение заказов и поддержание актуальной информации о клиентах, услугах и задействованных ресурсах.

Разрабатываемое программное обеспечение должно автоматизировать ключевые процессы предприятия, включая:

* управление проектами строительства;
* оформление заказов на предоставление услуг;
* отслеживание выполнения работ;
* взаимодействие с клиентами;
* предприятие предоставляет несколько основных услуг;
* строительство: клиенты могут заказать строительные услуги, такие как возведение зданий, монтаж инженерных систем и другие строительные работы;
* проектирование и планирование: услуги по разработке проектов, планированию строительства, расчету стоимости материалов и трудозатрат.

Целевая аудитория:

* менеджеры проекта: сотрудники, отвечающие за управление строительными проектами, которые будут использовать программное обеспечение для планирования работ, управления ресурсами, составления графиков и контроля выполнения задач;
* клиенты: физические лица или компании, заинтересованные в заказе строительных услуг. Они будут использовать систему для выбора услуг, оформления заказов и получения информации о ходе выполнения работ;
* рабочие бригады: сотрудники, непосредственно выполняющие строительные работы, включая прорабов, начальников участков и рабочих. Они будут использовать систему для получения информации о назначенных задачах и сроках выполнения;
* администраторы: сотрудники, управляющие базой данных заказов, услуг и ресурсов, которые будут вести учет и обновление информации.

Таким образом, разрабатываемое программное обеспечение должно обеспечить прозрачное и эффективное взаимодействие между всеми категориями пользователей, предоставляя удобные и интуитивно понятные инструменты для выполнения их задач.

## 1.2 Выбор средств разработки

При разработке программного обеспечения для предоставления строительных услуг необходимо выбрать инструменты, которые обеспечат эффективное выполнение проекта. В этом разделе будут рассмотрены основные компоненты разработки, включая язык программирования, операционную среду, систему управления базами данных и среду разработки.

### 1.2.1 Язык программирования

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Выбор языка программирования играет основоположную роль в проектировании и реализации ПО. Рассмотрим два наиболее подходящих для данного проекта языка:

1. Python:

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нем программ.

Рассмотрим преимущества и недостатки данного языка программирования

Преимущества:

1. Простота и читабельность: Python имеет понятный синтаксис, близкий к естественному языку, что делает его идеальным для новичков и способствует быстрой разработке.
2. Большая стандартная библиотека: Python поставляется с обширной библиотекой модулей, что упрощает выполнение различных задач, таких как работа с файлами, обработка данных и сетевое программирование.

Недостатки:

1. Медленное выполнение: по сравнению с компилируемыми языками, Python работает медленнее из-за интерпретируемого характера.
2. Высокое потребление памяти: Python может требовать больше оперативной памяти, особенно при работе с большими объемами данных или в высоконагруженных системах.
3. C#:

С# – объектно-ориентированный язык программирования общего назначения. Разработанный для платформы .NET, подходящий для разработки приложений на Windows. С его помощью можно быстро создавать приложения с богатым пользовательским интерфейсом, а также серверные решения.

Рассмотрим преимущества и недостатки данного языка программирования

Преимущества:

1. Интеграция с .NET: C# отлично интегрирован с платформой .NET, что предоставляет доступ к большому количеству библиотек, фреймворков и инструментов для разработки различных типов приложений, включая веб-приложения, десктопные программы и мобильные приложения.
2. Строгость типизации: C# — это статически типизированный язык, что помогает обнаруживать ошибки на этапе компиляции, улучшая стабильность и безопасность программ.

Недостатки:

1. Зависимость от платформы Windows: хотя с появлением .NET Core и .NET 5/6 C# стал кроссплатформенным, многие его функции исторически были сильно завязаны на экосистему Windows, что все еще может создавать сложности при разработке под другие ОС.
2. Большие ресурсоемкие приложения: Программы на C# могут потреблять больше ресурсов, особенно в сравнении с языками низкого уровня (например, C++), что может быть проблемой для высокопроизводительных систем.

В результате анализа языков программирования для разработки программного обеспечения для предоставления строительных услуг было принято решение выбрать C#. Этот язык был выбран по следующим причинам:

* Интеграция с .NET и поддержка Windows: C# отлично подходит для разработки бизнес-приложений, особенно тех, которые должны работать на Windows. Для программного обеспечения для предоставления строительных услуг, где, возможно, используются различные системы для управления заказами, учета клиентов и финансами, возможности платформы .NET предоставляют гибкость и надежность;
* Объектно-ориентированный подход: для управления данными о строительных услугах C# предоставляет мощные инструменты объектно-ориентированного программирования, что позволяет легко моделировать реальные объекты, такие как строительные заказы и взаимодействия с клиентами;
* Поддержка сложных бизнес-логик: C# позволяет создавать сложные системы с применением строгой типизации и расширенных возможностей управления данными. Это упрощает разработку надежного программного обеспечения с минимальным количеством ошибок, что особенно важно в системах управления заказами, складами и учетными системами, которые могут использоваться в программном обеспечении для предоставления строительных услуг;
* Гибкие UI-технологии: C# поддерживает разработку пользовательских интерфейсов через Windows Forms, WPF или веб-приложения через ASP.NET, что позволяет создать удобное и функциональное программное обеспечение для сотрудников и клиентов.

Таким образом, выбор C# в сочетании с Windows обеспечивает максимальную продуктивность разработки и позволяет эффективно решать задачи, связанные с созданием надежного и функционального программного обеспечения для данного проекта.

### 1.2.2 Система управления базами данных

Система управления базами данных (СУБД) — это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными. СУБД обеспечивает пользователям и приложениям интерфейс для работы с данными, позволяя выполнять операции по их хранению, извлечению, обновлению и удалению.

Для хранения данных необходимо выбрать надежную систему управления базами данных. Рассмотрим два наиболее подходящих варианта для данного проекта:

1. PostgreSQL

PostgreSQL — это мощная объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом. Она известна своей гибкостью, надежностью и поддержкой стандартов SQL. PostgreSQL широко используется в приложениях, требующих сложной обработки данных, высокой производительности и масштабируемости.

Рассмотрим преимущества и недостатки данной СУБД

Преимущества:

1. Открытый исходный код: PostgreSQL является свободной и открытой СУБД, что позволяет пользователям свободно изменять и распространять её код в соответствии с лицензией.
2. Расширенные возможности: PostgreSQL поддерживает широкий спектр функций, включая сложные запросы, индексы, триггеры, хранимые процедуры и многое другое, что делает его мощным инструментом для различных задач.
3. Масштабируемость: СУБД хорошо масштабируется и способна обрабатывать большие объемы данных, что делает её подходящей для проектов с различными масштабами и потребностями.

Недостатки:

1. Высокая цена: PostgreSQL может требовать дополнительных ресурсов и инвестиций в оборудование для обеспечения высокой.
2. производительности, что может повлечь за собой значительные затраты для предприятий.
3. Высокое потребление оперативной памяти: при работе с крупными объемами данных PostgreSQL может потреблять значительное количество оперативной памяти, что может стать проблемой в случае ограниченных ресурсов сервера.
4. Microsoft SQL Server — это реляционная система управления базами данных (СУБД), разработанная компанией Microsoft. SQL Server используется для хранения и управления данными, поддерживает стандартный язык запросов SQL и широко применяется в корпоративных и бизнес-системах. Эта СУБД предоставляет множество инструментов для обработки, анализа и визуализации данных.

Рассмотрим преимущества и недостатки данной СУБД

Преимущества:

1. Интеграция с продуктами Microsoft: SQL Server отлично взаимодействует с другими продуктами Microsoft, такими как Windows Server, Azure, Visual Studio и Power BI. Это делает его естественным выбором для компаний, использующих Microsoft-технологии.
2. Высокая производительность и масштабируемость: SQL Server поддерживает работу с большими объемами данных и обеспечивает высокую производительность для корпоративных приложений. Он оптимизирован для сложных транзакций и аналитических запросов, что делает его подходящим для масштабируемых решений.
3. Инструменты бизнес-аналитики: Встроенные инструменты, такие как SQL Server Reporting Services (SSRS), Integration Services (SSIS) и Analysis Services (SSAS), делают SQL Server мощным инструментом для анализа данных и отчетности.

Недостатки:

1. Высокая стоимость: Коммерческая версия SQL Server может быть дорогой, особенно для крупных предприятий, которым требуется лицензирование на основе количества ядер процессора. Хотя существуют бесплатные версии (SQL Server Express), они ограничены в функциональности и масштабируемости.
2. Зависимость от Windows: хотя с появлением SQL Server для Linux зависимость от Windows уменьшилась, все же большинство функций и поддержка тесно интегрированы с Windows, что ограничивает гибкость при использовании других операционных систем.
3. Высокие системные требования: SQL Server требует значительных ресурсов для обеспечения высокой производительности, особенно при работе с большими базами данных и высокими нагрузками. Это может стать проблемой для небольших компаний или проектов с ограниченным бюджетом на инфраструктуру.

Таким образом, выбор Microsoft SQL Server в качестве системы управления базами данных обеспечивает необходимую производительность, надежность и гибкость для успешной реализации проекта, направленного на создание эффективного программного обеспечения для предоставления строительных услуг.

### 1.2.3 Среда разработки

**Среда разработки (Integrated Development Environment, IDE)** — это программное обеспечение, предназначенное для облегчения процесса разработки программного обеспечения. IDE объединяет в себе набор инструментов, необходимых для написания, редактирования, компиляции, отладки и тестирования кода, что делает процесс разработки более удобным и эффективным.

Выбор среды разработки (IDE) важен для удобства и эффективности работы над проектом. Рассмотрим две популярные среды:

1. Project Rider

JetBrains Rider — это мощная интегрированная среда разработки (IDE) от компании JetBrains, предназначенная для разработки на языке C# и других языках, поддерживаемых платформой .NET. Rider объединяет в себе возможности ReSharper и платформы IntelliJ IDEA, предоставляя удобный интерфейс и богатый функционал для разработки приложений на различных технологиях .NET, включая .NET Core, .NET Framework, ASP.NET, Xamarin и Unity.

Преимущества:

1. Поддержка множества платформ: Rider поддерживает разработку для .NET Framework, .NET Core, Mono и Unity, что делает его универсальным инструментом для разработки как настольных, так и мобильных приложений, игр и веб-приложений.
2. Быстрая и легкая IDE: Rider, как правило, работает быстрее и использует меньше ресурсов по сравнению с Visual Studio, что делает его подходящим выбором для разработчиков с более ограниченными системными ресурсами.
3. Встроенные инструменты рефакторинга и анализа кода: Rider использует возможности ReSharper, предоставляя мощные инструменты для улучшения качества кода, автоматического рефакторинга, быстрого поиска и исправления ошибок. Это помогает повысить продуктивность разработчика.

Недостатки:

1. Коммерческая лицензия: Rider является платным продуктом, и хотя существует бесплатный пробный период, его стоимость может быть значительным фактором для индивидуальных разработчиков или малых компаний.
2. Проблемы с интеграцией: в некоторых случаях Rider может иметь проблемы с интеграцией с определенными инструментами или библиотеками, что может потребовать дополнительных усилий для настройки и исправления ошибок.
3. Зависимость от ресурсов: несмотря на то, что Rider оптимизирован для производительности, он все же требует значительных ресурсов (ОЗУ и процессор), особенно при работе с большими проектами, что может быть проблемой на менее мощных машинах.
4. Visual Studio

Visual Studio — это интегрированная среда разработки (IDE) от Microsoft, предназначенная для создания приложений на различных языках программирования, включая C#, VB.NET, C++, F# и другие. Она поддерживает разработку для платформ .NET, ASP.NET, Azure, а также создание настольных, веб- и мобильных приложений. Visual Studio предлагает множество инструментов и функций для упрощения процесса разработки, от редактирования кода до отладки и тестирования.

Преимущества:

1. Широкая функциональность: Visual Studio предлагает множество встроенных инструментов для разработки, включая редактор кода, отладчик, профайлер, средства для работы с базами данных и интеграцию с системами контроля версий.
2. Интеграция с другими продуктами Microsoft: IDE прекрасно интегрируется с другими решениями Microsoft, такими как Azure, Office и Power BI, что упрощает разработку и развертывание облачных приложений.
3. Поддержка различных языков программирования: Visual Studio поддерживает множество языков, включая C#, VB.NET, C++, F#, JavaScript и Python, что делает её универсальным инструментом для разработки.

Недостатки:

1. Сложность освоения для новичков: из-за обширного функционала и большого количества возможностей, новичкам может потребоваться время для освоения всех аспектов среды.
2. Проблемы с совместимостью: иногда возникают проблемы с совместимостью между различными версиями Visual Studio и проектами, особенно при использовании устаревших технологий.
3. Частые обновления: хотя обновления обычно приносят улучшения, иногда они могут вызывать нестабильность или проблемы с совместимостью, требуя дополнительного времени для исправления.

В результате анализа интегрированных сред разработки для создания программного обеспечения подходит С# для проекта по предоставлению строительных услуг.

Кроме того, встроенные средства для отладки и тестирования помогают выявлять и устранять ошибки на ранних этапах разработки. Это гарантирует высокое качество конечного продукта, что особенно важно в строительной отрасли, где ошибки могут привести к значительным затратам и задержкам.

Интуитивно понятный интерфейс делает процесс разработки более удобным и эффективным, что будет полезно для команд, работающих над проектом. Также стоит отметить возможность расширяемости: Visual Studio поддерживает множество расширений и библиотек, что позволяет легко интегрировать дополнительные функции и инструменты для управления проектами и ресурсами в строительстве.

Широкое сообщество разработчиков и обширная документация обеспечивают поддержку и ресурсы, которые могут быть полезны при решении проблем или внедрении новых функций. Наконец, возможность работы с системами контроля версий (например, Git) упрощает совместную работу нескольких разработчиков над проектом, что делает процесс разработки более организованным.

Таким образом, использование Visual Studio в проекте по предоставлению строительных услуг позволит создать эффективное, надежное и масштабируемое программное обеспечение, соответствующее требованиям вашей компании и клиентов.

## 1.3 Описание предметной области

Предметная область данной курсовой работы охватывает управление магазином по продаже строительных услуг. Основная цель системы — обеспечить эффективное взаимодействие между клиентами и услугами, предлагаемыми магазином, а также управлять внутренними процессами, связанными с предоставлением строительных услуг.

1. Пользователи системы

Система строительных услуг предполагает наличие различных пользователей, каждый из которых имеет свою роль и уровень доступа:

* администраторы: Ответственные за управление пользователями. Администраторы имеют полный доступ к функционалу системы, включая управление проектами, бригадами, ресурсами, а также аналитическими отчетами и финансовыми данными;
* менеджеры: работают непосредственно с клиентами, оформляют их заказы и управляют их выполнением. Менеджеры имеют доступ к информации о клиентах, проектах и рабочих бригадах. Они контролируют процесс выполнения строительных работ и координируют работу между клиентом и бригадами;
* клиенты: могут просматривать каталог строительных услуг, заказывать необходимые работы. Клиенты имеют ограниченный доступ, позволяющий им управлять своими заказами и просматривать их статус;

1. Услуги по проекту

Магазин по предоставлению строительных услуг предлагает широкий спектр строительных и ремонтных услуг. Услуги делятся на категории в зависимости от типа работ, что упрощает выбор для клиентов.

Список услуг может включать:

* проектирование: Разработка чертежей и планов для строительства и ремонта зданий;
* строительные работы: Выполнение строительных работ, включая возведение зданий, ремонт помещений, реконструкцию объектов;
* ремонтные работы: Восстановление жилых и коммерческих помещений, ремонт фасадов, крыши и других частей здания;
* ландшафтные работы: Услуги по благоустройству территорий, создание садов и парков.

Каждая услуга имеет детальное описание, цену и сроки выполнения.

1. Заказы и статусы

Клиенты могут оформлять заказы на строительные услуги. Каждый заказ включает в себя название проекта, услуга, стоимость и сроки выполнения. Заказы также связаны с конкретным клиентом и ответственным менеджером.

Заказы могут иметь различные статусы, такие как:

* «Создан»
* «В обработке»
* «Назначена бригада»
* «Выполняется»
* «Завершен»
* «Отменен»

Эти статусы помогают отслеживать текущее состояние заказа и обеспечивают эффективное управление процессом выполнения работ.

1. Цели и задачи системы

Основные цели системы по предоставлению строительных услуг:

* обеспечение удобного интерфейса для пользователей с возможностью быстрого доступа к информации о строительных услугах;
* автоматизация процессов оформления заказов, управления проектами и назначением бригад;
* обеспечение актуальности информации о предоставляемых услугах и сроках выполнения работ.

Таким образом, данная система призвана оптимизировать управление строительными проектами, улучшить взаимодействие с клиентами и повысить качество услуг, предоставляемых компанией.