Содержание

[1 Общая часть 2](#_Toc210126782)

[1.1 Описание предметной области 2](#_Toc210126783)

[1.2 Анализ методов решения 5](#_Toc210126784)

[1.3 Обзор средств программирования 5](#_Toc210126785)

[1.4 Описание выбранного языка программирования 8](#_Toc210126786)

[2 Специальная часть 11](#_Toc210126787)

[2.1 Постановка задачи 11](#_Toc210126788)

[2.2 Проектирование приложения 11](#_Toc210126789)

[2.3 Текст программы 12](#_Toc210126790)

[2.1 Описание алгоритма 13](#_Toc210126791)

[ПРИЛОЖЕНИЕ А 14](#_Toc210126792)

[ПРИЛОЖЕНИЕ Б 23](#_Toc210126793)

[ПРИЛОЖЕНИЕ В 24](#_Toc210126794)

# Общая часть

Описание предметной области

Производственно-сервисное предприятие «Чистая Планета» оказывает услуги стирки, химической чистки и ремонта одежды и текстильных изделий. Компания занимается оказанием услуг и реализует их через сеть пунктов приема-выдачи (партнеров), которые взаимодействуют с конечными клиентами.

С целью оптимизации деятельности компании разрабатывается система,

в которой заказчик выделил несколько подсистем:

* + - услуги и прайс-лист;
    - склад и расходные материалы;
    - производство (стирка и чистка);
    - сотрудники;
    - партнеры (пункты приема-выдачи).

В данной области определены следующие основные составляющие:

* 1. Партнеры (Пункты приема-выдачи): тип (розничный пункт, корпоративный клиент), наименование, юридический адрес, ИНН, ФИО руководителя, контактные данные (телефон, email), логотип, рейтинг, адреса пунктов, история оказанных услуг для последующего расчета бонусов или скидок.

Партнеры получают комиссию от стоимости заказов, которая зависит от общего объема оказанных услуг через их пункты. Бонусная система мотивирует партнеров привлекать больше клиентов. Партнеры работают с розничными клиентами, корпоративными заказчиками и интернет-агрегаторами.

Взаиморасчеты прачечной с партнерами осуществляются в рублях.

* 1. Менеджеры: поиск и регистрация новых партнеров, решение об изменении рейтинга партнера, ведение истории изменений рейтинга, прием заявок от партнеров, формирование для них коммерческого предложения на

основании статистики истории заказов, полученной от аналитиков компании.

* 1. Заказ (Заявка): если партнер ранее сдавал вещи в чистку и предоставил данные об услугах, то на основании этих данных и текущей загрузки производственного цеха менеджер формирует предложение по срокам и стоимости. Если работа ведется с новым партнером, то менеджер предоставляет прайс-лист и принимает заказ. Затем менеджер должен указать стоимость и дату выполнения каждой указанной в заказе услуги. После согласования списка услуг с количеством, стоимостью и сроком выполнения заказ считается созданным.

Партнер может отменить заказ до момента внесения предоплаты. Если предоплата не поступает в течение трех дней, то заказ отменяется автоматически, о чем обязательно уведомляется партнер. После поступления предоплаты услуги, указанные в заказе, запускаются в работу. По готовности всего заказа менеджер уведомляет партнера о возможности выдачи. В момент получения заказа (путем доставки или самовывоза) партнер производит полную оплату. После проверки качества выполнения услуг партнером заказ считается выполненным.

* 1. Сотрудники: ФИО, дата рождения, паспортные данные,

банковские реквизиты, должность, состояние здоровья (для допуска к работе с химическими веществами).

* 1. Производственный персонал: допуск к работе с конкретным

оборудованием (стиральные машины, машины химчистки, утюги, катки). Технолог производства занимается контролем технологических процессов, качеством выполнения услуг, заказом расходных материалов у поставщиков.

* 1. Доступ: двери производственных и складских помещений открываются при использовании карты сотрудника, система турникетов передает данные о перемещении сотрудников в приложение для формирования аналитиками статистики.
  2. Материалы (расходные средства): тип (моющее средство,

растворитель, отбеливатель, пакеты, теги-ярлыки), наименование, поставщик, количество в упаковке, единица измерения (литры, кг, штуки), описание, изображение, стоимость, количество на складе, минимальный запас, история изменений количества материалов на складе.

* 1. Склад: регистрация поступлений материалов, резервирование

материалов для выполнения заказов, отпуск и списание материалов со склада

в производство, прием готовых заказов от производства и их выдача

партнерам, получение информации о текущих остатках и движении товарно-материальных ценностей по складу.

* 1. Поставщики: тип (химия, упаковка, оборудование), наименование,

ИНН, история поставок материалов. Менеджеры контролируют список

поставщиков, аналитики формируют информацию о качестве поставляемых

материалов для принятия решения о прекращении работы с определенными

компаниями.

* 1. Услуги: код услуги, тип (стирка, химчистка, ремонт),

наименование, описание, изображение (например, иконка), минимальная

стоимость для партнера, норма времени на выполнение, себестоимость

(расчетная), номер цеха, количество персонала на выполнение, необходимые

материалы для оказания услуги (нормы расхода).

## Анализ методов решения

Первый и основной вопрос, который требует наибольшего внимания это выбор основы технологии, которая будет реализовывать БД. Выбор по своей сути стоит между SQL БД и NoSQL БД.

SQL реализация будет заключаться в разработки большого количества отношений и их связей. NoSQL реализация может иметь множество других форм реализации, но наиболее интересная и подходящей будет документо-ориентированная, позволяющая организовывать свою структуру файлов-документов и их связей.

Выбор будет сделан в пользу SQL (то есть реляционной модели), поскольку пользование в системе предполагает большое количество не регулярно изменчивых по структуре данных, то их хранение наиболее оптимально будет организовать в отношениях SQL баз данных.

Преимущество SQL баз данных заключаются в хранении предсказуемых, структурированных данных, где важны надежность, точность и соблюдение целостности, в то время как преимущества документо-ориентированных БД это возможность хранения и обработки типизированных, но при этом иногда различающихся по структуре данных.

Поскольку разработанная система не предполагает регулярного обновления приложения и форматов хранимых данных, то и преимущества документо-ориентированных БД несущественны для данного проекта. Таким образом решение будет реализовано средствами SQL.

## Обзор средств программирования

Перед реализацией информационной системы следует сделать два основных выбора: СУБД и технологии для реализации приложения.

СУБД существует великое множество, но выбор СУБД ограничим 4-мя популярными: MySQL, PostgreSQL, Microsoft SQL Server, SQLite.

Выбор будет сделан в пользу PostgreSQL, поскольку:

* Microsoft SQL Server это платный лицензионный продукт, который обладает высоким разнообразием функционала, который в текущем проекте не востребован.
* SQLite не подойдёт, поскольку требуется работа с сервером и скорее всего подразумевается многопользовательский доступ, который SQLite не обеспечивает.
* В свою очередь выбор между PostgreSQL и MySQL почти не играет роли, но PostgreSQL имеет свободную лицензию MIT, в то время как MySQL распространяется под двойной лицензией и GPL/commercial, что означает уверенность в том, что ПО не станет платным. Кроме того PostgreSQL универсален и с его помощью можно настроить БД под почти любые задачи. Такой выбор подтверждается статистикой показанной ниже. (рисунок 1)

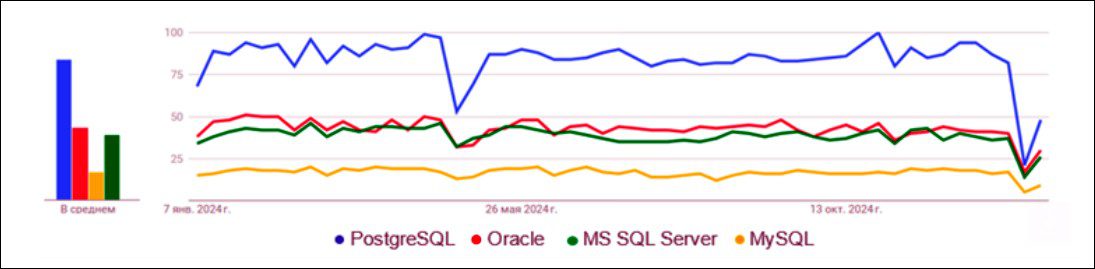


Рисунок 1 - Динамика популярности СУБД в России в 2024 г., Google Trends.

Выбор технологии реализации приложения состоит из выбора языка программирования, а также выбора фреймворка для работы с базой данных.

Возможные языки программирования: C++, C#, python, Java.

C++ — это компилируемый, статически типизированный язык программирования общего назначения, разработанный как расширение языка C, сочетающий низкоуровневый контроль с поддержкой объектно-ориентированного и обобщенного программирования.

C# — это компилируемый статически типизированный язык программирования, поддерживает автоматическую сборку мусора, полиморфизм и обобщённые типы. Ключевые особенности — его универсальность для разработки веб-приложений (ASP.NET), игр (Unity), десктопных и мобильных приложений, а также тесная интеграция с платформой .NET.

Python – это высокоуровневый, интерпретируемый язык программирования общего назначения с простым синтаксисом, поддерживающий несколько парадигм программирования (ООП, процедурное, функциональное).

Java – это многоплатформенный (кроссплатформенный), объектно-ориентированный язык программирования общего назначения, который позволяет создавать приложения для различных платформ, таких как серверы, компьютеры и мобильные устройства.

Среди этих языков наибольшим преимуществом обладает C#, поскольку работа с C# почти всегда подразумевает работу с платформой .Net, на которой доступны к применению широкий спектр библиотек для работы с SQL БД, в частности ORM библиотек. Именно ORM становится главным фактором выбора средств разработки и c++ не подходит для этих целей, поскольку сам язык не ориентирован в своей основе на ООП. Python в свою очередь ориентирован на ООП, но зачастую имеет проблемы с производительностью, которые в свою очередь решают тем, что прибегают к работе с более низкоуровневыми программирование на том же C++. Java в свою очередь почти полностью аналогичен по своей философии языку программирования C#, однако выбор в пользу C# сделан, поскольку на C# присутствует Entity Framework – фреймворк для реализации ORM, и хотя на Java тоже есть свои ORM фреймворки, Entity Framework разрабатывается непосредственно той же организацией, которой и разрабатывается весь язык C#, что означает наиболее приспособленную работу Entity Framework с средствами языка C#, по сравнению с фреймворками на Java, разрабатываемыми сторонними разработчиками.

## Описание выбранного языка программирования

* + 1. Общие сведения

C# — это язык программирования, созданный Microsoft для платформы .NET. Он сочетает подход объектно-ориентированного программирования с элементами функционального программирования, обеспечивая безопасность, высокую производительность и удобную интеграцию с внешними сервисами и библиотеками. Среди ключевых особенностей C# — строгая типизация, автоматическое управление памятью (сборщик мусора) и встроенные механизмы для работы с многопоточностью.

* + 1. Элементы языка
* Переменные – именованные участки памяти, хранящие данные для работы программы, строго типизированы;
* Типы данных: определённые структуры данных (не путать с типом данных struct), позволяющая ограничить то, как хранимая двоичная информация будет интерпретирована. Самые широко используемые типы данных это int – целочисленные, double – вещественные числа, string – строковые, пользовательские классы и структуры, записи – сложные типы данных, описываемые пользователем, bool – булево, а также enum – перечисления;
* Операторы: обеспечивают выполнение различных операций, язык поддерживает перегрузку операторов, позволяющий пользователю реализовать собственную логику для операторов. Самые распространённые используемые операторы обознаются следующим образом: +, -, \*, /, ^, &, &&, |, ||, >, <, !, !=, ==, <=, >=;
* Управляющие конструкции: особые языковые конструкции позволяющие управлять потоком программы, например ветвление (операторы if, switch), повторение или циклы (do while, while, for, foreach), операторы перехода (go to).
  + 1. Способы структурирования программы

Приложение заключено в сборку, в которой хранятся классы, в которых описываются методы, поля и свойства. Поток программы начинается в методе Main() класса Program.

Фундаментальным средством разграничения структурирования контекста программы является понятие пространства имён (namespace), в коде эти пространства имён можно распознать по фигурным скобкам, обозначающие блоки программы. Пространства имён позволяет ограничить контекст, то есть область видимости переменных в пределах пространства имён. Таким образом при использовании одно и того же имени переменной в разных блоках программы происходить коллизии, то есть обращения к одному и тому же участку памяти, происходить не будет.

Методы содержат в себе какую-то ограниченную логику, позволяющие разработчику использовать ранее написанные методы как самостоятельные дискретные логические единицы и позволяющие редуцировать повторное копирование и использование одного и того же участка кода.

Сборки (Assembly) представляют из себя главные единицы развертывания приложения, хранят в себе информацию обо всех пользовательских типах и их методах.

* + 1. Встроенные элементы

C# на платформе .Net или .Net Framework содержит большое разнообразие встроенных элементов:

* системные классы: классы из пространства имен System, например, LINQ, DateTime, Math, которые предоставляют технологии для работы с различными типами данных;
* обработчики событий: позволяют запускать определённые методы переданные через делегат по срабатыванию каких-либо событий;
* исключения: использование конструкций try catch обеспечивают механизм обработки ошибок, позволяя разработчикам управлять приложением при возникновении таковых.
  + 1. Средства отладки

C# тесно интегрирован со средой разработки Visual Studio, поэтому основные средства отладки следующие:

* точки останова. Конкретные марки, при достижения которых потоком исполнение программы останавливается в режиме отладки и позволяет в последствии запустить в обычном режиме или пошаговом;
* окно переменных. Если написать отслеживаемые значения на специальной панели или при наведении на соответствующую переменную её закрепить, то можно отследить конкретные значения;
* стек вызовов. Позволяет отслеживать последовательность вызовов методов, что помогает в поиске ошибок;
* отладка пошагово. Позволяет поочередно выполнять строки кода и наблюдать за каждым изменением отслеживаемого значения или убедиться в корректности ветвления.

# Специальная часть

Постановка задачи

## Проектирование приложения

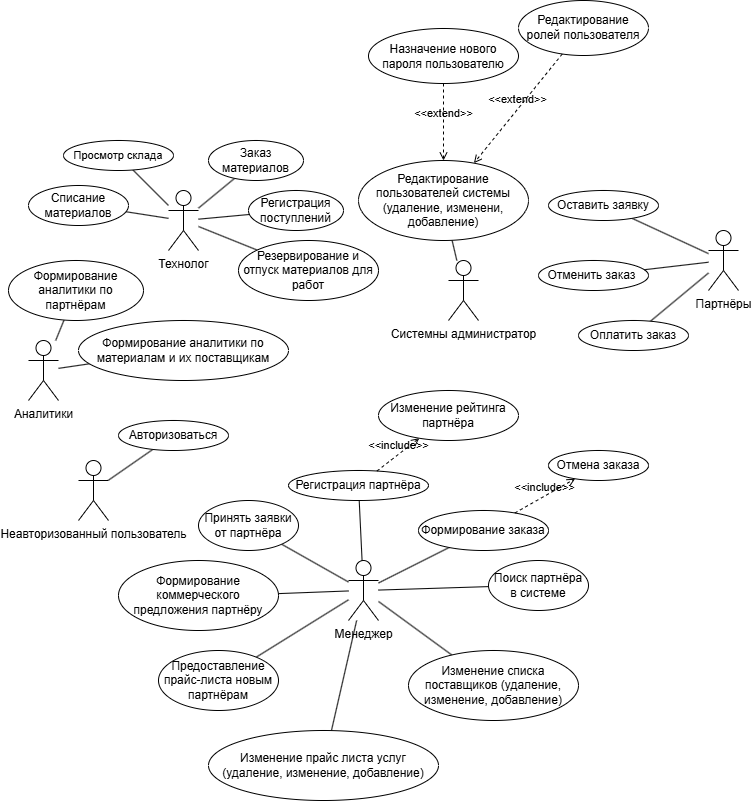


Рисунок – UML диаграмма прецендентов использования системы.

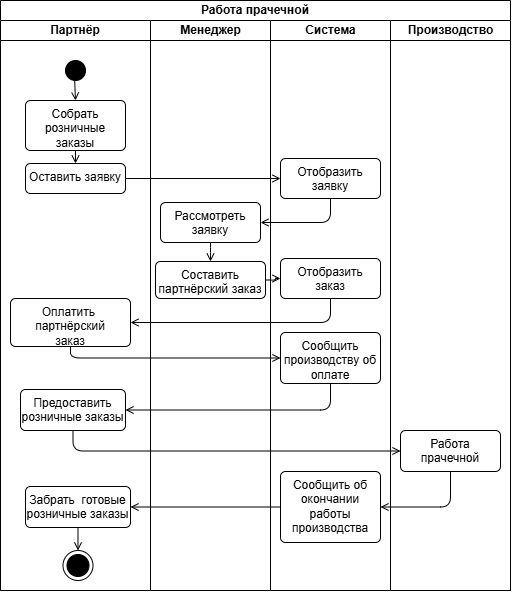


Рисунок – Диаграмма активности работы прачечной

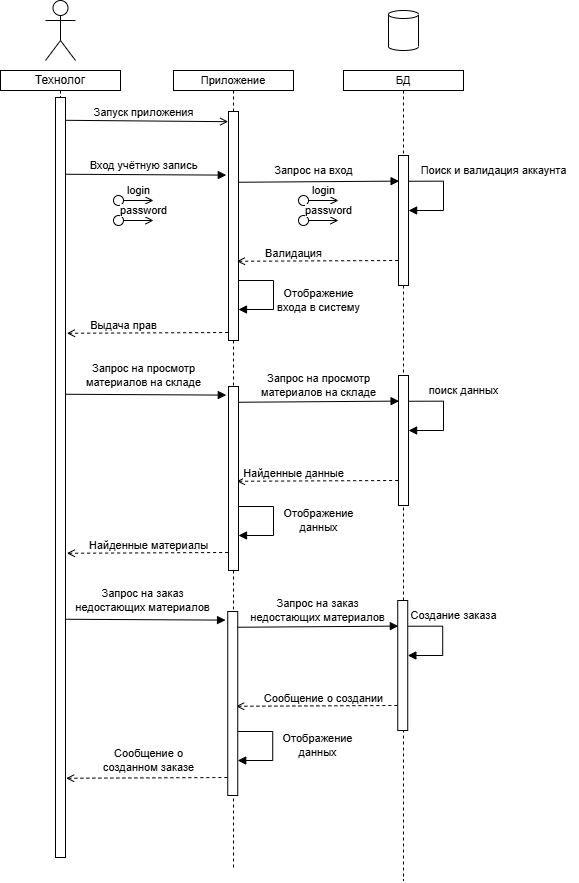


Рисунок – Диаграмма последовательности заказа материалов Технологом

## Текст программы

Текст программы основного игрового модуля представлен в приложении В.

## Описание алгоритма

Для обеспечения безопасности хранимых паролей их необходимо шифровать, для этого будет использоваться функция HashPassword(), представленная на блок-схеме ниже. (рисунок 5)

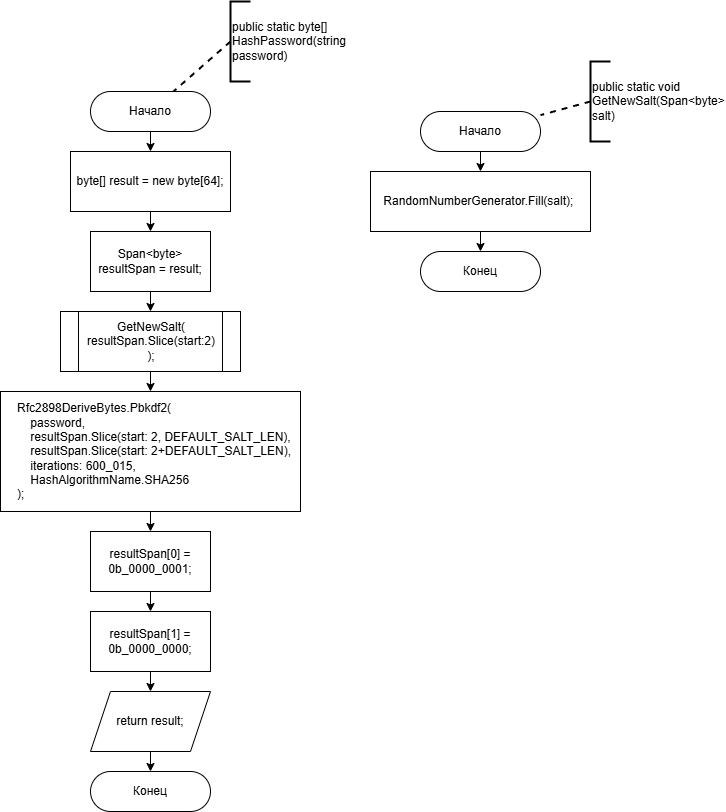


Рисунок – Блок-Схема алгоритма хэширования пароля

ПРИЛОЖЕНИЕ А

(обязательное)

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский государственный политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Институт среднего профессионального образования

Информационная система для производственно-сервисного предприятия “Чистая Планета”

**Техническое задание**

Листов \_\_7\_\_

ВЫПОЛНИЛ

Студент группы 22290907/1097

\_\_\_\_\_\_\_\_\_\_ Моисеев Е.В.

\_\_\_\_.\_\_\_\_\_\_\_\_\_. 2025

2025

1. **ВВЕДЕНИЕ**
   1. Полное наименование программной разработки: Информационная система для производственно-сервисного предприятия «Чистая Планета».
   2. Информационная система для производственно-сервисного предприятия «Чистая Планета» представляет из себя защищённую базу данных и интегрированное с базой данных приложение.
   3. Программа предназначена для доступа в компании сотрудникам с целью реализации их бизнес-задач.
2. **ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ**

Разработка ведётся на основании задания к курсовому проекту по профессиональному модулю ПМ 02 «Осуществление интеграции программных модулей» и утверждена Председателем предметно-цикловой комиссии отделения информационных технологий ФГАОУ ВО «СПбПУ» Института среднего профессионального образования.

1. **НАЗНАЧЕНИЕ РАЗРАБОТКИ**
   1. Основное назначение программного продукта: оптимизация деятельности компании-заказчика.
   2. Эксплуатационное назначение программного продукта: программа предназначена для узкого круга пользователей - сотрудников компании “Чистая планета”.
2. **ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ**
   1. Требования к функциональным характеристикам:

* ИС должна осуществлять хранение следующих сущностей: услуги и прайс-лист, склад и расходные материалы, производство (стирка и чистка), сотрудники, партнеры (пункты приема-выдачи);
* Данные о партнёре включают в себя: тип (розничный пункт, корпоративный клиент), наименование, юридический адрес, ИНН, ФИО руководителя, контактные данные (телефон, email), логотип, рейтинг, адреса пунктов, история оказанных услуг для последующего расчета бонусов или скидок;
* ИС должна реализовывать бонусную систему, при которой партнёры получают комиссию от стоимости заказов;
* Все заказы в ИС должны осуществляться в рублях;
* ИС должна реализовывать роль менеджера с соответствующему этой роли функционалу: поиск и регистрация новых партнеров, изменение рейтинга партнера, ведение истории изменений рейтинга, прием заявок от партнеров, формирование для них коммерческого предложения, формирование предложения партнёру с указанием срока и стоимости оказываемых услуг, предоставляет прайс-лист новым партнёрам;
* Заказ формируется менеджером, оформленный заказ включает в себя все согласованные услуги и даты выполнения к каждой услуге, стоимость к каждой услуге;
* ИС должна обеспечивать возможность отмены заказа до внесения предоплаты в течение 3-ёх дней;
* Система должна обеспечить автоматическую отмену заказа, если он не оплачен в течение 3-ёх дней и уведомление об отмене партнёру;
* ИС хранит следующие данные о сотруднике: дата рождения, паспортные данные, банковские реквизиты, должность, состояние здоровья (для допуска к работе с химическими веществами);
* ИС должна обеспечивать для производственного персонала разграничение допуска к работе с конкретным оборудованием;
* ИС должна обеспечивать формирование статистики для аналитиков о перемещении сотрудников при использовании их карт сотрудников с системой турникетов;
* Пользователям с ролью Технолога должен быть обеспечен доступ к заказам материалов (расходных средств);
* ИС хранит следующую информацию о материалах (расходных средствах): тип (моющее средство, растворитель, отбеливатель, пакеты, теги-ярлыки), наименование, поставщик, количество в упаковке, единица измерения (литры, кг, штуки), описание, изображение, стоимость, количество на складе, минимальный запас, история изменений количества материалов на складе;
* ИС должна организовывать работу склада и реализовывать следующие функции при работе с материалами: регистрация поступлений материалов, резервирование материалов для выполнения заказов, отпуск и списание материалов со склада в производство, прием готовых заказов от производства и их выдача партнерам, получение информации о текущих остатках и движении товарно-материальных ценностей по складу;
* ИС должна хранить следующие сведения о поставщике: тип (химия, упаковка, оборудование), наименование, ИНН, история поставок материалов;
* ИС должна обеспечивать доступ менеджерам по изменению списка поставщиков (добавление, удаление, изменение);
* ИС хранит следующие сведения об услуге: код услуги, тип (стирка, химчистка, ремонт), наименование, описание, изображение (например, иконка), минимальная стоимость для партнера, норма времени на выполнение, себестоимость (расчетная), номер цеха, количество персонала на выполнение, необходимые материалы для оказания услуги (нормы расхода);
  1. Требования к надежности:
* Использование лицензированного программного обеспечения;
* Проверка программы на наличие вирусов;
* ИС должна реализовывать обработку исключительных ситуаций в приложении;
* ИС должна уведомлять пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, предупреждать о неотвратимых операциях. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления;
* Организация бесперебойного питания.
  1. Требования к составу и параметрам технических средств

Для нормального функционирования данной информационной системы необходим компьютер, клавиатура, мышь и следующие технические средства:

- процессор Intel или другой совместимый с архитектурой х86, 64 битный;

- объем свободной оперативной памяти ~2 Гб;

- объем необходимой памяти на жестком диске ~150Гб;

- стандартный VGA-монитор или совместимый;

- стандартная клавиатура;

- манипулятор «мышь».

* 1. Требования к информационной и программной совместимости

Для полноценного функционирования данной системы необходимо наличие операционной системы новее Microsoft Windows 10 или совместимой. Язык интерфейса – русский.

* 1. Для работы системы требуются пакеты .Net Framework.
  2. Требования к маркировке и упаковке

Программа должна поставляться в виде: исходного кода приложения, исполняемых файлов, файлов скрипта БД и прочих графических, текстовых файлов.

* 1. Требования к транспортировке и хранению

Необходимо загрузить исходный код метода расчета материала в отдельный репозиторий с названием, совпадающим с названием проекта. Все результаты должны быть переданы путем загрузки файлов на предоставленный репозиторий системы контроля версий.

* 1. Специальные требования

Теоретическая часть включает подробное описание работы с приложением, разработанные на этапе проектирования программной системы. Практическая часть включает разработку и реализацию программных модулей программного продукта с использованием среды программирования.

1. **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

Технико-экономические показатели не рассчитываются.

1. **СТАДИИ И ЭТАПЫ РАЗРАБОТКИ**

В таблице А.1 представлены стадии и этапы разработки.

Таблица А.1 – Стадии и этапы разработки

| **Содержание стадии** | **Содержание этапа** | **Срок 2025 г.** | | **Форма отчетности** |
| --- | --- | --- | --- | --- |
| **начало** | **конец** |
| Техническое задание | Составление технического задания | 29.09.2025 | 30.09.2025 | Техническое задание |
| Эскизный проект | Проектирование программы | 01.10.2025 | 03.10.2025 | UML-диаграммы |
| Рабочий проект | Разработка спецификаций | 04.10.2025 | 06.10.2025 | Спецификации отдельных компонентов |
| Составление программы | 06.10.2025 | 08.10.2025 | Программная документация |
| Приёмо-сдаточные испытания | 09.10.2025 | | Протокол испытаний |
| Приёмка | Защита результата учебной практики | 10.10.2025 | | Оценка за учебную практику |

1. **ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**
   1. Порядок контроля

Контроль выполнения должен осуществляться руководителем учебной практики (преподавателем) в соответствие с п.7.

* 1. Порядок приемки

Приемка должна осуществляться с участием руководителя после проведения приемо-сдаточных испытаний. В результате защиты учебной практики должна быть выставлена оценка за курсовое проектирование.

ПРИЛОЖЕНИЕ Б

(обязательное)

**Руководство по стилю**

Все экранные формы пользовательского интерфейса должны иметь заголовок. Кроме того, на главной форме должен быть установлен логотип (представлен в ресурсах). Логотип не искажать: не менять изображение, пропорции изображения, цвет.

Для приложения должна быть установлена иконка (дана в ресурсах), если это реализуемо в рамках платформы.

Использовать шрифт Franklin Gothic Medium.

В качестве основного фона используется белый цвет, дополнительного фона используется цвет #E0FFFF. Для акцентирования внимания пользователя на целевом действии интерфейса используется цвет #00CED1.

В таблице Б.1 представлены примеры исползуемых цветов.

Таблица Б.1 – Примеры импользуемых цветов

|  |  |  |
| --- | --- | --- |
| Основной фон | Дополнительный фон | Акцентирование внимания |
|  | #E0FFFF | #00CED1 |
|  |  |  |

ПРИЛОЖЕНИЕ В

(обязательное)

**Листинг кода основного модуля приложения**