**Содержание**

[Введение 2](#_Toc449196913)

[1 Аналитическая часть 3](#_Toc449196914)

[1.1 Описание приложения 3](#_Toc449196915)

[1.2 Анализ требований 3](#_Toc449196916)

[1.3 Предварительная оценка трудоемкости разработки 3](#_Toc449196917)

[2 Технологическая часть 4](#_Toc449196918)

[2.1 Описание проектирования логики и данных 4](#_Toc449196919)

[2.2 Определение функциональных типов по данным 4](#_Toc449196920)

[2.3 Описание проекта интерфейса приложения 4](#_Toc449196921)

[2.4 Определение транзакционных функциональных типов 4](#_Toc449196922)

[2.5 Расчет количества функциональных точек 4](#_Toc449196923)

[2.6 Определение основных технико-экономических показателей 5](#_Toc449196924)

[Заключение 6](#_Toc449196925)

[Использованные источники информации 7](#_Toc449196926)

**Введение**

Голосовая обработка текста становиться все более важной в нашем современном мире .Мы видим ее использование в различных сферах,начиная от домашних устройств,закончивая медицинскими исследованиями.моя работа состоит в том чтобы рассмотреть принцыпы и технологии,используемые в голосовом обработчике текста,ии оценить их воздействие на наше повседневное общение и взаимодействие

В ходе данного курсового проекта была создана программа по обработке текса(нейро блокнот).Данное приложение будет собирать голосовые данные в течении дня и конвертировать их в текстовые данные .Будет классифицировать их по ключевым словам и сохранять ее на облаке

Аналитическая часть

Голосовой обработчик текста — это важный инструмент, который поможет людям с огранничеснными возможностями, а также тем, кому неудобно писать текст на клавиатуре, выражать свои мысли и коммуницировать. Каждый человек имеет право на свободу слова и доступ к информации.

Голосовой обработчик создает разные возможности для всех, улучшая качество жизни и обеспечивая доступ к образованию, работе и общению

Основанием для разработки проекта является учебный план специальности 09.02.07 Информационные системы и программирование (квалификация выпускника - Специалист по информационным ресурсам).

* 1. **Функции программного продукта**

Описание предметной области, цель и задачи ПП

Описание

***Цель-*** помочь людям не терять интересные мысли, идеи и слова в их жизни. Голосовой обработчик смог бы их записать и просортировать по ключевым словам, и категориям занятий

**Задачи ПП**

1. Собирать голосовые данные в течении дня
2. Конвертировать их в текстовый формат
3. Классифицировать их, по ключевым словам, и категориям

# Сохранять данные на облако либо в telegramАналитическая часть

1. **Описание приложения**

Задачей курсового проекта является разработка и создание клиентского приложения. Данная система должна состоять из:

* программы, осуществляющей операции с базой данных, включая запись и чтение.

Целью данного курсового проекта является создание голосового обработчика текста.

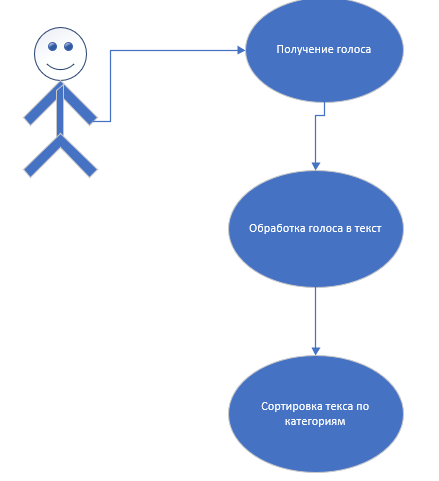
1. **Анализ требований**

Система должна обеспечивать следующие функции:

1. Запись голоса.
2. Конвертация голоса в текст
3. Обработка текста по категориям
4. Выполнение обработки текста

Основные требования, предъявляемые к системе:

В ходе анализа требований была создана диаграмма прецедентов, отображенная на рисунке 1.



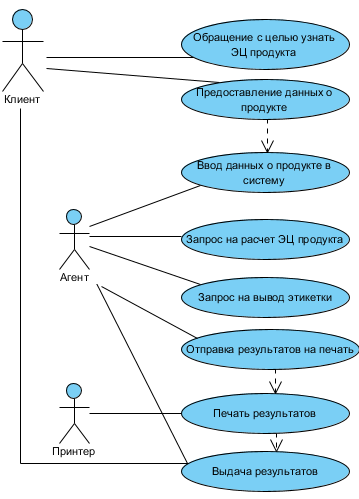
**

Рисунок 1 - Диаграмма прецедентов

1. **Предварительная оценка трудоемкости разработки**

На основании перечня функций, выявленных на этапе анализа требований была выполнена экспертная оценка трудоемкости разработки приложения, с учетом его функций, представленная в таблице 1.

Таблица 1 - Экспертная оценка трудоемкости разработки приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функция** | **Лучший случай** | **Наиболее вероятный** | **Худший случай** | **Ожидаемый случай** |
| Разработка базы данных | 3 | 6 | 9 | 22,67 |
| Создание хранимых процедур | 8 | 9 | 10 | *6* |
| Разработка интерфейса | 10 | 24 | 30 | *9* |
| Документация | 16 | 30 | 31 | 27,83 |
| **Итого** | **37** | **69** | **80** | **65,5** |

1. **Технологическая часть**
2. **Описание проектирования логики и данных**

По результатам проектирования была составлена диаграмма классов проекта, представленная на рисунке 2.

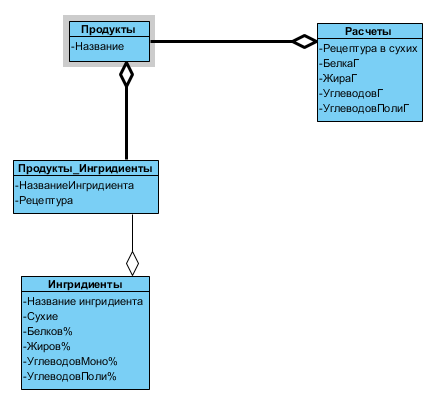


Рисунок 2 - Диаграмма классов

Следующим шагом проектирования является составление ER-диаграммы, показанной на рисунке 3. Диаграмма отображает состав таблиц базы данных проекта, а также связи между ними.

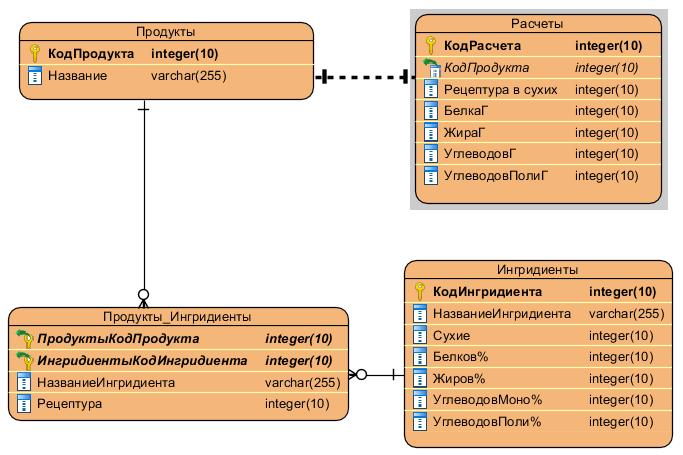


Рисунок 3 - ER Диаграмма

Для оценки действий актера и отклика системы был создан основной успешный сценарий, представленный на рисунке 4, который помогает понять взаимодействие между сотрудником и системой.

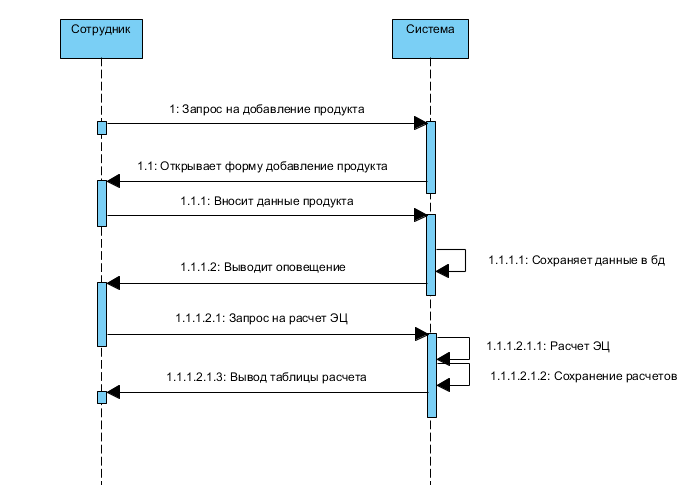


Рисунок 4 - Основной успешный сценарий

1. **Определение функциональных типов по данным**

На основании ER-диаграммы выполнили оценку функциональных типов по данным. Сначала определили тип данных системы: EIF или ILF. Затем произвели подсчет функциональных точек, связанных с данными, определили сложность данных. Полученный результат представлен в таблице 2.

Таблица 2 - Оценка функциональных типов по данным

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Информационный объект** | **Тип** | **RET** | **DET** | **Сложность** | **Коэффициент** |
| Продукты | ILF | 1 | 2 | Low | 7 |
| ИнгредиентыПродукта | ILF | 1 | 1 | Low | 7 |
| Ингредиенты | ILF | 1 | 7 | Low | 7 |
| Расчеты(таблица) | ILF | 1 | 7 | Low | 7 |
| Расчеты | EIF | 1 | 6 | Low | 5 |

Коэффициент определяется в зависимости от принадлежности объекта к ILF или EIF.

1. **Описание проекта интерфейса приложения**

Эскизы форм приложения были разработаны в программе MS Visio. На рисунке 5 представлен макет формы «Главная», который содержит таблицу «Продукты», кнопки для добавления, изменения, удаления продукта, кнопки для вывода форм расчета и этикетки, а также кнопка выхода из приложения.

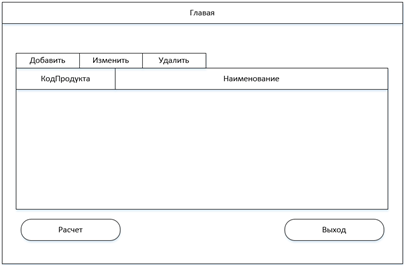


Рисунок 5 - Макет формы "Главная"

На рисунке 6 представлен макет формы «ДобавлениеПродукта», который содержит поле для ввода названия продукта, а также кнопку для сохранения введённых данных.

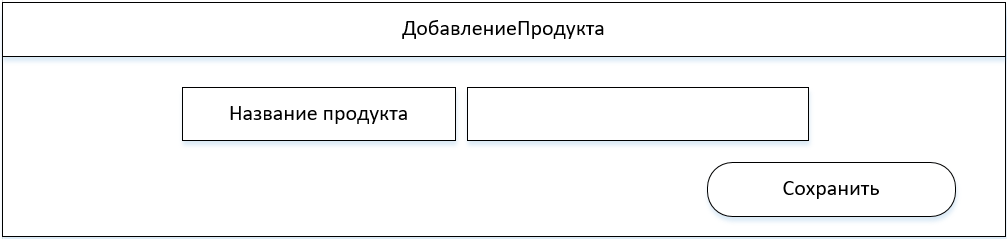


Рисунок 6 - Макет формы "ДобавлениеПродукта"

На рисунке 7 представлен макет формы «ИзменениеПродукта», который имеет поле, где отображается название продукта, таблица с ингредиентами продукта, кнопки для добавления и удаления ингредиента, добавления рецептуры, а также кнопки подтверждения и отмены действия.

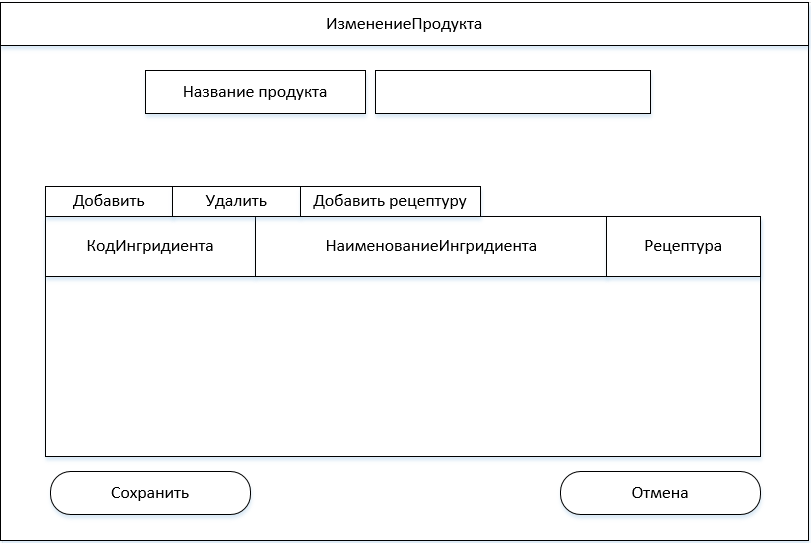


Рисунок 7 - Макет формы "Изменение продукта"

1. **Схема перемещений по приложению**

Схема перемещений по приложению, которая представлена на рисунке 7, представляет собой переходы с формы «Главная» к 3 другим формам: «ИзменениеПродукта», «ДобавлениеПродукта» и «Расчеты». С формы «ИзменениеПродукта» можно перейти к формам «Ингредиент» и «ДобавлениеРецептуры», а с формы «Ингредиент» к форме «ДобавлениеИнгредиента».

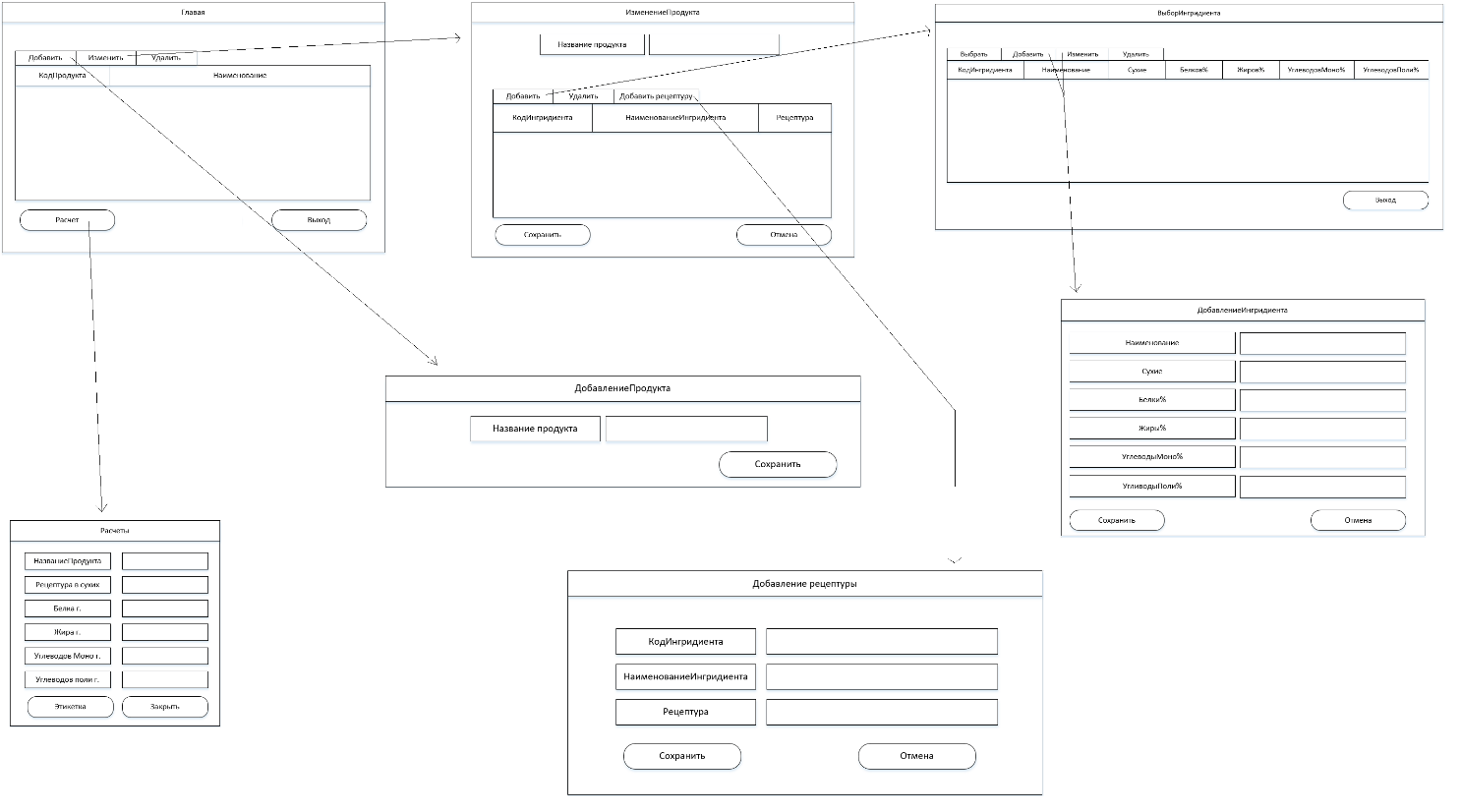


Рисунок 8 - Схема перемещений

1. **Определение транзакционных функциональных типов**

Следующий шаг анализа по методу функциональных точек производился для каждой формы приложения. Были определены типы транзакций, выполнена оценка сложности транзакций, а также определение сложности и расчет коэффициента. Все расчеты представлены в таблицах 3-8.

Таблица3 - Форма Главная

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Вывод продуктов | EО | 1 | 1 | Простая | 4 |
| Добавление продукта | EI | 2 | 1 | Простая | 3 |
| Изменение продукта | EI | 2 | 1 | Простая | 3 |
| Удаление продукта | EI | 1 | 1 | Простая | 3 |
| Открытие формы Расчет | EI | 1 | 1 | Простая | 3 |

Таблица 4 - Форма Продукт

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Ввод названия продукта | EI | 1 | 1 | Простая | 3 |
| Добавление ингредиентов продукта | EI | 3 | 1 | Простая | 3 |
| Удаление ингредиента продукта | EI | 1 | 1 | Простая | 3 |
| Вывод ингредиентов продукта | EО | 1 | 1 | Простая | 4 |
| Добавление рецептуры | EI | 2 | 1 | Простая | 3 |
| Запись продукта | EI | 1 | 2 | Простая | 3 |

Таблица 5 - Форма Ингредиент

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Вывод ингредиентов | EО | 1 | 1 | Простая | 4 |
| Выбор ингредиента | EI | 1 | 1 | Простая | 3 |
| Добавление ингредиента | EI | 2 | 6 | Средняя | 4 |
| Изменение ингредиента | EI | 2 | 6 | Средняя | 4 |
| Удаление ингредиента | EI | 1 | 1 | Простая | 3 |

Таблица 6 - Форма Добавление/Изменение ингредиента

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Ввод наименования | EI | 1 | 1 | Простая | 3 |
| Ввод сухих | EI | 1 | 1 | Простая | 3 |
| Ввод Белков% | EI | 1 | 1 | Простая | 3 |
| Ввод Жиров% | EI | 1 | 1 | Простая | 3 |
| Ввод углеводов Моно% | EI | 1 | 1 | Простая | 3 |
| Ввод углеводов Поли% | EI | 1 | 1 | Простая | 3 |
| Запись ингредиента | EI | 1 | 1 | Простая | 3 |

Таблица 7 - Форма Рецептура

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Ввод рецептуры | EI | 1 | 1 | Простая | 3 |
| Запись рецептуры | EI | 1 | 1 | Простая | 3 |

Таблица 8 - Форма Расчет

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Транзакция** | **Тип** | **FTR** | **DET** | **Сложность** | **Коэффициент** |
| Вывод расчетов | EО | 1 | 1 | Простая | 3 |

1. **Расчет количества функциональных точек**

В таблице 9 представлен анализ функциональных точек - стандартный метод измерения размера программного продукта с точки зрения пользователей системы. Метод предназначен для оценки на основе логической модели объема программного продукта количеством функционала.

Таблица 9 - Сводная таблица определения количества функциональных точек

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Категория функциональных типов** | **Простые** | **Средние** | **Сложные** | **Кол-во точек** |
| Количество внутренних логических файлов | 4\*7 | 0 | 0 | 28 |
| Количество внешних интерфейсных файлов | 2 |  |  |  |
| Количество входных элементов | 20\*3 | 2\*4 | 0 | 68 |
| Количество выходных элементов | 4\*2 | 0 | 0 | 8 |
| Количество внешних запросов | 0 | 0 | 0 | 0 |
| **Количество функциональных точек (UFP)** | | | | 104 |

Помимо функциональных требований на продукт накладываются общесистемные требования, которые ограничивают разработчиков в выборе решения и увеличивают сложность разработки. Для учета этой сложности применяется фактор выравнивания, представленный в таблице 10.

Таблица 10 - Факторы выравнивания

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование фактора** | **Значение** |
| **1** | Обмен данными | 2 |
| 2 | Распределенная обработка данных | 2 |
| 3 | Производительность | 3 |
| 4 | Ограничения по аппаратным ресурсам | 0 |
| 5 | Транзакционная нагрузка | 1 |
| 6 | Интенсивность взаимодействия с пользователем | 2 |
| 7 | Эргономика | 1 |
| 8 | Интенсивность изменения данных (ILF) пользователями | 3 |
| 9 | Сложность обработки | 1 |
| 10 | Повторное использование | 5 |
| 11 | Удобство инсталляции | 1 |
| 12 | Удобство администрирования | 1 |
| 13 | Портируемость | 5 |
| 14 | Гибкость | 2 |
|  | **Суммарное значение коэффициентов (М)** | 29 |

Расчет значения фактора выравнивания

VAF = (0,65 + (29 \* 0,01)) = 0,94

Расчет количества выровненных функциональных точек

AFP = 104 \* 0,94 = 97.76

1. **Определение основных технико-экономических показателей**

Расчет количества функциональных точек

F=104

Показатель LOC для языка C#

LOC=53

Для определения технико-экономических показателей необходимо определить тип системы. Данная система относится ко второму типу, она реализована на языке C#.

После расчета количества функциональных точек нужно определить размерность программного обеспечения. Размерность для конкретного языка программирования определяется с учетом нормативов, по формуле:

(1)

R = f × LOC

Размерность программного обеспечения рассчитана по формуле (1)

R= 104\*53=5 512

Значения параметров A и E определяются из таблицы коэффициентов математической модели оценки трудозатрат в зависимости от типа программной системы.

Оценка трудозатрат проводится с помощью степенной функции базовой модели COCOMO. Значения параметров *A* и *E* определяются из таблицы коэффициентов математической модели оценки трудозатрат в зависимости от типа программной системы.

Единица измерения R в данной формуле - тысяча строк.

(2)

Расчет количества человек в месяц произведен по формуле (2)

Средняя численность сотрудников, занятых в проекте, срок реализации которого 4 месяца, составляет

(3)

Средняя численность сотрудников рассчитана по формуле (3)

Таким образом, метод функциональных точек определил следующие основные технико-экономические показатели:

* трудозатраты на разработку системы за 4 месяца составят 5 человеко-месяцев;
* необходимые людские ресурсы при реализации системы за 4 месяца составят 2 чел.

1. **Заключение**

До начала реальной части проекта произвели расчет трудоемкости работы. Было выявлено какое количество человеко-часов необходимо для разработки проекта.

В результате работы была создана логическая модель базы данных в среде Visual Paradigm 15.1 и реализована в Microsoft SQL Server 2017. Разработан пользовательский интерфейс с формами для работы с данными. Информационная система, состоящая из:

* базы данных, разработанной в MS SQL Server 2017.
* клиентской части, разработанной средствами MS Visual Studio 2017.

В программе предусмотрены возможности:

* записи данных;
* удаления данных;
* редактирования данных;

В ходе работы закреплена технология создания форм и запросов в MS SQL Server 2017, навыки использования языков SQL и C#, и технологию для создания интерфейса пользователя – Entity. Были изучены навыки расчета необходимого времени и количества человек на разработку приложения.

1. **Использованные источники информации**
   1. ГОСТ Р ИСО 9000-2008 «Системы менеджмента качества»;
   2. ГОСТ Р ИСО/МЭК 12207-99 «Процессы жизненного цикла программных средств»;
   3. Голощапов Алексей Леонидович. MICROSOFT VISUAL STUDIO 2010 - 2-е изд.,перераб.и доп. - СПб.: БХВ-Петербург, 2012. - 448с.
   4. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения: Учебник. 4-у изд. Стандарт третьего поколения. – СПб.: Питер, 2012. – 608 с.
   5. Панюкова Т. А. Документирование программного обеспечения: В помощь техническому писателю: Учебное пособие. - М.: Книжный дом «ЛИБРОКОМ», 2012. - 264 с.
   6. Панюкова Т. А. Проектирование программное обеспечение: В помощь техническому писателю: Учебное пособие. - М.: Книжный дом «ЛИБРОКОМ», 2012. - 362 с.
   7. Фримен Адам. ASP.NET 4.5 с примерами на C# 5.0 для профессионалов– Москва: Вильямс, 2014. - 1056c.