**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 создать программу по преобразованию голоса в текст(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
4. Сохранять данные на облакоАналитическая часть
5. **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.

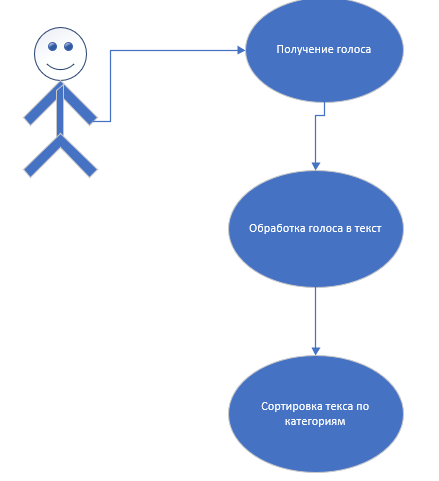
1. **Анализ требований**

Система должна обеспечивать следующие функции:

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

Основные требования, предъявляемые к системе:

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



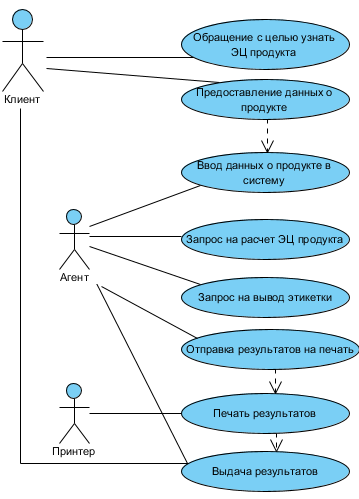
**

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 текстовой базы данных | 3 | 6 | 9 | 22,67 |
| Создание хранимых процедур | 8 | 9 | 10 | *6* |
| Сохранение данных на облако | 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**

* Выбор технологий и инструментов: Для разработки проекта будут использоваться следующие технологии и инструменты:
* Язык программирования Python, так как он является одним из самых популярных и простых в изучении языков программирования;
* Библиотека Keras для создания нейронных сетей, поскольку она предоставляет удобный интерфейс для создания и обучения моделей;
* Фреймворк TensorFlow для работы с нейронными сетями, так как он обладает высокой производительностью и хорошей поддержкой;
* Библиотеки numpy и pandas для работы с данными, так как они предоставляют широкий набор функций для обработки и анализа данных.
* Архитектура системы: Система будет состоять из двух основных компонентов: модуля распознавания речи и модуля проверки орфографии.
* Модуль распознавания речи будет использовать нейронную сеть для преобразования аудиосигнала в текст. Нейронная сеть будет обучаться на большом количестве аудиозаписей с текстом, чтобы научиться распознавать речь.
* Модуль проверки орфографии будет использовать словарь русского языка и алгоритмы для проверки правильности написания слов. Этот модуль будет проверять каждый введенный пользователем символ и сообщать о возможных ошибках.
* Алгоритмы распознавания речи: Для распознавания речи будет использоваться алгоритм глубокого обучения, основанный на нейронных сетях. Этот алгоритм будет обучаться на больших объемах данных и сможет распознавать речь с высокой точностью.
* Алгоритм проверки орфографии: Для проверки орфографии будут использоваться алгоритмы, основанные на словарях и правилах грамматики. Эти алгоритмы будут проверять каждый символ введенного текста на соответствие словарю и правилам грамматики русского языка.
* Пользовательский интерфейс: Интерфейс будет представлять собой простую форму, где пользователь сможет ввести текст и нажать кнопку “Проверить”. После нажатия кнопки текст будет проверяться на орфографию и, если все правильно, отправляться на сервер для распознавания.
* Тестирование и отладка: После создания системы необходимо провести тестирование и отладку. Для этого будут использоваться различные тесты и методы отладки, такие как unit-тесты, интеграционные тесты и анализ кода.

1. **Описание проекта интерфейса приложения**
2. **Расчет количества функциональных точек**
3. **Определение основных технико-экономических показателей**
4. **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.