**Техническое задание по созданию**

**проекта «Приложение для здорового образа жизни»**

## НАИМЕНОВАНИЕ ИС

**Полное наименование системы:** Приложение для здорового образа жизни

**Условное обозначение системы:** Система.

**Условное обозначение подсистемы:** Приложение, Подсистема.

1. ОБЩИЕ ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Задача системы:

* Обеспечить пользователей актуальной информацией о калорийности продуктов, физических упражнениях и советах по здоровому образу жизни.

Система должна:  
Регистрация пользовательских данных в системе.

Получение и отображение данных о калорийности продуктов.

Получение и отображение данных о физических упражнениях.

Получение и отображение советов по здоровому образу жизни.

1. **ОПИСАНИЕ СИСТЕМЫ**
   1. Пользовательские роли

Система должна поддерживать следующую ролевую модель, а также возможность управления ролями и их правами.

Основные ролевые группы и их состав:

* Пользователь: имеет доступ ко всем функциям приложения.
* Администратор: может добавлять и редактировать данные.
  1. **Пользователи приложения**

Эта ролевая группа включает в себя пользователей системы, обладающих первичным без ограниченным доступом.

* Пользователь - получает доступ к информации о калорийности продуктов, упражнениях и советах.
* Администратор - управляет содержанием приложения и его настройками.
  1. **Система управления мероприятиям**

Система управления обеспечивает функции создания, управления и поддержки.

* 1. Система управления пользователями

Система управления пользователями позволяет пользователям регистрироваться в системе.

1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К СИСТЕМАМ
   1. Система управления данными о калорийности продуктов

**Система управления данными о калорийности продукта должна:**

* **З**апрашивать данные о калорийности с веб-страницы.
* Парсинг HTML-кода страницы для извлечения информации.
* Отображать данные в упорядоченной таблице с колонками: Продукт, Калории, Белки, Жиры, Углеводы.
  1. **Система управления данными о физических упражнениях**

**Система управления данными о физических упражнениях:**

* **З**апрашивать данные об упражнениях с веб-страницы.
* Парсинг HTML-кода страницы для извлечения информации.
* Отображать данные в упорядоченной таблице с колонками: Упражнение, Повторы, Подходы.
  1. **Система управления советами по здоровому образу жизни**

**Система управления советами по здоровому образу жизни:**

* **З**апрашивать данные с веб-страницы.
* Парсинг HTML-кода страницы для извлечения информации.
* Отображать данные в упорядоченной таблице с колонкой: Советы.
  1. **Система управления пользователями**

**Система управления пользователями должна:**

* Регистрация пользователя: позволяет пользователям регистрироваться и редактировать свой профиль.
* Личный кабинет пользователя: содержит поля для заполнения личной информации, таких как фамилия, имя, отчество, email, пароль, пол, телефон и согласие с правилами использования.

1. **ИНТЕРФЕЙСЫ**
   1. **Веб-приложение**

В данном разделе описаны основные требования к веб приложению.

**Общие требования к веб-приложению**

Приложение должно отвечать общим функциональным требованиям, представленным в разделе «Общие функциональные требования».

**Административная часть**

* 1. **Графический интерфейс пользователя (GUI)**

Главное окно: включает кнопки для получения данных о калорийности продуктов, физических упражнениях и советах по здоровому образу жизни.

Окно отображения данных: отображает данные в виде таблицы с соответствующими колонками.

* 1. **Кнопки интерфейса**
* Кнопка "Получить данные о калорийности продуктов": вызывает функцию get\_calories\_button\_click.
* Кнопка "Получить данные о физических упражнениях": вызывает функцию get\_exercises\_button\_click.
* Кнопка "Получить советы по здоровому образу жизни": вызывает функцию get\_health\_tips\_button\_click.

1. **ВНЕШНИЕ СЕРВИСЫ**

Система должна поддерживать внешние сервисы, такие как:

* Запрос данных: осуществляется с веб-страниц с использованием библиотеки requests.
* Парсинг данных: осуществляется с использованием библиотеки BeautifulSoup.

1. ПЛАН ГРАФИК РАБОТ

Разработка и тестирование функций получения данных: 2 недели.

Разработка графического интерфейса: 2 недели.

Интеграция функций и интерфейса: 1 неделя.

Тестирование и отладка: 1 неделя.

Развертывание и выпуск: 1 неделя.