BiteRight Software Design Specification (SDS)

Project Name: BiteRight
Prepared By: Hana Amr, Nada Mohamed, Farah Mostafa, Youssef Ahmed
Date: 9/11/2024

1. Introduction

1.1 Purpose

The purpose of this SDS document is to outline the design, architecture, and technical specifications for the development of BiteRight, a personalized nutrition website. This document provides detailed guidance for developers, testers, and stakeholders regarding the system's structure, functionality, and design choices.

1.2 Scope

The scope of BiteRight includes providing users with personalized meal plans based on their unique health goals and preferences. Key functionalities are as follows:

- User Registration and Authentication
- Profile Setup
- Calories Calculation
- BMI Calculation
- Personalized Meal Plan Generation
- Daily and Weekly Meal Logging
- User Feedback Mechanism
- Goal Setting and Adjustment
- Tips and Tricks
- User Support and “Contact Us”

---

2. System Overview

2.1 System Architecture

The BiteRight system will be built with client-server architecture. Key components include:

- **Frontend**: Built it using HTML, CSS for styling and JavaScript framework (React) to create an interactive and responsive user interface.

- **Backend**: Developed with Flask to handle business logic, API calls, and authentication.
- **Database**: Kaggle dataset.

---

3. Functional Requirements

3.1 Architectural Design

The BiteRight system is designed following a **client-server architecture**, where:

- The **Frontend** communicates with the **Backend** using RESTful APIs.
- The **Backend** interacts with the **Database** to store and retrieve data for user profiles and meal plans.

3.2 Data Flow

1. **User Interaction:** User performs actions (e.g: profile setup).
2. **Request Processing:** Frontend sends requests to the backend.
3. **Data Handling:** Backend processes request and interacts with the database.
4. **Response:** Backend responds to the frontend, updating the UI.

4.Database Design

4.1 Database Schema

The BiteRight system uses a **relational database** with key tables for managing user information, meal plans, and feedback. Key tables are as follows:

-Users Table: Stores user profiles and authentication data.

- user_id: Unique identifier.
- name: User's name.
- email: Contact information.
- age, height, weight: User’s health metrics.

- **Meals Table:** Stores meal plans with calorie and nutrient data.

- meal_id: Unique identifier for each meal.
- user_id: Foreign key linking to Users.
- calories: Total calories for the meal.

-**Feedback Table:** Stores user feedback on meal plans.

- feedback_id: Unique identifier.
- user_id: Foreign key linking to Users.
- comments: User feedback.

5. Technology Stack

- **Frontend:** React.js

- **Backend:** Flask

-**Database:** MySQL, Keggle

-**Hosting:** oneDrive or Normal Pc

6.Testing Plan

6.1 Unit Testing
Each module, including profile setup and meal generation, will undergo unit testing to ensure proper functionality.
6.2 Integration Testing
Integration tests will ensure smooth communication between frontend, backend, and database(dataset) components.
6.3 User Acceptance Testing (UAT)
End users will test the system to ensure it meets requirements, with feedback collected for final adjustments.
6.4 Performance Testing
Stress and load testing will validate system performance under expected user load.

---

7. Conclusion

This SDS for BiteRight outlines a robust, scalable, and user-centric design. By adhering to the specifications detailed here, the development team can create a reliable platform that supports personalized health goals and provides value to its users.