Project Report

Health & Diet Analyzer for Hostel Students

Submitted by: ARYA UPADHYAY

Registration No: 24BSA10044

Department: Computer Science / Software Engineering

Date: November 24, 2025

Table of Contents

1. Executive Summary

This project, titled **"Health & Diet Analyzer for Hostel Students,"** is a software application designed to help students living in hostels manage their nutritional intake. Hostel life often leads to irregular eating habits and a lack of nutritional awareness. This system provides a platform for students to log their daily meals, analyze nutritional content, and receive simple health recommendations.

The application is built using **Java** technologies, utilizing a **Spring Boot** backend for API management and a **JavaFX** frontend for the user interface, demonstrating core software engineering principles like modular design and multi-layered architecture.

2. Introduction

Hostel students frequently face challenges in maintaining a balanced diet due to limited food

options and lack of monitoring. The Health & Diet Analyzer serves as a digital assistant that allows users to track what they eat and understand the health implications.

The system is designed to be modular, consisting of a backend server that processes data and a client-side application that users interact with. It ensures that students can easily register, log meals, and view their health statistics.

3. Problem Statement

- **Lack of Awareness:** Students often consume cafeteria food without knowing its nutritional value.
- **Manual Tracking:** Existing methods (paper diaries) are tedious and prone to error.
- **Health Risks:** Poor diet can lead to fatigue or long-term health issues, which negatively impacts academic performance.

This project aims to solve these problems by automating the tracking process and providing instant feedback.

4. System Analysis

4.1 Objectives

- To develop a user-friendly interface for logging daily meals (Breakfast, Lunch, Dinner, Snacks).
- To calculate daily caloric and nutritional intake.
- To provide basic dietary recommendations based on the logged data.
- To implement a secure login system for user data privacy.

4.2 Existing System

Currently, most students either do not track their diet or use generic mobile apps that do not account for specific hostel mess menus. These generic apps often require complex inputs that are not suitable for the quick-paced life of a student.

4.3 Proposed System

The proposed system is tailored for quick entry. It connects a desktop client (JavaFX) to a robust server (Spring Boot). Data is stored in a structured format (JSON or Database), allowing for persistent tracking over time.

5. System Design

5.1 System Architecture

The system follows a Client-Server architecture:

1. **Client Layer (JavaFX):** Handles user interaction and displays data.

2. **API Layer (Spring Boot):** Exposes REST endpoints (POST, GET) to handle logic.
3. **Data Layer:** Stores user profiles and meal logs.

Flow: User Input $\rightarrow$ JavaFX Client $\rightarrow$ REST API $\rightarrow$ Data Store

5.2 Use Case Design

The primary actor in the system is the **Student/User**.

- **Register/Login:** User authenticates to access their personal dashboard.
- **Log Meal:** User inputs food items consumed during specific meal times.
- **View Dashboard:** User views a summary of their total intake.
- **Get Recommendations:** System suggests improvements (e.g., "Eat more protein") based on the data.

5.3 Sequence Design

A typical interaction flow (Logging a Meal):

1. **Client:** User opens the app and submits meal data.
2. **Client** $\rightarrow$ **Server:** Sends a POST /meals request.
3. **Server** $\rightarrow$ **Database:** Validates and stores the meal entry.
4. **Database** $\rightarrow$ **Server:** Confirms storage.
5. **Server** $\rightarrow$ **Client:** Returns a "Success" message to the user.

Retrieving Summary:

1. **Client** $\rightarrow$ **Server:** Sends GET /summary.
2. **Server** $\rightarrow$ **Database:** Requests aggregated data.
3. **Server:** Calculates totals and health risks.
4. **Server** $\rightarrow$ **Client:** Returns JSON data containing the summary and risk assessment.

6. Technology Stack

- **Programming Language:** Java (JDK 17+)
- **Frontend:** JavaFX (for Desktop GUI)
- **Backend Framework:** Spring Boot (REST API)
- **Data Storage:** JSON File System / SQL Database (extensible)
- **Tools:** Maven/Gradle for build automation, Git for version control.

7. Implementation Details

The project is implemented in several modules:

1. **Auth Module:** Handles user registration and secure login sessions.
2. **Meal Service:** Accepts food inputs (e.g., "Roti", "Dal", "Rice") and maps them to approximate nutritional values.

3. **Analysis Engine:** A logic component that sums up calories, proteins, and carbs for the day and compares them against standard recommended values (RDA).
4. **UI Controller:** JavaFX controllers that manage the windows for Login, Dashboard, and Entry forms.

8. Conclusion

The **Health & Diet Analyzer** successfully addresses the need for a simple, student-focused diet tracking tool. By integrating a JavaFX frontend with a Spring Boot backend, the project demonstrates the practical application of full-stack development principles.

The system enables students to take control of their health by providing transparent data about their dietary habits. Future enhancements could include a mobile app interface, integration with the hostel mess menu for easier logging, and AI-driven detailed health reports.

End of Report