Software Requirements Specification

For

Intelligent Food Recommendation Engine

Date:24th March 2022

Prepared by

Specialisation	SAP ID	Student Name
AI & ML	500075162	ARRA SAI NITISH REDDY
AI & ML	500075774	ARPIT
AI & ML	500075813	BHAVESH
AI & ML	500075953	ARIHANT JAIN

Department of Informatics
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Table of Contents

1. INTRODUCTION

1.1. Purpose of the Project

It is very challenging to eat right and exercise right. Some try to count calories but there is no quick and accurate way to do so. We aim to bring a balanced diet into the life of people and still make them feel good and happy about their diet.

1.2. Target Beneficiary

All the people who are looking for a balanced diet and want to enjoy their meal can go for this intelligent recommendation engine which will keep them happy as well as fit while consuming their favourite dish or meal.

1.3. Project Scope

The proposed solution is an Intelligent Self-Learning System using evolutionary programming and reinforcement learning which prescribes to the user an exceptionally customised day to day eating plan which holds the *calories under control* and *amplifies the client's satisfaction from food consumption*.

Aimed at developing a self-learning engine that can dynamically adapt itself according to the data provided by the user and his/her nutritional requirements.

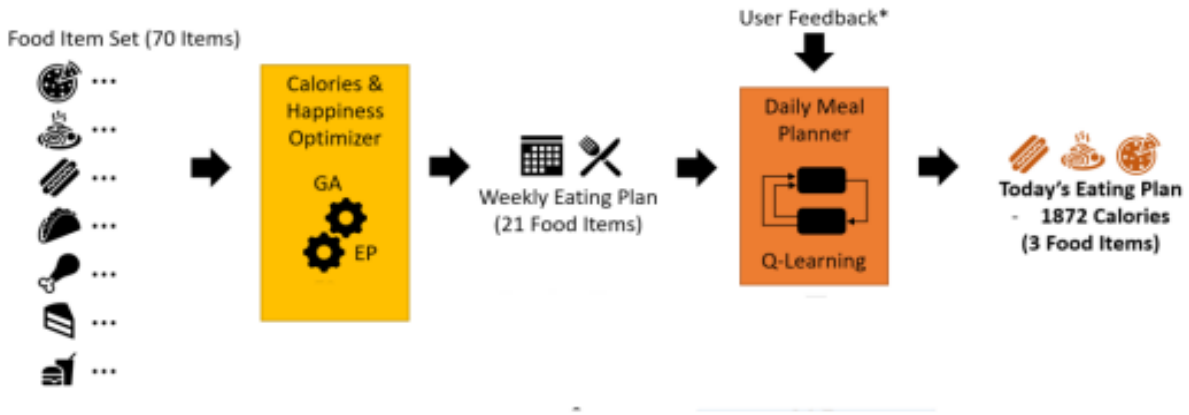
1.4. References

- Genetic Algorithms - GeeksforGeeks
- Intelligent food planning: personalized recipe recommendation | Request PDF
- http://datajobstest.com/data-science-repo/Genetic-Algorithm-Guide-[Tom-Mathew].pdf
- https://www.researchgate.net/publication/278965698_Automatic_dietary_menu_planning_based_on_evolutionary_algorithm
- https://www.myfitnesspal.com/

2. PROJECT DESCRIPTION

2.1. Reference Algorithm

The solution comprises



Calories and Happiness Optimizer:

The Calories and Happiness Optimizer selects 21 food items from the food listing comprising 70 initial food item sets. Each food item has a utility score elicited from the user representing how much the user likes that food item. The higher the utility score, the happier the user will feel after consuming the food item. The goal of the optimizer is to maximise the total utility score (thus maximising happiness) and to keep within a specific daily calorie intake. The 21 food items represent the eating plan for a week and must not have repeated food items. Various Evolutionary Computation techniques are used to perform the optimization for comparison. From week to week, a rule can be imposed such that not more than 10 food items can be repeated.

Daily Meal Planner

The Daily Meal Planner takes the weekly eating plan and uses Q-Learning to learn the user's preference on what food they like or do not like to eat in close succession. An example is that if the user ate chicken rice during lunch, he may enjoy fried rice less for dinner as compared to if he had something else for lunch. The output of the Daily Meal Planner is an optimised sequence for the 21 food items organised into 7 x 3 meals a day

■ Module 1: Calories and Happiness Optimizer

Evolutionary Computation techniques were used to select 21 food items from 70 food items to maximise utility and keep within a target daily calories intake.

Genetic Algorithm

- Genetic algorithm is suitable for this task because the task is a combinatorial optimization problem with a maximum calorie constraint.
    - **Chromosome Representation** – Each chromosome has 70 genes that take on binary '0' or '1' indicating if a particular food item is selected. Only 21 genes can be '1'.
    - **Fitness Function** – The fitness function is the summation of utility scores for the 21 food items represented in each chromosome.
    - **Constraints** – There are two constraints.

1. The solution must only select 21 meals (21 ones in the final solution) based on the assumption of 3 meals a day i.e. 21 meals a week.
2. The allocated calorie budget cannot be exceeded.

(1) is a hard constraint and (2) is a soft constraint with penalties if violated.

- **Crossover Operation** – The crossover operation uses 2 randomly selected crossover points.
- **Mutation Operation** – The mutation operation random toggles the gene value based on rate of mutation. After mutation, a special handling is implemented to adjust the gene value such that there are only 21 genes with '1'.
- **Replacement Strategy** – The fittest half parent population is combined with the fittest half offspring population to form the next generation.

Evolutionary Programming

- Similar to Genetic Algorithms, Evolutionary Programming is suitable for this task because the task is a combinatorial optimization problem with a maximum calorie constraint.

- **Chromosome Representation** – Each chromosome has 70 genes that take on binary ‘0’ or ‘1’ indicating if a particular food item is selected. Only 21 genes can be ‘1’.
- **Fitness Function** – The fitness function is the summation of utility scores for the 21 food items represented in each chromosome.
- **Constraints** – There are two constraints.

1. The solution must only select 21 meals (21 ones in the final solution) based on the assumption of 3 meals a day i.e. 21 meals a week.
2. The allocated calorie budget cannot be exceeded.

(1) is a hard constraint and (2) is a soft constraint with penalties if violated.

Mutation Operation – The mutation operation randomly toggles the gene value based on the rate of mutation. After mutation, special handling is implemented to adjust the gene value such that there are only 21 genes with ‘1’.

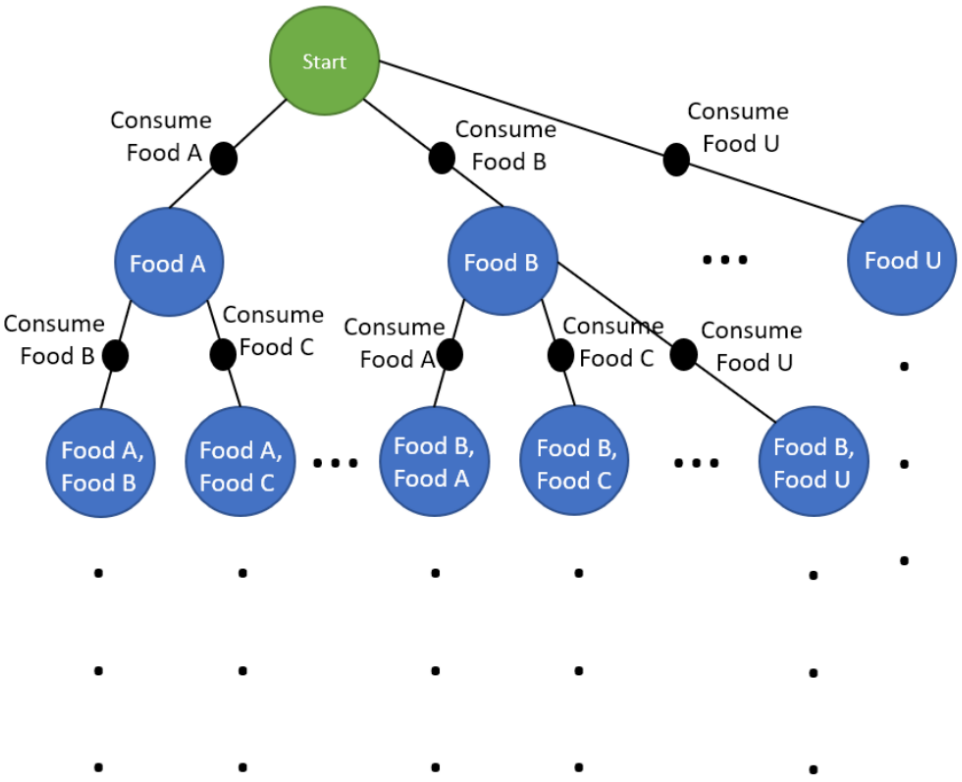
Mutation Rate – The mutation rate is reduced by 10% after every 10% completion of the total generation iterations.

Replacement Strategy – N-tournament is conducted between randomly picked N parents and N offspring and the chromosome with the highest fitness score is selected to form the new generation.

■ Module 2: Daily Meal Planner:

The Daily Meal Planner uses Q-Learning, a form of reinforcement learning to learn the optimised sequence to consume the 21 food items recommended by the Calories and Happiness Optimizer. It avoids sequencing similar food one after another which diminishes the enjoyment for the second food item due to users getting sick of similar food. Reinforcement learning is suitable for this task because user preference changes over time and reinforcement learning can continuously learn the user preference from user feedback.
Q-learning is selected because we are solving a model-free problem and we want the planner to learn an optimal greedy policy.

- **MDP Modelling:**
    - States – States represent the sequence of food consumption. Examples of states are, etc.
    - Action – An action is an act of consuming a specific food item.
    - State Transition – State transition is deterministic. When an action is performed to consume a specific food, the next state will be all previously consumed food plus the current consumed food.
    - Reward – Reward is calculated by multiplying the user feedback score (after the action of consuming a specific food) with the utility value of the food item. The reward signal is estimated during training.



- **Reward Estimation:**
  - The main challenge in developing the system is getting real users to give feedback after consuming a meal. Even if we can get a large enough pool of users who are willing to help train the system, they only eat 3 main meals a day and it will take a very long time to train the system.
  - To overcome the problem, the team has come up with a method to estimate the reward signal. The estimation is based on the hypothesis that food with similar main ingredients will have similar carbohydrate, fat and protein composite. The estimated reward signal is the utility score of the food currently consumed multiple by a discount factor determined by the Euclidean distance between the composition of the food item currently consumed and the most recent food consumed in the past.
  - The approach is to train the system up to a baseline level using the estimated reward signal and later cutover to production to learning the finer user preference.

2.2. Characteristic of Data:

- Our data is a listing of food items with their respective calories, utility and carbohydrate, fat and protein composite.
- The calories and carbohydrate, fat and protein composite information are taken from https://www.myfitnesspal.com/food/search
- The utility value is arbitrarily populated for the development of the system. In a production setting, these values will be elicited from the user.

SWOT ANALYSIS

STRENGTHS

- The method comprises various models to provide with optimal diet plane
- The food menu and utility score is customizable according to user preference
- Calorie intake is calculated individually as every person is diverse

WEAKNESSES

- The diet plan is prepared in the sets of 7 days as the resulting feedback is used to recondition for the upcoming week
- The method does not involve the medical history of the person

OPPORTUNITIES

- The model can provide a healthy eating plan that gives your body the nutrients it needs every day while staying within your daily calorie goal for weight loss
- The diet plan generated can be utilized to live a long and healthy life

THREATS

- In a worst-case scenario, the final solution may not be optimal as the evolution of the diet plane can get astray

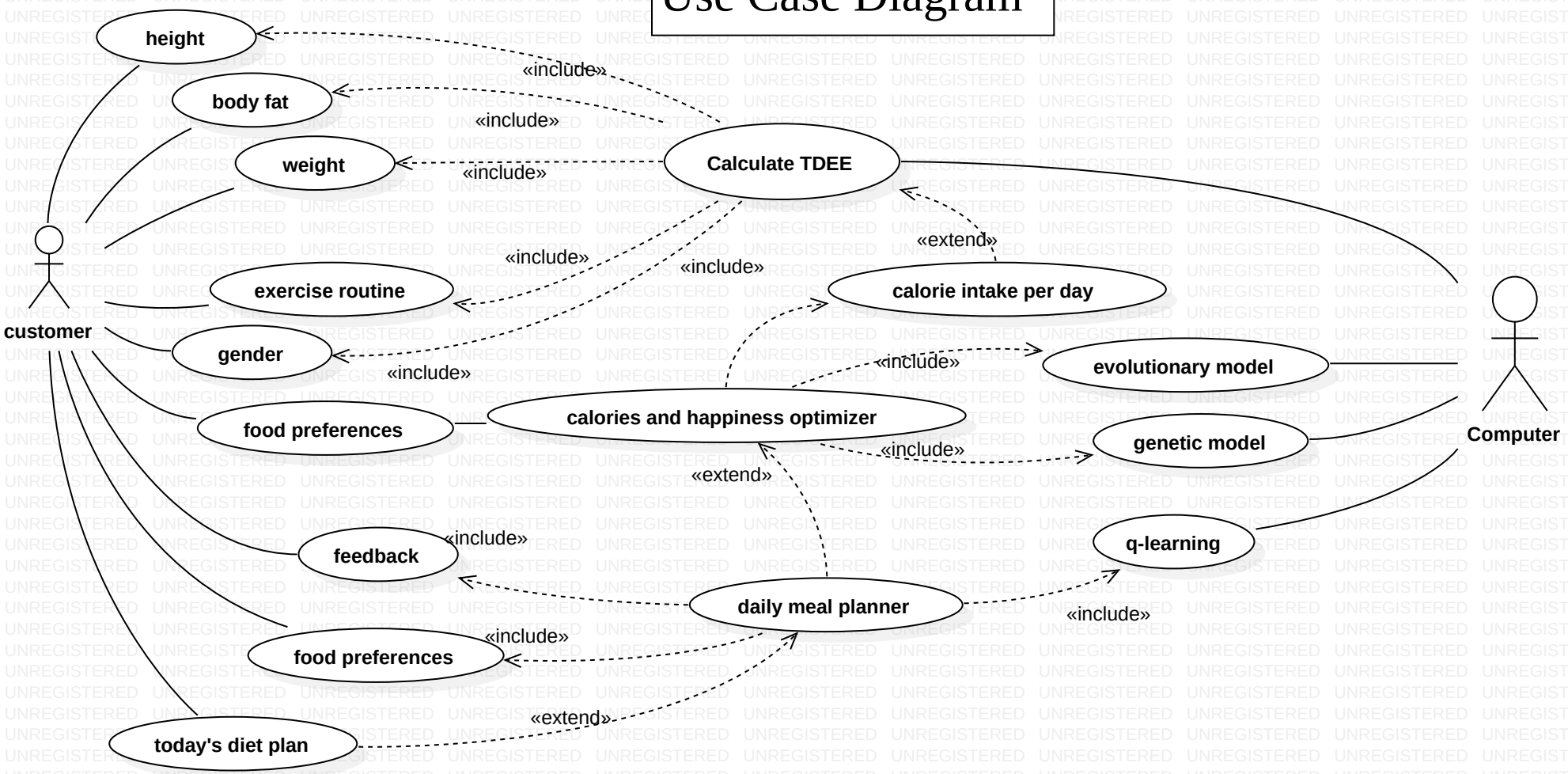
2.4. Project Features

- Allows users to use a personalised diet recommendation system as per their preferences to achieve desired nutritional goals
- Unlike traditional diet engines available on the market, use input from the user and his/her preference to customise a weekly eating plan
- The main system components have been developed such that when in production, the system is capable of continuous self-learning and self-evolving according to the changing environment such as changing user food preference and new food items.
-

2.5. Design and Implementation Constraints

- Team has considered calculation on basis of initial *70 food item* dataset with their respective nutritional and utility scores
- Current implementation supports generation of week-wise food menu with *3 meals a day (7 days x 3 meals/day)*
- Initial *TDEE calculation* for a finding of maintenance calories of an individual is done using *Harris-Benedict* formula

Use Case Diagram



2.7. Assumption and Dependencies

3. SYSTEM REQUIREMENTS

3.1. User Interface:

- User input is provided by the dashboard which includes age, height, weight, gender by which the calory intake per week is calculated
- The user also provides a dataset that represents the menu with items and their utility score which represents the appetence of the food item

3.2. Software Interface:

- The solution comprises (1) Calories and Happiness Optimizer, (2) Daily Meal Planner
- The Calories and Happiness Optimizer selects 21 food items from the food listing comprising 70 initial food item sets
- The Daily Meal Planner uses Q-Learning, a form of reinforcement learning to learn the optimized sequence to consume the 21 food items recommended by the Calories and Happiness Optimizer

3.3. Database Interface

- This program requires a Flat-File Database or simply put a file as our database system
- The data of the user's appetence is considered in form of the CSV files which contains the food items with the nutrition information

4. NON-FUNCTIONAL REQUIREMENTS

4.1. Performance requirements

- Considering the worst-case scenario of the model that is O(g(n*m + n*m + n)) can be considered as the optimal performance requirement

4.2. Software Quality Attributes

4.3. Quality Attributes	Performance
adaptability	Medium
availability	Medium
correctness	Medium
flexibility	High
interoperability	Low
maintainability	High
portability	High
reliability	Medium
reusability	High
robustness	High
testability	Medium
usability	High