

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛТЕХНІЧНИЙ ІНСТИТУТ»**

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри

_____ О.Р. Чертов

«____» _____ 2015 р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності 6.040301 «Прикладна математика»

на тему: Автоматизована система складання раціону харчування людини

Виконав: студент 4 курсу, групи КМ-12 №0110

Борисенко Владислав Денисович

Керівник асистент Терещенко І.О.

Консультант з нормоконтролю ст. викл. Мальчиков В.В.

Рецензент к. т. н. доцент Орлова М.М.

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2015

**Національний технічний університет України
«Київський політехнічний інститут»**

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти – перший (бакалаврський)

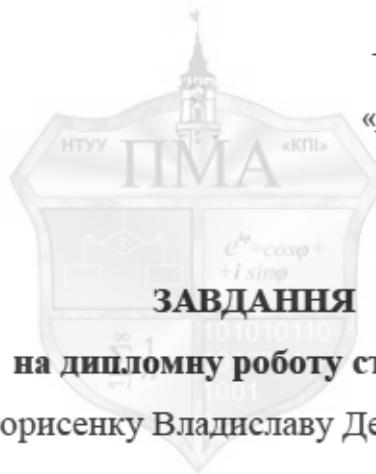
Спеціальність 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О. Р. Чертов

« ____ » _____ 2015 р.



Борисенку Владиславу Денисовичу

1. Тема роботи «Автоматизована система для складання раціону харчування людини»,

керівник роботи – асистент Терещенко Ігор Олександрович,

затверджені наказом по університету від «19» травня 2015 р. № 1039 – С.

2. Термін подання студентом роботи “12” червня 2015р.

3. Вихідні дані до роботи

- база даних продуктів та страв
- персональні дані користувача

4. Зміст роботи

- вивчити літературні джерела за тематикою дослідження;
- провести порівняльний аналіз математичних методів для складання раціону харчування;
- обґрунтувати вибір обраного методу;
- реалізувати математичне програмне забезпечення для складання раціону харчування людини;
- Оформити документацію до дипломної роботи

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

- ілюстрації роботи програми
- презентація до дипломної роботи

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	ст.викладач Мальчиков В.В.		

7. Дата видачі завдання 28» жовтня 2014

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за тематикою роботи	20.11.2014	
2.	Порівняльний аналіз математичних методів складання раціону	30.11.2014	
3.	Розроблення математичної моделі для вирішення задачі	30.12.2014	
4.	Проектування архітектури розроблюваних програмних засобів	01.03.2015	
5.	Визначення складу та форматів вихідних даних та результатів для кожної програми	09.03.2015	
6.	Програмна реалізація обраних математичних методів	13.03.2015	
7.	Розробка клієнт-серверного додатку	27.03.2015	
8.	Перевірка та тестування моделі	16.04.2015	
9.	Підготовка графічної частини дипломної роботи	25.04.2015	
10.	Оформлення документації дипломної роботи	20.05.2015	

Студент _____

Борисенко В.Д.

Керівник роботи _____

Терещенко І.О.

АНОТАЦІЯ

Дана дипломна робота виконана на здобуття освітньо-кваліфікаційного рівня “Бакалавр” та присвячена розробці автоматизованої системи складання раціону харчування людини.

Метою роботи є створення програмного забезпечення для складання раціону харчування людини.

В рамках дипломної роботи проведено аналіз існуючих методів вирішення поставленої задачі, проведено їх аналіз, виділено їх недоліки.

Робота стосується задачі складання раціону харчування. У якості алгоритмів оптимізації раціону було обрано симплекс-метод та генетичний алгоритм. Реалізовані алгоритми здатні знаходити оптимальне рішення поставлених перед ними задач.

Робота виконана на 53 аркушах, має посилання на список використаних літературних джерел з 11 найменувань, а також наведено 12 рисунків, 4 таблиці та 2 додатки.

Ключові слова: раціон харчування, симплекс-метод, генетичний алгоритм.

ABSTRACT

This degree work is completed to obtain educational bachelor degree and is devoted to developing of mathematical and applied software for food planning.

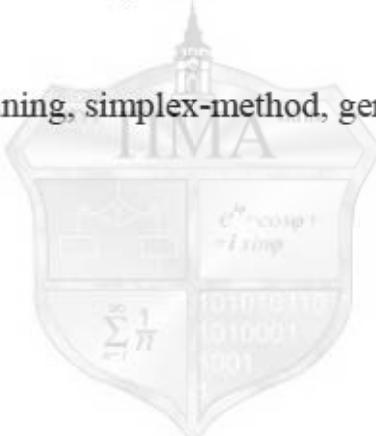
The main goal of this work is a development of human food planning system .

In this work were described , analized and showed shortcomings of existing methods for solution of food planning problem and software solutions.

The task was completed using simplex-method and genetic algorithm. Implemented algorithms are capable of finding optimal solution of specific problems.

The work is written on 53 papers, contains references to the list of used literature with 11 items and also provides 12 figures, 4 tables and 2 supplementials with source code.

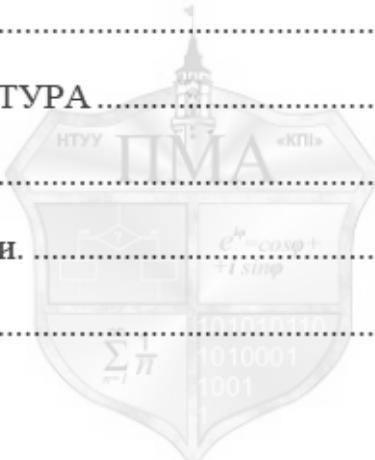
Keywords: food planing, simplex-method, genetic algorithm.



ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ	9
ВСТУП	10
1 ПОСТАНОВКА ЗАДАЧІ	11
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	12
2.1 Опис предметної області.....	12
2.2 Огляд існуючих систем зіставлення раціону харчування	12
2.3 Огляд технологій розробки.....	14
2.4 Вибір технологій розробки	18
2.5 Висновки.....	18
3 ОГЛЯД МАТЕМАТИЧНИХ МЕТОДІВ	20
3.1 Формалізована постановка задачі, опис множин, визначення параметрів задачі	20
3.2 Порівняльний аналіз існуючих методів розвязку задачі	21
3.3 Вибір та обґрунтування вибору рішення	23
3.4 Розв'язок задачі з використанням симплекс-методу	23
3.5 Розв'язок задачі з використанням генетичного алгоритму	26
3.6 Висновки.....	30
4 СТРУКТУРА І ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	31
4.1 Опис клієнтської частини	31
4.2 Опис серверної частини	32
4.3 Категорії користувачів	33

4.4 Класи даних	33
4.5 Матриця елементарних подій	34
4.6 Бізнес-правила	35
4.7 Інфологічне проектування	36
4.8 Керівництво користувача	38
5 ТЕСТУВАННЯ КОМПОНЕНТІВ СИСТЕМИ	43
5.1 Контрольні приклади	43
5.2 Висновок	48
ВИСНОВКИ	49
ВИКОРИСТАНА ЛІТЕРАТУРА	51
ДОДАТКИ	53
Додаток А. Код програми	53
Додаток Б. Презентація	60



ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

БД – База Даних;

ГА – генетичний алгоритм;

ЗЛП – задача лінійного програмування;

ЗЦЛП -задача ціличисельного лінійного програмування;

ІС – Інформаційна Система;

КЗЛП – канонічна задача лінійного програмування;

СКБД – система керування базами даних;

ERD (Entity-relationship diagram) – діаграма зв'язків між сутностями;



ВСТУП

Сучасне життя, з її наростаючим темпом, повне стресів і постійного поспіху. Тому дуже багато людей веде малоактивний спосіб життя. На роботу ми ідемо на машині, метро або автобусі і цілий день проводимо в офісі, сидячи за комп'ютером зі згорбленою спиною. Протягом робочого дня мало хто дозволяє собі нормальній сніданок або обід.

Неправильний режим харчування, мала фізична активність та куріння є найпоширенішими факторами ризику для здоров'я людини, чому сприяє необізнаність про принципи раціонального харчування. Це призводить до різкого дисбалансу харчових речовин, і, як наслідок, до порушення процесів регуляції обміну речовин, гіповітамінозам з подальшим формуванням виражених форм розладів шлунково-кишкового тракту, нервових хвороб та інших патологічних станів [1].

Саме тому дуже важливим є питання полегшення зіставлення правильного режиму та раціону харчування.

В даній роботі розглядаються відомі засоби зіставлення раціону харчування, та фактори, які впливають на його зіставлення, та описується автоматизована система зіставлення раціону харчування людини, яка в подальшому може бути підсистемою загальної системи підтримки здоров'я людини.

1 ПОСТАНОВКА ЗАДАЧІ

Метою даної роботи є дослідження існуючих методів побудови раціону харчування людини з урахуванням фізіологічних особливостей організму, розробка програмного забезпечення для складання раціону харчування людини.

В якості предметної області використовується база даних страв ресторанної їжі [2], а також персональні дані користувача.

При розробленні даної системи повинні бути розв'язані такі завдання:

- а) проведення аналізу існуючих рішень, позначення їх недоліків
- б) проведення огляду та порівняльного аналізу існуючих методів складання раціону людини
- в) вибір методу для розв'язання задачі
- г) розробка програмного забезпечення на базі обраного методу
- д) тестування розробленої системи

Реалізована система має задовільняти наступним вимогам:

- а) система має бути зроблена на базі клієнт-серверної архітектури.
- б) має бути реалізована взаємодія з базою даних.
- в) система повинна мати декілька типів користувачів
- г) система повинна мати зручний інтерфейс
- д) має бути реалізована реєстрація та аутентифікація користувачів в системі

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Опис предметної області

Раціональне харчування являється харчуванням, основними принципами якого є наступне [3]:

1. Рівновага між енергією, яка надходить з їжею і енергією, що витрачається людиною в процесі життедіяльності
2. Задоволення потреб організму в певній кількості і співвідношення харчових речовин (білків, жирів і вуглеводів).

Для розрахунків необхідні дані про: зріст, вагу, стать, вікову групу, стиль життя та діагнози людини. Також можуть враховуватися побажання самої людини.

Дані про продукти та страви повинні включати в себе інформацію про харчову цінність та вміст корисних речовин на фіксовану кількість грам, або, у випадку страв, на одну одиницю страви.

2.2 Огляд існуючих систем зіставлення раціону харчування

2.2.1 MyFitnessPal

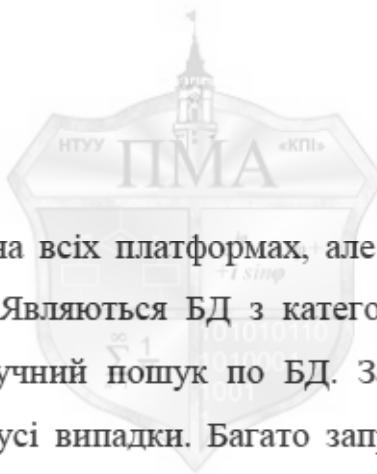
Android-програма, розроблена для слідкування за схудненням. Реалізована функція ведення щоденника, проставлення цілей, ведення статистики. Система реалізована як соціальна мережа, яка обертається навколо правильного харчування. Користувач може не тільки записувати свої

досягнення, а і, як в соц. мережах, може розповісти про них друзям. Має велику базу даних страв та продуктів[4].

Недоліки :

- Не має функції надавання порад користувачеві.
- Не має функції прогнозування.
- Окрім ведення історії та збору статистики, фактично, більше нічого не робить.
- Не враховує хвороби користувача

2.2.2 Збірники дієт



Реалізовані майже на всіх платформах, але у більшості на мобільних ОС, та як веб-сторінки. Являються БД з категоризованими стандартними дієтами. Реалізований зручний пошук по БД. Зазвичай зберігають в собі велику кількість дієт на усі випадки. Багато запропонованих дієт складені дипломованими дієтологами.

Недоліки:

- Не мають функції порадників(треба знати, яку саме дієту шукати)
- Ніяк не взаємодіють з персональними даними користувача

Прикладом такої програми може слугувати безкоштовний сбірник дієт на Google Play, розроблений як Android-аплікація[5].

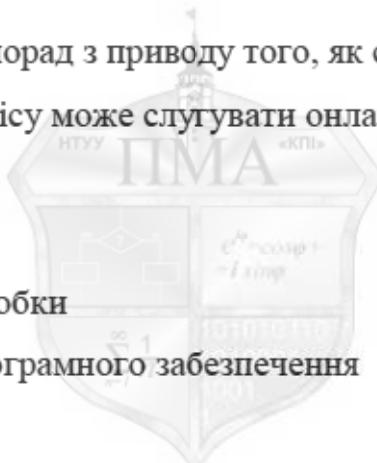
2.2.3 Сервіси для розрахунку дієти

Більшість розроблені як веб-сторінки. В них зазвичай реалізована тільки одна функція – розрахунок денної норми калорій в залежності від параметрів користувача. Іноді додатково розраховуються інші поживні речовини(білки, жири, вуглеводи тощо).

Недоліки:

- Не зберігають інформацію про користувача (кожного разу доводиться вводити свої дані)
- Не надають ніяких порад з приводу того, як саме треба харчуватись.

Прикладом такого сервісу може слугувати онлайн-калькулятор дієт [6].



2.3 Огляд технологій розробки

2.3.1 Огляд архітектур програмного забезпечення

Клієнт-серверна архітектура, яка широко застосовується при роботі з базами даних в мережі, відома вже давно і найчастіше застосовувалась у великих організаціях. Ця технологія все частіше приваблює погляди розробників програмного забезпечення, оскільки в світі нагромаджено величезну кількість інформації по різноманітних питаннях і найчастіше ця інформація зберігається в базах даних[7].

Переваги використання:

- Дозволяють організовувати мережі з великою кількістю робочих станцій;

- Забезпечують централізоване управління обліковими записами користувачів, безпекою та доступом, що спрощує мережне адміністрування;
- Ефективний доступ до мережевих ресурсів;
- Користувачеві потрібен один пароль для входу в мережу і для отримання доступу до всіх ресурсів, на які поширюються права користувача.

Поряд з перевагами мережі клієнт – серверної архітектури мають і ряд недоліків:

- Несправність сервера може зробити мережу непрацездатною, як мінімум втрату мережевих ресурсів;
- Вимагають кваліфікованого персоналу для адміністрування;
- Мають вищу вартість мереж і мережевого обладнання.

Стаціонарна архітектура використовується тоді, коли неважливе розподілення даних і планується використання функціоналу лише одним користувачем і зберігання програм та даних на одному комп’ютері. Ніяких переваг для використання в даній задачі ця архітектура не має, оскільки розроблювана система розрахована на багатьох користувачів, в яких є доступ до єдиної БД.

2.3.1 Огляд мов програмування

Java — об'єктно-орієнтована мова програмування [8]. Синтаксис мови багато в чому схожий на С та С++. У офіційній реалізації, Java

програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Переваги застосування для поставленої задачі:

- зручність розробки інтерфейсу через JavaFX фреймворк;
- написаний код може бути повторно використаний для схожих задач;
- незалежність від ОС.

Недоліки застосування для поставленої задачі:

- складно реалізувати математичні обчислення;
- необхідність встановлення JVM (Java virtual machine) для роботи.

MATLAB — пакет прикладних програм для чисельного аналізу, а також мова програмування, що використовується в даному пакеті [9].

Переваги застосування для поставленої задачі:

- мова розроблена для математичних та наукових обчислень;
- має готові реалізації багатьох алгоритмів оптимізації;
- підтримка обчислення даних великої розмірності.

Недоліки застосування для поставленої задачі:

- складність поєднання розробленого математичного забезпечення з прикладними програмами, написаними на інших мовах високого рівня.

Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою [10].

Переваги застосування для поставленої задачі:

- зручність наукових обчислень;

Недоліки застосування для поставленої задачі:

- менша швидкодія написаного коду в порівнянні з іншими мовами програмування, такими як Java.
- потребує інтерпретатора Python на комп'ютері.

2.3.2 Огляд СКБД

SQLite — полегшена реляційна система керування базами даних [11].

Втілена у вигляді бібліотеки.

Переваги застосування для поставленої задачі:

- проста у використанні;
- простота збереження резервних копій бази даних.

Недоліки застосування для поставленої задачі:

- недостатньо захищена зо зовнішніх атак;
- не пристосовна для зберігання великих об'ємів даних.

Oracle - об'єктно-реляційна система керування базами даних від Oracle Corporation [12]. Ця СКБД має великі та потужні можливості не лише для зберігання даних, а і для їх захисту, аналітики та реорганізації.

Переваги застосування для поставленої задачі:

- здатна зберігати великі об'єми даних;
- має велику кількість механізмів захисту даних

Недоліки застосування для поставленої задачі:

- менш швидкі запити до БД ніж в інших СКБД

2.4 Вибір технологій розробки

Архітектурою розроблюваного програмного забезпечення було обрано клієнт-серверну модель, оскільки вона найповніше відповідає поставленим перед системою задачам.

Мовою розробки програмного забезпечення було обрано Java, оскільки вона має велику кількість бібліотек, за допомогою яких можна зручно спроектувати систему та реалізувати весь функціонал.

Програма розробляється на базі JavaFX фреймворку, який дозволяє будувати зручний інтерфейс, використовувати стилі, та запускати програму через браузер, що може бути використаним для розміщення програми на веб-сторінці.

У якості СКБД було обрано Oracle, оскільки система буде зберігати велику кількість даних і ці дані зручно описуються реляційною моделлю. До того ж, розроблені зручні механізми інтеграції Oracle Database в Java.

2.5 Висновки

Було проаналізовано існуючі рішення, пов'язані з слідкуванням за харчуванням людини, та виявлено їхні основні недоліки:

- відсутність надання порад з приводу раціону харчування
- відсутність врахування хвороб користувача
- неможливість збереження персональних даних користувача
- відсутність великої бази даних страв та хвороб, та їх взаємодії

Ці недоліки усуваються створенням системи на базі клієнт-серверної архітектури. Як мову розробки було обрано Java. Серверна частина буде реалізована на базі СКБД Oracle.



3 ОГЛЯД МАТЕМАТИЧНИХ МЕТОДІВ

3.1 Формалізована постановка задачі, опис множин, визначення параметрів задачі

Задача зіставлення раціону харчування людини може бути формалізована таким чином [13]:

$$Z = \sum_{j=1}^n c_j x_j \quad (3.1.1)$$

$$c_j = \sum_{i=1}^m a_{ij} \quad (3.1.2)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \quad i=1, \dots, m \quad (3.1.3)$$

$$x_j \geq 0 \quad j=1, \dots, n \quad (3.1.4)$$

Де:

j – індекс відповідного продукту/страви/інгредієнту

c – ціна за одиницю продукту

x – вага, або кількість одиниць продукту

a_{ij} – вміст i -го нутрієнту в j -му продукті

b_i – обмеження на вміст i -го нутрієнту в меню

Z – функція, яку мінімізуємо

3.2 Порівняльний аналіз існуючих методів розвязку задачі

Основні пункти, які повинні бути враховані під час зіставлення раціону харчування є [14]:

1. Дієтичні обмеження, наприклад інгредієнти, на які у користувача алергія, або які не мають бути вживані за інших медичних причин.
2. Харчова цінність, наприклад, кількість жиру або білка яка міститься в страві, або потрібна користувачеві.
3. Денна норма споживання корисних речовин користувачем.
4. Вподобання самого користувача.

Але основною задачею, являється саме процес складання раціону. Його можна сформулювати як обернену проблему пакування рюкзака:

Задача полягає у наповненні рюкзака, що здатен витримати деяку максимальну масу, предметами, кожен з яких має вартість (або корисність) та масу. Необхідно обрати об'єкти в такий спосіб, аби максимізувати сумарну вартість (або користь), але не перевищити максимально припустиму масу.

Генетичний алгоритм та симплекс-метод – основні методи вирішення поставленої задачі.

3.2.1 Симплекс-метод

Стандартний метод вирішення задач лінійного програмування. Оскільки задача планування зводиться якраз до задачі лінійного програмування, симплекс-метод є очевидним вирішенням. Його плюсами є простота,

швидкість та точність роботи. Мінусами є складність врахування вподобань користувача, обмеженість розмірності конкретної задачі, а також те, що запропоноване алгоритмом рішення буде єдиним. Найбільшим недоліком даного методу є те, що використовуючи даний метод дуже складно вийти за рамки звичайної задачі лінійного програмування. Наприклад, врахувати узгодженість чи неузгодженість продуктів (хліб та масло, огірок та молоко), або вподобання користувача до цього продукту стає дуже складною задачею.

3.2.2 Генетичний алгоритм

Складніше, але більш елегантне і більш широке вирішення проблеми.

Еволюційні алгоритми – напрям в штучному інтелекті, який включає числові і комбінаторні проблеми оптимізації. Він використовує ітераційні процеси, такі як зріст або розвиток популяції потенційних рішень поставленої задачі.

Генетичні алгоритми здатні працювати в умовах, які постійно змінюються, та враховувати вподобання користувача, навіть такі, як колір продуктів, з яких складається меню [15]. Одним з найважливіших плюсів є те, що за один запуск алгоритму, може бути запропонована множина оптимальних рішень.

В нашому випадку, критеріями оптимального рішення є:

- Чи задоволяє рішення усім обмеженням
- Наскільки воно задоволяє конфліктуочим умовам, таким як: ціна, якість інгредієнтів, естетичні стандарти, та інші фактори.

В рамках задачі лінійного програмування генетичний алгоритм повинен працювати гірше, ніж симплекс-метод, але оскільки для нього не являється

проблемою вирішення задач великої розмірності, він може бути виправданою альтернативою точному методу.

3.3 Вибір та обґрунтування вибору рішення

Алгоритмом реалізації задачі складання раціону харчування було вирішено обрати і симплекс-метод, і генетичний алгоритм та провести їх порівняльний аналіз на конкретних задачах.

Очікується, що симплекс-метод буде отримувати найоптимальніше рішення задачі, але за більший проміжок часу ніж генетичний алгоритм. Генетичний алгоритм зможе вирішувати поставлені задачі швидше ніж симплекс метод, але знайдене рішення не завжди буде найоптимальнішим.

3.4 Розв'язок задачі з використанням симплекс-методу

Оскільки продукт, чи страва не являються неперервними величинами, а їхні параметри зберігаються в базі даних у форматі кількість на порцію то ця задача являється задачею цілочисельного лінійного програмування.

Загальна задача цілочисельного лінійного програмування [16]:

$$\text{Мінімізувати } Z = \sum_{j=1}^n c_j x_j \quad (3.4.1)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \quad i=1, \dots, m \quad (3.4.2)$$

$$x_j \geq 0 \quad j=1, \dots, n \quad (3.4.3)$$

$$x_j \in N \quad (3.4.4)$$

Де:

j – індекс відповідного продукту/страви/інгредієнту

c_j – ціна за одиницю продукту

x – кількість порцій продукту

a_{ij} – вміст i -го нутрієнту в j -му продукті

b_i – обмеження на вміст i -го нутрієнту в меню

n – кількість страв та продуктів в задачі

m – кількість нутрієнтів

Оскільки в базі немає даних про вартість продуктів, а також оскільки мінімізувати витрати користувача не є кінцевою ціллю, c_j (3.4.1) є сумою усіх корисних речовин на одну одиницю відповідного продукту.

$$c_j = \sum_{i=1}^m a_{ij} \quad (3.4.5)$$

Таким чином, мінімізація цільової функції Z здійснюється не по одному параметру, а по всім корисним речовинам, присутнім в продуктах та стравах, що призводить до рівномірного врахування усіх корисних речовин.

Область допустимих рішень ЗЛП представляє опуклий багатогранник. Оптимальне рішення знаходить в вершинах багатогранника. В основі симплекс-методу лежить цілеспрямований перебір вершин області припустимих рішень. При цьому рух до оптимальної вершині здійснюється по сусідніх вершин області.

КЗЛП:

$$\text{Мінімізувати } Z = \sum_{j=1}^n c_j x_j \quad (3.4.6)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad i=1, \dots, m \quad (3.4.7)$$

$$x_j \geq 0 \quad j=1, \dots, n \quad (3.4.8)$$

1) $\max(z) = -\min(-z)$;

2) Перетворення нерівностей в рівності здійснюється шляхом введення додаткових невід'ємних змінних:

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \rightarrow \sum_{j=1}^n a_{ij} x_j - x_{n+i} = b_i, \quad x_{n+i} \geq 0; \quad (3.4.9)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \rightarrow \sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i, \quad x_{n+i} \geq 0; \quad (3.4.10)$$

3) Усі вільні члени $b_i \geq 0$.

Пошук оптимального значення СМ:

- 1) Приведення ОЗЛП до КЗЛП.
- 2) Вибір початкового допустимого базисного рішення.
- 3) Перехід від початкового базисного рішення до суміжного допустимому базисному рішенням з кращим значенням цільової функції.
- 4) Продовження пошуку допустимих базисних рішень, що поліпшують значення цільової функції. Якщо існує допустиме базисне рішення

поліпшити не можна воно є оптимальним.

В даного метода є один суттєвий недолік: при розв'язку задач великої розмірності метод сильно сповільнюється. Але, оскільки користувач в переважній кількості випадків заздалегідь має невеликий набір продуктів, на яких буде здійснюватися задача пошуку оптимального раціону, не має сенсу використовувати менш повільні алгоритми.

У тому випадку, коли користувач усе ж має великий набір з продуктів для зіставлення раціону, буде використовуватися менш точний, але швидший генетичний алгоритм.

3.5 Розв'язок задачі з використанням генетичного алгоритму



При описі генетичних алгоритмів використовуються визначення, запозичені з генетики. Наприклад, мова йде про популяцію особин, а в якості базових понять застосовуються ген, хромосома, генотип, фенотип, алель. Також використовуються відповідні цим термінам визначення з технічного лексикону, зокрема, ланцюг, двійкова послідовність, структура [17].

Популяція — це множина особин.

Особини, що входять в популяцію, у генетичних алгоритмах представляються хромосомами з закодованими в них множинами параметрів задачі, тобто рішень, які інакше називаються точками в просторі пошуку (search points).

Хромосоми (інші назви — ланцюжки або кодові послідовності) — це впорядковані послідовності генів.

Ген (який також називається властивістю, знаком чи детектором) — це атомарний елемент генотипу, зокрема, хромосоми.

Генотип або структура — це набір хромосом даної особини. Отже, особинами популяції можуть бути генотипи або одиничні хромосоми (в досить поширеному випадку, коли генотип складається з однієї хромосоми).

Дуже важливим поняттям у генетичних алгоритмах вважається функція пристосованості (fitness function), яка інакше називається функцією оцінки. Вона являє міру пристосованості даної особини в популяції. Ця функція відіграє найважливішу роль, оскільки дозволяє оцінити ступінь пристосованості конкретних особин у популяції і вибрати з них найбільш пристосовані (тобто мають найбільші значення функції пристосованості) відповідно з еволюційним принципом виживання «найсильніших» (які найкраще пристосувалися).

Особа x – вектор розмірності n .

$$\text{Фітнес-функція:} \quad (3.5.1)$$

$$F(x) = f(x) + \Phi(g(x), h(x))$$

$$f(x) = \sum_{j=1}^n c_j x_j \quad (3.5.2)$$

$$c_j = \sum_{i=1}^m a_{ij} \quad (3.5.3)$$

$$\Phi = 10^{20} \sum_{j=1}^n |g_j(x)| \quad (3.5.4)$$

$$g_i(x) : \sum_{j=1}^n a_{ij} x_j \geq b_i \quad i=1, \dots, m \quad (3.5.5)$$

$$x_j \geq 0 \quad j=1, \dots, n \quad (3.5.6)$$

$$x_j \in N \quad (3.5.7)$$

Де:

j – індекс відповідного продукту/страви/інгредієнту

c_j – ціна за одиницю продукту

x – кількість порцій продукту

a_{ij} – вміст i -го нутрієнту в j -му продукті

b_i – обмеження на вміст i -го нутрієнту в меню

n – кількість страв та продуктів в задачі

m – кількість нутрієнтів

В загальному випадку фітнес-функція генетичного алгоритму має зростати – та особина, в якої ця функція найбільша і є найбільш пристосованою, тобто є найбільш оптимальним рішенням. Але, в даному випадку було вирішено зробити навпаки. Оскільки дана модель напряму виходить з моделі для симплекс-методу, немає сенсу цільову функцію. Таким чином функція пристосованості має зменшуватися від покоління до покоління.

Для того, щоб особи в популяції не виходили за обмеження задачі було застосовано метод штрафних функцій. Функція $\Phi(x)$ - штрафна функція нескінченного бар'єру. Та особа, що виходить за обмеження отримує максимальне значення функції пристосованості, що означає, що вона є найменш пристосованою і не являється оптимальним рішенням.

Хромосоми зазвичай являються двійковими величинами, але в даному випадку це погане рішення. Номер хромосоми в генотипі особи – номер продукту в конкретній задачі. Значення хромосоми – кількість порцій відповідного продукту в раціоні.

Виділяються такі етапи генетичного алгоритму:

- a) Створення початкової популяції;
- b) Обчислення функції пристосованості для осіб популяції (оцінювання)
- c) Повторювання до виконання критерію зупинки алгоритму:
 1. Вибір індивідів із поточної популяції (селекція)
 2. Схрещення або/та мутація
 3. Обчислення функції пристосуваності для всіх осіб
 4. Формування нового покоління

Було розроблено два види мутацій:

- a) Відновлювальна мутація [18]. Оскільки існує дуже велика імовірність того, що уся початкова популяція буде розташована за межами поставлених умов, був введений даний вид мутації. Деякий відсоток осіб з популяції будуть змінюватись до тих пір, доки не будуть виконані усі умови задачі. Даний вид мутацій надзвичайно важливий на початковому етапі алгоритму. Пізніше, коли усі особи знаходяться в необхідних межах, даний вид мутацій майже не відбувається.
- b) Звичайна випадкова мутація. Деякий відсоток осіб матиме гени, змінені випадковим чином. На початку алгоритму майже не має впливу, але пізніше вносить різноманітність в популяцію.

Схрещення відбувається наступним чином:

- 1) Випадковим чином відбирається два набори особин з поточної популяції
- 2) В кожному з наборів відбирається найпристосованіша особина. Ці дві особини – батьки.
- 3) Гени нової особини утворюються з генів батьків. Кожний ген запозичується в одного з батьків, обраного випадковим чином.

Було вирішено застосувати принцип елітарності – найпристосованіша особа популяції поточного покоління в будь-якому випадку матиме місце в популяції наступного.

Критерій зупинки алгоритму: алгоритм працює до тих пір, доки найбільш пристосована особа останнього покоління має функцію менше ніж в найбільш пристосованої особи передостаннього.

3.6 Висновки

Було розглянуто два методи вирішення задачі складання раціону харчування людини – генетичний алгоритм та симплекс-метод. Сама задача формалізується як задача лінійного програмування.

Головними критеріями, за якими був обраний симплекс-метод, були точність, простота реалізації, та швидкість його роботи на задачах невеликої розмірності. Також однією з головних причин обрання даного методу було те, що поставлення задача розрахунку раціону харчування повністю являється стандартною ЗЛП.

Головним критерієм обрання генетичного алгоритму є швидкість його роботи на задачах великої розмірності. Але, оскільки даний метод не є точним, він може не знаходити найбільш оптимальні рішення поставлених перед ним задач.

4 СТРУКТУРА І ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Система реалізована на основі клієнт-серверної архітектури. Використана мова програмування – Java, JavaFX фреймворк.

4.1 Опис клієнської частини

Клієнт розроблений на основі моделі MVC (Model-View-Controller).

Вигляд вікна інтерфейсу описується у відповідному файлі .fxml формату.

До моделі входять такі класи:

- User
- Diagnosis
- Product
- SimplexDiet
- DataBaseConnector

Класами генетичного алгоритму є:

- GeneticAlgorithm
- Individual
- Population

Класи User, Diagnosis та Product використовують JavaDataBaseConnector (jdbc) для підключення до бази даних Oracle. В цих класах реалізована взаємодія клієнту з сервером та базою даних.

Класи SimplexDiet та GeneticAlgorithm реалізують логіку та алгоритми виконання симплекс-методу та генетичного алгоритму відповідно.

В SimplexDiet використовується java-бібліотека chocosolver для зручної імплементації алгоритму симплекс-методу [19].

4.2 Опис серверної частини

Серверною частиною є база даних, розроблена за допомогою СКБД Oracle. Модель даної системи легко описується в рамках реляційної моделі даних. Логічна модель ERD (entity relationship diagram) зображена на рисунку 4.2.1.

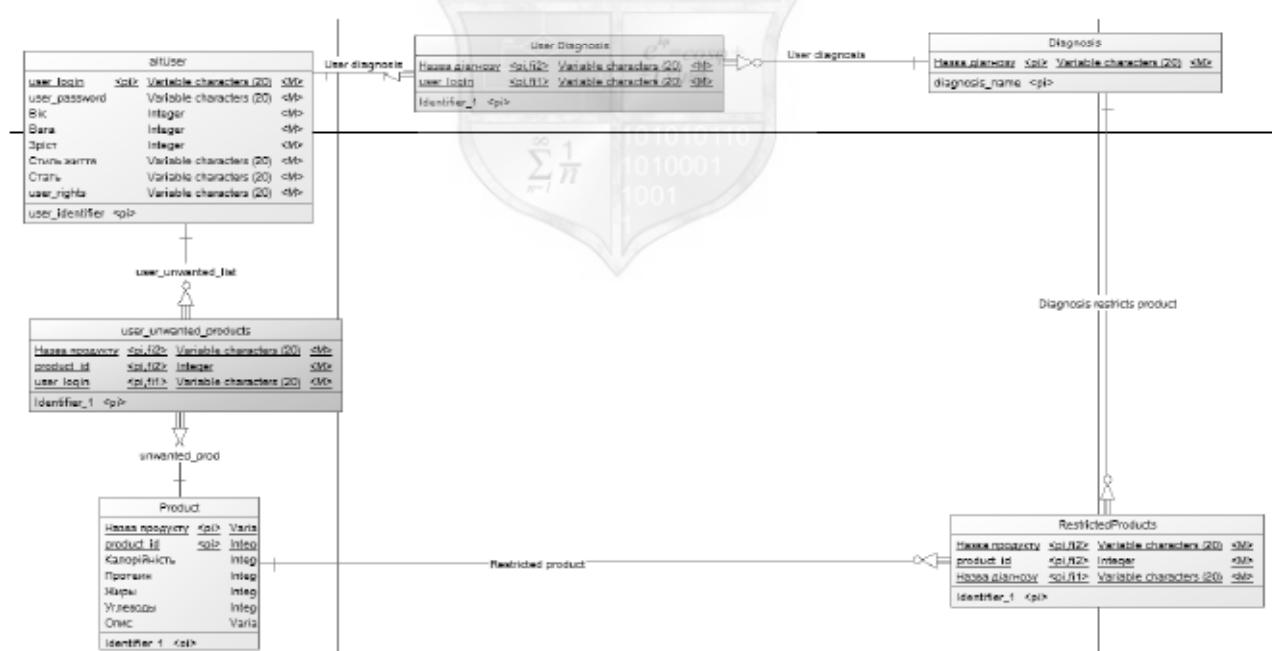


Рисунок 4.2.1 - ERD

Усі запити клієнту до БД відбуваються через класи моделі User, Diagnosis та Product.

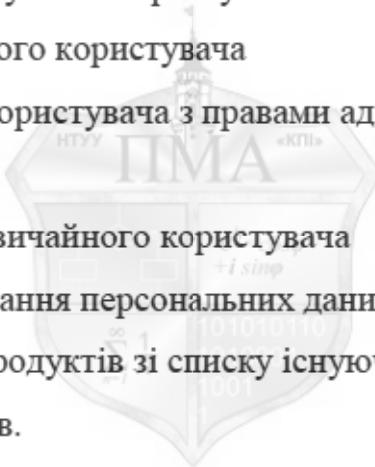
4.3 Категорії користувачів

В системі існують дві категорії користувачів:

- адміністратор;
- звичайні користувачі.

Права та обмеження для адміністратора

- а) Видалення існуючого користувача;
- б) Перегляд списку існуючих користувачів.
- в) Зміна даних існюючого користувача
- г) Створення нового користувача з правами адміністратора



Права та обмеження для звичайного користувача

- а) внесення та редагування персональних даних;
- б) вибір діагнозів та продуктів зі списку ісочущих;
- в) перегляд результатів.

4.4 Класи даних

Для коректної роботи системі потрібні наступні дані:

- дані про користувачів;
- дані про продукти;
- дані про діагнози.

4.5 Матриця елементарних подій

У таблиці 4.1 представлена елементарні події із описом відповідної реакції системи на неї.

Таблиця 4.1 – Матриця елементарних подій

№	Опис події	Тип події	Реакція на подію
1	Користувач хоче ввести інформацію про себе	N	Надати форму для заповнення
2	Користувач вводить інформацію про себе	N	Проаналізувати інформацію, виділити ту, що впливає на результат
5	Користувач хоче дізнатися результат	N	Вивести на екран результат виконання програми
6	Користувач хоче зберегти результат	N	Зберегти результат.
7	Зірвалося підключення до БД	NN	Перевірити дані на їхню цілісність. Відновити пошкоджені дані. Відновити підключення.

4.6 Бізнес-правила

4.6.1 Об'єкт «діагноз»

Бізнес-логіка об'єкту «діагноз» включає в себе:

- 1) Неможна вдруге додати вже існуючий в базі даних діагноз
- 2) Назва (тип – Char) діагнозу має бути науковою.
- 3) Діагноз має забороняти хоча б один продукт, інакше не має змісту його зберігати – для даних системи такий діагноз рівний здоровій людині.

4.6.2 Об'єкт «користувач»



Бізнес-логіка об'єкту «користувач» включає в себе:

- 1) Користувач характеризується повністю введеним набором характеристик: вага, зріст, вік, стать та стиль життя.
- 2) У користувача є логін та пароль
- 3) В базі даних не може бути двох чи більше користувачів з однимаковим логіном.
- 4) Характеристики, логін та пароль не можуть бути порожніми

4.6.3 Об'єкт «продукт»

Бізнес-логіка об'єкту «продукт» включає в себе:

- 1) У продукту є назва.

- 2) Назва не може бути порожньою.
- 3) В базі даних не може бути двох чи більше продуктів з однаковою назвою.
- 4) У продукту є наступні характеристики: калорійність, вміст білків, жирів, вуглеводів.

4.7 Інфологічне проектування

1. User

- 1.1. Зберігається інформація, яка має вплив на результат роботи програми.
- 1.2. До цієї сутності має доступ процес коригування результатів, та процес видачі результату.
- 1.3. Структура

Таблиця 4.2 – Структура сутності User

Назва атрибуту	Код атрибуту	Тип атрибуту
Вік	user_age	Int
Вага	user_weight	Int
Зріст	user_height	Int
Стиль життя	user_lifestyle	Enum
Стать	user_gender	Enum

2. Diagnosis

- 2.1. Зберігається назва діагнозу
- 2.2. До цієї сутності має доступ процес коригування результатів

2.3. Структура

Таблиця 4.4 – Структура сутності Diagnosis

Назва атрибуту	Код атрибуту	Тип атрибуту
Назва	diagnosis_name	Text

3. Product

3.1. Зберігається інформація про продукт, його склад, поживність.

3.2. До цієї сутності має доступ процес зіставлення меню.

3.3. Структура

Таблиця 4.3 – Структура сутності Product

Назва атрибуту	Код атрибуту	Тип атрибуту
Назва	product_name	Text
Калорійність	product_calorie	integer
Білки	product_protein	integer
Жири	product_fat	integer
Вуглеводи	product_carbohydrates	integer
Id	product_id	integer
Опис продукту	product_descr	Text

4.8 Керівництво користувача

При запуску клієнту користувач побачить вікно логіну. У випадку, якщо користувач ще не зареєстрований, є кнопка «Sign Up», що переведе користувача на вікно реєстрації. Інакше, користувач вводить свої логін та пароль у відповідні поля, та тисне на кнопку «Sign In».

В залежності від типу користувача з'являється наступне вікно:

- Користувач – адміністратор. З'являється вікно управління користувачами.
- Користувач без привілеїв. З'являється вікно з вкладками, кожна з яких відповідає за окрему задачу:
 - вкладка перегляду/зміни персональної інформації
 - вкладка управління діагнозами *cosip + i simp*
 - вкладка управління раціоном продуктів
 - вкладка зіставлення плану харчування
 - вкладка налаштувань та локауту

На випадок, коли користувач не розуміє, чому він не може зробити ту, чи іншу дію реалізована система підказок – треба навести на проблемне поле курсор, та з'явиться підказка.

На скріншотах 4.8.1-4.8.7 зображені інтерфейси клієнтської частини системи.

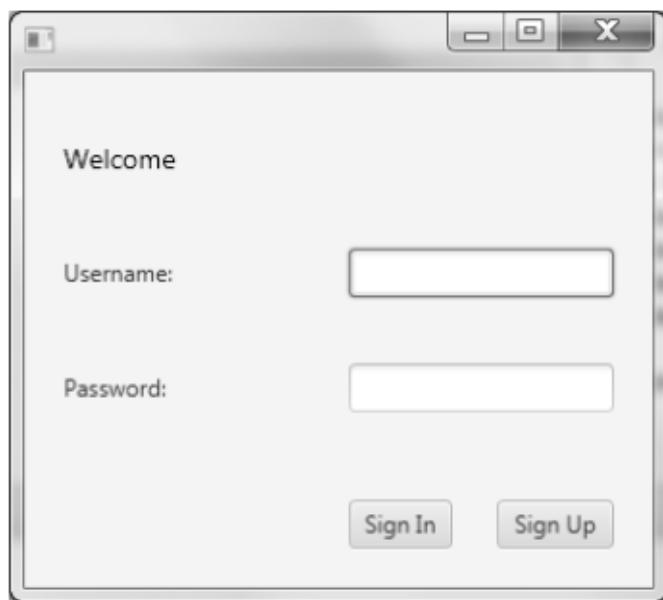


Рисунок 4.8.1 - Вікно логіну



Рисунок 4.8.2 - Вікно перегляду персональних даних

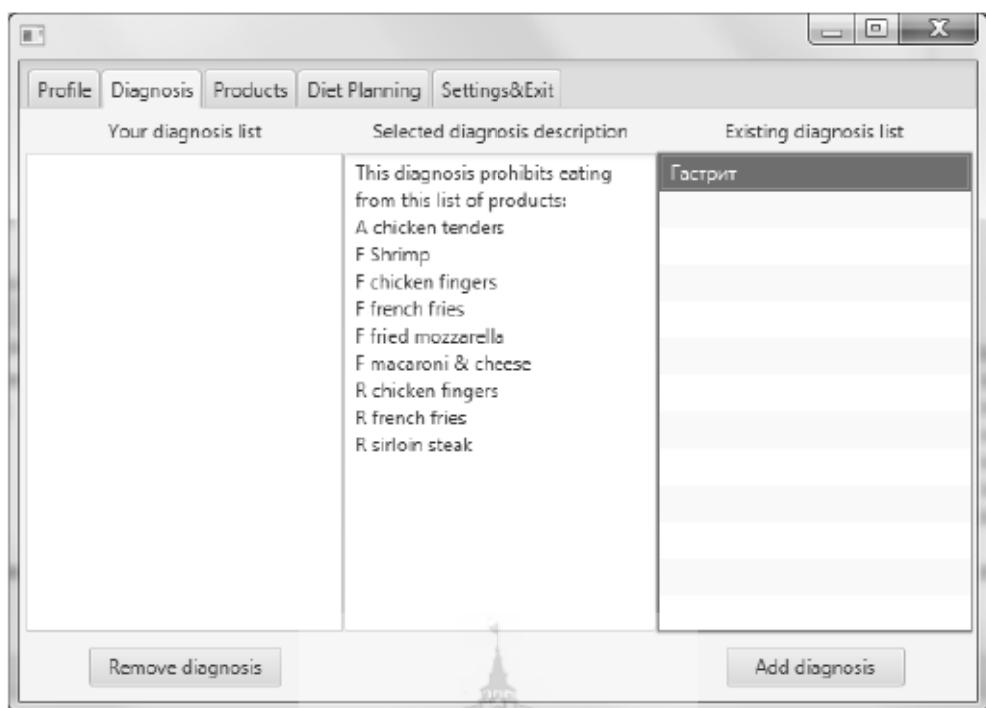


Рисунок 4.8.3 - Вікно перегляду та роботи з існуючими діагнозами

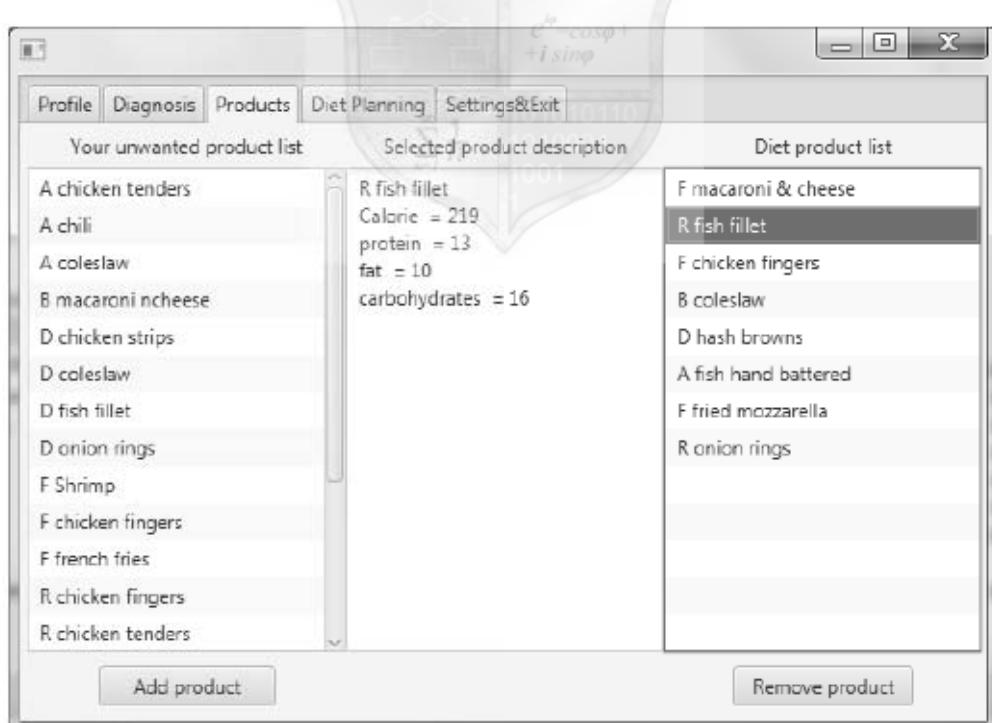


Рисунок 4.8.4 - Вікно перегляду списку продуктів, з яких буде складатися
раціон харчування

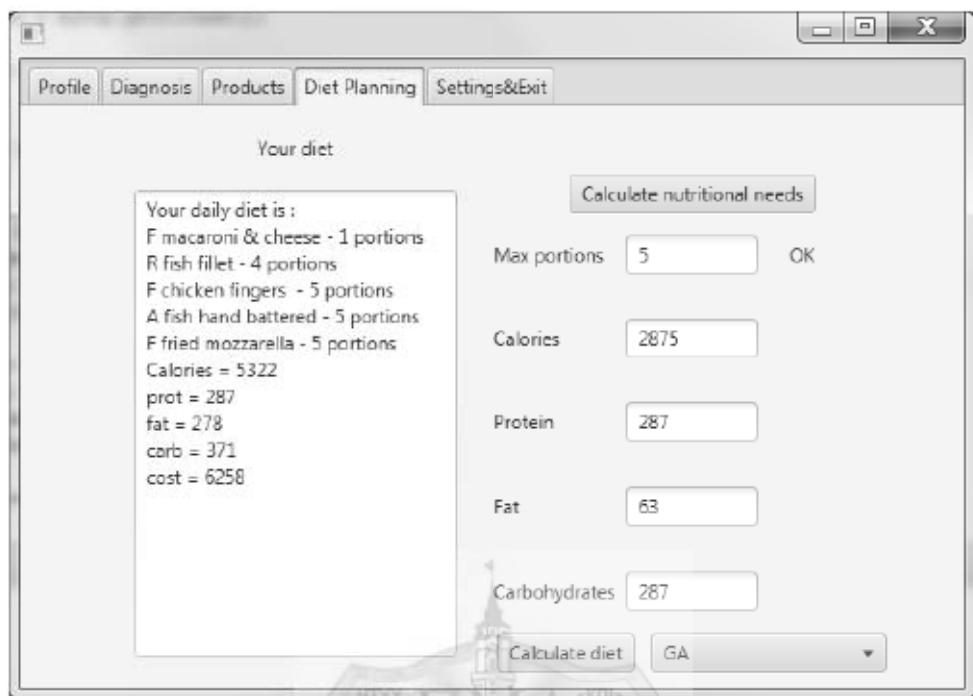


Рисунок 4.8.5 - Вікно отримання результатів

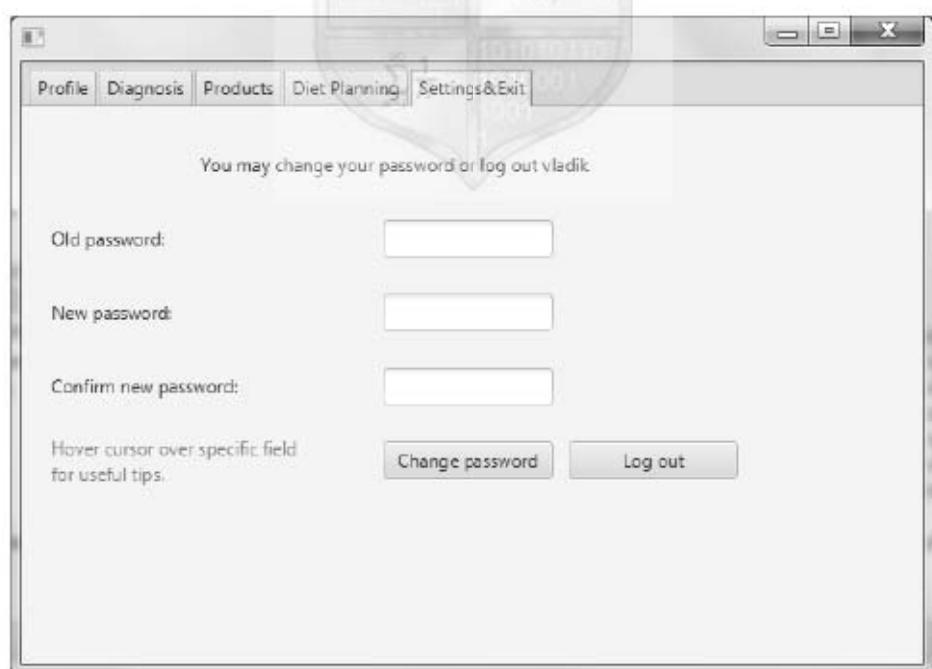


Рисунок 4.8.6 - Вікно зміни паролю та логауту

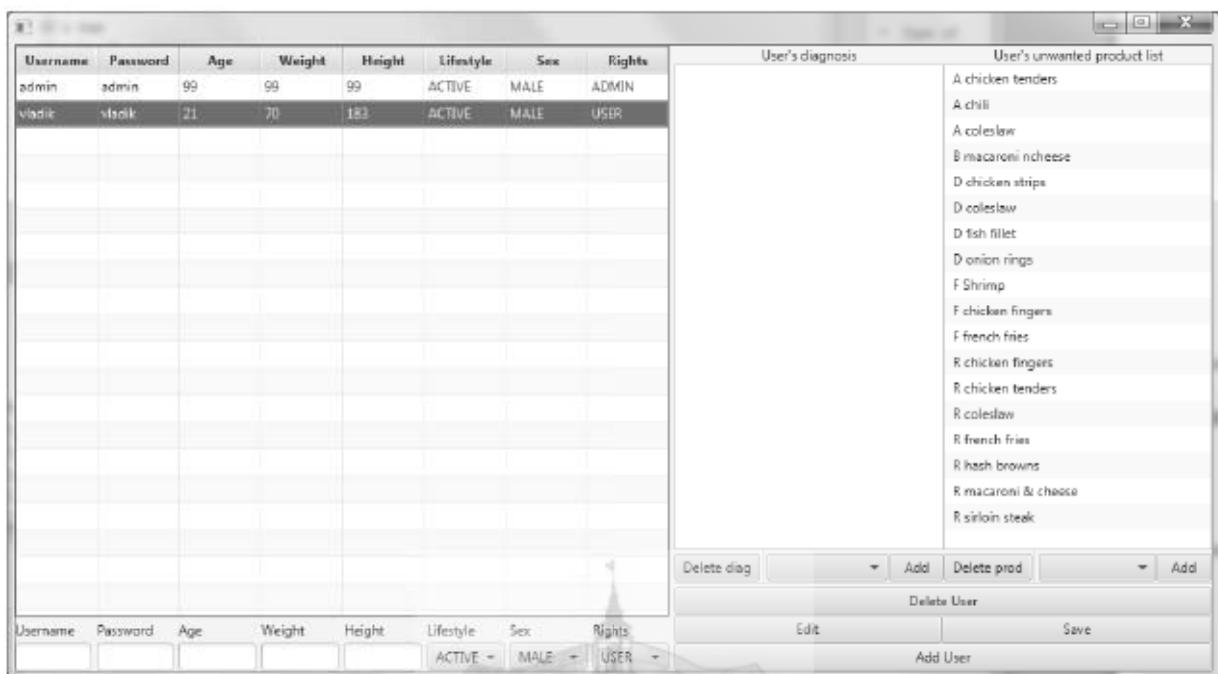


Рисунок 4.8.7 - Вікно роботи адміністратора

5 ТЕСТУВАННЯ КОМПОНЕНТІВ СИСТЕМИ

В даному розділі будуть наведені контрольні приклади, які цілком описані як задачі лінійного програмування.

Відповідю являється вектор X , елементами якого є кількість порцій відповідного продукту.

Оскільки в системі результати виводяться у вигляді більш зрозумілому для звичайного користувача, скріншоти, які будуть наведені, будуть показувати результат виконання обох методів в режимі відладки. Момент зупинки програми – одразу після отримання рішення.

5.1 Контрольні приклади



Контрольний приклад 1. Перевірка роботи симлес-методу та генетичного алгоритму

Мінімізувати $Z = 146 * x + 258 * y + 385 * z + 201 * d + 233 * k + 240 * j + 391 * h + 419 * l$

$$\left\{ \begin{array}{l} 121 * x + 219 * y + 330 * z + 175 * d + 197 * k + 202 * j + 333 * h + 356 * l \geq 2875 \\ 5 * x + 13 * y + 18 * z + 0 * d + 2 * k + 13 * j + 15 * h + 4 * l \geq 287 \\ 3 * x + 10 * y + 20 * z + 13 * d + 8 * k + 9 * j + 18 * h + 19 * l \geq 63 \\ 17 * x + 16 * y + 17 * z + 13 * d + 26 * k + 16 * j + 25 * h + 40 * l \geq 287 \\ x_i \leq 5 \end{array} \right.$$

Відповідь:

$$X = (1, 4, 5, 0, 0, 5, 5, 0)$$

Симплекс-метод:

Max portions	5	OK	solution = [int8]@1562{
Calories	2875	OK	[0] = 1
Protein	287	OK	[1] = 4
Fat	63	OK	[2] = 5
Carbohydrates	287	OK	[3] = 0
			[4] = 0
			[5] = 5
			[6] = 5
			[7] = 0

Рисунок 5.1.1 - Результат роботи симплекс-методу

Генетичний алгоритм:

```

Generation: 1 Fittest: 7418 Genes: 2 3 5 1 1 5 5 2
Generation: 2 Fittest: 7150 Genes: 3 5 4 0 3 5 4 1
Generation: 3 Fittest: 6771 Genes: 4 4 5 0 2 5 4 0
Generation: 4 Fittest: 6323 Genes: 4 5 5 0 0 4 4 0
Generation: 5 Fittest: 6305 Genes: 4 4 5 0 0 5 4 0
Generation: 6 Fittest: 6258 Genes: 1 4 5 0 0 5 5 0
Generation: 7 Fittest: 6258 Genes: 1 4 5 0 0 5 5 0
Solution found!

```

Рисунок 5.1.2 - Результат роботи генетичного алгоритму – найбільш оптимальний випадок

```

Generation: 1 Fittest: 7030 Genes: 0 5 5 1 3 4 5 0
Generation: 2 Fittest: 6944 Genes: 3 3 5 0 1 5 5 1
Generation: 3 Fittest: 6571 Genes: 0 5 5 1 0 5 5 0
Generation: 4 Fittest: 6571 Genes: 0 5 5 1 0 5 5 0
Solution found!

```

Рисунок 5.1.3 - Результат роботи генетичного алгоритму – прийнятне, але не найкраще рішення

Контрольний приклад 2. Тестування роботи симплекс-методу та генетичного алгоритму

Мінімізувати $Z = 146 * x + 258 * y + 385 * z + 201 * d + 233 * k + 240 * j + 391 * h + 419 * l$

$$\left\{ \begin{array}{l} 121 * x + 219 * y + 330 * z + 175 * d + 197 * k + 202 * j + 333 * h + 356 * l \geq 2875 \\ 5 * x + 13 * y + 18 * z + 0 * d + 2 * k + 13 * j + 15 * h + 4 * l \geq 287 \\ 3 * x + 10 * y + 20 * z + 13 * d + 8 * k + 9 * j + 18 * h + 19 * l \geq 63 \\ 17 * x + 16 * y + 17 * z + 13 * d + 26 * k + 16 * j + 25 * h + 40 * l \geq 287 \\ x_i \leq 7 \end{array} \right.$$

Відповідь: $\mathbf{X} = (0, 7, 6, 0, 0, 7, 0, 0)$

Симплекс-метод:



Рисунок 5.2.1 - Результат роботи симплекс-методу

Генетичний алгоритм:

```

Generation: 1 Fittest: 6701 Genes: 4 5 6 0 1 3 4 0
Generation: 2 Fittest: 6522 Genes: 2 5 5 2 0 6 3 0
Generation: 3 Fittest: 6258 Genes: 1 4 5 0 0 5 5 0
Generation: 4 Fittest: 6105 Genes: 1 7 4 0 0 6 3 0
Generation: 5 Fittest: 5811 Genes: 1 5 7 0 0 7 0 0
Generation: 6 Fittest: 5811 Genes: 1 5 7 0 0 7 0 0
Solution found!

```

Рисунок 5.2.2 - Результат роботи генетичного алгоритму – прийнятне, але не найкраще рішення

Контрольний приклад 3. Складання рациону здорового користувача

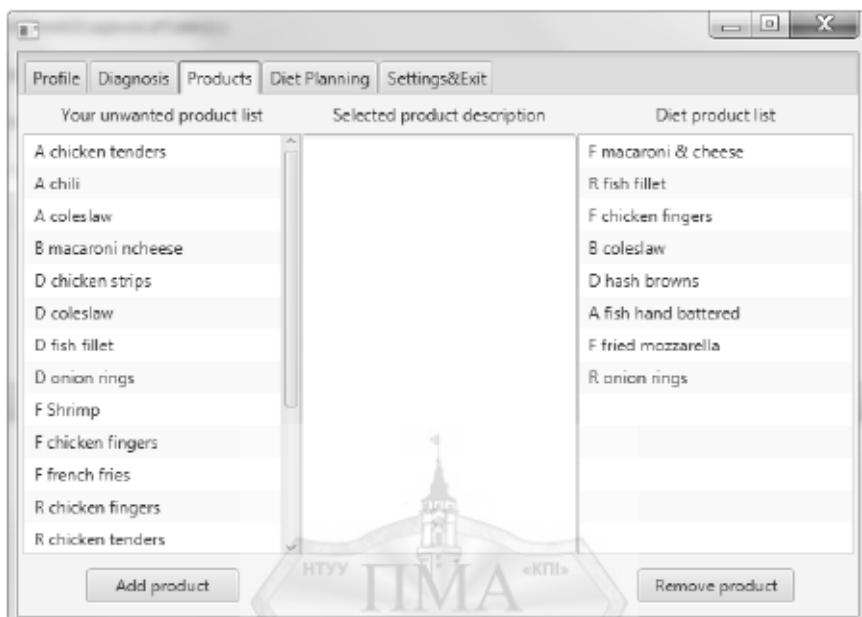


Рисунок 5.3.1 – Список продуктів користувача

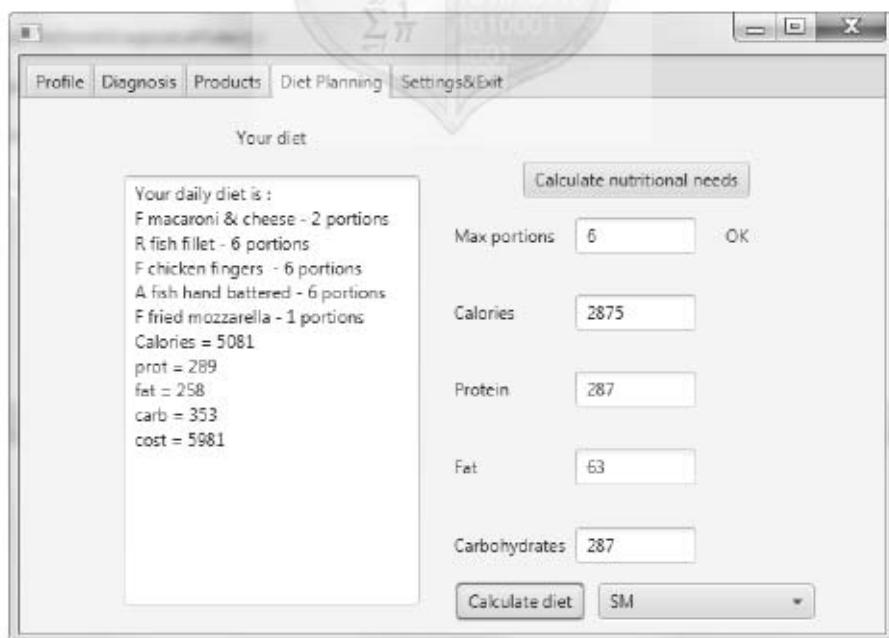


Рисунок 5.3.2 – Результат роботи - план харчування користувача

Контрольний приклад 4. Складання раціону харчування користувача, хворого на гастрит.

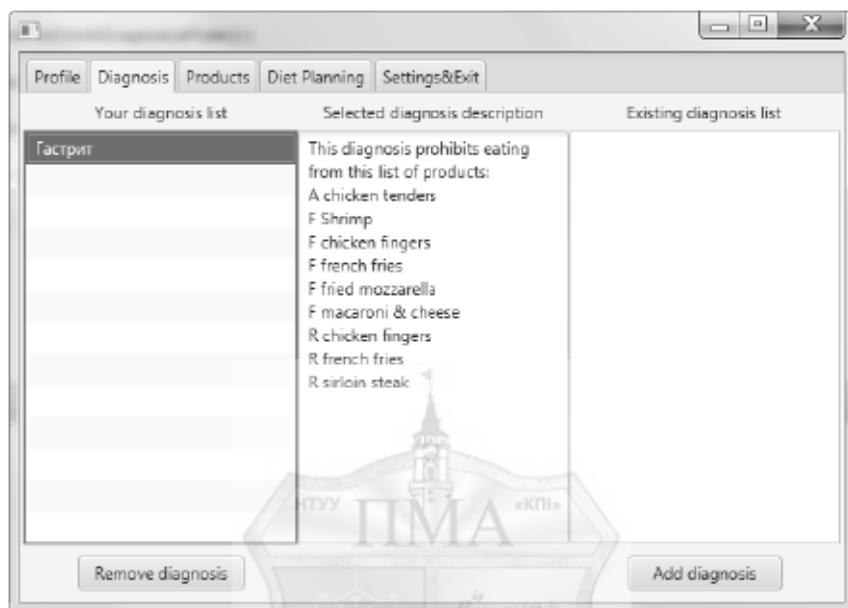


Рисунок 5.4.1 – Список продуктів, які не можна їсти користувачу

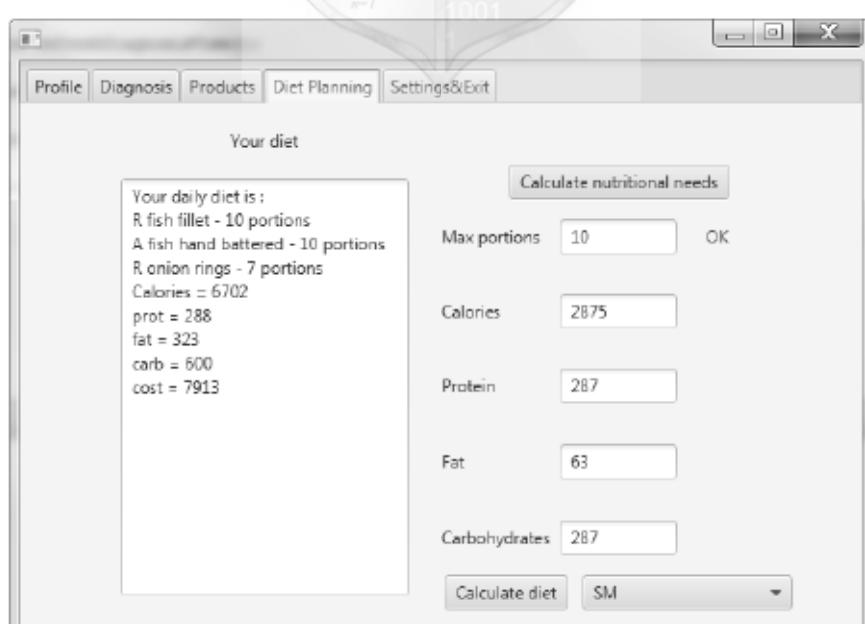


Рисунок 5.4.2 – План харчування користувача з гастритом

5.2 Висновок

Дані контрольні приклади підтверджують те, що симплекс-метод є найкращим методом для вирішення задач малої розмірності. Генетичний же алгоритм хоч і здатний знаходити найбільш оптимальне рішення, але частіше він буде приходити до рішення, яке є близьким до найбільш оптимального.

Для того, щоб покращити роботу генетичного алгоритму, можна збільшити вірогідність мутацій чи збільшити популяцію. Але обидва ці способи можуть значно збільшити час роботи алгоритму. Найбільш доцільним на даний момент буде кількаразовий запуск алгоритму, та обрання найкращого результату з отриманих.



ВИСНОВКИ

В даній роботі була розроблена автоматизована система зіставлення раціону харчування людини. Предметною областью є дані про користувачів системи, та набір даних про продукти, страви та діагнози, які можуть обмежувати вживання деяких з продуктів.

Алгоритмами складання раціону було обрано симплекс-метод і генетичний алгоритм, оскільки кожний із них може працювати краще за інший за різних умов. Було проведено порівняльний аналіз обох алгоритмів, та виведено умови, за яких кожний з алгоритмів показує свій найкращий результат.

Система може:

- зберігати персональні дані користувача
- розраховувати раціон харчування користувача
- враховувати дані користувача під час розрахунку раціону харчування, наприклад, хвороби чи стиль життя
- вираховувати dennу норму споживання корисних речовин

В майбутньому, система може покращена таким чином:

- ведення статистики (як в MyFitnessPal)
- підключення соц.. мережей
- вдосконалення генетичного алгоритму (врахування специфічних вподобань користувача, узгодженість продуктів один з одним)
- розширення бази даних їжі та хвороб, збільшення кількості їхніх параметрів
- підключення бази даних дієт

Дана система об'єднує найкращі риси вже існуючих систем, та має великі перспективи на розвиток.

Система реалізована на мові програмування Java.



ВИКОРИСТАНА ЛІТЕРАТУРА

1. Анализ и динамическая коррекция качества питания/Антоненко С.В./
«Тульский Государственный Университет», Тула - 2008, С. 11-12
2. База данных їжі та страв. Група Restaurant Foods [Електронний ресурс] –
Режим доступу: <http://ndb.nal.usda.gov/ndb/foods>
3. Автоматизация составления рациона питания/ Коробова К.А., Щербо
Н.М./ Журнал «Октябрь», г. Могилев - 2005, С. 1-2
4. Офіційний сайт програми myfitnesspal [Електронний ресурс] - Режим
доступу: www.myfitnesspal.com/ru
5. Сторінка безкоштовного сбірника дієт в магазині Google Play
[Електронний ресурс] - Режим доступу:
<https://play.google.com/store/apps/details?id=com.nvdevelopment.ref.dietrus.free&hl=ru>
6. Сторінка онлайн-калькулятору дієт [Електронний ресурс] - Режим
доступу: www.dietaonline.ru/articles/calc_calories_howto.php
7. Лекція. Архітектурні особливості розробки і проектування веб-аплікацій
[Електронний ресурс].-
<http://www.intuit.ru/studies/courses/611/467/lecture/20744>
8. Офіційна сторінка Java [Електронний ресурс] - Режим доступу:
<https://java.com/>
9. Офіційна сторінка Matlab [Електронний ресурс] - Режим доступу:
www.mathworks.com/products/matlab/
10. Офіційна сторінка Python [Електронний ресурс] - Режим доступу:
<https://www.python.org/>

- 11.Офіційна сторінка SQLite [Електронний ресурс] - Режим доступу:
<https://www.sqlite.org/>
- 12.Офіційна сторінка Oracle Database [Електронний ресурс] - Режим доступу:
<https://www.oracle.com/database/>
- 13.Основы исследования операций/ Вагнер Г.М./Изд. «Мир», Москва – 1972,
том 1, С. 104 - 110
- 14.An Evaluation of a Meal Planning System: Ease of Use and Perceived
Usefulness/ Johan Aberg//Linköpings universitet, Linköping, Sweden, C. 3-4
- 15.Computer-Based Dietary Menu Planning: How to Support It by Complex
Knowledge?/ Barbara Koroušić Seljak//Jožef Stefan Institute, Ljubljana,
Slovenia, C. 2-3
- 16.Основы исследования операций/ Вагнер Г.М./Изд. «Мир», Москва – 1972,
том 2, С. 246 - 250
- 17.Загальні відомості про генетичні алгоритми [Електронний ресурс] - Режим
доступу: http://www.znannya.org/?view=ga_general
- 18.Вирішення задачі пакування рюкзаку за допомогою Генетичного
Алгоритму [Електронний ресурс] - Режим доступу:
<http://www.intuit.ru/studies/courses/14227/1284/lecture/24170%3Fpage%3D4>
- 19.Бібліотека для реалізації симплекс-методу[Електронний ресурс] - Режим
доступу: <http://choco-solver.org>