

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДРАХУНКУ КАЛОРІЙ
CALORIE COUNTER SOFTWARE

Виконав: студент(ка) 4 курсу, групи КНТ-119сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Крутікова К.Д.

(прізвище та ініціали)

Керівник Гофман Є.О.

(прізвище та ініціали)

Рецензент Коцур М.І.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Крутікової Катерини Дмитрівни

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для підрахунку калорій. Calorie Counter Software

керівник проєкту (роботи) Гофман Євгеній Олександрович, к.т.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 7 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Гофман Є.О., доцент		
Нормоконтроль	Белова А.В., асистент		

7. Дата видачі завдання “ 25 ” квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки		
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	5 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	6 тиждень	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Крутікова К.Д.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Гофман Є.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
89 с., 36 рис., 3 табл., 4 дод., 17 джерел.

JAVA, POSTGRESQL, ЗДОРОВЕ ХАРЧУВАННЯ, МОВА
ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, КАЛОРІЯ

Об'єкт дослідження – процес розробки програмного забезпечення для підтримки здорового харчування.

Предмет дослідження – програмні засоби для підрахунку калорій.

Метою роботи є розробка програмного забезпечення для надання інформації про кількість калорій у їжі.

Матеріали, методи та технічні засоби: мова програмування Java, середовище розробки IntelliJ IDEA, система керування базами даних PostgreSQL.

Результати. Розроблено програмне забезпечення підрахунку кількості калорій, що виконує функції контролю процесів споживання та витрат калорій.

Практична цінність роботи полягає у розробці програмного забезпечення для надання інформації про кількість калорій у спожитій їжі.

Висновки. Розроблено програмне забезпечення для підрахунку кількості калорій та контролю процесів споживання та витрат калорій.

Галузь використання – здорове харчування, дієтологія.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 89 pages, 36 figures, 3 tables, 4 appendixes, 17 sources.

JAVA, POSTGRESQL, HEALTHY EATING, PROGRAMMING LANGUAGE, SOFTWARE, CALORIES

The object of research is the process of software development for a healthy diet maintaininig.

The subject of the research is software for counting calories.

The purpose of this work is to develop software to provide information about the number of calories in food.

Materials, methods and technical tools: Java programming language, IntelliJ IDEA development environment, PostgreSQL database management system.

Results. The software tools for control the consumption and consumption of calories have been developed.

The practical value of the work lies in the development of software to provide information on the number of calories in the food consumed.

Conclusions. The software for counting calories and controlling the processes of consumption and consumption of calories is developed.

Scope of use – healthy eating, nutrition.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Поняття про раціональне харчування та розрахунок кількості калорій .	11
1.2 Дослідження програмних засобів для розрахунку кількості калорій	22
1.3 Висновки за розділом 1	30
2 Матеріали і методи	31
2.1 Вибір мови програмування.....	31
2.2 Вибір середовища розробки для створення програмного забезпечення для підрахунку калорій.....	35
2.3 Висновки за розділом 2	37
3 Опис програми	38
3.1 Схема взаємодії основних компонентів програмного забезпечення для підрахунку калорій	38
3.2 Опис основних розроблених компонентів та модулів	39
3.3 Функціонування програмного забезпечення для підрахунку калорій	40
3.4 Висновки за розділом 3	43
4 Експлуатація, тестування та експериментальне дослідження програми.....	45
4.1 Призначення й умови застосування програми	45
4.2 Характеристики програми	46
4.3 Інструкція по експлуатації програми.....	46
4.3.1 Звернення до програми.....	46
4.3.2 Вхідні й вихідні дані.....	48
4.3.3 Повідомлення.....	49
4.4 Тестування програмного забезпечення для підрахунку калорій.....	49
4.5 Висновки за розділом 4	53
Висновки.....	54
Перелік джерел посилання	56

Додаток А Технічне завдання.....	58
Додаток Б Опис програми	62
Додаток В Текст програми	66
Додаток Д Слайди презентації	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

BMR – basal metabolic rate;
COM – component object model;
CMS – content management system;
CRUD – create, read, update, delete;
CSRF – cross site request forgery;
GUI – graphical user interface;
HDD – hard disk drive;
HTML – hypertext markup language;
IMDB – in-memory database;
J2EE – java 2 enterprise edition;
JS – javascript;
JVM – java virtual machine;
MVC – model-view-controller;
SQL – structured query language;
UML – unified modeling language;
XSS – cross-site scripting;
ООР – основний обмін речовин;
ОС – операційна система;
ПЗ – програмне забезпечення;
СКБД – система керування базами даних.

ВСТУП

Проблема здорового харчування та розроблення дієт для підтримки нормального стану організму людей є дуже важливою у наш час. Підрахунок калорій у процесі розроблення дієт для лікування різних хвороб є найважливішим показником при дотриманні правил здорового і раціонального харчування, що забезпечує прийнятний стан здоров'я людини та її здатність до праці.

У наш час існує досить багато різного програмного забезпечення, що дозволяють виконувати підрахунок калорій різноманітної їжі. Проте деякі програмні продукти мають недоліки у таких компонентах як синхронізація з сервером, лаконічність та привабливість інтерфейса, багатоплатформність, аналіз даних та їх подання у графічному вигляді, інші програми є досить дорогими для звичайних користувачів, що суттєво ускладнює їх застосування на практиці. Тому наявність недоліків існуючого програмного забезпечення обумовлює доцільність розробки нових програм для розрахунку калорій при споживанні та плануванні вживання певної їжі, що дозволить забезпечити здоровий спосіб життя.

У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для надання інформації про кількість калорій у їжі.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розрахунку кількості калорій спожитої їжі;
- здійснити проектування програмного забезпечення для надання інформації про для надання інформації про кількість калорій у їжі, що споживається;
- створити програмне забезпечення для розрахунку кількості калорій спожитої їжі;

– виконати тестування розробленого програмного забезпечення для надання інформації про кількість калорій у їжі, що споживається.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про раціональне харчування та розрахунок кількості калорій

Здоров'я та працездатність людини неможливо забезпечити без науково обґрунтованого раціонального харчування. Правильне, або раціональне харчування – це таке харчування, яке забезпечує зміцнення та покращення здоров'я, фізичних і духовних сил людини, попередження та лікування захворювань [1]–[7].

Кількість поживних речовин, необхідних людині, залежить від багатьох факторів: маси тіла, виконуваної роботи, віку, статі, умов навколишнього середовища. З їжі та води людина отримує поживні речовини. Сюди входять білки, жири, вуглеводи, мінерали, ціна вітамінів орлістат і неперетравлювані харчові волокна [1]–[7].

Склад і калорійність харчових продуктів нині добре вивчені, відомі найважливіші аспекти фізіології харчування людини. Це дозволяє точно орієнтуватися в питаннях раціонального харчування. Однак, дієтологія виявилася саме тим вченням, в якому, на жаль, немає згоди серед світових авторитетів з найелементарніших питань (наприклад, режим харчування тощо) [1]–[7].

Теорія раціонального, індивідуально збалансованого харчування визначає три напрямки діяльності: по-перше, кількість поживних речовин відповідно до їх калорійності відповідає енергетичним витратам організму; по-друге, якісний склад і співвідношення різних поживних речовин, необхідних для нормальної життєдіяльності організму; по-третє, основним принципом раціонального харчування є вживання різноманітної їжі, збалансованої за її кількістю та якістю окремо для кожної людини або для окремих груп людей, які виконують однорідну роботу, ведуть певний спосіб життя в даних умовах середовища [1]–[7].

Калорійність харчового раціону повинна відповідати витраті енергії.

Характеристика фізіологічної цінності основних харчових продуктів така [1]–[7]:

– молоко та кисломолочні продукти. Молоко є одним з найцінніших продуктів харчування. У ньому є всі необхідні речовини для нормального росту і розвитку організму. Молоко містить близько 100 різних компонентів і близько 20 амінокислот. У молочному жирі багато ненаситних жирних кислот, вітамінів. Він знаходиться в футлярі для емульгування. У молоці є холестерин і лецитин, які знаходяться в хорошому збалансованому стані. У молоці присутній молочний цукор – лактоза, яка сприяє засвоєнню кальцію. У молоці різні мінеральні речовини, оптимальна концентрація кальцію і фосфору. У молоці присутні вітаміни А, D, B2, B6. Кисломолочні продукти – це вершки, сметана, сир, йогурт. Засвоюється краще за молоко (за годину засвоєння 91% кисломолочних продуктів і лише 32% молока [1]–[7];

– м'ясо та м'ясні продукти. Це основне джерело білків, у ньому є жири, екстрактивні та мінеральні речовини, вітаміни. Відварене м'ясо краще впливає на органи травлення, ніж смажене на грилі і тушковане. Телятина, яловичина, кролик, курка, індичка перетравлюються краще, ніж м'ясо свині, гусака, качки [1]–[7];

– зернові продукти (зерно, хліб). Крупи виготовляють із зернових культур. У ньому багато вуглеводів, деяка кількість білків і невелика кількість жирів. У злаках є фосфор і кальцій, залізо, магній, вітаміни групи В. Існують крупи з вівса, проса, гречки, ячменю. Хліб містить 40-50% вуглеводів, 5-8% білків. У хлібі присутні білки. Чорний хліб більше впливає на секрецію шлунка, ніж білий хліб; свіжий хліб більше впливає на секрецію шлунка, ніж черствий хліб і шматок підсушеного хліба [1]–[7];

– овочі, фрукти та ягоди. Овочі є основним джерелом вуглеводів у разі цукру, крохмалю, пектину. У моркві присутня близько 6 % вуглеводів, у буряку близько 8 % вуглеводів. В овочах міститься 75-95% води. В овочах також у великій кількості присутні вітаміни та мінеральні речовини. Овочі стимулюють секрецію шлунка, моторну функцію травного тракту, вироблення

жовчі. Це сприяє засвоєнню білків, жирів, вуглеводів і вітамінів. Плоди та ягоди мають багато вітамінів, мінеральних солей. Близько 85% води міститься у фруктах і дрібних плодах (ягодах). У них є органічні речовини та пектин [1]–[7].

Засвоєння їжі – це співвідношення між кількістю їжі, яку засвоює організм, до загальної кількості їжі, яку було спожито. Освоюється близько 95% продуктів тваринного походження, близько 80% продуктів рослинного походження, близько 82-90% сумішей. Засвоєння білка з продуктів тваринного походження становить 97 %, із рослинних – 85 %, із сумішей – 92 % [1]–[7].

Ізодинаміка поживних речовин та їх калорійність. Амінокислоти, жирні кислоти, моносахариди з'єднуються в енергетичному обміні. Все це має свою енергетичну цінність. Калорійний коефіцієнт жирів 9,0 ккал/г, білків 4,0 ккал/г, вуглеводів 3,75 ккал/г. Але необхідно врахувати, що необхідно споживати жири, білки, вуглеводи, оскільки у випадку, коли споживається тільки один продукт, може бути лише заміщення енергії і організму для життєдіяльності необхідне все харчування [1]–[7].

Принципи складання поживних раціонів з урахуванням кількості калорій [1]–[7]:

– відповідність калорійності денного раціону втратам енергії
Калорійність добового раціону повинна дорівнювати енергетичним витратам організму. Людині першої професійної групи необхідно 40 ккал/кг ваги; людині другої професійної групи потреба 43 ккал/кг ваги; людині третьої професійної групи потреба 46 ккал/кг ваги; особам четвертої професійної групи потрібно 53 ккал/кг ваги; людині п'ятої професійної групи потрібно 61 ккал/кг ваги [1]–[7];

– режим харчування Якщо ми їмо чотири рази на день: сніданок повинен складати 25-30% добової калорійності, обід (після вечері) або перед вечерею повинен мати 10-15%, вечеря повинна мати 40-45% добової калорійності, вечеря повинна бути 20 % від добової калорійності. Якщо ми їмо 3 рази на день: сніданок має становити 30% добової калорійності, вечеря

повинна мати 45-50% добової калорійності, а вечеря повинна мати 20-25% добової калорійності. Час між сніданком і вечерею має становити 5-6 годин; час між вечерею і вечерею має становити 6-7 годин. Оптимальна тривалість між прийомами їжі 4-5 годин, вночі 8-10 годин [1]–[7];

– повнота і узгодженість раціону – їжа повинна мати оптимальне співвідношення білків, жирів, вуглеводів, вітамінів, мінеральних речовин. Оптимальна норма білка – 1,5 г/кг ваги на добу. Білки повинні давати 14-15% добової калорійності. Оптимальна норма жиру – 1,5 г/кг ваги на добу. Жири повинні давати 28-30% добової калорійності. Середня норма вуглеводів – 400-500 грам на добу. Вони забезпечують 50-60% добової калорійності. Співвідношення білків до жирів та вуглеводів становить 1:0,8:3 або 1:1:4 для людей, які мають високу розумову активність; 1:1:5 для людей з фізичною активністю [1]–[7];

– врахування індивідуально-фізіологічних особливостей організму. Сніданок необхідно починати з салату, який активізує травну секрецію. Далі має бути їжа з основним джерелом енергії та харчування, наприкінці має бути загальнозміцнюючий напій (чай, кава, какао). Вечеря потрібно починати з салату (коли у вас менше апетиту). Треба пам'ятати, що вечеря починається не з їжі, яка зменшує секрецію шлунка. Вечеря повинна складатися з їжі, яка добре перетравлюється (риба, молоко, яйця), напоїв, які не стимулюють центральну нервову систему [1]–[7].

Метаболізм вуглеводів: жири містять переважно вуглець і водень, трохи кисню, а іноді й інші атоми. Трьома основними формами жиру, що містяться в їжі, є гліцериди (переважно триацилгліцерин «тригліцерид», форма, в якій жир зберігається як паливо), фосфоліпіди та стерини (переважно холестерин). Жири забезпечують 9 кілокалорій на грам (ккал/г), порівняно з 4 ккал/г для вуглеводів і білків. Триацилгліцерин, у формі хіломікронів (мікроскопічних ліпідних частинок) або інших ліпопротеїнів, не поглинається безпосередньо жодною тканиною, а повинен гідролізуватися поза клітиною до жирних кислот і гліцерину, які потім можуть потрапити в клітину [1]–[7].

Жирні кислоти надходять з дієти, адипоцитів (жирових клітин), вуглеводів і деяких амінокислот. Після перетравлення більшість жирів переноситься в кров у вигляді хіломікронів. Основними шляхами метаболізму ліпідів є ліполіз, бетаокислення, кетоз і ліпогенез. Ліполіз (розщеплення жирів) і бета-окислення відбувається в мітохондріях. Кетоз виникає, коли швидкість утворення кетонів печінкою перевищує здатність тканин їх окислювати. Виникає при тривалому голодуванні і при вживанні великої кількості жирів при нестачі вуглеводів [1]–[7].

Жири є концентрованим джерелом енергії, оскільки вони дають вдвічі більше енергії, ніж вуглеводи або білки за вагою. Функції жирів полягають у тому, щоб входити до структури клітин, утворювати захисну подушку та теплоізоляцію навколо життєво важливих органів, переносити жиророзчинні вітаміни та забезпечувати резервний запас енергії. Три незамінні ненасичені жирні кислоти включають: лінолеву, лінолінову та арахідонову та мають 2, 3 та 4 подвійні зв'язки відповідно. Насичені жири разом з холестерином причетні до атеросклерозу, затвердіння артерій. З цієї причини в раціоні слід зменшити кількість насичених жирів (тваринних) і збільшити кількість ненасичених жирів (рослинні) [1]–[7].

Жир надходить у структуру мембран клітин. Жир необхідний для структури стероїдних гормонів. Жири – джерело енергії. Перетворення жирів в організмі людини, всмоктуються у разі жирних кислот з коротким і довгим ланцюгом і гліцерину з травного тракту. Жирні кислоти з коротким ланцюгом і гліцерин надходять у кров, потім до жирового депо, серця, печінки. Жирні кислоти з довгим ланцюгом надходять в слизову оболонку кишечника, де тригліцериди всмоктуються в хілімікронах. Потім вони потрапляють в лімфу і кров [1]–[7].

Регуляція жирового обміну: центральний механізм регуляції діє на гіпоталамус. Це активізує гіпофіз, який виробляє гормон росту; активізує щитовидну залозу та надниркові залози. Гіпоталамус має парасимпатичний та симпатичний центри. Парасимпатичний вплив, інсулін, глюкокортикоїди

мають симанаболічну дію. впливи, гормони щитовидної залози, глюкагон, адреналін, гормон росту мають катаболічний ефект [1]–[7].

Вітаміни та мінерали змушують організм людей працювати правильно. Незважаючи на те, що людина отримує вітаміни та мінерали з їжі, яку споживає щодня, деякі продукти містять більше вітамінів і мінералів, ніж інші [1]–[7].

Вітаміни діляться на дві категорії: жиророзчинні і водорозчинні. Жиророзчинні вітаміни — А, D, Е і К — розчиняються в жирі і можуть зберігатися в організмі. Водорозчинні вітаміни — С і вітаміни В-комплексу (такі як вітаміни В6, В12, ніацин, рибофлавін і фолат) — повинні розчинитися у воді, перш ніж організм зможе їх засвоїти. Через це тіло людини не може зберігати ці вітаміни. Будь-який вітамін С або В, який тіло людини не використовує під час проходження через систему, втрачається. Тому щодня людині потрібен свіжий запас цих вітамінів [1]–[7].

Вітаміни — це органічні речовини (виробляються рослинами чи тваринами), мінерали — це неорганічні елементи, які надходять із ґрунту та води та поглинаються рослинами чи поїдаються тваринами. Тіло людини потребує більшої кількості деяких мінералів, таких як кальцій, щоб рости та залишатися здоровими. Інші мінерали, такі як хром, мідь, йод, залізо, селен і цинк, називаються мікроелементами, тому що людині потрібна лише невелика кількість їх щодня [1]–[7].

Вітаміни та мінерали зміцнюють імунну систему, підтримують нормальний ріст і розвиток, а також допомагають клітинам і органам виконувати свою роботу. Наприклад, морква корисна для очей. Морква сповнена речовин, які називаються каротиноїдами, які тіло перетворює у вітамін А, що допомагає запобігти проблемам з очима [1]–[7].

Вітамін К сприяє згортанню крові та знаходиться у зелених листових овочах, брокколі та соєвих бобах. Щоб мати міцні кістки, людині потрібно їсти такі продукти, як молоко, йогурт і зелені листові овочі, які багаті мінералом кальцію [1]–[7].

Вживання різноманітної їжі – найкращий спосіб отримувати всі необхідні щодня вітаміни та мінерали, а також правильний баланс вуглеводів, білків, жирів і калорій. Цілі або необроблені продукти, такі як свіжі фрукти та овочі, цільні зерна, нежирні молочні продукти, нежирне м'ясо, риба та птиця, є найкращим вибором для забезпечення поживними речовинами, необхідними організму людини, щоб залишатися здоровим і правильно рости [1]–[7].

У випадку, якщо людина є вегетаріанцем, потрібно ретельно планувати дієту, яка містить вітаміни та мінерали, які містяться в основному в м'ясі. Найкращими джерелами мінералів цинку та заліза є м'ясо, риба та птиця. Однак можна отримати цинк і залізо в сушених бобах, насінні, горіхах і зелених листових овочах, таких як капуста [1]–[7].

Вітамін B12, важливий для виробництва еритроцитів, не міститься в рослинній їжі. Якщо людина не їсть м'ясо, його можна знайти у яйцях, молоці та інших молочних продуктах, а також збагачених сухих сніданках. Веганам (вегетаріанцям, які взагалі не вживають продуктів тваринного походження, включаючи молочні продукти) може знадобитися приймати вітамінні добавки [1]–[7].

Хоча всі вони вважаються мікроелементами, вітаміни та мінерали відрізняються за основними ознаками. Вітаміни є органічними і можуть розщеплюватися під дією тепла, повітря або кислоти. Мінерали є неорганічними і зберігають свою хімічну структуру [1]–[7].

Мінерали в ґрунті та воді легко потрапляють у тіло людини через рослини, рибу, тварин і рідини, які ви споживаєте. Але важче переносити вітаміни з їжі та інших джерел у ваше тіло, оскільки приготування, зберігання та просте перебування на повітрі можуть інактивувати ці більш крихкі сполуки [1]–[7].

Вітамін А забезпечує нормальний ріст і розвиток, бере участь у синтезі пігменту зору, забезпечує безпечну адаптацію очей до світла [1]–[7].

Вітамін А робить набагато більше, ніж допомагає бачити в темряві. Він стимулює виробництво та активність білих кров'яних тілець, бере участь у

відновленні кісток, допомагає підтримувати здоров'я ендотеліальних клітин (ті, що вистилають внутрішні поверхні тіла), а також регулює ріст і поділ клітин. Ця остання роль змушувала дослідників роками досліджувати зв'язок між вітаміном А і раком. Зокрема, дослідники досліджували, чи можуть люди знизити ризик розвитку раку, приймаючи добавки бета-каротину, одного з кількох попередників, які організм може трансформувати у вітамін А, або приймаючи активну форму вітаміну А (також званого ретинолом або попередньо сформованим вітаміном) [1]–[7].

Нині Інститутом медицини рекомендоване споживання вітаміну А становить 900 мкг ретинолу для чоловіків (еквівалентно 3000 МО) і 700 мкг ретинолу для жінок (еквівалентно 2333 МО). Верхня межа споживання вітаміну А з ретинолом становить 3000 мікрограмів, але таке високе споживання може підвищити ризик перелому стегна або порушити корисну дію вітаміну D [1]–[7].

Джерела їжі: багато сухих сніданків, соки, молочні продукти та інші продукти збагачені ретинолом (також відомим як попередньо сформований вітамін А). Багато фруктів і овочів, а також деякі харчові добавки також містять бета-каротин та інші апрекурсори вітаміну, які організм може перетворити у вітамін А. Найкраще вибрати полівітамінну добавку, яка містить весь або переважну більшість вітаміну А у формі бета-каротин [1]–[7].

Хоча можна отримати занадто мало вітаміну А, легко отримати занадто багато попередньо сформованого вітаміну А (ретинолу) з добавок. Вважається безпечним споживання до 3000 мікрограмів попередньо сформованого вітаміну А, що більш ніж у три рази перевищує поточну рекомендовану добову норму. Однак є деякі докази того, що цей вітамін А може збільшити ризик перелому стегна або деяких вроджених вад. Іншою причиною уникати надлишку попередньо сформованого вітаміну А є те, що він може перешкоджати корисній дії вітаміну D [1]–[7].

На відміну від попередньо сформованого вітаміну А, бета-каротин не є токсичним навіть при високому рівні споживання. За потреби організм може

утворювати вітамін А з бета-каротину, і немає необхідності стежити за рівнем споживання, як це відбувається з попередньо сформованим вітаміном А. Тому краще вибрати полівітамінну добавку, яка містить всю або переважну більшість його вітамін А у вигляді бета-каротину; багато виробників полівітамінів вже зменшили кількість попередньо сформованого вітаміну А у своїх продуктах. Курцям слід уникати одноразових високих доз бета-каротину, оскільки деякі рандомізовані дослідження за участю курців пов'язували прийом високих доз із підвищеним ризиком раку легенів. Немає вагомих причин приймати окремі добавки з бета-каротином [1]–[7].

Вітамін D регулює обмін кальцію і фосфору. Вітамін Е має антиоксидантну дію на ліпіди клітин. Вітамін К забезпечує нормальні процеси гемостазу) [1]–[7].

Замість того, щоб легко проникати в кров, як більшість водорозчинних вітамінів, жиророзчинні вітаміни потрапляють в кров через лімфатичні канали в стінці кишечника (див. ілюстрацію). Багато жиророзчинних вітамінів проходять через організм лише під супроводом білків, які діють як носії [1]–[7].

Всмоктування жиророзчинних вітамінів: вживається їжа, що містить жиророзчинні вітаміни. Їжа перетравлюється кислотою шлунка, а потім потрапляє в тонкий кишечник, де перетравлюється далі. Жовч необхідна для засвоєння жиророзчинних вітамінів. Ця речовина, що виробляється в печінці, надходить в тонкий кишечник, де розщеплює жири. Потім поживні речовини всмоктуються через стінку тонкої кишки. Після всмоктування жиророзчинні вітаміни потрапляють в лімфатичні судини, перш ніж потрапляти в кров. У більшості випадків жиророзчинні вітаміни повинні бути поєднані з білком, щоб переміщатися по організму. Ці вітаміни використовуються в усьому організмі, але надлишки відкладаються в печінці та жирових тканинах. Оскільки необхідна додаткова кількість цих вітамінів, ваше тіло використовує запаси, викидаючи їх у кров із печінки [1]–[7].

Чотири принципи раціонального харчування [1]–[7]:

– необхідно вживати їжу 3-4 рази день, крім того можна робити 1-2 перекуси щодня, щоб підтримувати високу швидкість метаболізму і збалансоване споживання поживних речовин. Їжу або калорійні закуски можна вживати з інтервалом приблизно в 4 години. Необхідно планувати їжу заздалегідь і дбати про те, щоб якісні страви завжди були в наявності. Найважливіші вікна можливостей відбуваються вранці, а також під час і після тренування [1]–[7];

– вода є основою харчової піраміди та є найбільш поширеною та важливою поживною речовиною (складає понад дві третини маси тіла). Необхідно запобігати спразі, а не втамовувати її. Спрага відстає від реальної потреби; і насправді зневоднення вже погіршує вашу продуктивність до того часу, коли ви відчуваєте спрагу. Оскільки воду можна втратити швидше, ніж її замінити, необхідно пити рідину при кожній нагоді. Вуглеводно-білково-електролітний напій під час і після тренування також може мінімізувати катаболізм і втому [1]–[7];

– необхідно вживати різноманітну їжу, наголошуючи на тій, що лежить в основі піраміди харчових орієнтирів, особливо складні, багаті клітковиною, нерафіновані вуглеводи, нежирні білки та необхідні ненасичені жири. Кожна група продуктів містить важливі біологічно активні сполуки. Необхідно побудувати своє харчування за правилом 3-2-1. 3 частини вуглеводів. Вуглеводи живлять нервово-м'язову систему, запасні білки та основний жировий обмін. Вуглеводи забезпечують 4,1 калорій на грам і повинні складати 55-65% калорійності кожного прийому їжі. 2 частини білка: амінокислоти є структурними будівельними блоками для росту та відновлення тканин, забезпечують енергією, буферний кислотно-лужний баланс і регулюють баланс рідини та об'єм плазми. Вони забезпечують 4,35 калорій на грам і повинні складати 15-25% споживання калорій. 1 частина жиру. Незамінні жирні кислоти виконують життєво важливі функції: енергію, транспортування/зберігання поживних речовин, гормональну/клітинну структуру, амортизацію, захист та ізоляцію. Вони забезпечують 9 калорій на

грам, що вдвічі більше, ніж вуглеводи або білки. Жири повинні складати 20-25% споживання калорій, більшість з яких є ненасиченими маслами з рослинних горіхів [1]–[7];

– спосіб приготування їжі так само важливий, як і те, що людина їсть. Краще вживати запечену, варену, смажену, сиру або приготовану на пару їжу [1]–[7].

Кожному організму для нормального функціонування, щоб бути здоровим, необхідно споживати певну кількість калорій. Це значною мірою залежить від кількості фізичної активності, яку людина виконує щодня [1]–[7].

Деякі фактори, які впливають на кількість калорій, необхідних людині для збереження здоров'я, включають вік, вагу, зріст, стать, рівень фізичної активності та загальний стан здоров'я. Наприклад, фізично активний 25-річний чоловік, який має зріст 1,8 метри, потребує значно більшого споживання калорій, 1,5 метрова сидяча 70-річна жінка. Дорослим чоловікам, як правило, потрібно вживати 2000-3000 калорій на день для підтримки ваги, тоді як дорослим жінкам потрібно приблизно 1600-2400 калорій [1]–[7].

Організму не потрібно багато калорій, щоб просто вижити. Однак споживання занадто малої кількості калорій призводить до поганого функціонування організму, оскільки воно витрачатиме калорії лише на функції, необхідні для виживання, і ігнорує ті, які необхідні для загального здоров'я та благополуччя [1]–[7].

У наш час вже є добре відомою кількість калорій, що міститься у певній їжі [1]–[7]. Так, наприклад, калорійність білого цукру складає 17 калорій в 1 чайній ложці, калорійність консервованої квасолі – 96 калорій у 100 г, калорійність оливкової олії – 163 калорії в 1 столовій ложці, калорійність курячої грудки – 164 калорії на 100 г і т.ін. [1]–[7].

Тому з метою підтримки здорового способу харчування у наш час використовується програмне забезпечення для розрахунку кількості калорій.

1.2 Дослідження програмних засобів для розрахунку кількості калорій

У наш час використовується багато програмних засобів для підтримки здорового харчування та розрахунку кількості калорій. Виконаємо деякі з них.

Провідну позицію у списку найбільш популярних застосунків для підрахунку калорій займає My FitnessPal. За заявою розробників, програма має найбільшу базу даних, яка поповнюється щодня. У застосунку реалізований повний комплект функцій: зручна статистика та звіти про динаміку, сканер, статистика [8].

Ще одним зручним моментом My FitnessPal є повна синхронізація з сайтом: заповнювати щоденник можна як з комп'ютера, так і з телефону [8]. Додаток безкоштовний, проте окремі додаткові функції доступні тільки платно (рис. 1.1).

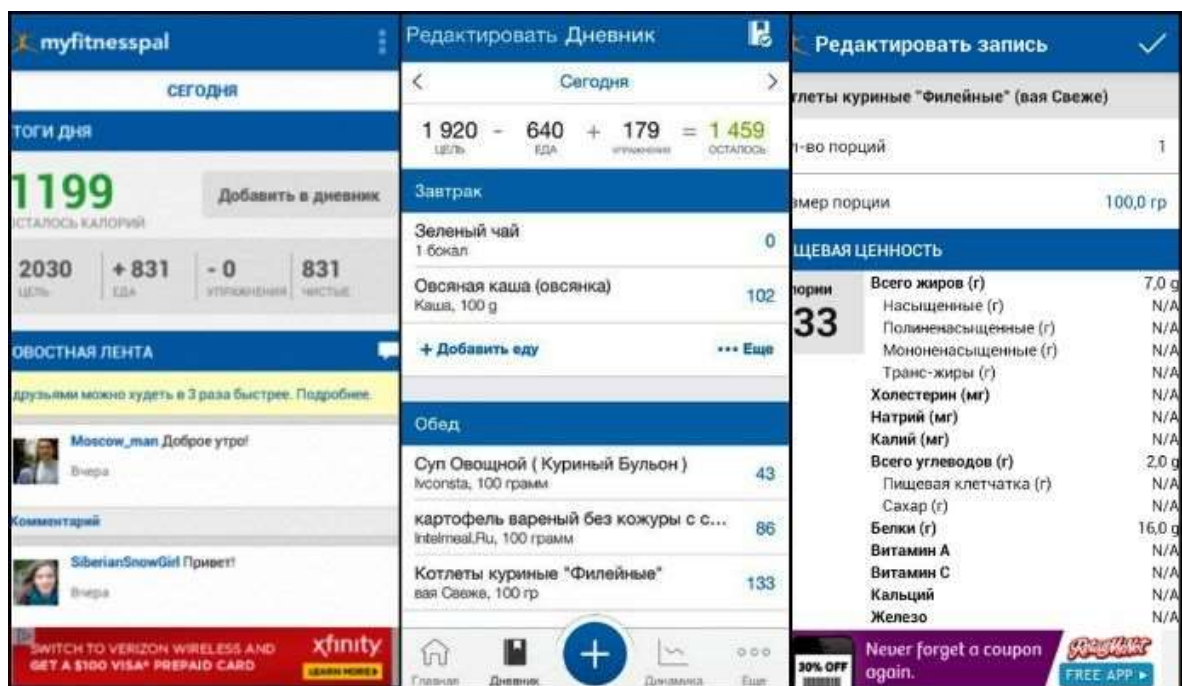


Рисунок 1.1 – Сервіс My FitnessPal

Основні переваги сервісу My FitnessPal [8]:

- зручний інтерфейс;

- наявність сканеру їжі та штрихкодів;
- наявність рекомендацій для тренувань.

Недоліки сервісу My FitnessPal [8]:

- присутній лише під ОС (операційну систему) Android;
- відсутність аналізу даних та їх подання у графічному вигляді.

Програмне забезпечення Lose It! є трекером здоров'я, який включає простий у використанні щоденник харчування та журнал вправ. Також є можливість підключати крокомір або інший фітнес-пристрій. Виходячи з ваги, зросту, віку та цілей, програмне забезпечення Lose It! надає індивідуальну рекомендацію щодо споживання калорій. Програмне забезпечення Lose It! містить повну базу даних харчових продуктів і піктограму, що представляє кожну їжу (рис. 1.2). Харчовий щоденник простий і зручний у використанні. Додати нові продукти не складно [9].

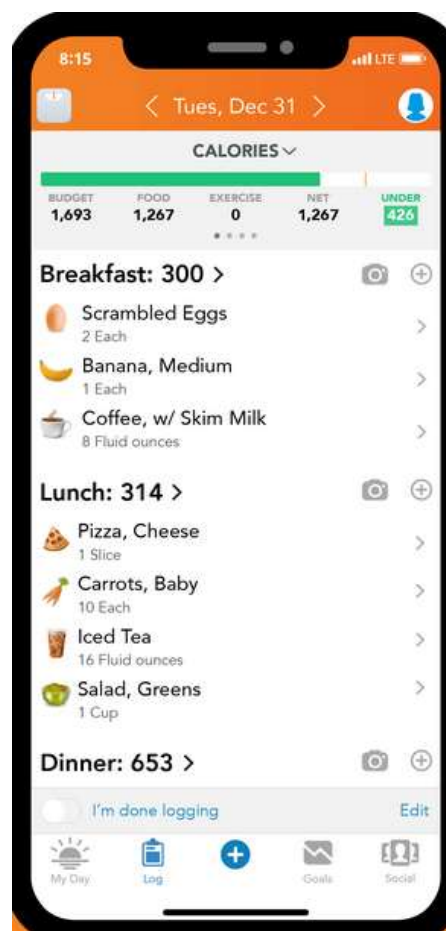


Рисунок 1.2 – Програмне забезпечення Lose It!

Крім того, у застосунку Lose It! є сканер штрих-коду для упакованих продуктів, а звичайні продукти зберігаються для швидкого введення пізніше. Програмне забезпечення Lose It! відображає зміни ваги на графіку, надає доступ до активної чат-спільноти та зберігає щоденні та тижневі підсумки. Його вкладка під назвою «виклики» дозволяє вам брати участь у дістичних завданнях або приготувати свої власні [9].

Перевагами програмного забезпечення Lose It! є такі [9]:

- має базу даних харчових продуктів із популярними ресторанами, продуктовими магазинами та фірмовими продуктами, усі з яких перевірено командою експертів;
- програмне забезпечення Lose It! дозволяє встановлювати нагадування для запису їжі та закусок.

Недоліками програмного забезпечення Lose It! є такі [9]:

- важко зареєструвати домашні страви або розрахувати їх харчову цінність;
- застосунок може бути складним у навігації;
- не відстежує мікроелементи.

Програмне забезпечення FatSecret є безкоштовним лічильником калорій, який містить харчовий щоденник, базу даних харчування, здорові рецепти, журнал вправ, таблицю ваги та журнал [10].

Сканер штрих-коду допомагає відстежувати упаковані продукти. На домашній сторінці показано загальне споживання калорій, а також розподіл вуглеводів, білків і жирів, які відображаються як за день, так і для кожного прийому їжі. FatSecret пропонує місячний підсумковий перегляд, який показує загальну кількість спожитих калорій за день і загальні середні значення за кожен місяць. Ця функція може бути зручною для відстеження вашого загального прогресу.

Цей лічильник калорій дуже зручний у використанні. Він також включає чат-спільноту, де користувачі можуть обмінюватися історіями успіху та отримувати поради, рецепти тощо (рис. 1.3).



Рисунок 1.3 – Програмне забезпечення FatSecret

FatSecret пропонує функцію під назвою «виклики», за допомогою якої користувачі можуть створювати або брати участь у дієтичних завданнях у закритій групі людей.

Перевагами програмного забезпечення FatSecret є такі [10]:

- база даних продуктів харчування є вичерпною, включаючи багато продуктів харчування супермаркетів та ресторанів;
- продукти, надіслані іншими користувачами, виділені, щоб користувачі могли перевірити, чи точна інформація;
- FatSecret може представляти чисті вуглеводи, які можуть стати в нагоді для тих, хто дотримується низьковуглеводної дієти;
- програмне забезпечення є безкоштовним.

Недоліком програмного забезпечення FatSecret є такі [10] є те, що інтерфейс досить захащений і заплутаний.

Програмне забезпечення Cronometer дозволяє легко відстежувати свій раціон, вправи та вагу тіла [11]. Воно пропонує точні розміри порцій і корисну

базу даних вправ. Якщо людина є вагітною, то можна вибрати індивідуальний профіль на основі більшої потреби в калоріях. Також Cronometer може враховувати те, що людина дотримується певної дієти, наприклад, палео дієти, дієти з низьким вмістом вуглеводів або вегетаріанської дієти з низьким вмістом жиру. Це змінює рекомендації щодо макронутрієнтів. Харчовий щоденник дуже простий і зручний у використанні (рис. 1.4). Під ним можна знайти стовпчасту діаграму, яка показує розподіл вуглеводів, жирів і білків за цей день разом із загальними спожитими калоріями [11].

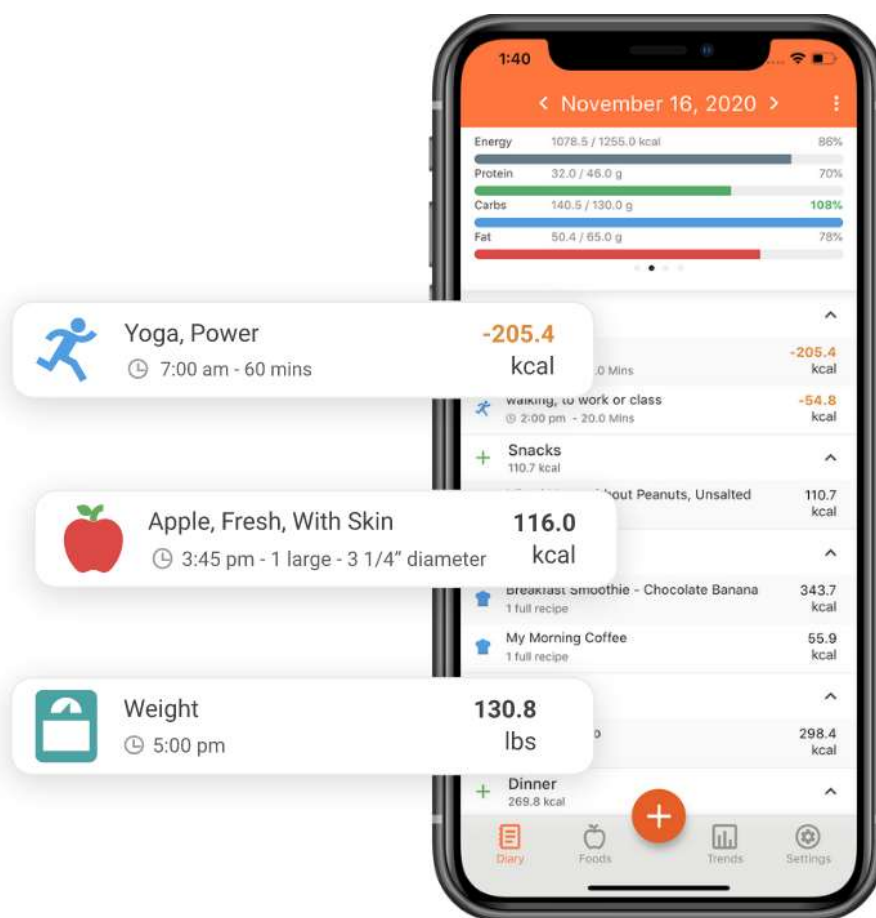


Рисунок 1.4 – Програмне забезпечення Cronometer [11]

Cronometer особливо корисний для відстеження мікроелементів, таких як вітаміни та мінерали [11].

Перевагами програмного забезпечення Cronometer є такі [11]:

- простий у використанні;

– є можливість синхронізувати дані з пристроїв охорони здоров'я з програмою та імпортувати вагу, відсоток жиру в організмі, дані про сон і діяльність;

– відстеження всіх мікроелементів, такі як вітаміни, мінерали та мікроелементи.

Недоліками програмного забезпечення Cronometer є такі [11]:

- не ділить харчовий щоденник на прийоми їжі;
- рецепт домашнього приготування можна додати лише на веб-сайт, а не в додаток. Однак після цього їжа буде доступна в додатку;
- не має соціальної спільноти користувачів;
- не є безкоштовним.

Програмне забезпечення Lifesum – це застосунок для підрахунку калорій, який порадує привабливим дизайном. У програмі велика база, можливість додавання рецептів із зазначенням порцій і пристрій для зчитування штрих-кодів (рис. 1.5). Lifesum також запам'ятовує, які продукти ви з'їли, і це додатково спрощує контроль харчування [12].

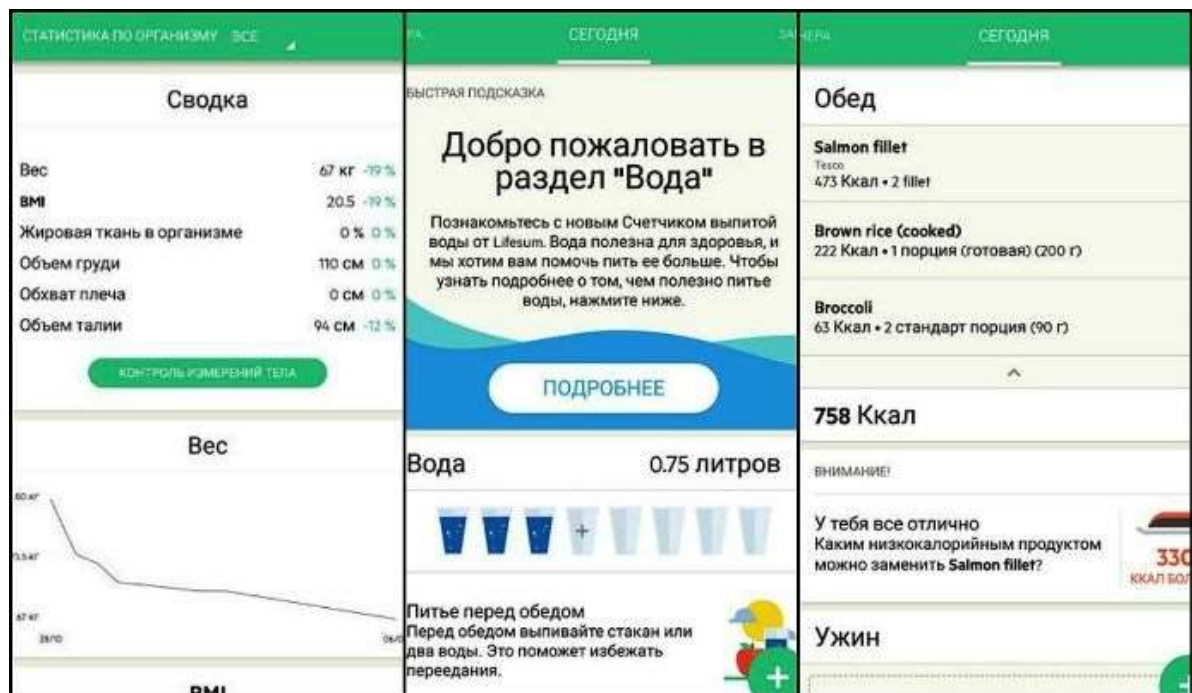


Рисунок 1.5 – Програмний продукт LifeSum [12]

Програма безкоштовна, але можна придбати преміум-аккаунт, в якому вам відкриється доступ до додаткової інформації по продуктам, облік обсягів тіла і відсоток жирової тканини, рейтинг продуктів. У безкоштовної версії цього функціоналу немає [12].

Основні переваги програмного продукту LifeSum [12]:

- додавання рецептів та групування продуктів у блюда;
- зчитування штрих-кодів;
- облік антропометричних даних (платна версія).

Недоліки застосунку LifeSum [12]:

- часткова безкоштовність;
- відсутня синхронізація даних с сервером (збереження даних користувача на пристрої).

Таким чином, у наш час використовується багато програмних засобів для підтримки здорового харчування та розрахунку кількості калорій. Порівняльний аналіз програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій наведено у табл. 1.1.

З приведеної таблиці видно, що переважна більшість програмних засобів для підтримки здорового харчування та розрахунку кількості калорій не надають можливість їх використання безкоштовно, що для більшості користувачів є досить суттєвим недоліком. Також недоліком існуючого програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій є те, що воно не підтримують локалізацію деяких країн, зокрема і України. Отже доцільною є розробка програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій, що усуває зазначені недоліки.

Таблиця 1.1 – Програмне забезпечення для підтримки здорового харчування та розрахунку кількості калорій

Критерій, за яким порівнюють	My FitnessPal	Lose It!	FatSecret	Cronometer	Lifesum
Наявність сканеру їжі та штрихкодів	+	—	—	+	—
База даних продуктів	—	+	+	+	+
Безкоштовність	—	—	+	—	—
Зручність інтерфейсу	+	—	+	+	—
Наявність рекомендацій для тренувань	+	—	—	—	+
Можливість роботи з різними ОС	—	+	—	—	—

Таким чином, необхідно створити програмне забезпечення для підтримки здорового харчування та розрахунку кількості калорій. При розробці програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій необхідно забезпечити такі функціональні вимоги:

- реєстрація, авторизація та аутентифікація користувачів;
- CRUD (create, read, update, delete) операції з користувачами (права адміністратора);
- CRUD операції з їжею;
- фільтрація за датою та часом;

- побудова графіку розподілу калорій;
- розрахунок числа калорій, яке необхідно скоротити в раціоні за антропометричними даними.

1.3 Висновки за розділом 1

Проаналізовано про раціональне харчування та розрахунок кількості калорій. Визначено, що правильне або раціональне харчування – це таке харчування, яке забезпечує зміцнення та покращення здоров'я, фізичних і духовних сил людини, попередження та лікування захворювань. Проаналізовано принципи складання поживних раціонів з урахуванням кількості калорій. Визначено, що з метою підтримки здорового способу харчування у наш час використовується програмне забезпечення для розрахунку кількості калорій.

Виконано дослідження програмних засобів для розрахунку кількості калорій, зокрема проаналізовано таке програмне забезпечення: My FitnessPal, Lose It!, FatSecret, Cronometer, LifeSum.

Визначено, що переважна більшість програмних засобів для підтримки здорового харчування та розрахунку кількості калорій не надають можливість їх використання безкоштовно, що для більшості користувачів є досить суттєвим недоліком. Отже доцільною є розробка програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій, що усуває зазначені недоліки.

Визначено функціональні вимоги, яким повинно задовольняти програмне забезпечення для підтримки здорового харчування та розрахунку кількості калорій.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

Для створення програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій було обрано мову програмування Java [13]-[16].

Java є відносно молодого мовою програмування, яка широко використовується для розроблення програмного забезпечення різного призначення. На даний момент для використання Java є можливість вибирати між чотирма варіантами поставки [13]-[16]:

- Java Standard Edition – стандартний набір засобів розробки Java, яких достатньо при створенні більшості програмного забезпечення;
- Java Enterprise Edition – набір засобів розробки Java для створення більшості великих корпоративних програм;
- Java Micro Edition – набір засобів розробки Java для створення мобільних програмних застосунків;
- Java Card Edition – набір засобів розробки Java для створення більшості інтелектуальних карт (смарт-карт).

Поточна версія Java станом офіційно називається Java Standard Edition і складається з пакету розробки Java (JDK, Java Development Kit) і середовища виконання Java (JRE, Java Runtime Environment) [13]-[16].

Пакет Java Development Package (JDK) містить усі інструменти та інструменти програмування, необхідні для розробки програмного забезпечення Java [13]-[16].

Для запуску створених програм Java потрібне середовище виконання Java (JRE). Середовище виконання Java також називають віртуальною машиною [13]-[16].

Java – це спеціальна мова програмування, яка заснована на синтаксисі мов програмування C і C++. З самого початку розробки Java було приділено багато уваги тому те, щоб не виникали серйозні помилки програмування. У

Java немає покажчиків, тому програміст не може випадково отримати доступ до неправильної адреси пам'яті. Java високо цінується серед програмістів як легка в освоєнні, об'єктно-орієнтована, безпечна мова програмування, яка не залежить від архітектури комп'ютера. В останні роки розробникам Java вдалося зробити мову програмування все більш потужною [13]-[16].

Java — це мова програмування, яка з повною підставою позбавляє певних функцій C++, таких як покажчики, множинне успадкування та файли заголовків. Натомість багато розширених функцій, таких як обробка винятків, паралельність і графічний інтерфейс користувача, були вбудовані в мову програмування Java [13]-[16].

Java містить усі властивості сучасних об'єктно-орієнтованих мов програмування. Мова програмування Java особливо характеризується такими особливостями [13]-[16]:

- мова дуже схожа до мов C і C++, зокрема, оператори if, break і continue, цикли for, while і do;
- типи даних – більшість популярних типів даних вбудовані в Java. Логічний тип даних уже існує з логічним значенням;
- масиви у Java інтерпретуються як об'єкти, а доступ до масивів перевіряється на правильність під час виконання. Багатовимірні структури можна моделювати шляхом вкладення простих масивів;
- немає оператора goto: це усуває поширене джерело помилок;
- визначення класів: в Java об'єкти створюються з класів. Спеціальні конструктори ініціалізують об'єкти, а структура відповідає за знищення об'єктів;
- видимість: ключові слова private, protected та відкриті обмежують видимість елементів;
- управління пам'яттю: Java має автоматичне керування пам'яттю, так званий збирач сміття. Невикористана пам'ять знову звільняється під час виконання;
- спадкування: у Java можна успадковувати властивості від класів;

– обробка винятків: для того, щоб мати можливість обробляти помилки під час виконання контрольованим способом, об'єкти винятків були введені в Java. Вони дозволяють перехоплювати й обробляти помилки під час виконання [13]-[16].

Програми Java можна запускати в браузері: за допомогою аплетів можна запускати Java-додатки в браузері (Firefox, Internet Explorer, Safari або Opera). Це стало можливим завдяки тегу аплету мови Інтернету HTML [13]-[16]. Програми Java можуть виконуватися лише браузерами з підтримкою Java. Для цього в браузер повинні бути інтегровані інтерпретатор Java (віртуальна машина) та бібліотека часу виконання. Аплети полегшують розповсюдження програм Java в Інтернеті, незалежно від обладнання та операційної системи [13]-[16].

Java-аплети мають деякі обмеження безпеки, щоб запобігти пошкодженню системи під час роботи аплету. Забороняється виконувати файлові операції на локальному комп'ютері. Також неможливо запустити зовнішні програми [13]-[16].

Аплет завантажується через завантажувач класів браузера, а потім виконується безпосередньо. За допомогою аплету можна виводити текст, графіку, аудіо, відео та діалоги [13]-[16].

За допомогою мови програмування Java можна дуже легко і зручно створювати графічні інтерфейси користувача. Для цього бібліотека часу виконання Java містить три компоненти [13]-[16]:

- Abstract Windowing Toolkit (AWT): для відображення елементарної графіки та віконних функцій;
- Swing Toolset: для відображення складних графічних інтерфейсів користувача;
- Java 2D API: для складних завдань обробки зображень і графічних маніпуляцій.

За допомогою набору інструментів абстрактних вікон текст, лінії, кола, прямокутники та багатокутники можна малювати в будь-якому місці вікна

[13]-[16]. Звичайно, вихідні графічні елементи можна відобразити будь-яким кольором. Для визначення потрібного кольору використовується колірна модель RGB (червоний, зелений, синій). У Java також можна використовувати колірну модель HSB (Hue, Saturation, Brightness) [13]-[16].

Для розробників аудіододатків Java пропонує спеціальний звуковий API, за допомогою якого можна відтворювати аудіофайли та керувати ними. Дані зображення можна зручно редагувати та відображати за допомогою пакета зображень [13]-[16].

З графічними інтерфейсами користувача, оскільки користувач взаємодіє з програмним забезпеченням, інструкції користувача повинні бути зафіксовані. Java вирішує це за допомогою подій, які генеруються при виконанні користувачем певних дій і передаються програмі Java [13]-[16].

Програма оцінює вхідні події, а потім виконує відповідні функції. Подіями можуть бути записані такі події [13]-[16]:

- клацання миші;
- натискання клавіш;
- вікно та діалогове введення.

Події можна надсилати на будь-який об'єкт. Однак умова завжди полягає в тому, що інтерфейс одержувача повідомлення реалізований в об'єкті [13]-[16]. Swing Toolset є першим вибором для програмістів із вимогливими графічними інтерфейсами користувача. Він має багато графічних елементів, які використовуються в сучасних інтерфейсах користувача [13]-[16].

Ще однією великою перевагою Swing Toolset є висока переносимість на різні платформи [13]-[16].

Звичайно, програми, які керуються за допомогою командного рядка, також можна запрограмувати на Java. У цьому випадку Java забезпечує текстовий вихід із звичайним набором функцій C/C++ [13]-[16]. Одним з найважливіших класів є клас String. Він містить усі методи, необхідні для оцінки та обробки рядків символів. Крім класу String, в бібліотеці класів Java вже реалізовано багато інших важливих класів [13]-[16].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для підрахунку калорій наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для підрахунку калорій

Критерій порівняння мов програмування	Мова програмування		
	Java	C++	PHP
Зручність роботи з масивами даних	+	+	+—
Засоби для роботи з базами даних	+	+—	+
Засоби паралельного програмування	+	+—	—
Зручність для створення ПЗ для підрахунку калорій	+	+—	+—
Зручність керування пам'яттю	+	+	—

Отже, для реалізації програмного забезпечення для підрахунку калорій обрано мову програмування Java.

2.2 Вибір середовища розробки для створення програмного забезпечення для підрахунку калорій

Оскільки в якості мови програмування було обрано Java, то для створення програмного забезпечення для підрахунку калорій доцільно обрати середовище розробки IntelliJ IDEA, яке досить ефективно підтримує розробку програм на мові Java.

IntelliJ IDEA — це швидкий і надійний інструмент для веб-розробки Java. Це середовище розробки Java розроблено JetBrains. Він постійно оновлюється та доповнюється відповідно до потреб ринку [13]-[17].

IntelliJ IDEA поставляється в двох версіях. Безкоштовна версія призначена для студентів, викладачів і розробників плагінів. Платні – для організацій, що розробляють комерційні додатки [13]-[17].

Для створення програмного забезпечення для підрахунку калорій доцільно використовувати версію Ultimate. Він інтегрує такі фреймворки, як Spring MVC (контролер перегляду моделі), J2EE, React, Angular JS (javascript), CMS (системи керування вмістом), такі як Drupal, WordPress, Laravel [13]-[17].

Платна версія має додаткову підтримку контролю версій, наприклад Perforce, ClearCase і Visual SourceSafe. Він має багато варіантів для налаштування стратегії розгортання за допомогою Tomcat, TomEE, GlassFish, JBoss, WebLogic, WebSphere, Geronimo, Jetty і Virgo. На відміну від Eclipse і NetBeans, він підтримує розумний рефакторинг і розуміє контекст [15]-[17].

Ця версія підтримує NPM, WebPack, Gulp і Grunt. Він включає в себе такі інструменти, як діаграми UML (Уніфікована мова моделювання) або матриця залежностей [15]-[17].

Таким чином, IntelliJ IDEA — це інтелектуальна, контекстно-залежна IDE для роботи з Java та іншими мовами JVM, такими як Kotlin, Scala та Groovy, у всіх видах програм. Крім того, IntelliJ IDEA Ultimate може допомогти розробляти повноцінні веб-додатки завдяки своїм потужним інтегрованим інструментам, підтримці JavaScript і пов'язаних технологій, а також розширеній підтримці популярних фреймворків, таких як Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus, Helidon. Крім того, можна розширити IntelliJ IDEA за допомогою безкоштовних плагінів, розроблених JetBrains, що дозволить вам працювати з іншими мовами програмування, включаючи Go, Python, SQL, Ruby і PHP [17].

Обґрунтування вибору середовища розробки для підрахунку калорій наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для підрахунку калорій

Критерій порівняння середовищ розробки	Середовище розробки		
	IntelliJ IDEA	Eclipse	MS Visual Studio
Зручність для створення програмного забезпечення для підрахунку калорій	+	+—	+—
Розуміння контексту(autocomplete)	+	—	+—
Інтелектуальний рефакторинг	+	—	+—
Кросплатформність	+	+—	+
Розробка GUI	+	+—	+

Таким чином, для створення програмного забезпечення для підрахунку калорій обрано середовище розробки IntelliJ IDEA.

2.3 Висновки за розділом 2

Для створення програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій було обрано мову програмування Java, яка широко використовується для розроблення програмного забезпечення різного призначення та є легкою в освоєнні, об'єктно-орієнтованою, безпечною мовою програмування, яка не залежить від архітектури комп'ютера.

Оскільки в якості мови програмування було обрано Java, то для створення програмного забезпечення для підрахунку калорій доцільно обрати середовище розробки IntelliJ IDEA, яке досить ефективно підтримує розробку програм на мові Java.

3 ОПИС ПРОГРАМИ

3.1 Схема взаємодії основних компонентів програмного забезпечення для підрахунку калорій

У розробленому програмному забезпеченні для підрахунку калорій основними структурними компонентами є:

- серверний компонент, що реалізує основні функції програмного забезпечення для підрахунку калорій;
- модуль вебінтерфейсу, що забезпечує взаємодію користувача з розробленою програмою;
- база даних про вміст калорій у продуктах споживання.

Користувач програмного забезпечення для підрахунку калорій використовує вебінтерфейс для взаємодії з програмою. Модуль вебінтерфейсу, використовуючи програмні засоби, розташовані на сервері, передає та отримує інформацію про вміст калорій у наборі їжі, що вживається користувачем. При цьому основна інформація про вміст калорій у конкретній їжі знаходиться у базі даних.

Схему взаємодії основних модулів програмного забезпечення для підрахунку калорій зображено на рис. 3.1.

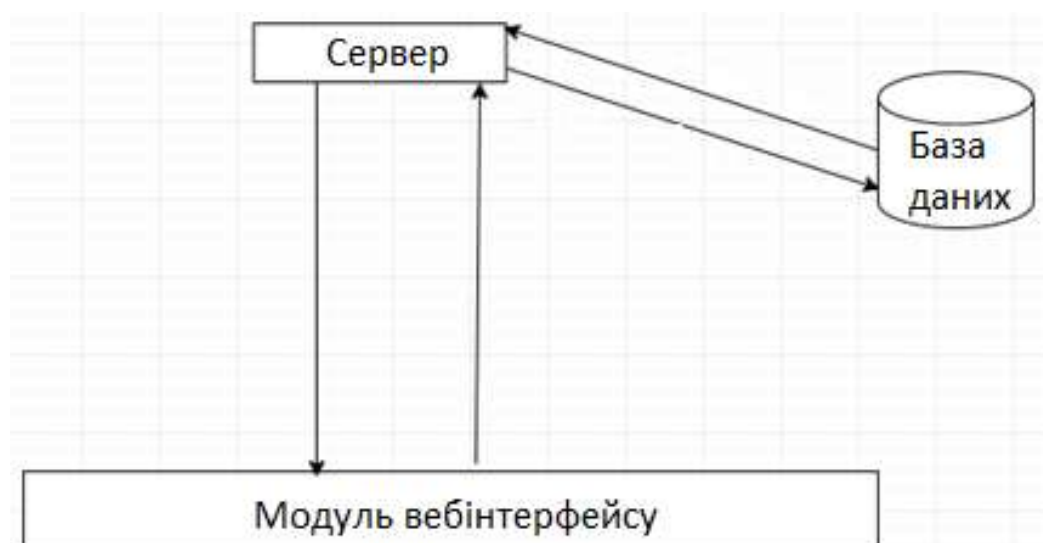


Рисунок 3.1 – Схема взаємодії основних модулів програмного забезпечення для підрахунку калорій

Відповідно до наведеної схеми на стороні клієнта розташовано компонент системного інтерфейсу (вебінтерфейсу), який надається кінцевому користувачу. Цей рівень не має прямого підключення до бази даних (у зв'язку з вимогами безпеки та масштабованості), не завантажений основною бізнес-логікою (вимоги до масштабованості) і не підтримує стан програми (вимоги надійності). На цьому рівні здійснюється лише найпростіша бізнес-логіка: перевіряється інтерфейс авторизації, дійсність значень і їх відповідність формату даних.

На програмному рівні на сервері зосереджено більшість бізнес-логіки програмного забезпечення для підрахунку калорій. Цей рівень координує роботу програми для підрахунку калорій, обробляє команди, приймає логічні рішення та обчислення. Він обробляє дані між двома іншими рівнями програмного забезпечення.

За межами серверного рівня залишаються лише логічні елементи з бази даних про вміст калорій у їжі.

Рівень даних забезпечує зберігання даних у базі даних, що реалізована за допомогою СКБД PostgreSQL, підключення до цього компонента здійснюється тільки на рівні сервера програмного застосунку.

3.2 Опис основних розроблених компонентів та модулів

У програмному забезпеченні для підрахунку калорій реалізовано різні компоненти та модулі. Основні з них описано нижче.

Компонент Model у програмному забезпеченні для підрахунку калорій є шаром даних, який містить первинні доменні об'єкти і сутності. Компонент Model у програмному забезпеченні для підрахунку калорій характеризується знаннями про себе саму і не знає про те, які контролери і уявлення даних використовуються у програмі для підрахунку калорій. Об'єкти, що реалізують один з шаблонів проектування (Transfer Objects) у програмному забезпеченні

для підрахунку калорій використовуються для передачі даних між підсистемами програмного застосування.

В обов'язки модулю вебінтерфейсу програмного забезпечення для підрахунку калорій входить відображення даних, цей модуль має код, який реалізує деяку бізнес-логіку програмного забезпечення для підрахунку калорій, зокрема він містить пакети, які містять контролери, реалізовані за допомогою паттерна «Єдина точка входу» (Front Controller [10]-[12]).

Патерн Front Controller у програмному забезпеченні для підрахунку калорій об'єднує обробку запитів, пропускаючи запити через обробник. Цей об'єкт містить загальну логіку поведінки програмного забезпечення для підрахунку калорій, яка може бути змінена в реальному часі за допомогою декораторів. Після обробки запиту контролер програмного забезпечення для підрахунку калорій звертається до конкретного об'єкта для відпрацювання конкретного поведінки.

Вебінтерфейс програмного забезпечення для підрахунку калорій містить пакети meal (їжа) та user (користувач), які містять Rest та Ajax контролери, за допомогою яких налаштована маршрутизація таблиць з користувачами та їжею, а також зв'язок з FrontController.

Пакет json у програмному забезпеченні для підрахунку калорій є допоміжним та допомагає пересилати дані до серверу та назад. Конвертуються дані у програмному забезпеченні для підрахунку калорій за допомогою реалізації gson converter від google у пакеті converter.

3.3 Функціонування програмного забезпечення для підрахунку калорій

Схема функціонування розробленої програми для підрахунку калорій зображено на рис. 3.2.

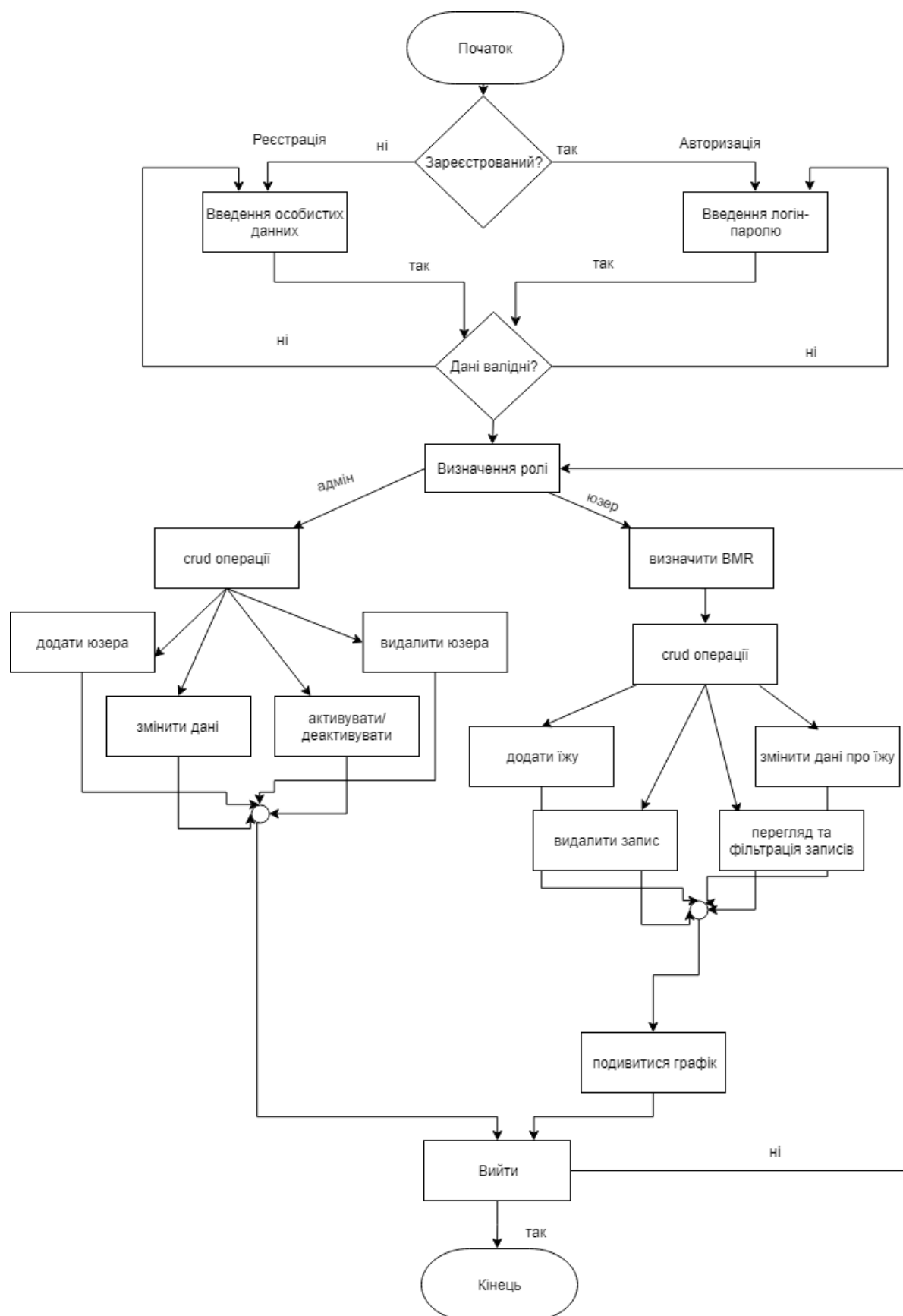


Рисунок 3.2 – Схема функціонування розробленої програми для підрахунку калорій

Як видно зі схеми основними функціями, що реалізовано у програмному забезпеченні для підрахунку калорій, є:

- авторизація користувача програми підрахунку калорій;
- реєстрація користувача програми підрахунку калорій;
- валідація користувача програми підрахунку калорій;
- аутентифікація користувача програми підрахунку калорій;
- маніпулювання списками (користувачі/їжа) – CRUD (create, read, update, delete) операції;
- визначення базового метаболічного рівня (BMR) як кількості калорій, що було отримано організмом у стані спокою за добу;
- перегляд статистики стосовно споживання їжі.

На початку роботи з програмою для підрахунку калорій користувач має на вибір один з двох режимів – реєстрація та авторизація користувача. Якщо користувач програмного забезпечення для підрахунку калорій вже зареєстрований, то він не зможе створити запис з тими ж даними.

Для авторизації у програмному забезпеченні для підрахунку калорій потрібні логін та пароль вже зареєстрованого користувача, а також необхідно, щоб обліковий запис не було деактивовано адміністратором.

Для реалізації у програмному забезпеченні для підрахунку калорій використовуються Spring MVC та Spring Security [10]-[12]. Модуль Spring Security підключається до нашого додатку на базі Spring MVC. Користувач програмного забезпечення для підрахунку калорій потрапляє на початкову сторінку login. За прийом даних та авторизацію користувача за допомогою інформації з бази даних відповідає сконфігурований Spring Security. Заздалегідь також налаштовані ролі користувачів (ADMIN та USER) у СКБД PostgreSQL та HSQLDB. За допомогою авторизації встановлюються і реалізуються права доступу до ресурсів.

Перевірка даних при роботі з програмою для підрахунку калорій відбувається за допомогою Hibernate Validator. Перевірка даних полягає у

валідації об'єктів і обробці повідомлень про порушення оголошених рамок (constraintViolations).

Про можливі помилки та повідомлення при роботі з програмою для підрахунку калорій детальніше розписано у наступному розділі.

Для адміністраторів програмного забезпечення для підрахунку калорій доступні CRUD функції таблиці користувачів.

Адміністратори програмного забезпечення для підрахунку калорій можуть переглядати користувачів, додавати та видаляти їх, змінювати дані, керувати правами доступу (активний або неактивний користувач).

Для користувачів програмного забезпечення для підрахунку калорій доступні CRUD операції над таблицею з інформацією про вміст калорій у спожитій їжі. Кожен адміністратор програмного забезпечення для підрахунку калорій також є користувачем.

Для користувачів програмного забезпечення для підрахунку калорій можлива фільтрація даних по даті і часу, при цьому колір записів у таблиці їжі залежить від того, чи перевищує сума калорій за день норму. На основі таблиці будується графік розподілу калорій за часом.

Визначення базового метаболічного рівня (BMR) відбувається за антропометричними даними (зріст, маса тіла та іншого).

3.4 Висновки за розділом 3

Розроблено програмне забезпечення для підрахунку калорій, у якому основними структурними компонентами є серверний компонент, що реалізує основні функції програмного забезпечення для підрахунку калорій, модуль вебінтерфейсу, що забезпечує взаємодію користувача з розробленою програмою, а також база даних про вміст калорій у продуктах споживання.

Запропоновано схему взаємодії основних модулів програмного забезпечення для підрахунку калорій. Користувач програмного забезпечення для підрахунку калорій використовує вебінтерфейс для взаємодії з програмою.

Модуль вебінтерфейсу, використовуючи програмні засоби, розташовані на сервері, передає та отримує інформацію про вміст калорій у наборі їжі, що вживається користувачем. При цьому основна інформація про вміст калорій у конкретній їжі знаходиться у базі даних.

Розроблено схему функціонування розробленої програми для підрахунку калорій.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для розрахунку кількості спожитих калорій, зберігання та аналізу інформації про споживання, зіставлення калорійності з'їденого з BMR. Програма написана на мові Java.

Програмне забезпечення для підрахунку калорій забезпечує виконання таких функцій:

- реєстрація, авторизація та аутентифікація користувачів;
- CRUD операції з користувачами (права адміністратора);
- CRUD операції з їжею;
- фільтрація за датою та часом;
- побудова графіку розподілу калорій;
- розрахунок числа калорій, яке необхідно скоротити в раціоні за антропометричними даними.

Для експлуатації програмного забезпечення для підрахунку калорій необхідно таке апаратне та програмне забезпечення:

- підтримка сервером PostgreSQL версії не нижче 9.0;
- підтримка сервером HSQLDB версії не нижче 2.4.0;
- процесор із тактовою частотою 1,5 GHz та більше;
- оперативна пам'ять 4 ГБ та більше;
- HDD (hard disk drive), 80 Гігабайт, не менше.

Крім того, на стороні клієнта необхідна наявність на пристрої користувача доступу до Інтернету та веб-браузера.

Функціональні характеристики розробленого програмного забезпечення для підрахунку калорій наведено у технічному завданні (додаток А). Опис програмного забезпечення для підрахунку калорій та текст програмного забезпечення для підрахунку калорій наведені у додатках Б та В.

4.2 Характеристики програми

Режим роботи програмного забезпечення для підрахунку калорій є постійним, цілодобовим. Програма для підрахунку калорій виконана за допомогою мови програмування Java та середовища програмування IntelliJ IDEA.

Час відновлення програмного забезпечення для підрахунку калорій після відмови, викликаної збоєм операційної системи, не перевищує прийняттого часу (декілька хвилин), що потрібні для «підняття» серверу та завантаження ресурсів.

Відмови програмного забезпечення для підрахунку калорій внаслідок некоректних дій користувача при взаємодії з програмою через вебінтерфейс неприпустимі.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Є два варіанти запуснути програмне забезпечення для підрахунку калорій: локальний запуск або запуск на PaaS-платформі. Адміністратор програмного забезпечення для підрахунку калорій повинен бути впевнений, що застосунок запущено або за допомогою Tomcat, або на Heroku (рис. 4.1).

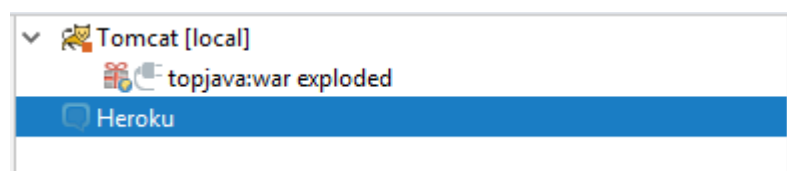


Рисунок 4.1 – Сервери застосунку

Деплой на Heroku відбувається з гілки kurs github'у (рис. 4.2).

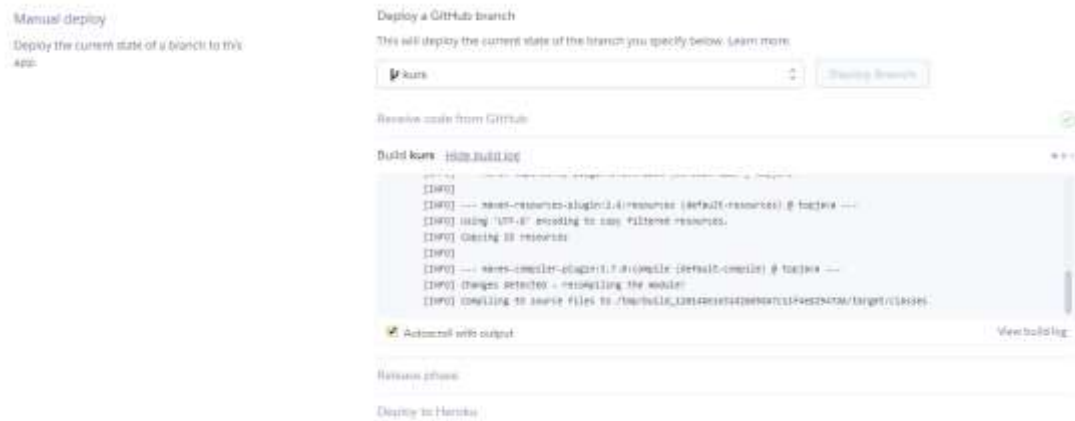


Рисунок 4.2 – Деплой додатку на Heroku

Після цього в адмін-панелі (рис. 4.3) з'явиться повідомлення про успішне розвертання додатку, та користувачу воно стане доступне за посиланням (рис. 4.4).

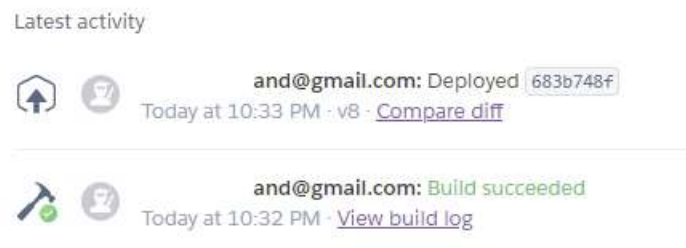


Рисунок 4.3 – Повідомлення в адмін-панелі Heroku

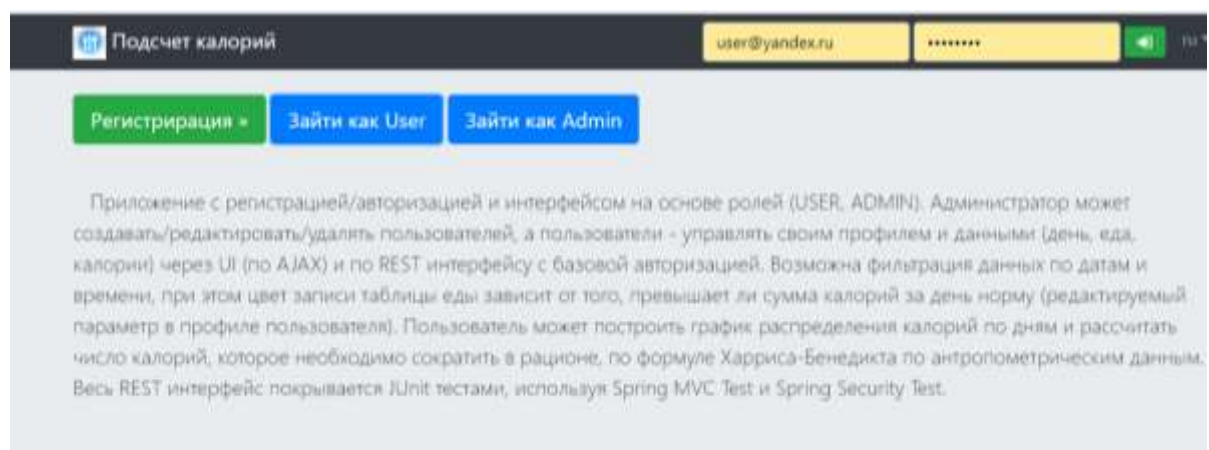


Рисунок 4.4 – Доступ до застосунку після розвертання за допомогою PaaS-платформи Heroku

Локальний запуск програмного забезпечення для підрахунку калорій відбувається за допомогою Tomcat або TomEE з середовища розробки (рис. 4.5).

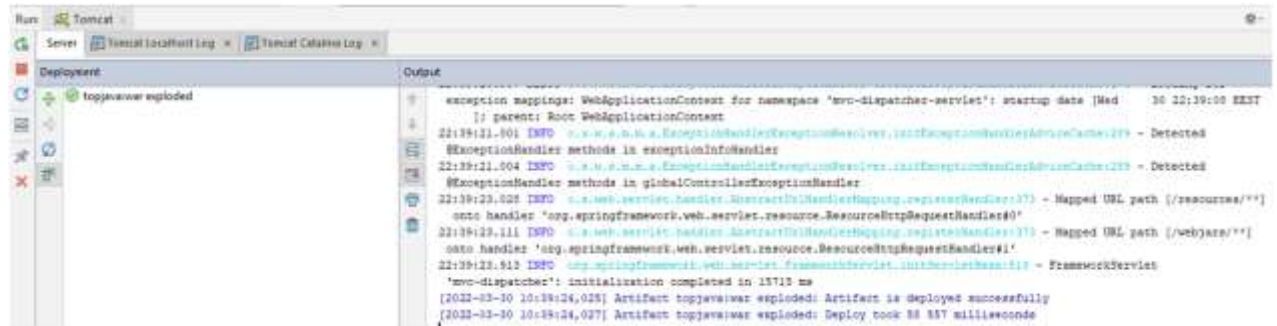


Рисунок 4.5 – Запуск застосунку локально з середовища розробки

Після успішного розвертання у браузері відкриється програмне забезпечення для підрахунку калорій за посиланням: localhost (рис. 4.6).

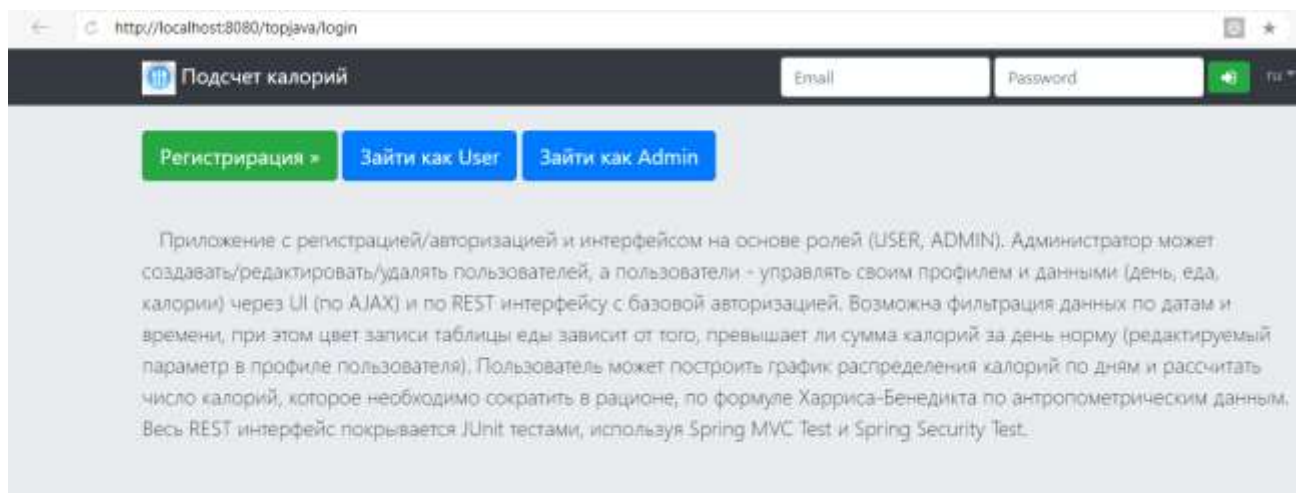


Рисунок 4.6 – Доступ до застосунку після локального розвертання

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для підрахунку калорій є дані, що користувач/адміністратор вводить у спеціалізовані форми на сайті. Інформація з них потрапляє у СКБД PostgreSQL у разі якщо вона валідна.

Вихідними даними програмного забезпечення для підрахунку калорій є профільтровані з СКБД дані, що можуть бути подані у вигляді таблиць або графічному вигляді за бажанням користувача.

4.3.3 Повідомлення

Під час роботи із програмою для підрахунку калорій користувач може отримати різні повідомлення, серед них можуть бути такі: Bad credentials (користувач ввів невірні дані під час реєстрації), Неверный формат поля (користувач ввів некоректні дані у поле, тобто дані не того типу, що очікується) та інші.

4.4 Тестування програмного забезпечення для підрахунку калорій

Тестування програмного забезпечення для підрахунку калорій проводитиметься за таким сценарієм:

- користувач програмного забезпечення для підрахунку калорій пробує увійти до програми із невалідними даними;
- користувач програмного забезпечення для підрахунку калорій отримає повідомлення про введення невірних даних під час реєстрації та буде змушений перейти до процесу повторної реєстрації;
- в ході реєстрації, користувач програмного забезпечення для підрахунку калорій спробує ввести некоректну інформацію;
- після успішної реєстрації користувач програмного забезпечення для підрахунку калорій змінює/видаляє/редагує записи у таблиці бази даних, що містить їжу;
- після цього користувач програмного забезпечення для підрахунку калорій виконує ті ж дії, проте з некоректними даними;
- якщо кількість записів бази даних є ненульовою (непустою), то можна перейти до відповідної сторінки та подивитися на статистику;

- коли користувач програмного забезпечення для підрахунку калорій є адміністратором (має відповідні права) йому доступні додаткові операції з таблицею, що містить інформацію про користувачів;
- сторінка входу до програмного забезпечення для підрахунку калорій не доступна коли користувач є авторизованим;
- користувач програмного забезпечення для підрахунку калорій попадає на сторінку входу після натискання на кнопку вихід.

Тестування програмного забезпечення для підрахунку калорій буде проводитися за допомогою апаратних засобів, описаних у підрозділі 4.1.

На початку тестування програмного забезпечення для підрахунку калорій користувач пробує увійти у систему із некоректними даними. Виникає помилка, яка зображена на рис. 4.7.

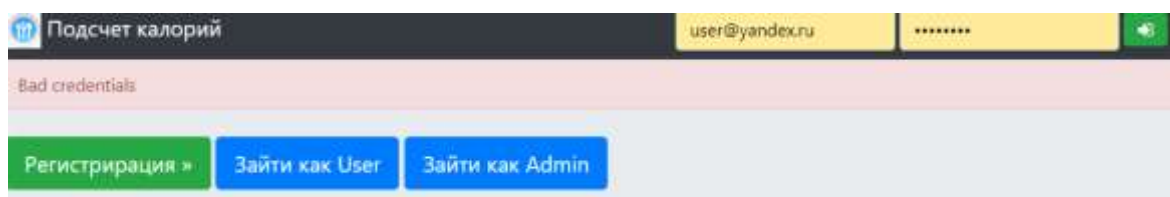


Рисунок 4.7 – Помилка при авторизації

Після цього відбувається реєстрація користувача, та тестується валідність полів вводу (рис. 4.8 – 4.9).

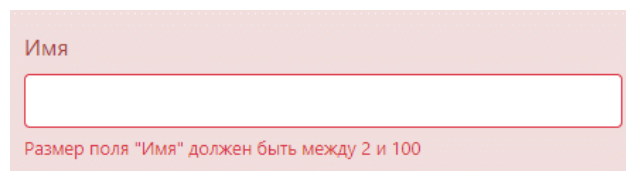


Рисунок 4.8 – Помилка при вводі імені

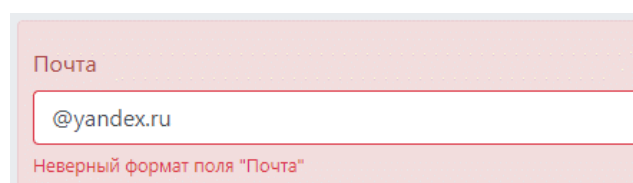


Рисунок 4.9 – Помилка при вводі email

Після цього зареєструємося з валідними даними (рис. 4.10).

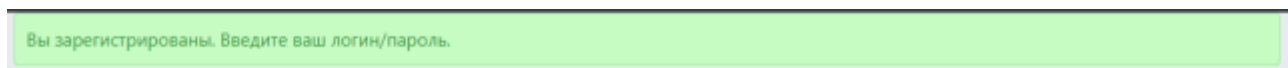


Рисунок 4.10 – Успішна реєстрація

Після реєстрації входимо в обліковий записи (рис. 4.11).

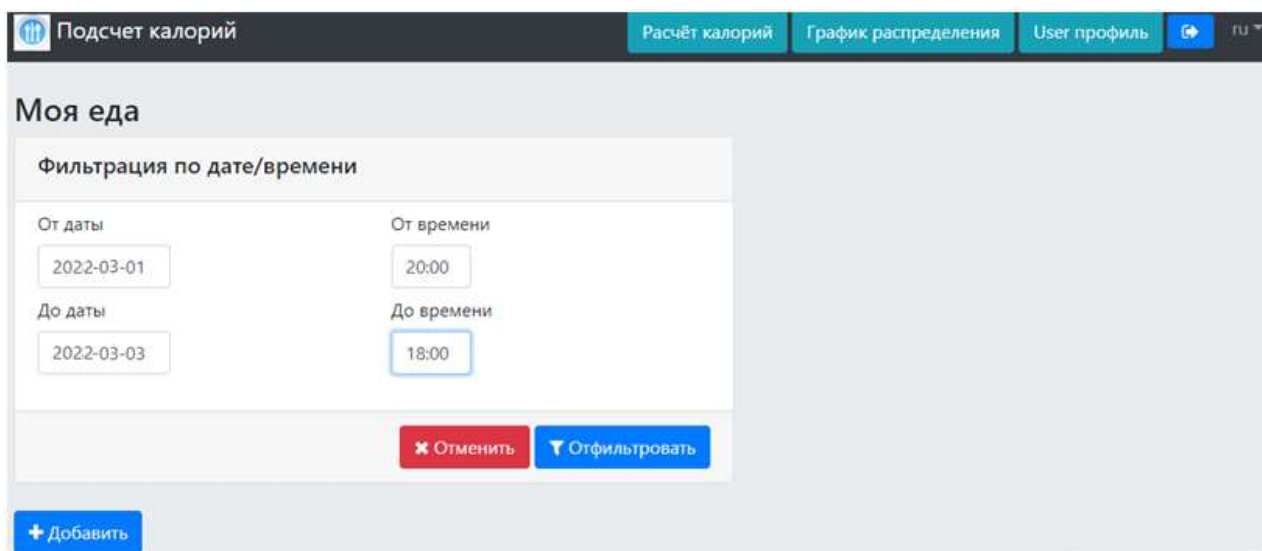


Рисунок 4.11 – Головна сторінка після реєстрації

Користувач може виконувати CRUD операції з їжею. Для кожної операції присутня фільтрація (рис. 4.12).

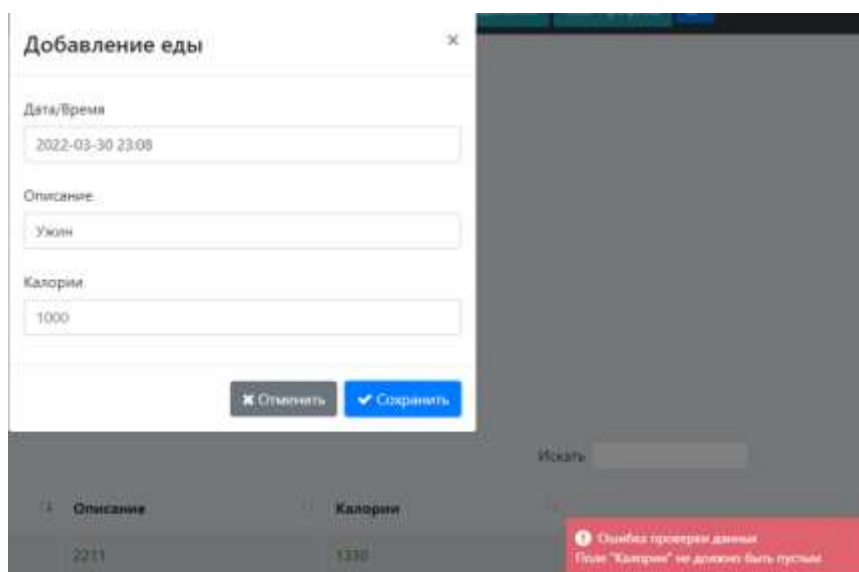


Рисунок 4.12 – Спроба додати їжу

Після цього користувач може продивитися статистику розподілу калорій за днями (рис. 4.13).

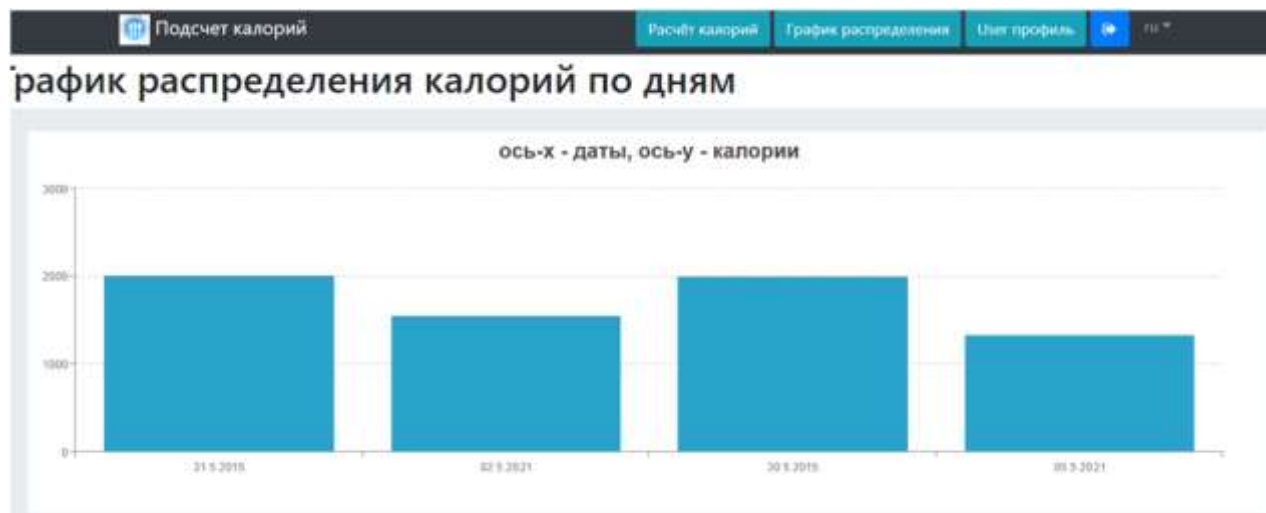


Рисунок 4.13 – Перегляд статистики

Якщо у користувача змінилися антропометричні дані, то він може заново розрахувати ООР для себе (рис. 4.14).

Расчет по формуле ХАРРИСА-БЕНЕДИКТА

Определение количества калорий, которое необходимо сократить в рационе, чтобы похудеть на определенное количество килограммов – это больше искусство, чем наука. Но формула Харриса-Бенедикта поможет вам им овладеть!

Мужчина Женцина

Вес:

Поле "Вес" не должно быть пустым

Рост:

Поле "Рост" не должно быть пустым

Возраст:

Поле "Возраст" не должно быть пустым

Рисунок 4.14 – Розрахунок BMR

Адміністратор також може керувати записами користувачів (рис. 4.15).



Рисунок 4.15 – Користувачі в панелі адміністратора

Результати тестування програмного забезпечення для підрахунку калорій показали, що воно відповідає технічному завданню та може використовуватися за призначенням.

4.5 Висновки за розділом 4

Описано програмне забезпечення для підрахунку калорій. Виконано тестування розробленого програмного забезпечення для підрахунку калорій. Результати тестування програмного забезпечення для підрахунку калорій підтвердили його працездатність.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для підтримки здорового харчування.

Проаналізовано поняття про раціональне харчування та розрахунок кількості калорій. Визначено, що правильне або раціональне харчування – це таке харчування, яке забезпечує зміцнення та покращення здоров'я, фізичних і духовних сил людини, попередження та лікування захворювань. Проаналізовано принципи складання поживних раціонів з урахуванням кількості калорій. Визначено, що з метою підтримки здорового способу харчування у наш час використовується програмне забезпечення для розрахунку кількості калорій.

Виконано дослідження програмних засобів для розрахунку кількості калорій, зокрема проаналізовано таке програмне забезпечення: My FitnessPal, Lose It!, FatSecret, Cronometer, LifeSum.

Визначено, що переважна більшість програмних засобів для підтримки здорового харчування та розрахунку кількості калорій не надають можливість їх використання безкоштовно, що для більшості користувачів є досить суттєвим недоліком. Отже доцільною є розробка програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій, що усуває зазначені недоліки.

Визначено функціональні вимоги, яким повинно задовольняти програмне забезпечення для підтримки здорового харчування та розрахунку кількості калорій.

Для створення програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій було обрано мову програмування Java, яка широко використовується для розроблення програмного забезпечення різного призначення та є легкою в освоєнні, об'єктно-

орієнтованою, безпечною мовою програмування, яка не залежить від архітектури комп'ютера.

Оскільки в якості мови програмування було обрано Java, то для створення програмного забезпечення для підрахунку калорій доцільно обрати середовище розробки IntelliJ IDEA, яке досить ефективно підтримує розробку програм на мові Java.

Розроблено програмне забезпечення для підрахунку калорій, у якому основними структурними компонентами є серверний компонент, що реалізує основні функції програмного забезпечення для підрахунку калорій, модуль вебінтерфейсу, що забезпечує взаємодію користувача з розробленою програмою, а також база даних про вміст калорій у продуктах споживання.

Запропоновано схему взаємодії основних модулів програмного забезпечення для підрахунку калорій. Користувач програмного забезпечення для підрахунку калорій використовує вебінтерфейс для взаємодії з програмою. Модуль вебінтерфейсу, використовуючи програмні засоби, розташовані на сервері, передає та отримує інформацію про вміст калорій у наборі їжі, що вживається користувачем. При цьому основна інформація про вміст калорій у конкретній їжі знаходиться у базі даних.

Розроблено схему функціонування розробленої програми для підрахунку калорій.

Описано програмне забезпечення для підрахунку калорій. Виконано тестування розробленого програмного забезпечення для підрахунку калорій. Результати тестування програмного забезпечення для підрахунку калорій підтвердили його працездатність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Physiological basis of rational nutrition [Electronic resource]. – Access mode : [https://tdmuv.com/kafedra/internal/normal_phiz/classes_stud/en/stomat/2course/5 cycle physiology of digestion system and kidneys/02 physiological basis of rational nutrition.htm](https://tdmuv.com/kafedra/internal/normal_phiz/classes_stud/en/stomat/2course/5cyclephysiologyofdigestionsystemandkidneys/02physiologicalbasisofrationalnutrition.htm).
2. Dietology the theory of rational nutrition [Electronic resource]. – Access mode : <https://zmgtennis.com/dietology-the-theory-of-rational-nutrition.html>.
3. Four Principles of Rational Nutrition [Electronic resource]. – Access mode : <https://www.thefreelibrary.com/Four+Principles+of+Rational+Nutrition.-a066454390>.
4. Рациональное питание – основа долголетия [Электронный ресурс]. – Режим доступа : <http://www.16gp.by/pravilnoe-pitanie-zalog-zdorovya-i-dolgoletiya/514-khochesh-kak-mozhno-dolshe-sokhranit-zdorove-strojnoe-i-krasivoe-telo-pitaj-sya-pravilno/>.
5. What is the organization of rational nutrition [Electronic resource]. – Access mode : <https://jdmsale.ru/en/pavement/v-chem-zaklyuchaetsya-organizaciya-racionalnogo-pitaniya-rebenka-racionalnoe-pitanie-detei-doshkolnogo.html>.
6. The importance of milk and dairy products consumption as a part of rational nutrition [Electronic resource]. – Access mode : <https://potravinarstvo.com/journal1/index.php/potravinarstvo/article/view/1050>.
7. Rational Nutrition Management and Balanced Diet of Beneficiaries [Electronic resource]. – Access mode : <https://rdehub.uniwa.gr/en/ffs/rational-nutrition-management-and-balanced-diet-of-beneficiaries/>.
8. My FitnessPal [Electronic resource]. – Access mode : <https://www.myfitnesspal.com/ru>.
9. Lose It! [Electronic resource]. – Access mode : <https://www.loseit.com/?correlationId=97b992a3-290f-405f-bacb-bd6c0e7e2441>
10. FatSecret [Electronic resource]. – Access mode : <https://www.fatsecret.com/?correlationId=504e2b80-ef60-4474-8b8d-e6f30abba94e>.

11. Cronometer [Electronic resource]. – Access mode :
<https://cronometer.com>.
12. Lifesum [Electronic resource]. – Access mode : <https://lifesum.com>.
13. Was ist Java [Electronic resource]. – Access mode :
<https://www.programmierenlernenhq.de/was-ist-java-einfuehrung-in-die-programmiersprache-java/>.
14. Java (Programmiersprache) [Electronic resource]. – Access mode :
<https://www.it-treff.de/it-lexikon/java-programmiersprache>.
15. Java Programmieren lernen [Electronic resource]. – Access mode :
<https://lerneprogrammieren.com/java-programmieren-lernen/>.
16. Was ist Java-Programmierung [Electronic resource]. – Access mode :
<https://www.dev-insider.de/was-ist-java-programmierung-a-569186/>.
17. What is IntelliJ IDEA [Electronic resource]. – Access mode :
<https://www.jetbrains.com/idea/features/>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення для підрахунку калорій може використовуватися для підтримки процесу здорового харчування та здорового образу життя.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для підрахунку калорій», затверджене наказом Національного університету «Запорізька політехніка» № 102 від 27 квітня 2022 р.

A.2 Призначення розробки

Програмний продукт для підрахунку калорій призначений для надання інформації про кількість спожитих калорій при споживанні їжі.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підрахунку калорій повинно забезпечити виконання таких функцій:

- реєстрація, авторизація та аутентифікація користувачів;
- CRUD операції з користувачами (права адміністратора);
- CRUD операції з їжею;
- фільтрація за датою та часом;
- побудова графіку розподілу калорій;
- розрахунок числа калорій, яке необхідно скоротити в раціоні за антропометричними даними.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програми для підрахунку калорій повинен бути зручним для використання на практиці. Взаємодія користувача з програмою для підрахунку калорій можлива за допомогою форм користувача.

A.3.3 Вимоги до надійності

Програмне забезпечення для підрахунку калорій повинно забезпечити можливість оброблення критичних ситуацій.

A.3.4 Умови експлуатації та до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для підрахунку калорій необхідно таке апаратне та програмне забезпечення:

- підтримка сервером PostgreSQL версії не нижче 9.0;
- підтримка сервером HSQLDB версії не нижче 2.4.0;
- процесор із тактовою частотою 1,5 GHz та більше;
- оперативна пам'ять 4 ГБ та більше;
- HDD (hard disk drive), 80 Гігабайт, не менше.

Крім того, на стороні клієнта необхідна наявність на пристрої користувача доступу до Інтернету та веб-браузера.

A.3.5 Вимоги до маркування і пакування

Програма для підрахунку калорій може бути записана на будь-якому носії інформації. На пакуванні повинна бути назва програми – «Програмне забезпечення для підрахунку калорій».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для підрахунку калорій є дані, що користувач/адміністратор вводить у спеціалізовані форми на сайті. Інформація з них потрапляє у СКБД PostgreSQL у разі якщо вона валідна.

Вихідними даними програмного забезпечення для підрахунку калорій є профільтровані з СКБД дані, що можуть бути подані у вигляді таблиць або графічному вигляді за бажанням користувача.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Програма призначена для розрахунку кількості спожитих калорій, зберігання та аналізу інформації про споживання, зіставлення калорійності з'їденого з BMR. Програма написана на мові Java.

Б.2 Функціональне призначення

Програмне забезпечення для підрахунку калорій забезпечує виконання таких функцій:

- реєстрація, авторизація та аутентифікація користувачів;
- CRUD операції з користувачами (права адміністратора);
- CRUD операції з їжею;
- фільтрація за датою та часом;
- побудова графіку розподілу калорій;
- розрахунок числа калорій, яке необхідно скоротити в раціоні за антропометричними даними.

Б.3 Опис логічної структури

У розробленому програмному забезпеченні для підрахунку калорій основними структурними компонентами є:

- серверний компонент, що реалізує основні функції програмного забезпечення для підрахунку калорій;
- модуль вебінтерфейсу, що забезпечує взаємодію користувача з розробленою програмою;
- база даних про вміст калорій у продуктах споживання.

Користувач програмного забезпечення для підрахунку калорій використовує вебінтерфейс для взаємодії з програмою. Модуль вебінтерфейсу, використовуючи програмні засоби, розташовані на сервері, передає та отримує

інформацію про вміст калорій у наборі їжі, що вживається користувачем. При цьому основна інформація про вміст калорій у конкретній їжі знаходиться у базі даних.

На стороні клієнта розташовано компонент системного інтерфейсу (вебінтерфейсу), який надається кінцевому користувачу. Цей рівень не має прямого підключення до бази даних (у зв'язку з вимогами безпеки та масштабованості), не завантажений основною бізнес-логікою (вимоги до масштабованості) і не підтримує стан програми (вимоги надійності). На цьому рівні здійснюється лише найпростіша бізнес-логіка: перевіряється інтерфейс авторизації, дійсність значень і їх відповідність формату даних.

На програмному рівні на сервері зосереджено більшість бізнес-логіки програмного забезпечення для підрахунку калорій. Цей рівень координує роботу програми для підрахунку калорій, обробляє команди, приймає логічні рішення та обчислення. Він обробляє дані між двома іншими рівнями програмного забезпечення.

За межами серверного рівня залишаються лише логічні елементи з бази даних про вміст калорій у їжі.

Рівень даних забезпечує зберігання даних у базі даних, що реалізована за допомогою СКБД PostgreSQL, підключення до цього компонента здійснюється тільки на рівні сервера програмного застосунку.

Б.4 Використані технічні засоби

Для експлуатації програмного забезпечення для підрахунку калорій необхідно таке апаратне та програмне забезпечення:

- підтримка сервером PostgreSQL версії не нижче 9.0;
- підтримка сервером HSQLDB версії не нижче 2.4.0;
- процесор із тактовою частотою 1,5 GHz та більше;
- оперативна пам'ять 4 ГБ та більше;
- HDD (hard disk drive), 80 Гігабайт, не менше.

Крім того, на стороні клієнта необхідна наявність на пристрої користувача доступу до Інтернету та веб-браузера.

Б.5 Виклик і завантаження

Є два варіанти запустити програмне забезпечення для підрахунку калорій: локальний запуск або запуск на PaaS-платформі. Адміністратор програмного забезпечення для підрахунку калорій повинен бути впевнений, що застосунок запущено або за допомогою Tomcat, або на Heroku.

Б.6 Вхідні та вихідні дані

Вхідними даними до програмного забезпечення для підрахунку калорій є дані, що користувач/адміністратор вводить у спеціалізовані форми на сайті. Інформація з них потрапляє у СКБД PostgreSQL у разі якщо вона валідна.

Вихідними даними програмного забезпечення для підрахунку калорій є профільтровані з СКБД дані, що можуть бути подані у вигляді таблиць або графічному вигляді за бажанням користувача.

ДОДАТОК В
Текст програми

```

//Текст файла Meal.java:
@SuppressWarnings("JpaQlInspection")
@NamedQueries({
    @NamedQuery(name = Meal.ALL_SORTED, query = "SELECT m
FROM Meal m WHERE m.user.id=:userId ORDER BY m.dateTime DESC"),
    @NamedQuery(name = Meal.DELETE, query = "DELETE FROM Meal
m WHERE m.id=:id AND m.user.id=:userId"),
    @NamedQuery(name = Meal.GET_BETWEEN, query = "SELECT m
FROM Meal m " +
        "WHERE m.user.id=:userId AND m.dateTime BETWEEN
:startDate AND :endDate ORDER BY m.dateTime DESC"),
//    @NamedQuery(name = Meal.UPDATE, query = "UPDATE Meal m
SET m.dateTime = :datetime, m.calories= :calories," +
//        "m.description=:desc where m.id=:id and
m.user.id=:userId")
})
@Entity
@Table(name = "meals", uniqueConstraints =
{@UniqueConstraint(columnNames = {"user_id", "date_time"}, name =
"meals_unique_user_datetime_idx")})
public class Meal extends AbstractBaseEntity {
    public static final String ALL_SORTED = "Meal.getAll";
    public static final String DELETE = "Meal.delete";
    public static final String GET_BETWEEN = "Meal.getBetween";

    @Column(name = "date_time", nullable = false)
    @NotNull
    @DateTimeFormat(pattern = DateTimeUtil.DATE_TIME_PATTERN)
    private LocalDateTime dateTime;

    @Column(name = "description", nullable = false)
    @NotBlank
    @Size(min = 2, max = 120)
    @SafeHtml(groups = {View.Web.class})
    private String description;

    @Column(name = "calories", nullable = false)
    @Range(min = 10, max = 5000)
    @NotNull
    private Integer calories;

    @ManyToOne(fetch = FetchType.LAZY)

```

```

@JoinColumn(name = "user_id", nullable = false)
@OnDelete(action = OnDeleteAction.CASCADE)
@NotNull(groups = View.Persist.class)
private User user;

public Meal() {
}

public Meal(LocalDate dateTime, String description, int
calories) {
    this(null, dateTime, description, calories);
}

public Meal(Integer id, LocalDate dateTime, String
description, int calories) {
    super(id);
    this.dateTime = dateTime;
    this.description = description;
    this.calories = calories;
}

public LocalDate getDateTime() {
    return dateTime;
}

public String getDescription() {
    return description;
}

public int getCalories() {
    return calories;
}

public LocalDate getDate() {
    return dateTime.toLocalDate();
}

public LocalTime getTime() {
    return dateTime.toLocalTime();
}

public void setDateTime(LocalDate dateTime) {
    this.dateTime = dateTime;
}

```

```

    }

    public void setDescription(String description) {
        this.description = description;
    }

    public void setCalories(Integer calories) {
        this.calories = calories;
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    @Override
    public String toString() {
        return "Meal{" +
            "id=" + id +
            ", dateTime=" + dateTime +
            ", description='" + description + '\'' +
            ", calories=" + calories +
            '}';
    }
}

//Текст класу RootController.java
@Controller
public class RootController extends AbstractUserController {
    @Autowired
    private MealService service;

    @GetMapping("/")
    public String root() {
        return "redirect:meals";
    }

    //    @Secured("ROLE_ADMIN")
    @PreAuthorize("hasRole('ROLE_ADMIN')")

```

```

@GetMapping("/users")
public String users() {
    return "users";
}

@GetMapping(value = "/login")
public String login() {
    return "login";
}

@GetMapping("/meals")
public String meals() {
    return "meals";
}

@GetMapping("/profile")
public String profile(ModelMap model,
@AuthenticationPrincipal AuthorizedUser authorizedUser) {
    model.addAttribute("userTo", authorizedUser.getUserTo());
    return "profile";
}

@PostMapping("/profile")
public String updateProfile(@Valid UserTo userTo,
BindingResult result, SessionStatus status,
@AuthenticationPrincipal AuthorizedUser authorizedUser) {
    if (result.hasErrors()) {
        return "profile";
    }
    try {
        super.update(userTo, authorizedUser.getId());
        authorizedUser.update(userTo);
        status.setComplete();
        return "redirect:meals";
    } catch (DataIntegrityViolationException ex) {
        result.rejectValue("email",
EXCEPTION_DUPLICATE_EMAIL);
        return "profile";
    }
}

@GetMapping("/register")
public String register(ModelMap model) {

```

```

        model.addAttribute("userTo", new UserTo());
        model.addAttribute("register", true);
        return "profile";
    }

    @PostMapping("/register")
    public String saveRegister(@Valid UserTo userTo,
    BindingResult result, SessionStatus status, ModelMap model) {
        if (result.hasErrors()) {
            model.addAttribute("register", true);
            return "profile";
        }
        try {
            super.create(UserUtil.createNewFromTo(userTo));
            status.setComplete();
            return
"redirect:login?message=app.registered&username=" +
userTo.getEmail();
        } catch (DataIntegrityViolationException ex) {
            result.rejectValue("email",
EXCEPTION_DUPLICATE_EMAIL);
            model.addAttribute("register", true);
            return "profile";
        }
    }

    @GetMapping("/norm")
    public String norm(ModelMap model) {
        model.addAttribute("norm", new TempInfo());
        return "norm";
    }

    @PostMapping("/norm")
    public String saveHello(@ModelAttribute("norm") @Valid
    TempInfo tempInfo, BindingResult result) {
        if (result.hasErrors()) {
            return "norm";
        } else {
            int norm;
            if (tempInfo.isSex())//male
            {
                norm = (int) (655.1 + 9.6 * tempInfo.getWeight()
+ 1.85 * tempInfo.getHeight() - 4.68 * tempInfo.getAge());

            } else //female

```

```

        {
            norm = (int) (66.47 + 13.75 *
tempInfo.getWeight() + 5.0 * tempInfo.getHeight() - 6.74 *
tempInfo.getAge());
        }
        AuthorizedUser authorizedUser = AuthorizedUser.get();
        UserTo userTo = authorizedUser.getUserTo();
        userTo.setCaloriesPerDay(norm);

        super.update(userTo, authorizedUser.getId());
        authorizedUser.update(userTo);

        System.out.println(norm);
        return "meals";
    }
}

@GetMapping("/plot")
public String plot(ModelMap model) {
    int userId = AuthorizedUser.id();
    model.addAttribute("list",
MealsUtil.getWithExceeded(service.getAll(userId),
AuthorizedUser.getCaloriesPerDay()));
    return "plot";
}

@PostMapping("/plot")
public String savePlot() {
    return "meals";
}
}

```

```

//Текст класы InMemoryAdminRestControllerTest.java
public class InMemoryAdminRestControllerTest {
    private static ConfigurableApplicationContext appCtx;
    private static AdminRestController controller;

    @BeforeClass
    public static void beforeClass() {
        appCtx = new
ClassPathXmlApplicationContext("spring/spring-app.xml",
"spring/mock.xml");
        System.out.println("\n" +
Arrays.toString(appCtx.getBeanDefinitionNames()) + "\n");
    }
}

```

```

        controller = appCtx.getBean(AdminRestController.class);
    }

    @AfterClass
    public static void afterClass() {
//          May cause during JUnit "Cache is not alive
//          (STATUS_SHUTDOWN)" as JUnit share Spring context for speed
//          http://stackoverflow.com/questions/16281802/ehcache-
//          shutdown-causing-an-exception-while-running-test-suite
//          appCtx.close();
    }

    @Before
    public void setUp() throws Exception {
        // re-initialize
        InMemoryUserRepositoryImpl repository =
appCtx.getBean(InMemoryUserRepositoryImpl.class);
        repository.init();
    }

    @Test
    public void testDelete() throws Exception {
        controller.delete(UserTestData.USER_ID);
        Collection<User> users = controller.getAll();
        Assert.assertEquals(users.size(), 1);
        Assert.assertEquals(users.iterator().next(), ADMIN);
    }

    @Test(expected = NotFoundException.class)
    public void testDeleteNotFound() throws Exception {
        controller.delete(10);
    }
}

```

```

//Текст класы MealServiceImpl.java
@Service
public class MealServiceImpl implements MealService {

    private final MealRepository repository;

```

```

@Autowired
public MealServiceImpl(MealRepository repository) {
    this.repository = repository;
}

@Override
public Meal get(int id, int userId) {
    return checkNotFoundWithId(repository.get(id, userId),
id);
}

@Override
public void delete(int id, int userId) {
    checkNotFoundWithId(repository.delete(id, userId), id);
}

@Override
public List<Meal> getBetweenDateTimes(LocalDateTime
startDateTime, LocalDateTime endDateTime, int userId) {
    Assert.notNull(startDateTime, "startDateTime must not be
null");
    Assert.notNull(endDateTime, "endDateTime must not be
null");
    return repository.getBetween(startDateTime, endDateTime,
userId);
}

@Override
public List<Meal> getAll(int userId) {
    return repository.getAll(userId);
}

@Override
public Meal update(Meal meal, int userId) {
    return checkNotFoundWithId(repository.save(meal, userId),
meal.getId());
}

@Override
public Meal create(Meal meal, int userId) {
    Assert.notNull(meal, "meal must not be null");
    return repository.save(meal, userId);
}

```

```

@Override
public Meal getWithUser(int id, int userId) {
    return checkNotFoundWithId(repository.getWithUser(id,
userId), id);
}
}

```

```

//Текст класы ValidationUtil.java
public class ValidationUtil {

    private ValidationUtil() {
    }

    public static void checkNotFoundWithId(boolean found, int id)
{
        checkNotFound(found, "id=" + id);
    }

    public static <T> T checkNotFoundWithId(T object, int id) {
        return checkNotFound(object, "id=" + id);
    }

    public static <T> T checkNotFound(T object, String msg) {
        checkNotFound(object != null, msg);
        return object;
    }

    public static void checkNotFound(boolean found, String arg) {
        if (!found) {
            throw new NotFoundException(arg);
        }
    }

    public static void checkNew(HasId bean) {
        if (!bean.isNew()) {
            throw new IllegalArgumentException(bean + " must
be new (id=null)");
        }
    }
}

```

```

    public static void assureIdConsistent(HasId bean, int id) {
        if (bean.isNew()) {
            bean.setId(id);
        } else if (bean.getId() != id) {
            throw new IllegalArgumentException(bean + " must
be with id=" + id);
        }
    }

    public static Throwable getRootCause(Throwable t) {
        Throwable result = t;
        Throwable cause;

        while (null != (cause = result.getCause()) && (result !=
cause)) {
            result = cause;
        }
        return result;
    }

    public static String getMessage(Throwable e) {
        return e.getLocalizedMessage() != null ?
e.getLocalizedMessage() : e.getClass().getName();
    }
}

```

```

// Текст класу GlobalExceptionHandler.java
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.NoHandlerFoundException;
import ru.javawebinar.topjava.util.ValidationUtil;
import
ru.javawebinar.topjava.util.exception.ApplicationException;
import ru.javawebinar.topjava.util.exception.ErrorType;

import javax.servlet.http.HttpServletRequest;

@ControllerAdvice

```

```

public class GlobalControllerExceptionHandler {
    private static final Logger log =
LoggerFactory.getLogger(GlobalControllerExceptionHandler.class);

    @Autowired
    private MessageUtil messageUtil;

    @ExceptionHandler(NoHandlerFoundException.class)
    public ModelAndView wrongRequest(HttpServletRequestRequest req,
NoHandlerFoundException e) throws Exception {
        return logAndGetExceptionView(req, e, false,
ErrorType.WRONG_REQUEST, null);
    }

    @ExceptionHandler(ApplicationException.class)
    public ModelAndView
applicationErrorHandler(HttpServletRequestRequest req,
ApplicationException appEx) throws Exception {
        return logAndGetExceptionView(req, appEx, true,
appEx.getType(), messageUtil.getMessage(appEx));
    }

    @ExceptionHandler(Exception.class)
    public ModelAndView defaultErrorHandler(HttpServletRequestRequest
req, Exception e) throws Exception {
        return logAndGetExceptionView(req, e, true,
ErrorType.APP_ERROR, null);
    }

    private ModelAndView
logAndGetExceptionView(HttpServletRequestRequest req, Exception e,
boolean logException, ErrorType errorType, String msg) {
        Throwable rootCause = ValidationUtil.getRootCause(e);
        if (logException) {
            log.error(errorType + " at request " +
req.getRequestURL(), rootCause);
        } else {
            log.warn("{} at request {}: {}", errorType,
req.getRequestURL(), rootCause.toString());
        }
        ModelAndView mav = new
ModelAndView("exception/exception");

```

```

        mav.addObject("typeMessage",
messageUtil.getMessage(errorType.getErrorCode()));
        mav.addObject("exception", rootCause);
        mav.addObject("message", msg != null ? msg :
ValidationUtil.getMessage(rootCause));
        return mav;
    }
}

```

```

// Текст класы AbstractUserController.java
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.env.Environment;
import ru.javawebinar.topjava.Profiles;
import ru.javawebinar.topjava.model.AbstractBaseEntity;
import ru.javawebinar.topjava.model.User;
import ru.javawebinar.topjava.service.UserService;
import ru.javawebinar.topjava.to.UserTo;
import
ru.javawebinar.topjava.util.exception.ModificationRestrictionExce
ption;

import java.util.List;

import static
ru.javawebinar.topjava.util.ValidationUtil.assureIdConsistent;
import static
ru.javawebinar.topjava.util.ValidationUtil.checkNew;

public abstract class AbstractUserController {
    protected final Logger log =
LoggerFactory.getLogger(getClass());

    @Autowired
    private UserService service;

    private boolean modificationRestriction;

    @Autowired
    public void setEnvironment(Environment environment) {

```

```

        modificationRestriction =
environment.acceptsProfiles(Profiles.HEROKU);
    }

    public List<User> getAll() {
        log.info("getAll");
        return service.getAll();
    }

    public User get(int id) {
        log.info("get {}", id);
        return service.get(id);
    }

    public User create(User user) {
        log.info("create {}", user);
        checkNew(user);
        return service.create(user);
    }

    public void delete(int id) {
        log.info("delete {}", id);
        checkModificationAllowed(id);
        service.delete(id);
    }

    public void update(User user, int id) {
        log.info("update {} with id={}", user, id);
        assureIdConsistent(user, id);
        checkModificationAllowed(id);
        service.update(user);
    }

    public void update(UserTo userTo, int id) {
        log.info("update {} with id={}", userTo, id);
        assureIdConsistent(userTo, id);
        checkModificationAllowed(id);
        service.update(userTo);
    }

    public User getByMail(String email) {
        log.info("getByEmail {}", email);
        return service.getByMail(email);
    }

```

```

    }

    public void enable(int id, boolean enabled) {
        log.info((enabled ? "enable " : "disable ") + id);
        checkModificationAllowed(id);
        service.enable(id, enabled);
    }

    private void checkModificationAllowed(int id) {
        if (modificationRestriction && id <
AbstractBaseEntity.START_SEQ + 2) {
            throw new ModificationRestrictionException();
        }
    }
}

```

```

// Текст класу style.css
tr[data-mealExceed="false"] {
    color: green;
}

tr[data-mealExceed="true"] {
    color: red;
}

.fa {
    cursor: pointer;
}

tr[data-userEnabled="false"] {
    opacity: 0.3;
}

.error, .message {
    padding: 10px;
    border-radius: 4px;
    font-size: 16px;
}

.error {
    color: #a94442;
    background-color: #f2dede;
}

```

```

        border: 1px solid #ebccd1;
    }

    .message {
        color: #2f9635;
        background-color: #c6fbc2;
        border: 1px solid #9feba6;
    }

    /*https://getbootstrap.com/docs/4.0/examples/sticky-
footer/sticky-footer.css*/
    html {
        position: relative;
        min-height: 100%;
    }
    body {
        margin-bottom: 60px; /* Margin bottom by footer height
*/
        .footer {
            position: absolute;
            bottom: 0;
            width: 100%;
            height: 60px; /* Set the fixed height of the footer here
*/
            line-height: 60px; /* Vertically center the text there
*/
            background-color: #f5f5f5;

```

ДОДАТОК Д
Слайди презентації



Рисунок Д.1 – Слайд 1

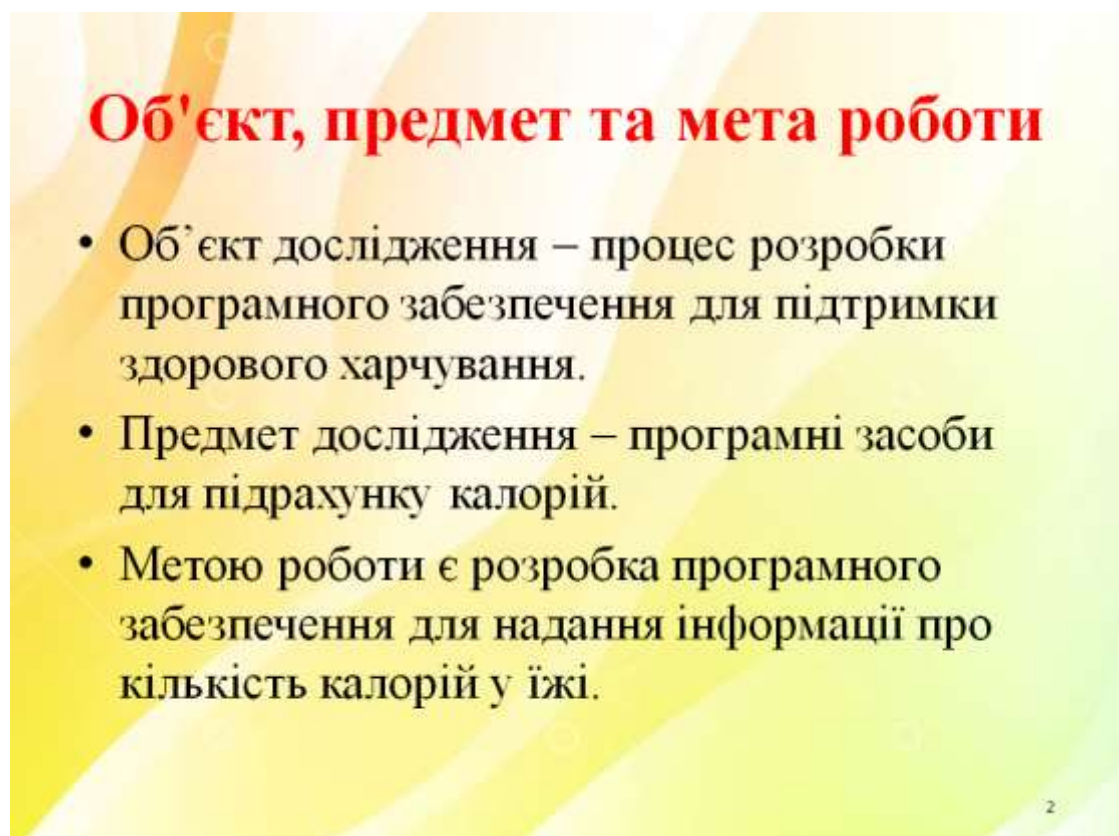


Рисунок Д.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для розрахунку кількості калорій спожитої їжі;
- здійснити проєктування програмного забезпечення для надання інформації про для надання інформації про кількість калорій у їжі, що споживається;
- створити програмне забезпечення для розрахунку кількості калорій спожитої їжі;
- виконати тестування розробленого програмного забезпечення для надання інформації про кількість калорій у їжі, що споживається.

3

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерій, за яким порівнюють	My FitnessPal	Lose It!	FatSecret	Cronometer	Lifesum
Наявність сканеру їжі та штрихкодів	+	–	–	+	–
База даних продуктів	–	+	+	+	+
Безкоштовність	–	–	+	–	–
Зручність інтерфейсу	+	–	+	+	–
Наявність рекомендацій для тренувань	+	–	–	–	+
Можливість роботи з різними ОС	–	+	–	–	–

4

Рисунок Д.4 – Слайд 4

Функціональні вимоги до розробки

Необхідно створити програмне забезпечення для підтримки здорового харчування та розрахунку кількості калорій. При розробці програмного забезпечення для підтримки здорового харчування та розрахунку кількості калорій необхідно забезпечити такі функціональні вимоги:

- реєстрація, авторизація та аутентифікація користувачів;
- CRUD (create, read, update, delete) операції з користувачами (права адміністратора);
- CRUD операції з їжею;
- фільтрація за датою та часом;
- побудова графіку розподілу калорій;
- розрахунок кількості калорій, яку необхідно скоротити в раціоні за антропометричними даними.

5

Рисунок Д.5 – Слайд 5

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Java	C++	PHP
Зручність роботи з масивами даних	+	+	+–
Засоби для роботи з базами даних	+	+–	+
Засоби паралельного програмування	+	+–	–
Зручність для створення ПЗ для підрахунку калорій	+	+–	+–
Зручність керування пам'яттю	+	+	–

6

Рисунок Д.6 – Слайд 6

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовище розробки		
	IntelliJ IDEA	Eclipse	MS Visual Studio
Зручність для створення програмного забезпечення для підрахунку калорій	+	+-	+-
Розуміння контексту (autocomplete)	+	-	+-
Інтелектуальний рефакторинг	+	-	+-
Кросплатформність	+	+-	+
Розробка GUI	+	+-	+

Рисунок Д.7 – Слайд 7

Схема взаємодії основних модулів програмного забезпечення для підрахунку калорій



Рисунок Д.8 – Слайд 8

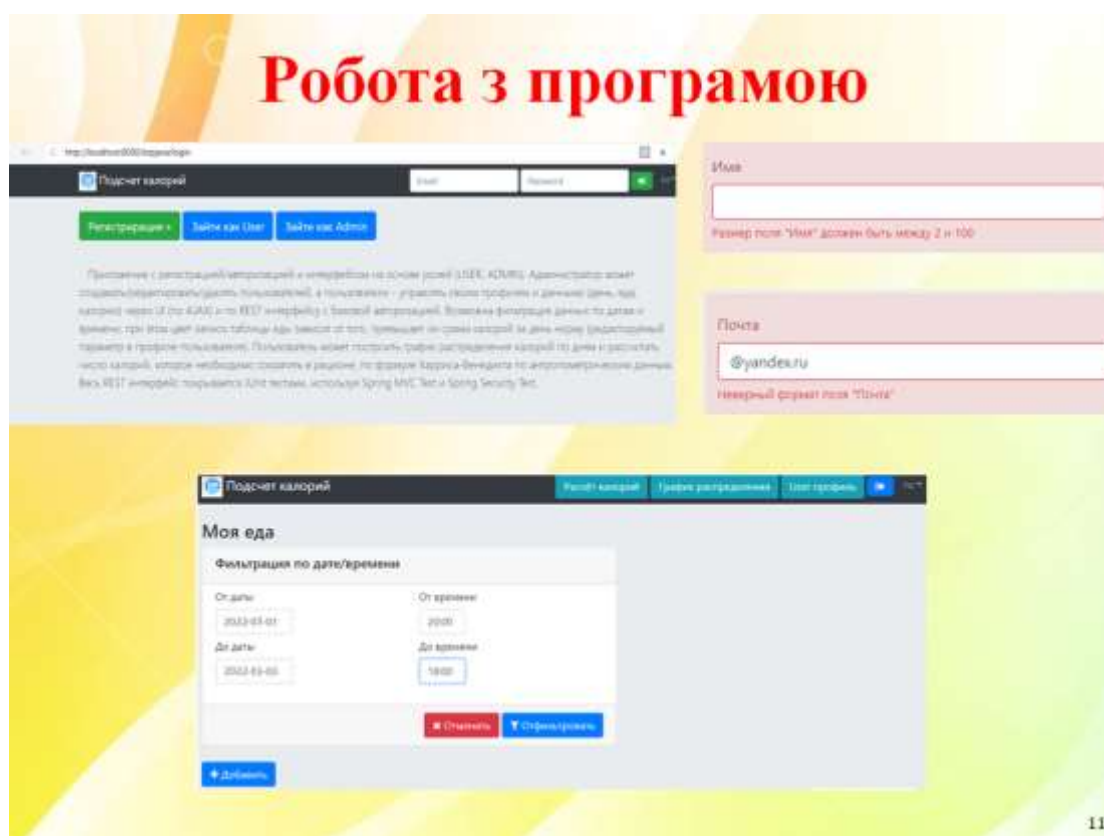


Рисунок Д.11 – Слайд 11

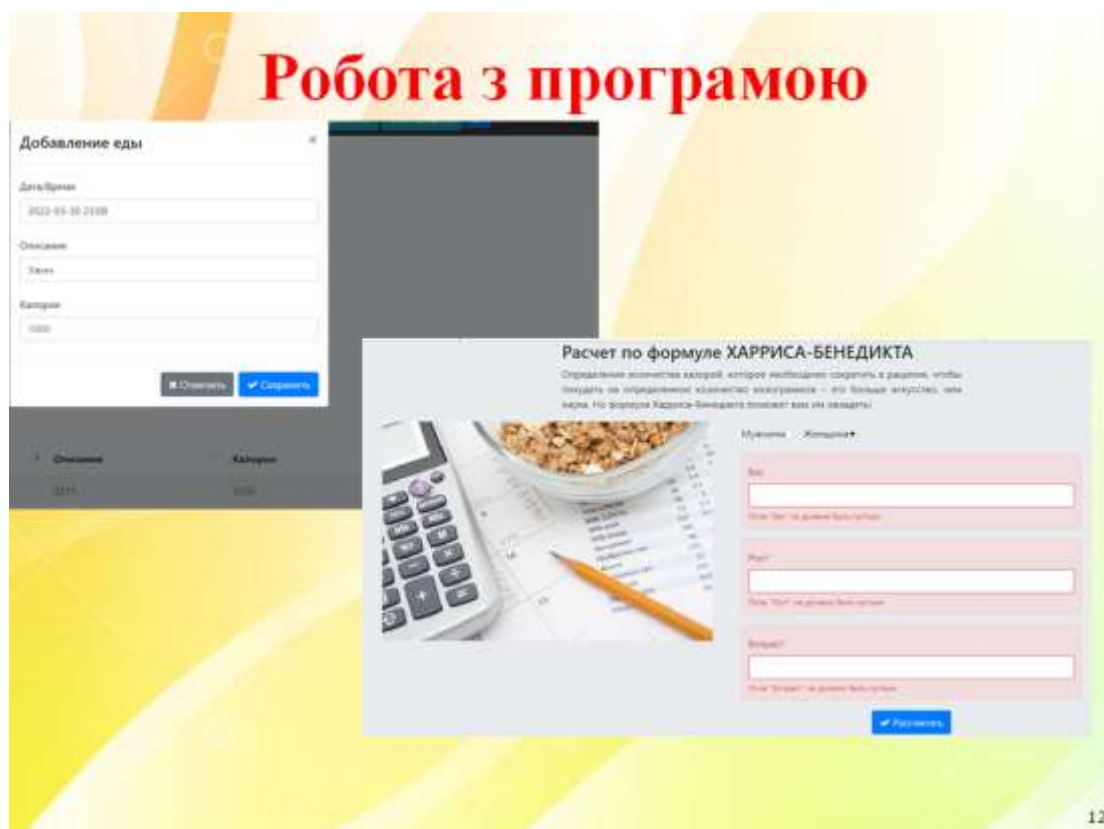


Рисунок Д.12 – Слайд 12

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для підтримки здорового харчування.

Проаналізовано поняття про раціональне харчування та розрахунок кількості калорій. Виконано дослідження програмних засобів для розрахунку кількості калорій, зокрема проаналізовано таке програмне забезпечення: My FitnessPal, Lose It!, FatSecret, Cronometer, LifeSum.

Визначено функціональні вимоги, яким повинно задовольняти програмне забезпечення для підтримки здорового харчування та розрахунку кількості калорій.

Розроблено програмне забезпечення для підрахунку калорій, у якому основними структурними компонентами є серверний компонент, що реалізує основні функції програмного забезпечення для підрахунку калорій, модуль вебінтерфейсу, що забезпечує взаємодію користувача з розробленою програмою, а також база даних про вміст калорій у продуктах споживання.

Виконано тестування розробленого програмного забезпечення для підрахунку калорій. Результати тестування програмного забезпечення для підрахунку калорій підтвердили його працездатність.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

13

Рисунок Д.13 – Слайд 13

Дякую за увагу!!!

14

Рисунок Д.14 – Слайд 14