

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«___» _____ 2019 р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

ІКС оптимізації харчування

Освітньо-кваліфікаційний рівень “бакалавр”
Спеціальність 123 “Комп’ютерна інженерія”

Керівник проекту:

_____ (підпис)

доц. Сафонова С.О.

_____ (ініціали, прізвище)

Консультант з охорони праці:

_____ (підпис)

ст.викл. Критська Я.О.

_____ (ініціали, прізвище)

Здобувач вищої освіти:

_____ (підпис)

Старовойтова А.В.

_____ (ініціали, прізвище)

Група:

_____ КІ-15бд

Севєродонецьк 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 123 "Комп'ютерна інженерія"
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
І.С. Скарга-Бандурова
« _____ » _____ 2019 р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Старовойтовій Аліні Вікторівні

(прізвище, ім'я, по батькові)

1. Тема роботи ІКС оптимізації харчування
- керівник проекту (роботи) Сафонова С.О., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
- затверджені наказом вищого навчального закладу від " 14 " 05 2019 р. № _____
2. Термін подання студентом роботи _____
3. Вихідні дані до роботи матеріали переддипломної практики
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області і постановка задачі.
Обробка математичної моделі оптимізації харчування.
Аналіз структур даних.
Пошук необхідних технологій для реалізації.
Розробка веб-додатку регулювання харчування. Тестування розробленої програмної системи. Охорона праці та безпека в надзвичайних ситуаціях.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст.викл. кафедри КНІ Критська Я.О.		

7. Дата видачі завдання _____

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання до роботи	14.05.19-16.05.19	
2	Аналіз завдання, огляд літератури	17.05.19-20.05.19	
3	Аналіз технічних засобів	21.05.19-24.06.19	
4	Розробка алгоритму	25.06.19-02.06.19	
5	Програмна реалізація	03.06.19-06.06.19	
6	Оформлення пояснювальної записки	07.06.19-09.06.19	
7	Підготовка презентації та доповіді	10.06.19-13.06.19	

Здобувач вищої освіти

_____ (підпис)

Керівник

_____ (підпис)

Старовойтова А.В.

_____ (прізвище та ініціали)

Сафонова С.О.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 101с., 50 рис., 4 табл., 13 бібліографічних джерел посилань, 2 додатка.

Об'єкт розробки: процеси регулювання та оптимізації харчування.

Мета роботи: розробка програмної системи регулювання харчування.

В проекті виконано:

- проведено аналіз предметної області;
- визначено особливості регулювання харчування;
- розроблено моделі регулювання та оптимізації харчування;
- обрано технологію та розроблено модулі програмного додатку регулювання харчування;
- перевірено роботу системи з використанням тестового матеріалу;
- здійснено аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкта, що впливають на персонал.

Під час розробки веб-додатку були використані: теорія проєктування no-SQL баз даних на прикладі MongoDB; основні принципи побудови архітектури веб-додатку з використанням Gradle; принципи контролю історії змінень файлів та їх версіонування за допомогою системи контролю версій Git; середовище розробки програмних систем IntelliJ IDEA, мови HTML та CSS, мова програмування Java та бібліотеки (Jsoup, Spring та інші).

Отримано наступні результати: розроблено помічника, здатного за короткий час запропонувати рецепт, виходячи з переваг, фінансових можливостей і особливостей харчування будь-якої людини.

Практичне значення, галузь застосування роботи: розроблений сервіс є помічником у регулюванні харчування.

Ключові слова: ВЕБ-ДОДАТОК, NO-SQL, НЕРЕЛЯЦІЙНА БАЗА ДАНИХ, БАГАТОМОДУЛЬНІСТЬ, СИСТЕМА КОНТРОЛЮ ВЕРСІЙ.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєверодонецьк, 93400.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області. Регулювання харчування.....	8
1.1 Актуальність задачі	8
1.2 Огляд існуючих систем.....	8
1.2.1 Система «МІЙ ДІЄТОЛОГ»	8
1.2.2 MyFitnessPal (веб-версія).....	20
1.3 Постановка задачі.....	28
2 Методи регулювання харчування.....	32
2.1 Математичні моделі регулювання та оптимізації харчування	32
2.2 Аналіз функціональних вимог до системи регулювання харчування	36
2.2.1 Аналіз системи зберігання та обробки рецептів	36
2.2.2 Структури даних.....	37
2.3 Розробка архітектури веб-додатку. Опис розроблених на даний момент модулів	40
2.4 Реалізація функціоналу	42
2.4.1 Опис класів та функцій	42
2.4.2 Методи обробки даних.....	47
3 Програмна реалізація. Аналіз отриманих результатів.....	48
3.1 Опис інструментів для розробки.....	48
3.1.1 Опис використовуваних мов програмування, розмітки	48
3.1.2 Опис використаних технологій	51
3.2 Аналіз функцій системи регулювання харчування.....	54
3.2.1 Опис задачі 1. Відображення загальної інформації про всі рецепти	56
3.2.2 Опис задачі 2. Повна інформація про певний рецепт.....	58
3.2.3 Опис задачі 3. Збереження інформації про рецепт у форматі .doc	59
3.2.4 Опис задачі 4. Видалення рецептів.....	61
3.2.5 Опис задачі 5. Редагування рецептів	61

3.2.6	Опис задачі 6. Додання нового рецепту до бази даних	63
3.2.7	Опис задачі 7. Пошук рецептів	64
3.2.8	Опис задачі 8. Наповнення бази даних рецептами через адміністратора	65
4	Охорона праці та безпека в надзвичайних ситуаціях	67
4.1	Загальні питання з охорони праці	67
4.1.1	Правові та організаційні основи охорони праці	68
4.1.2	Організаційно-технічні заходи з безпеки праці	68
4.2	Аналіз стану умов праці	69
4.2.1	Вимоги до приміщень	69
4.2.2	Вимоги до організації місця праці.....	70
4.2.3	Навантаження на напруженість процесу праці.....	71
4.3	Виробнича санітарія.....	71
4.3.1	Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) вибору	72
4.3.2	Пожежна безпека.....	73
4.4	Гігієнічні вимоги до параметрів виробничого середовища.....	75
4.4.1	Мікроклімат	75
4.4.2	Освітлення.....	76
4.4.3	Шум та вібрація, електромагнітне випромінювання.....	77
4.4.4	Вентилювання.....	78
4.5	Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	78
4.6	Розрахунок захисного заземлення(забезпечення безпеки будівлі).....	80
	Висновки до розділу 4	84
	Перелік корисних посилань до розділу 4	85
	Висновки	86
	Перелік джерел посилань	88
	Додаток А Комп'ютерна презентація	90
	Додаток Б Лістинг коду DailyMenuRecipeDocumentParser.java	98
	Додаток В Лістинг коду Grabber.java.....	101

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БЖВ – білки, жири і вуглеводи

ІМТ – індекс маси тіла

CSV – comma-separated values

IDE – integrated development environment

СУБД – система управління базами даних

FAO – food and agriculture organization

ВООЗ – всесвітня організація охорони здоров'я

JSON – JavaScript Object Notation

SQL – structured query language

NoSQL – not only SQL

DTO – data transfer object

JDBC – Java DataBase Connectivity

MVC – Model-View-Controller

HTML – Hyper Text Markup Language

CSS – Cascade Style Sheets

JS – JavaScript

ВСТУП

Потреба людини в регулюванні власного харчування в наш час підвищується при підвищенні рівня зацікавленості населення в цьому з різних причин. Спортсмени, які бажають схуднути, страждаючі від захворювань, змушені дотримуватися певних обмежень в харчуванні тимчасово або ж постійно, або просто ті, хто прагне до здорового способу життя. Контингент користувачів, яким так чи інакше потрібен контроль харчування, величезний, і потреба в якійсь інтелектуальній системі, яка допомагала б задовольнити вимоги вищеописаної цільової аудиторії в повній мірі, стає ребром [1].

Насправді, нинішні рішення на ринку так чи інакше намагаються покрити необхідність сучасної людини в контролі харчування, але всі страждають від тих чи інших недоліків, – змушують користувача вводити велику кількість інформації і витратити багато часу на фіксування кількості спожитих калорій, так званого БЖВ, рахувати власноруч ці самі калорії (наприклад, точно самостійно визначити калорійність борщу, напевно це завдання займе не одну хвилину), самостійно регулювати кількість грам в одній порції і так далі. Деякі і зовсім надають просто недостовірну інформацію (особливо коли занесені в базу даних рецепти ніяк не модеруються, і користувачі додають їх вільно, вводячи в оману інших).

Дипломна робота включає у себе аналіз існуючих рішень, постановку задачі, опис використаних технологій, спробу частково реалізувати власне рішення та аналіз отриманих результатів, висновок стосовно того, наскільки та чи інша технологія здатна спростити процес розробки запланованого функціоналу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. РЕГУЛЮВАННЯ ХАРЧУВАННЯ

1.1 Актуальність задачі

Питання, як видно з описаних у вступі проблем, є досить актуальним, – аудиторія, яка в тій чи іншій мірі потребує програмних рішень так званих помічників регулювання харчування, досить велика.

Велика кількість додатків пропонує доволі обмежений перелік дієт, розрекламованих зірками, але які не підтвердили своєї ефективності або які не є благонадійними з точки зору спеціалістів.

Розробка програмного забезпечення подібного профілю має достатній попит на сьогоднішньому ринку. Виходячи з цього, даний концепт додатку має всі шанси бути розвиненим до повноцінного проекту, здатного нести користь оточуючим і прибуток власнику за рахунок реклами, наприклад [2].

1.2 Огляд існуючих систем

Зважаючи на те, що ідея далеко не нова, треба порівняти вже існуючі рішення на ринку та оцінити їх недоліки та переваги.

1.2.1 Система «МІЙ ДІЄТОЛОГ»

Система «МІЙ ДІЄТОЛОГ» – це інструмент планування харчування, від складання раціонів до замовлення продуктів у інтернет-магазині, що опирається на наукові розробки та практичні методики дієтологів

Федеральної державної бюджетної установи науки «Федеральний дослідний центр харчування, біотехнології та безпеки їжі» (рис.1.1).

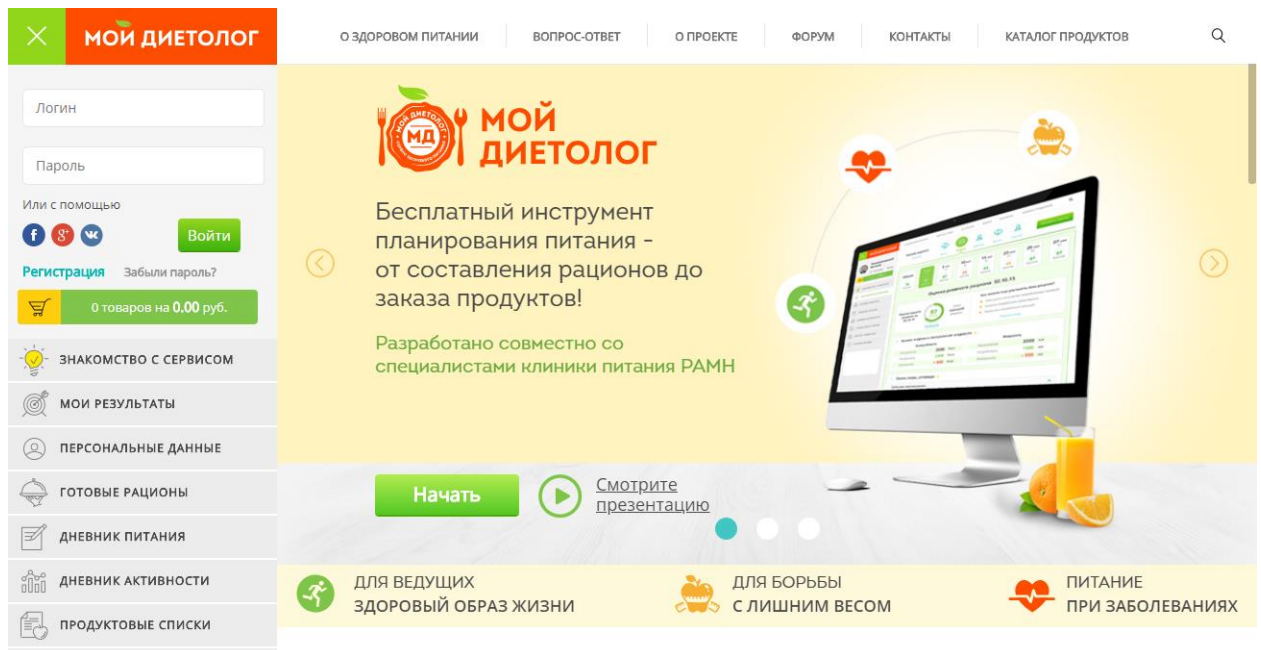


Рисунок 1.1 – Головна сторінка сайту <https://moydietolog.ru/>

Проект «МІЙ ДІЕТОЛОГ» створений спільними зусиллями представників наукових, громадських, комерційних і торгівельних організацій, об'єднані єдиною метою – запропонувати широкому колу користувачів реальну можливість перейти на правильне, здорове харчування. На сторінці «Про проект» власники сайту обіцяють, що навіть для початківця, що бажає перейти на правильне харчування, але не знає з чого почати, система стане реальним помічником організації здорового харчування.

Всі матеріали, опубліковані на сайті, складені експертами в області харчування – фахівцями Федеральної державної бюджетної установи науки «Федеральний дослідний центр харчування, біотехнології та безпеки їжі», практикуючими дієтологами та нутриціологами – вченими, які вивчають вплив харчування на здоров'я людини. Сайт запевняє, що їх багаторічний досвід дозволяє з упевненістю судити про принципи і правила здорового

харчування – саме ці принципи і алгоритми покладені в основу дієтологічного сервісу «МІЙ ДІЄТОЛОГ».

На сторінках сайту moydietolog.ru опубліковані матеріали, присвячені найбільш частим питанням харчування: надмірна вага, калорійність і склад продуктів, харчові забруднювачі, особливості дітей різних вікових груп, вплив харчування на виникнення харчової алергії, особливості харчування спортсменів і багато інших.

Всі права на матеріали сайту moydietolog.ru і сервіс «МІЙ ДІЄТОЛОГ» належать компанії ТОВ «Центр здорового харчування». Створювачі ресурсу стверджують, що ресурс є незалежним, і це дозволяє їм відкрито доносити до споживача достовірну інформацію, вільну від впливів різного характеру.

У 2015 році сервіс «МІЙ ДІЄТОЛОГ» пройшов незалежну експертну оцінку, визнаний інноваційним, а компанія отримала статус учасника проекту «Сколково» [3].

Сервіс «МІЙ ДІЄТОЛОГ» є безкоштовним російськомовним інструментом. Можливості змінити мову немає.

Слід зазначити, що на сайті, окрім безпосередньо функціоналу, присутня велика кількість загальної інформації: довідник для знайомства з сервісом, статті про здорове харчування, енциклопедія про продукти, їх склад, вітаміни та мінеральні речовини, харчові забруднювачі, харчові добавки та новини дієтології. Інформація структурована таким чином, що її відносно легко знайти інтуїтивно, і немає ніяких причин сумніватися у достовірності описаної інформації (рис. 1.2).

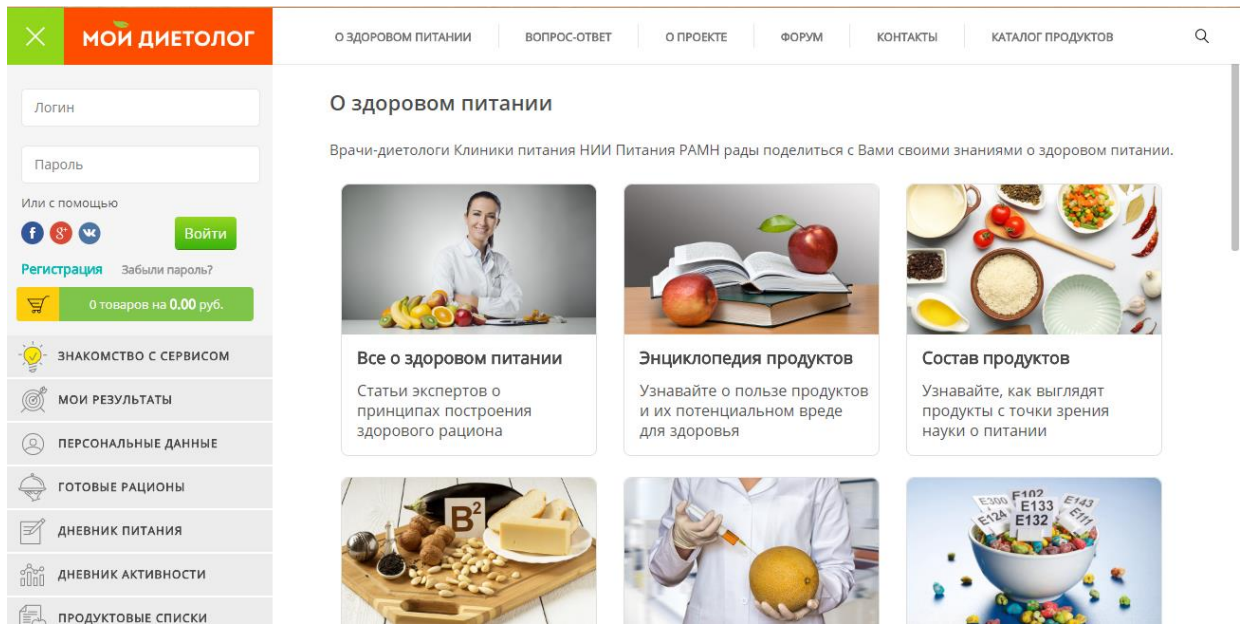


Рисунок 1.2 – Розділ «Про здорове харчування»

На сайті присутній також форум, який, по задумці авторів, повинен був стати зв'язком між авторами та користувачами, але, на жаль, не отримав попиту. На момент написання роботи там майже нема активності (лише 6 повідомлень, останнє написане у листопаді 2017 року).

У користувача є можливість створити власний аккаунт через безпосередню реєстрацію або за допомогою авторизації через соціальні мережі – Facebook, Google+ або Вконтакті.

Після авторизації у користувача є можливість описати персональні дані – ім'я, стать, кількість років, вага, зріст, рівень фізичної активності, свої захворювання та алергії, опція «Хочу скинути вагу» для тих, хто бажає споживати менше калорій, ніж є нормою для нього (рис.1.3).

Персональные данные показать подсказку

Не скажу Анна +

Имя:

Пол:

Возраст, лет: 21

Рост, см: 165

Текущий вес, кг: 75

Хочу сбросить вес:

Желаемый вес, кг: 58

Оцените вашу физическую активность

1 2 3 4

Регулярные физические нагрузки: фитнес или занятия спортом 2-3 раза в неделю, регулярные работы по дому или в саду, подвижные игры, либо работа с постоянными умеренными физическими нагрузками.

Особые состояния:

Для точного расчета, укажите, пожалуйста, находитесь ли вы в состоянии беременности или являетесь кормящей мамой.

Заболевания Показывать полный список

Аллергии Показывать полный список

Для того, чтобы дать вам правильные рекомендации, укажите, пожалуйста, имеются ли у вас какие-нибудь заболевания и / или аллергии - это поможет исключить продукты, которые вам употреблять не стоит.

Не скажу, у Вас избыточная масса тела (предупреждение)

ИМТ - 27.5

Потребность организма в калориях: **2517**

Ваша цель: снизить вес до 58 кг
Вам рекомендовано снижать вес в **3 этапа**
Цель данного этапа: похудеть на **8 кг** за **7 недель**
Общая продолжительность диеты: **14 недель**

Калорийность рациона на период диеты:
2017 ккал

[Посмотреть подобранные Вам рационы](#)

Используйте Дневник Питания для планирования, контроля и корректировки рациона

Рисунок 1.3 – Интерфейс персональных данных

Для жінок присутня опція «Вагітна» або «Мати, що годує дитину груддю». Є можливість зберегти декілька подібних профілів та переключатися між ними у час користування сервісом. Після натискання кнопки «Розрахувати» сайт розрахує ІМТ, добову потребу калорій, рекомендації щодо схуднення, калорійність на період дієти, та запропонує переглянути обрані системою дієти.

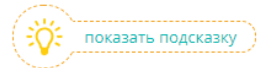
Є вкладка «Готові раціони», де присутня можливість відшукати рецепти, користуючись параметрами пошуку, такими як добова калорійність, вікова категорія, наявність захворювань та спеціальні теги (вегетаріанські,

високобілкові, низькокалорійні, низькохолестерінові, розвантажувальні дні). Потрібно зазначити, що сервіс дозволяє шукати лише раціони цілком, часто розраховані на декілька днів, тобто, нема можливості відшукати саме страву. Присутня можливість сортувати результат пошуку по калорійності та популярності. У кожного раціону присутні короткий опис, загальна кількість калорій, кількість переглядів, людей, що обрали цей раціон та відгуків, середній рейтинг (проте, відгуки майже відсутні, як і рейтинг).

Сервіс не дозволяє обрати бажані та небажані продукти, відфільтрувати пошук по кількості БЖВ або коштовності у фінансовому плані (хоча це могло бути логічним, зважаючи на наявність власного магазину продуктів).

Після натиску на назву раціону або гіперпосилання «Детальніше» іде перенаправлення на сторінку самого раціону (рис.1.4). Окрім описаної вище загальної інформації є можливість додати цей раціон до власного профілю. Нижче доволі зрозуміло описано дні раціону, послідовність прийомів їжі, страви, їх кількість (рис.1.5). Можна натиснути на страву та отримати детальну інформацію, – БЖВ, калорійність на 100 грам, рецепт приготування та склад. У користувача наявне вікно з кількістю грам у порції, яке можна редагувати. В залежності від цього параметра будуть розраховуватися кількість грам кожної складової. Після натискання кнопки «Обрати цей раціон» можна занести дані про нього, наприклад, коли користувач збирається годуватися цим раціоном, хто з профілів буде його дотримуватися.

Готовые рационы



Рационы и диеты различной калорийности

Рационы для людей, имеющих заболевания

Мои избранные рационы

Калорийность от 100 до 5000 ккал

Возрастная категория Дети (3-8 лет) Дети (8-14 лет) Взрослые (15-55 лет) Взрослые (старше 55 лет)

Вегетарианские Высокобелковые Низкокалорийные Низкохолестериновые
 Разгрузочные дни

Сбросить фильтры

Подобрано 3 рационов

Сортировать по: популярности калорийности↑



«Детский. Традиционный. Вариант 1» на 7 дней

☆☆☆☆☆

Рацион богат витаминами и минеральными веществами, необходимыми растущему организму. Каши, супы, йогурт, творог, фрукты, овощи, отварное мясо, рыба и «здоровые» сладости составят полноценное питание физически активного ребенка дошкольного возраста. Н...

В избранное [Подробнее](#)

1500

ккал/день
(средняя)

Просмотров	Выбрали	Отзывов
11	1	0

Рисунок 1.4 – Интерфейс пошуку раціонів

1 день 2 день 3 день 4 день 5 день 6 день 7 день

Первый завтрак

Второй завтрак


Обед

Полдник

Ужин


Поздний ужин


Первый завтрак

 200г


Манная каша молочная густая


Обед

 40г Бутерброд с маслом на белом хлебе

 150г Какао с молоком и сахаром

Второй завтрак

 130г Яблоко

 150г Вода питьевая столовая

Обед

Рисунок 1.5 – Зразок опису раціону

Внизу сторінки є можливість обрати продуктовий список на одну чи кілька персон, та після натискання кнопки «Сформувати» користувач буде перенаправлений на список покупок з магазину власників сайту. Він буде сформований у залежності від складових рецептів страв раціону. Людина має можливість скоригувати список, обравши продукти з мінімальною вартістю, продукти, що можна замінити, ті, що найкраще відповідають рецепту або ж якісь певні, що є до вподоби самому користувачеві.

Нижче буде описано кількість продуктів, їх загальна вартість, пропонування роздрукувати список або покласти усе у корзину. Напроти кожної позиції є поле вводу для бажаної кількості продукту у грамах або штуках, його ціна; присутня можливість замінити продукт на якийсь аналог. Якщо розгорнути вкладку «Додатково», можна додати до корзини «необов'язкові» продукти, тобто ті, що з великою долею ймовірності вже є вдома (вода, чай, цукор, сіль тощо). Майже під кожним найменуванням приводиться уточнення, для якої страви потрібен певний компонент (рис.1.6).

Выбрано 68		Стоимость ? 8 681 ₺		Распечатать список блюд		Положить в корзину 0	
Основные товары(48) ? — Свернуть				Количество		Стоимость	
Крупы и макароны							
<input checked="" type="checkbox"/>		Манная крупа Чистая крупа, 800г ♡ <i>Манная каша молочная густая, Пудинг из творога 9% Еще</i>	— 1 + ШТ	51.00 ₺	Похожие продукты		
<input checked="" type="checkbox"/>		Макароны Макфа Рожки гладкие, 450г ♡ <i>Куриный суп вермишелевый, Вермишель отварная с тертым сыром</i>	— 1 + ШТ	48.90 ₺	Похожие продукты		
<input checked="" type="checkbox"/>		Макароны Barilla Filini Vermicelles n.30, 500г ♡ <i>Макароны отварные</i>	— 1 + ШТ	86.10 ₺	Похожие продукты		
<input checked="" type="checkbox"/>		Рис Мистраль Кубань круглозерный, 900г ♡ <i>Тефтели из говядины с рисом паровые со сливочным маслом</i>	— 1 + ШТ	94.50 ₺	Похожие продукты		
<input checked="" type="checkbox"/>		Гречка Чистая крупа ядрица, 900г ♡ <i>Гречка рассыпчатая с молоком без сахара</i>	— 1 + ШТ	147.00 ₺			
<input checked="" type="checkbox"/>		Овсяная крупа Ярмарка, 700г ♡ <i>Овсяная молочная каша из хлопья</i>	— 1 + ШТ	69.40 ₺			

Рисунок 1.6 – Сформований з раціону список необхідних продуктів

Проте, важливо зазначити, що, нажаль, даний функціонал доступний лише для росіян, в обмеженому переліку міст. В Україні замовити продукти прямо з сайту не є можливим.

У розділі «Мої результати» (рис.1.7) користувач має змогу ознайомитися зі своїми результатами. Сайт на цій сторінці дає зведену інформацію, основується на отриманих даних про обрані раціони, фізичну активність, персональні дані про параметри тіла та спосіб життя, та надає оцінку відповідності обраного раціону потребам організму.

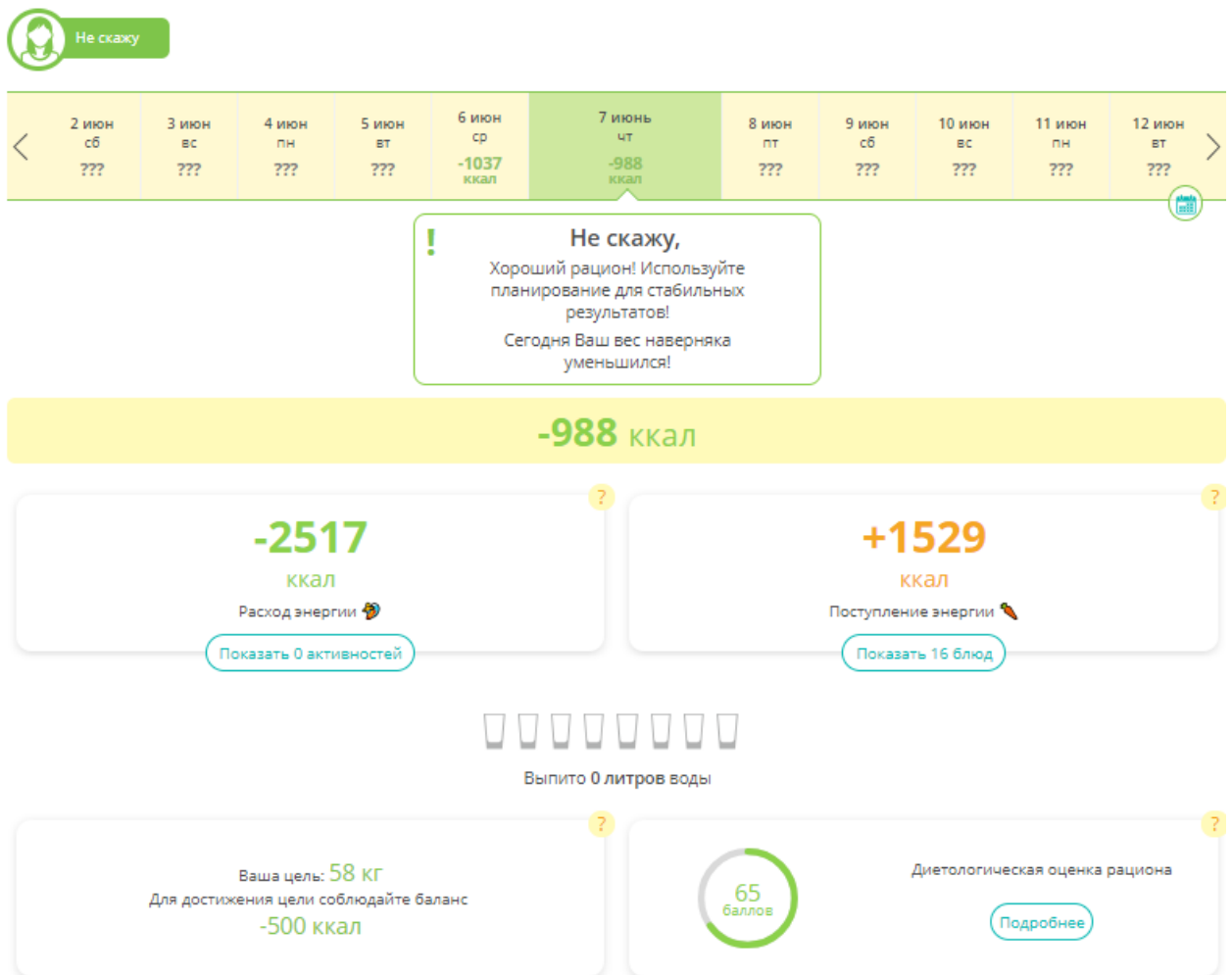


Рисунок 1.7 – Розділ «Мої результати»

У верхній частині сторінки сайт демонструє інформацію о різниці калорій по днях, а нижче надає рекомендацію стосовно того, наскільки

ефективною є різниця калорій, що спожив користувач, від тої, що він теоретично витратив. Нижче зазначені теоретично витрачені калорії, що розраховуються у залежності від віку користувача, його фізичної активності, нинішньої ваги, та споживані калорії, які він задає самостійно, помічаючи певний раціон як використаний. Після можна побачити, скільки води випито за день, кінцеву мету дієти, рекомендацію стосовно того, якої різниці слід дотримуватися, щоб худнути найбільш ефективно, та дієтологічна оцінка раціону, що розраховується в залежності від персональних даних та особливостей мети правильного харчування. Оцінка, більша за 70, вважається оптимальною.

У «Щоденнику харчування» (рис.1.8) зведена інформація про обраний раціон, його якість, що розраховується за допомогою дієтологічної оцінки, описані мета, БЖВ дня раціону та список страв на день. Червоним кольором позначаються продукти, що не бажано споживати при захворюваннях, вказаних у персональному профілі, а жовтим – страви, що можуть викликати алергічну реакцію.

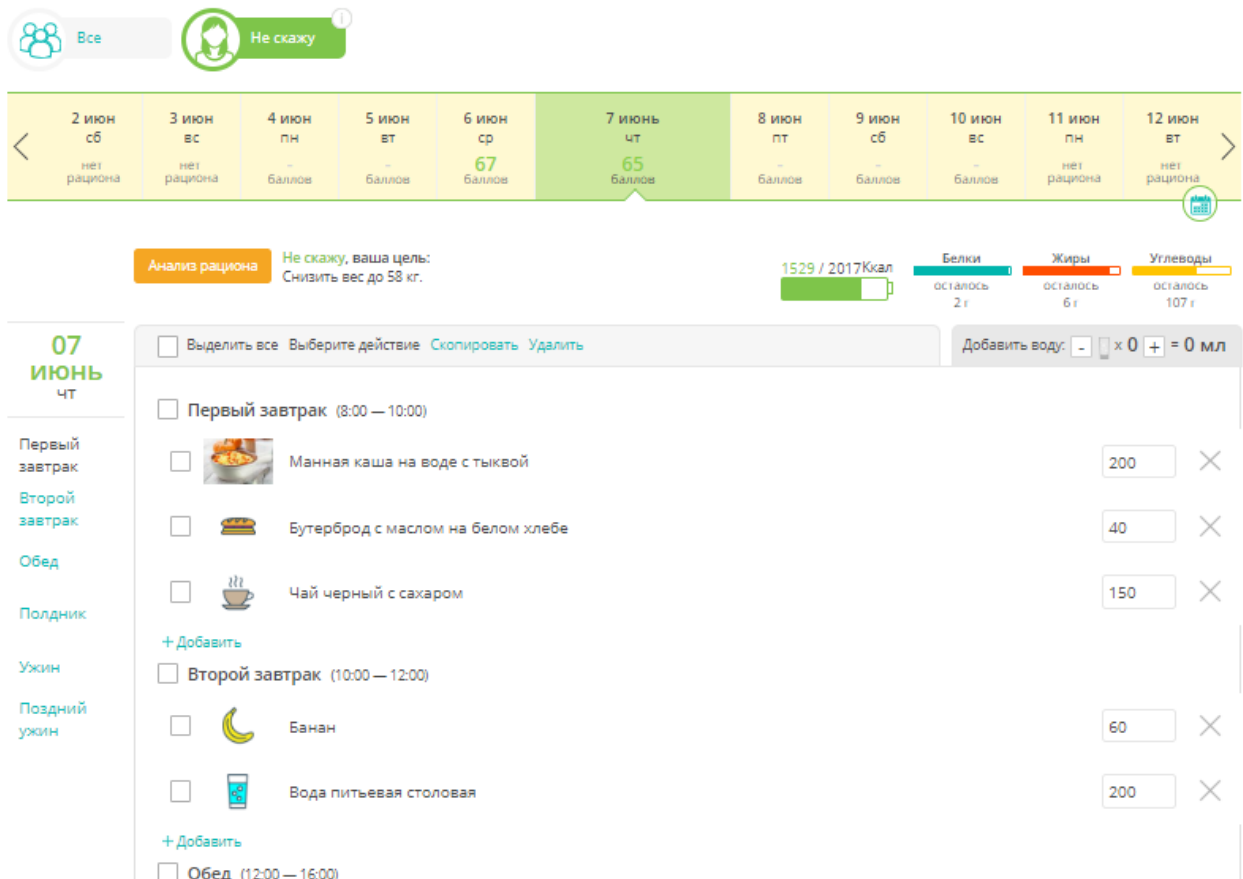


Рисунок 1.8 – «Щоденник харчування»

У користувача є можливість додати або видалити певні страви, скоригувати кількість грам у порції. Можна переглянути не тільки нинішній профіль, але й усі профілі, що додані у персональні дані.

Розділ «Щоденник активності» дозволяє встановити свій рівень активності та розрахувати кількість витрачених калорій за певний період часу, що вказується користувачем.

Витрачені калорії визначаються за допомогою тесту (рис.1.9). Він включає у себе питання, які по сенсу є коротким описом щоденної діяльності: різні види спорту та фізичних вправ, робота, домашні справи, сон, прибирання, прийоми їжі, дозвілля тощо. Напроти кожного пункту потрібно вказати, скільки годин у день уходить на певний вид заняття, таким чином, щоб у сумі вийшло рівно 24 години, ні більше, ні менше. Тест потрібно заповнити окремо для будніх днів та для вихідних.

Ваши будни

- Укажите, пожалуйста, какой у вас график работы
 - Пятидневная рабочая неделя
 - Шестидневная рабочая неделя
 - Сутки через трое
 - Сутки через двое
 - День через день
 - Все рабочие
- Выберите уровень физической нагрузки на вашей работе (используйте подсказки, чтобы точно определить к какому типу относится ваша работа)

1

Очень легкая физическая нагрузка: офисные работники, педагоги, руководители, секретари, ученые, инженерно-технические работники, IT-специалисты, водители легкового транспорта, врачи-терапевты, студенты и школьники.

2

3

4

5
- Сколько часов в день вы работаете?

ч 00 мин
- Сколько часов Вы спите?

ч 00 мин
- Сколько занимает дорога на работу?

ч 00 мин
- Продолжительность перерыва на обед?

Вы указали:
24 ч 0 мин
из 24

Рисунок 1.9 – Зразок тесту для визначення кількості витрачених калорій у день

Таким чином, проект «МІЙ ДІЄТОЛОГ» є досить якісним зразком помічника у здоровому харчуванні. Одними з найважливіших переваг є зручний, багатий на функціонал інтерфейс, можливість вказати захворювання та достовірна інформація щодо раціонів та їх переваг, що перевіряється досвідченими спеціалістами. Серед недоліків можна вказати, що пошук ведеться лише по окремих раціонах та самих параметрів пошуку не дуже багато – відсутні, наприклад, обмеження на складові раціонів, кількість грам бажаного БЖВ, середня коштовність продуктів, необхідних для приготування страв (і це при тому, що у ресурсу наявний власний інтернет-магазин).

1.2.2 MyFitnessPal (веб-версія)

MyFitnessPal – лічильник калорій від MyFitnessPal. Розробник завіряє, що його ресурс надає безкоштовний доступ до найбільшої бази даних продуктів – більше 3 мільйонів найменувань. Має мобільні версії для операційних систем Android та iOS, що синхронізуються автоматично з веб-версією [4].

Ресурс переведений на декілька мов, але української серед них нема. Проте є російська та англійська мови.

Наявні безкоштовна та платна версії додатку.

Безкоштовна версія пропонує наступний функціонал:

- встановлення денної мети за кількістю калорій, щоб полегшити досягнення мети по вазі;
- спеціальні цілі по БЖВ; цілі по макроелементах, які мають найбільше значення для користувача;
- якщо немає часу фіксувати споживані страви, можна просто записати кількість калорій.

Окрім цього, у користувача є можливість придбати платний варіант, що буде надавати наступні функції:

- зведення харчової цінності – можливість обрати вигляд, який найкраще допомагає досягати своїх цілей;
- аналіз страв – можливість дізнатися, у яких стравах найбільша кількість тих елементів, що цікавлять користувача;
- можливість експорту даних про прогрес, прийоми їжі, вправи у CSV-файли;
- мети по калоріях для різних страв – можливість знижувати денну мету у разі тренувань;
- різні мети на день – можливість встановлювати спеціальні цілі по калоріям та макроелементам на кожен день неділі;

- рівень вуглеводів, білків і жирів в стравах;
- можливість швидко додати вуглеводи, білки і жири;
- можливість встановити цілі по макроелементах в грамах або відсотках;
- пріоритетна служба підтримки;
- відсутність реклами.

Розглянемо безкоштовну версію ресурсу.

Під час реєстрації сайт пропонує ввести персональні дані, такі, як нинішня вага, бажана вага, зріст, стать, дата народження, країна, поштовий індекс, ім'я користувача, образ життя (сидячий, малорухомий, рухомий, дуже рухомий), кількість та тривалість тренувань, спосіб відображення інформації про енергію (кілокалорії або кілоджоулі) та мета стосовно ваги (підтримувати, набирати або ж скинути вагу). Є можливість обрати одиниці вимірювання ваги, зросту, пройденої відстані та об'єм випитої води.

Під час реєстрації (рис. 1.10), проте, вдалося потрапити на доволі дивну помилку. Якщо, маючи у одиницях виміру ваги кілограм, вказати бажану вагу меншою за норму, по думці системи, користувач, що знаходиться на російській версії сайту, побачить помилку, мало того що описану на англійській мові, так ще й одиниця вимірювання ваги указана у фунтах. Більше того, у помилці вказано, що ресурс чекає на дані, вказані в діапазоні від 111 до 149 фунтів (від 50,34 до 67,58 кілограм відповідно), проте, у вікні вказано 58 кілограм, що відповідає 127,868 фунтам, та, як наслідок, входить у діапазон.

Расскажите О Себе

Предоставленная вами информация будет использоваться для разработки вашего личного диетического и тренировочного профиля.

Изменить единицы измерения веса и роста (например, кг на фунты)

Текущий вес: 75 кг
 Рост: 185 см
 Целевой вес: 58 кг

The goal weight you've entered is classified as underweight. A normal weight range for your height is between 111 and 149 pounds. Please enter a goal weight of 111 pounds or greater.

Пол: Мужчина Женщина

Дата рождения: 11 июля 1996

Местоположение: Ukraine (non-Crimea)

Почтовый индекс:

Имя пользователя: Akira 4—30 знаков без пробелов

Опишите свой образ жизни:

- Сидячий: большую часть дня провожу сидя (банковский служащий, работник офиса и т. п.)
- Малоподвижный: большую часть дня провожу на ногах (учитель, продавец и т. п.)
- Подвижный: физические нагрузки большую часть дня (официант, курьер и т. п.)
- Очень подвижный: тяжелые физические нагрузки большую часть дня (спортсмен, плотник и т. п.)

Сколько раз в неделю вы планируете выполнять упражнения?

0 Тренировок в неделю

мин. на тренировку

Как вы хотите отслеживать затраченную энергию?

кДж ккал

Какова ваша цель?

Сбросить по 0,5 кг в неделю

Рисунок 1.10 – Реєстраційний лист з помилкою

Проте, якщо проігнорувати помилку та натиснути на «Продовжити», реєстрація пройде успішно. Опісля цього сайт підрахує рекомендовані цілі щодо харчування та фізичних вправ, та мету стосовно швидкості скидання ваги (рис.1.11).

Рекомендуемые Цели По Фитнесу И Питанию

Поздравляем! Ваш личный профиль питания и упражнений готов! Исходя из ваших ответов, мы подобрали вам соответствующие цели.

Цели По Питанию	Цель
«Чистые» потребленные калории* за день	1 340 ккал/день
Углеводов / день	168.0 g
Жиров / день	45.0 g
Белка / день	67.0 g

*Чистые потребленные калории = общее кол-во потребленных калорий - сожженные калории. Так что чем активнее вы проводите время, тем больше вы можете есть!

Фитнес-Цели	Цель
Сожжено калорий в неделю	0 ккал/нед.
Тренировок в неделю	0 тренировки
Минут на тренировку	0 мин.

Если вы будете следовать этому плану...

Прогнозируемая потеря веса 0,5 кг/нед.
Вам нужно сбросить 1,1 кг, 11 июля

Рисунок 1.11 – Рекомендації щодо скидування ваги

Важливо зазначити, що цей ресурс при однакових наданих персональних даних пропонує добову норму калорій у 1340 ккал/день, коли як проект «МІЙ ДІЄТОЛОГ», наприклад, оцінив її у 1702 ккал/день.

У проекті MyFitnessPal присутній форум, який, треба зазначити, достатньо активний. Сайт також пропонує додати друзів з метою худнути разом та обіцяє зберігати чутливі персональні дані у таємниці від них.

На головній сторінці сайту підвкладці «Головна» (рис.1.12) можна побачити добову мету у калоріях, прогрес її досягнення, можна додати вправу або страву, або написати щось у власній стрічці новин.

У підвкладці «Цілі» можна відредагувати власні добові норми БЖВ у процентному відношенні, калорії, співвіднести у процентному відношенні сніданок, полуденок, обід та вечерю, мету у плані тренувань та бажану потребу у макроелементах (насичені, поліненасичені, мононенасичені, транс-жири, холестерин, натрій, калій, клітковина, цукор, вітаміни А, С, кальцій та залізо). На жаль, більшість з описаного у підвкладці функціоналу доступна лише у платній версії.

Главная Цели Ввод данных Почта Профиль Мой блог Друзья Настройки

Итоги дня 1 день подряд

Нет фотографии
Загрузить фото

Осталось калорий Изменить

1340

Добавить Упражнение
Добавить Блюдо

0 кг СБРОСИЛ(А)

1340 ЦЕЛЬ 0 БЛЮДА - 0 УПРАЖНЕНИЯ = 0 ЧИСТЫЕ

Не забудьте подтвердить свой электронный адрес.
Мы отправили электронное письмо по адресу:
nataliia.manuilova@nure.ua

Отправить Заново
Или смените свой адрес email

Последние темы на форуме
Посмотреть все

- Strength workouts that keep heartrate up?
- How did you meet your SO?
- Daily Treadmill - total 40 km challenge
- What nobody tells you about losing weight
- Back at it with PCOS

Новостная лента

Поделитесь своими мыслями... Поделиться

Рисунок 1.12 – Вкладка «Головна»

У підвкладці «Введення даних» можна уточнити поточні дані (кількість кілограм, обхват шиї, талії та стегон). Це потрібно для моніторингу змін у динаміці.

Після натискання кнопки «Додати вправу» з вкладки «Головна» або на вкладку «Вправи» відкриється інша сторінка, де буде продемонстровано вже виконані вправи за день, кардіо- або силові вправи, їх вид, буде зазначена узагальнена інформація стосовно результатів виконання. До тренування можна додати якісь свої нотатки.

Якщо натиснути на «Додати вправу» напроти кардіо- або силової вправи, відкриється вікно з пошуком вправ. Після вводу назви та натискання кнопки «Пошук» програма дасть усі релевантні до уведеного запиту результати. Якщо серед результатів нема потрібного, є можливість додати власну вправу.

У підвкладці «База даних» вкладки «Вправи» можна знайти усі відомі додатку вправи. Можна здійснити пошук по запиту користувача, або скористатися готовим drop-down меню з метою обрати потрібну вправу. На жаль, вкладка лише дає можливість переглянути, скільки калорій «згорить» при зайнятті тим чи іншим видом тренування, але можливості звідси додати його у користувача немає.

Важливим недоліком є те, що відсутня будь-яка модерація найменувань вправ. Кожен користувач у змозі додати свої вправи, що за своєю ефективністю у плані спалених калорій не відповідають реальності.

У підвкладці «Мої вправи» (рис.1.13) користувач має змогу додати до своєї власної бази даних свої вправи або знайти щось з уже колись введеного ним. Щоб створити нову вправу, потрібно натиснути на кнопку «Створити вправу», ввести її опис, тип (кардіо, силова), час, необхідний для вправи та скільки калорій буде знищено. Знову ж таки, якщо скористатися пошуком та відкрити свою власну вправу, у користувача немає можливості додати її до виконаних прямо звідси.

Дневник упражнений База данных Мои упражнения Настройки

Дневник упражнений за: среда, 6 июня 2018 г.

Кардиоупражнения

	Мин.	Сожжено Калорий
Бег (бег), 10,7 км (5,6 мин на км)	5	69

[Добавить упражнение](#) | [Горячие ссылки](#)

Всего за день / Цель	5 / 0	69 / 0
Всего за неделю / Цель	5 / 0	69 / 0

Силовые Упражнения

	Подх.	Повторов За Подход	Вес/Подход
--	-------	--------------------	------------

[Добавить упражнение](#) | [Горячие ссылки](#)

Заметки к сегодняшней тренировке [Редактировать](#)

Рисунок 1.13 – Вкладка «Щоденник вправ»

Якщо відкрити посилання «Додати страву» з вкладки «Головна» або ж натиснути на вкладку «Їжа», то можна буде спостерігати першу підвкладку «Щоденник харчування» (рис.1.14). Користувач має можливість додати страву та побачити результат дотримання певної стратегії харчування за день, або додати якусь кількість випитої води. Є можливість спостерігати результати у кількості споживаних калорій, БЖВ, натрію та цукру. Знову ж таки, є можливість додати нотатки стосовно харчування. Після натискання кнопки «Завершити запис» буде сформовано кінцевий звіт за день.

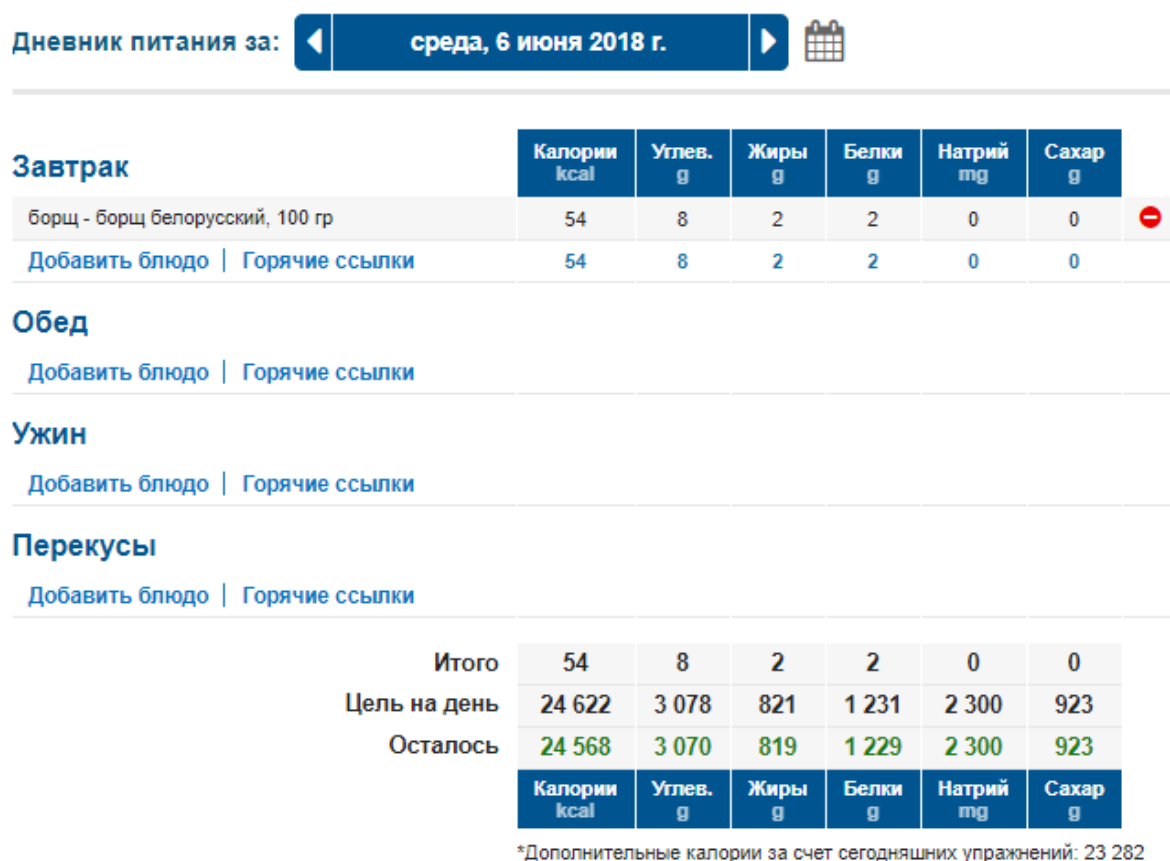


Рисунок 1.14 – Вкладка «Щоденник харчування»

По натисканню кнопки «Додати страву» або у підвкладці «База даних» знаходиться база даних страв. Страву можна знаходити за ключовими словами за допомогою пошуку (рис.1.15). Усі дані модеруються самими користувачами, і всі мають можливість редагувати дані про ту чи іншу

страву, якщо вважають дані недостовірними. За ступінь «достовірності» відповідає «народний» критерій – кількість людей, що підтвердили правдивість введених даних. Можна створити власну страву, давши їй назву та параметри кількості компонентів.

Важливим недоліком є те, що відсутня будь-яка модерація найменувань їжі. Більш того, немає ніяких описів рецептури, що позбавляє користувача будь-якої можливості зробити навіть припущення, яке саме найменування буде вірним у його випадку. Не є рідкими випадки, коли одне і те ж найменування зустрічається багато разів, і кожне з них містить різні параметри. Це, насправді, позбавляє будь-якого сенсу користуватися цією функцією, бо дані з великою ймовірністю недостовірні, що вводить користувача в оману.

Таблица калорийности в калориях и пищевая ценность продуктов

Поиск наша пищевая база данных по названию:

Найденные блюда:

Борщ есть борщ	Борщ, 100 гр, 57 ккал
холодный борщ	Холодный борщ, 100 г, 35 ккал
борщ белорусский	Борщ, 100 гр, 54 ккал
Борщ на курином бульоне	Борщ, 100 мл, 28 ккал
(обычный)	Борщ, 100 г, 82 ккал

Пищевая ценность

Порции:

ккал	0	Натрий	0 мг
Всего жиров	0 г	Калий	0 мг
Насыщенные	0 г	Всего углеводов	0 г
Полиненасыщенные	0 г	Пищевая клетчатка	0 г
Мононенасыщенные	0 г	Сахара	0 г
Транс-жиры	0 г	Белки	0 г
Холестерин	0 мг		
Витамин А	0%	Кальций	0%
Витамин С	0%	Железо	0%

*Суточная норма в процентах основана на диете на 2000 ккал. В зависимости от ваших потребностей в калориях, ваша суточная норма может оказаться выше или ниже указанной.

Рисунок 1.15 – Результаты пошуку за словом «борщ»

У підвкладці «Власні страви» користувач має змогу додати свою страву, по аналогії з власними вправами. Можна також шукати найменування серед своїх.

У підвкладці «Рецепти» користувач має можливість створити власний рецепт, указавши інгредієнти та необхідну їх кількість. Здавалось би, у цієї функції є можливість автоматично підрахувати кількість компонентів та калорій автоматично, але проблема у тому, що назви продуктів, що беруться з бази, також ніким не модеруються, що зводить якість інформації до нуля.

У вкладці «Графіки» користувач має можливість спостерігати за прогресом у динаміці за допомогою діаграм. Можна обирати будь-який власний параметр або кількість споживаних калорій/макроелементів. Графіки можна розширяти по днях або експортувати.

Таким чином, MyFitnessPal, не глядячи на наявність платної частини, демонструє себе куди гірше, ніж проект «МІЙ ДІЄТОЛОГ», наприклад. Більшість наданої ресурсом інформації про калорійність, наявність макроелементів, БЖВ, вартість вправ є попросту недостовірною, бо вводиться хаотично відкритим суспільством ресурсу без попередньої модерації. Відсутність складових рецепту у базі даних найменувань страв часто робить неможливим ідентифікувати, про яку страву йде мова (як приклад, можна спостерігати результати пошуку на рисунку 1.15, де під словом «борщ» вказана принципово різна кількість калорій та немає можливості перевірити, про який саме варіант борщу йде мова). «Народна» оцінка себе ніяк не виправдовує – відсутність експертної оцінки зводить на «ні» навіть сто згодних з даними чоловік, якщо у них немає ніякого досвіду у подібних речах. Як наслідок, встає питання, чи має взагалі сенс використання подібного додатку для здорового харчування та підтримання себе у формі. Як значну перевагу, можна було би зазначити можливість вводити свої страви та наявність вельми специфічних макроелементів, але корисність останньої функції знову ставиться під сумнів через відсутність достовірності інформації у цілому.

1.3 Постановка задачі

Планується розробити систему з метою позбутися від усіх вищеописаних недоліків. У користувача повинен бути додаток, за допомогою якого він зможе:

- відшукати рецепти, ґрунтуючись на бажаній кількості калорій, параметрах БЖВ, бажаних та небажаних продуктах у складі, середній ціні складових страви;
- мати можливість скласти свій профіль, в якому була б можливість вказати дозволені і заборонені продукти, свої параметри (вага, зріст, обсяги різних частин тіла), отримувати інформацію про спожиті страви, динаміці змін тих чи інших параметрах, узагальнених в таблиці\графіки;
- отримувати поради по харчуванню та профілі за замовчуванням для спортивного харчування, визначених дієт, пов'язаних з бажанням скинути вагу або захворюваннями шлунково-кишкового тракту;
- оновлювати інформацію про свої дані, щоб мати можливість спостерігати прогрес зміни обсягу частин тіла\ваги\кількості споживаних калорій в динаміці за допомогою графіків;
- відправляти відгуки про роботу системи через спеціальний інтерфейс з пропозиціями своїх рецептів, або ж просто ідеї щодо поліпшення сервісу;
- окрему увагу варто приділити безпеці, бо користувацькі персональні дані, дані про геолокацію для складання політики цін на продукти, дані про дієти, здоров'я чутливі.

Також планується додати функціонал для адміністраторів програми:

- можливість модерування наявної бази даних рецептів;

- можливість скористатися помічником-граббером, який буде, користуючись глобальним пошуком по Інтернету, пропонувати адміністратору можливі варіанти страв, які можна було б додати в базу даних програми.

Дизайн додатка повинен бути адаптивним та мінімалістичним, щоб його було зручно використовувати на мобільному пристрої.

З аналітичних задач, вирішення яких необхідне для реалізації поставлених перед нами завдань, варто виділити:

- визначити кількість необхідних грам продуктів-складових страви, враховуючи пропорції і обмеження за кількістю калорій і БЖВ;
- розрахувати ціну того чи іншого продукту, страви в цілому з огляду на регіон користувача, валюту країни, в якій він знаходиться, його місце розташування і, в ідеалі, фінансове становище клієнта;
- підрахунок змін параметрів людини і його харчування в динаміці, побудова графіків, ймовірності скинути певну кількість кілограм при заданих умовах харчування і фізичної активності;
- побудова грамотної архітектури бази даних з метою простого сортування даних за вказаними параметрами.

Узагальнюючи все вищесказане, треба розробити помічника, здатного за короткий час запропонувати рецепт, виходячи з переваг, фінансових можливостей і особливостей харчування будь-якої людини.

Мета роботи – розробка програмної системи регулювання харчування.

Для досягнення мети поставлено наступні задачі:

- провести аналіз предметної області;
- визначити особливості регулювання харчування;
- розробити моделі регулювання та оптимізації харчування;

- обрати технологію та розробити модулі програмного додатку регулювання харчування;
- перевірити роботу системи з використанням тестового матеріалу;
- зробити висновки.

2 МЕТОДИ РЕГУЛЮВАННЯ ХАРЧУВАННЯ

2.1 Математичні моделі регулювання та оптимізації харчування

Для оперативного аналізу фактичного харчування та констатації факту фактичних порушень у раціонах широко застосовуються різноманітні методи. Найбільш актуальним у сучасних умовах є використання комп'ютеризованих анкет та програм.

Вирішення задачі з оптимізації раціонів харчування різноманітних груп населення не може бути успішним без розробки критерію, що дозволяє координувати їх збалансованість відповідно до нормативних значень, рівнозначно враховуючи при цьому енергетичну цінність та всі параметри хімічного характеру кожного продукту.

У харчовій промисловості відомі методи застосування лінійних та дрібно-лінійних критеріїв раціональності, що використовуються при моделюванні багатокomпонентних білковомістячих рецептур, зокрема м'ясомолочних продуктів. Стосовно задачі, що розглядається, такі критерії призводять до віддалення від оптимального її рішення. У зв'язку з цим було запропоновано оптимізувати амінокислотний склад харчових продуктів, використовуючи апарат математичного програмування, що дозволяє визначити ступінь приближення амінокислотного складу до еталону FAO/WHO шляхом введення критерію відносної геометричної відстані.

Значення кожного еталонного параметру у розробленому алгоритмі оптимізації представлено у вигляді:

$$b_i (1 \leq i \leq p), \quad (2.1)$$

де b_i – нормативна величина i -го показника, p – кількість показників збалансованості продукту.

Задача з визначення кількості кожного продукту $x_j (1 \leq j \leq n)$, що входить у склад раціону, що оптимізується, вирішувалась таким чином, що значення усіх параметрів оптимізованого набору продуктів c_i по абсолютній величині та співвідношенню максимально наближались до відповідних величин b_i . Для цього розроблений інтегральний критерій якості (цільова функція), що демонструє ступінь збалансованості за допомогою управляючого вектору $\vec{X} = \{x_j\}$. Даний критерій оптимізації рівнозначно враховує значення енергетичної цінності та всіх параметрів хімічного складу харчових продуктів, по яким оцінюється збалансованість раціонів харчування.

Масова доля i -го показника у j -ому продукті, що входить у склад раціону, позначена a_{ij} . Ці числа зведені у матрицю $A = (a_{ij})$ розміром $p \times n$ та отриманий вектор хімічного складу оптимального раціону по формулі

$$\vec{C} = A \vec{x} = \left\{ \sum_{j=1}^n a_{ij} x_j \right\}. \quad (2.2)$$

Виходячи з фізіологічних норм харчування, на направляючий вектор \vec{X} можливе накладення обмежень:

$$0 \leq x_j \leq k_j, \quad (2.3)$$

де константи k_j відомі та задаються.

Враховуючи якісні відмінності готових страв та ринкових продуктів, що їх складають, найбільш доцільно вектор \vec{X} вважати таким, що відноситься до розрахунку готових продуктів.

Кількості ринкових продуктів $y_e (1 \leq e \leq m)$, що складають вектор $\vec{Y} = \{y_e\}$, можна розрахувати за формулою:

$$\vec{Y} = D\vec{X} = \left\{ \sum_{j=1}^n d_{ej}x_j \right\}, \quad (2.4)$$

де d_{ei} – масові долі е-го ринкового продукту у j-ому готовому продукті, що складає матрицю $D = (d_{ei})$.

З урахуванням квадрату модулю різності векторів хімічних сполук оптимізуемого раціону \vec{C} та еталону \vec{B} :

$$K = |\vec{C} - \vec{B}|^2 = \sum_{i=1}^p (c_i - b_i)^2, \quad (2.5)$$

запропонований наступний критерій оптимізації раціону по хімічному складу:

$$F = 100 - 100 \sqrt{\frac{1}{p} \sum_{i=1}^p \left(\frac{c_i}{b_i} - 1 \right)^2} \quad (2.6)$$

При повному збігу \vec{C} та \vec{B} (і тільки в цьому випадку) величина F досягає свого глобального максимуму 100%. Чим ближче F до 100%, тим більше збалансований раціон за своїм хімічним складом.

Можна показати, що екстремуми (2.5) та (2.6) досягаються шляхом одних і тих же оптимальних рішень \vec{X} . Таким чином, задача оптимізації раціонів харчування (2.5)-(2.6) зводиться до класу задач опуклого квадратичного програмування з цільовою функцією (2.5), що підлягає

мінімізації. Поряд з критерієм оптимізації (2.6), який повинен бути максимізованим, визначну роль у раціональному харчуванні грає енергетична вартість раціону Q .

Її можна розрахувати наступним чином:

$$Q = \vec{q}\vec{X} = \sum_{j=1}^n q_j x_j, \quad (2.7)$$

де q_j – калорійність одиниці маси j -го продукту, що використовується у раціоні; вектор $\vec{q} = \{q_j\}$. Тоді цільовою функцією у задачі (2.5)-(2.6) повинна виступати

$$K = |\vec{C} - \vec{B}| - Q = \sum_{i=1}^p (c_i - b_i)^2 - \sum_{j=1}^n q_j x_j, \quad (2.8)$$

яка також відноситься до класу опуклого квадратичного програмування.

Аналогічно до задачі (2.8) можна врахувати необхідність мінімізації вартості раціонів харчування та інші фактори, що мають важливе значення для конкретного користувача.

Аналіз вищеописаного алгоритму демонструє, що критерії якості, що застосовуються у ньому, не володіють наочними засобами оцінки власних показників збалансованості окремих компонентів хімічного складу, що є основним принципом, що забезпечує зручність комп'ютерного моделювання нових видів полікомпонентних рецептурних сумішей. Окрім цього, практичне дослідження показало, що:

- оціночна шкала, заснована на значеннях критерію (2.6), дає доволі розмиті відомості про ступінь приналежності харчового продукту до якісних груп «погана», «задовільна», «добра» або «відмінна» збалансованість;

- автори [5] не зазначають методи математичного програмування, на основі яких виконується оптимізація цільової функції у поставленій задачі, не приводять алгоритм, що їх реалізує та засоби програмної реалізації.

Таким чином, для представленого алгоритму невідома точність результатів, структура, продуктивність, програмна оболонка, а також ряд інших показників раціональності його застосування [5].

2.2 Аналіз функціональних вимог до системи регулювання харчування

2.2.1 Аналіз системи зберігання та обробки рецептів

Так як предметна область тісно пов'язана зі збереженням достатньо великої кількості даних, необхідно виявити певні сутності та встановити зв'язок між ними.

У рамках виконання роботи основний опір було зроблено на створення інформаційної системи-помічника у здоровому харчуванні, тому, очевидно, що одними з найважливіших сутностей будуть «страва», «рецепт» та «харчовий продукт». У кожній з цих сутностей наявні свої властивості. Розглянемо кожен з них окремо.

Харчовий продукт (далі будемо дотримуватися спрощеної назви «продукт») є продуктом у натуральному або переробленому вигляді, який людина здатна вживати у їжу. У нього є багато показників, проте упинимося на базових: назві, параметрах БЖВ, кількості грам та енергетичної цінності. В основному продукт гратиме роль складової інгредієнтів рецепту, але у певних випадках може бути споживаний і у своєму вихідному стані.

Рецепт (у даному сенсі мається на увазі саме кулінарний рецепт) є посібником з приготування кулінарного виробу. Він має інформацію про необхідні харчові продукти, їх пропорції та інструкції стосовно змішування

та обробки. Кулінарні рецепти описують механічну та теплову обробку інгредієнтів, засоби сервіровки готових виробів.

У нашій термінології рецепт має назву, орієнтовний час приготування, список необхідних складових та алгоритм приготування.

Страва є результатом слідування кулінарного рецепту, який, по суті, є тим самим харчовим продуктом, але більш складним за засобом приготування. У неї наявні параметри БЖВ (кількість, пропорції), загальна вага, загальна енергетична цінність.

2.2.2 Структури даних

Для зберігання даних про рецепти використовується документо-орієнтована база даних MongoDB.

MongoDB – це безкоштовна та відкрита версія платформи, орієнтована на документи. Класифікована як програма бази даних NoSQL, MongoDB використовує документи, подібні до JSON, зі схемами. MongoDB розроблена компанією MongoDB Inc., і видається під комбінацію загальної публічної ліцензії GNU Affero та ліцензії Apache [6].

MongoDB була обрана для зберігання рецептів, так як є досить простою у використанні документо-орієнтованою базою даних, яка зберігає дані у вигляді документів JSON. Для рецептів подібний опис є найкращим вибором, бо дані з неї можна витягувати і відправляти клієнту навіть без попередньої обробки. Більш того, з функціональної точки зору вона менша, ніж у реляційних баз даних (але більша достатньо великої кількості noSQL баз даних, наприклад, Redis, Neo4j та т.п.), але швидкість виконання запитів більша за рахунок відмови від моделі збереження даних пов'язаних між собою таблиць.

З мінусів даного підходу можна виділити відсутність зв'язності, і, як наслідок, відсутність таких переваг реляційних баз даних, як цілісність

даних, наявність повторюваної інформації (насправді, у нас є можливість створити, наприклад, ще одну колекцію та посилатися на дані з неї, але це негативно відобразиться на швидкості пошуку рецептів), та відсутність транзакцій у MongoDB (проте, розробники обіцяли ввести їх у найближчому майбутньому), що може призвести до таких феноменів, як «брудне» або фантомне читання, втрачене оновлення, зчитування, що не повторюється. Однак же, виключивши необхідність розподіляти дані по великій кількості таблиць, встановлюючи між ними зв'язок, первинні і зовнішні ключі, сильно спрощується робота з нею, що є більш істотною перевагою, ніж вищеописані недоліки.

Маємо єдину колекцію Recipes, що містить в собі масив рецептів. До речі кажучи, ще однією перевагою даного вибору є швидкість пошуку, що прямо пропорційна довжині документа і залежить від положення необхідної інформації в ньому. Колекція представляє собою документ JSON.

Опишемо усю інформацію, що знаходиться у колекції.

Одна одиниця інформації є сутністю «рецепт». Кожен рецепт має наступні властивості:

- `_id` – унікальний номер-ідентифікатор рецепту;
- `sourceLink` – посилання на сторінку, звідки був взятий рецепт;
- `recipeName` – назва рецепту;
- `imageLink` – посилання на зображення готової страви;
- `totalWeight` – загальна вага готової страви у грамах;
- `totalKcal` – загальна енергетична цінність готової страви у кілокалоріях;
- `totalProteins` – загальна кількість білків готової страви у грамах;
- `totalFats` – загальна кількість жирів готової страви у грамах;
- `totalCarbohydrates` – загальна кількість вуглеводів готової страви у грамах;
- `weightPerPortion` – вага однієї порції страви у грамах;

- kcalPerPortion – енергетична цінність однієї порції страви у кілокалоріях;
- proteinsPerPortion – кількість білків у одній порції, у грамах;
- fatsPerPortion – кількість жирів у одній порції, у грамах;
- carbohydratesPerPortion – кількість вуглеводів у одній порції, у грамах;
- kcalPer100g – кількість кілокалорій у 100 грамах;
- proteinsPer100g – кількість білків у 100 грамах;
- fatsPer100g – кількість жирів у 100 грамах;
- carbohydratesPer100g – кількість вуглеводів у 100 грамах;
- proteinsPercent – процентне співвідношення білків у страві;
- fatsPercent – процентне співвідношення жирів у страві;
- carbohydratesPercent – процентне співвідношення вуглеводів у страві;
- cookTime – орієнтовний час приготування страви;
- advancedCookTime – додатковий час, необхідний для приготування їжі (наприклад, час, що необхідно очікувати, поки торт настоїться, та крем «дасть» йому свій смак);
- preparation – алгоритм приготування страви;
- products – список продуктів.

Окрім цього, кожен продукт у списку має наступні властивості:

- productName – назва продукту;
- weight – вага продукту у грамах;
- kcal – енергетична цінність продукту у кілокалоріях;
- proteins – кількість білків у грамах;
- fats – кількість жирів у грамах;
- carbohydrates – кількість вуглеводів у грамах.

Під час роботи була використана IDE MongoDB Compass Community для створення колекції, дизайну архітектури бази даних, перегляду її

поточного стану, виконання таких дій, як видалення, редагування або додавання якихось даних.

Зовнішній вигляд JSON-документу можна спостерігати на рисунку 2.1.

```

_id: 119
sourceLink: "http://daily-menu.ru/dailymenu/recipes/view/119"
recipeName: "Блины классические тонкие"
imageLink: "http://daily-menu.ru/public/modules/dailymenu/dailymenurecipes/119/ec1..."
totalWeight: 586
totalKcal: 1142
totalProteins: 34
totalFats: 36
totalCarbohydrates: 173
weightPerPortion: 147
kcalPerPortion: 286
proteinsPerPortion: 8.4
fatsPerPortion: 9
carbohydratesPerPortion: 43.4
kcalPer100g: 198
proteinsPer100g: 5.2
fatsPer100g: 5.1
carbohydratesPer100g: 32
proteinsPercent: 12
fatsPercent: 28
carbohydratesPercent: 60
cookTime: "40 мин."
advancedCookTime: ""
preparation: "1. Молоко немного подогреть, смешать с кефиром 1:1.
2. Добавить яйцо,..."
products: Array
  0: Object
    productName: "Яйцо куриное"
    weight: 50
    kcal: 79
    proteins: 6
    fats: 6
    carbohydrates: 0

```

Рисунок 2.1 – Приклад вигляду документу JSON, у якому описаний рецепт

2.3 Розробка архітектури веб-додатку. Опис розроблених на даний момент модулів

На даний момент розробка багатомодульного додатку ведеться за допомогою інструменту Gradle, що дозволяє з мінімальними витратами за часом підключити всі зовнішні залежності до всього проекту або окремо до кожного модуля, описати залежність між власними сервісами і мікросервісами [7].

Розроблено 3 модуля: `models`, `recipe_grabber` і `web`.

Модуль `models` містить в собі моделі для роботи з базою даних, так звані DTO-об'єкти, класи сутностей, `bean`, і так далі. Призначення модулю полягає у зменшенні повторюваності коду – він не є самостійним та не несе ніякого безпосереднього функціоналу. Його можна імпортувати до інших модулів, використовуючи властивості Gradle, та користуватись класами-моделями.

Модуль `recipe_grabber` займається забором рецептів з сайту `daily-menu.com`. Є звичайним консольним додатком, призначеним для адміністратора. Його єдине призначення – автоматизоване наповнення бази даних рецептами, які пізніше будуть відображатися користувачеві через модуль `web`. Усе, що користувачеві потрібно, вказати діапазон ID рецептів через командну стрічку.

Модуль `web` містить в собі реалізацію сервісів для роботи з базою даних, контролерів для організації RESTful архітектури сайту і самі сторінки. Є самим великим за кількістю написаного коду модулем. З точки зору користувача відповідає за сторінки сайту, коректну роботу з компонентами веб-додатку та базою даних.

На рисунку 2.2 зображена діаграма компонентів. Службові бібліотеки (призначені для юніт-тестування або логування) не включені до діаграми.

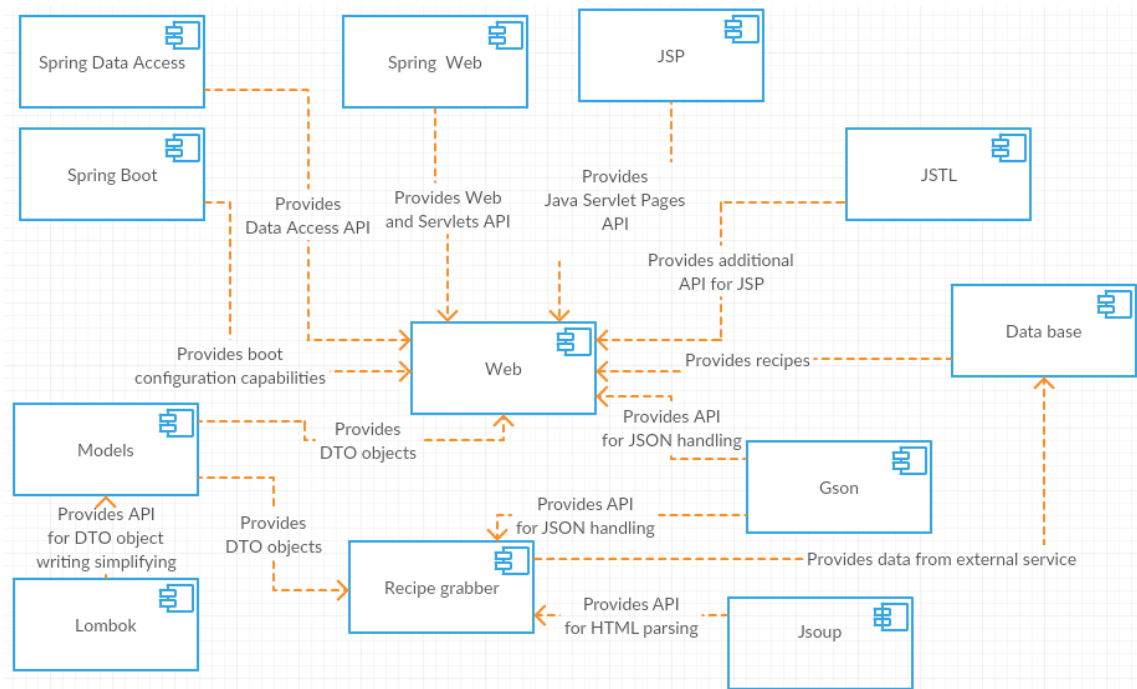


Рисунок 2.2 – Діаграма компонентів

2.4 Реалізація функціоналу

2.4.1 Опис класів та функцій

На даний момент існує часткова реалізація описаних в розділі 1 очікуваних функцій веб-додатку.

У модулі `models` є всього два класи, так звані моделі, `ProductDto` і `RecipeDto`. Їх призначення – утримання в собі результатів запитів або ж підготовка їх для, наприклад, додавання в таблицю. Не несуть у собі ніякої бізнес-логіки, мають лише поля. Анотація `@Data`, імпортована з бібліотеки `Lombok`, дозволяє автоматично генерувати геттери, сеттери, стандартні методи `equals`, `hashCode` та `toString` без участі розробника. Таким чином, коду та рутинної праці стає менше [8].

`RecipeDto` – об'єкт, здатний зберігати у собі сутність рецепту. Поля аналогічні представленню рецепту у базі даних (розділ 2.2.2).

ProductDto – об’єкт, здатний зберігати у собі сутність харчового продукту. Поля аналогічні представленню продукту у базі даних (розділ 2.2.2).

У модулі `recipe_grabber` присутні головний клас `Grabber` і сервіс для конвертації вмісту сторінки сайту `DailyMenuRecipeDocumentParser`.

Клас `Grabber` є головним, саме з нього починається робота модулю після запуску `.jar`-файлу.

У стандартному методі `main`, що приймає у себе масив аргументів (у нашому випадку очікуємо два цілих числа – перший та останній ID рецепту), відбувається виклик методу `getRecipe` класу `DailyMenuRecipeDocumentParser` для кожного ID рецепту, що попадає у діапазон зазначених чисел. Після того, як алгоритм отримає модель `RecipeDto`, вона буде сконвертована у JSON-об’єкт та відправлена на сервер бази даних, де буде додана у колекцію `Recipes`.

Приватний статичний метод `convert`, що приймає у себе об’єкт `RecipeDto` та повертає `DBObject`, власне, займається конвертацією типів даних для того, щоб дані можна було передати у базу даних через JDBC-драйвер.

Клас `DailyMenuRecipeDocumentParser` шукає необхідні частини рецепта і формує з них об’єкт класу вищеописаних моделей для їх подальшої конвертації в JSON. Для парсингу HTML використовується бібліотека `Jsoup`, а для конвертації в JSON – `Gson`.

Метод `getRecipe` має модифікатор доступу `package-private`, є статичним, приймає ID рецепту та повертає об’єкт класу `RecipeDto`. По факту, знаходить сторінку у Інтернеті, бере її HTML-код, знаходить усі необхідні компоненти, конвертує інформацію у потрібні формати, заносить дані до моделі та повертає її.

Приватний статичний метод `parseVariable` приймає об’єкт класу `Element` бібліотеки `Jsoup` та повертає `List<String>`. Він займається

конвертуванням елементу, що містить усі інгредієнти рецепту та розділяє його на список з інформацією про складові рецепту, що й повертає.

У модулі web є папка webapp, що містить сторінки для відображення кінцевому користувачеві, конфігурації, необхідні для коректної роботи сервісів та DAO, пакети web і utils. У свою чергу пакет web розподіляється на пакети controller, dao та service.

Пакет dao містить засоби для роботи з самою базою даних. CRUD-дії забезпечує інтерфейс RecipeDao і його реалізація RecipeDaoImpl. Для роботи використовується вищезгаданий Gson.

Метод addRecipe приймає об'єкт класу RecipeDto. Додає рецепт до бази даних.

Метод updateRecipe приймає об'єкт класу RecipeDto. Оновлює дані про вже існуючий рецепт.

Метод removeRecipe приймає ціле число. Видаляє рецепт з бази даних по його ID.

Метод getRecipeById приймає ціле число та повертає об'єкт класу RecipeDto. Шукає рецепт з бази даних по його ID.

Метод listRecipes повертає список усіх страв, що знаходяться у базі даних на даний момент.

Пакет service містить у собі інтерфейс RecipeService і його імплементацію RecipeServiceImpl для організації взаємозв'язку між контролером і DAO. Методи за описом нічим не відрізняються у DAO-класів, тільки DAO беруть дані безпосередньо з бази даних, тоді як сервісний клас використовує об'єкт класу DAO для користування його функціями.

Пакет controller містить в собі контролер RecipeController для організації REST запитів сайту до сервера, і в цьому досить сильно допомагає Spring WEB, – опис сервлетів витрачає набагато менше зусиль, залишаючи запас для менш рутинної роботи.

Контролер `listRecipes` реагує на GET-запит `/recipes`, додає до відповіді усі рецепти з бази даних та пересилає користувача на сторінку JSP, що знаходиться за адресою `/recipes`.

Контролер `addRecipe` реагує на POST-запит `/recipes/add`, приймає у себе модель `RecipeDto`, та викликає сервіс для того, щоб додати модель або оновити її у базі даних, після чого робить перенаправлення на JSP, що знаходиться за адресою `/recipes`. Рішенням, буде рецепт доданий або оновлений, робиться, виходячи з того, чи є ID рецепта у моделі або ні.

Контролер `deleteRecipe` реагує на GET-запит `/recipes/remove/{id рецепту}`, приймає у себе ціле число та викликає сервіс для того, щоб видалити рецепт з бази даних, після чого робить перенаправлення на JSP, що знаходиться за адресою `/recipes`.

Контролер `recipeData` реагує на GET-запит `/recipedata/{id рецепту}`, приймає у себе модель `RecipeDto` та викликає сервіс для того, щоб узяти потрібний рецепт, додає його до відповіді контролеру, після чого робить перенаправлення на JSP, що знаходиться за адресою `/recipedata`.

Контролер `downloadRecipeData` реагує на GET-запит `/recipedata/{id рецепту}/download` та викликає метод `saveToFile` класу `SaveRecipeToFileUtil` для того, щоб зберегти сторінку на комп'ютер користувача, після чого робить перенаправлення на JSP, що знаходиться за адресою `/recipedata/{id рецепту}`.

У пакеті `utils` міститься єдиний клас `SaveRecipeToFileUtil`, що займається збереженням даних рецептів в Word за допомогою Java COM Bridge. Він містить єдиний публічний, статичний метод `saveToFile`, що приймає у себе модель `RecipeDto`, який зберігає дані про рецепт у вигляді файлу на комп'ютері користувача.

Таким чином, веб-додаток має ознаки архітектури MVC.

Повну картину можна спостерігати за допомогою діаграми класів, що зображена на рисунках 2.3 та 2.4.

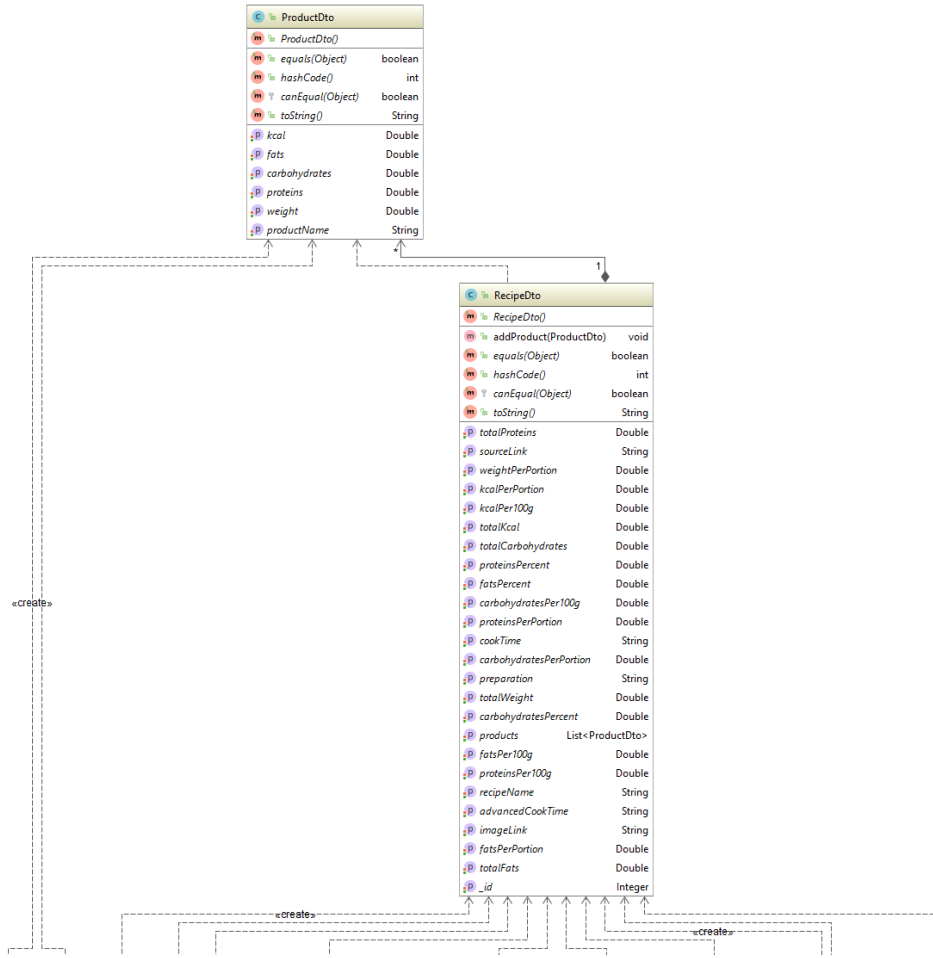
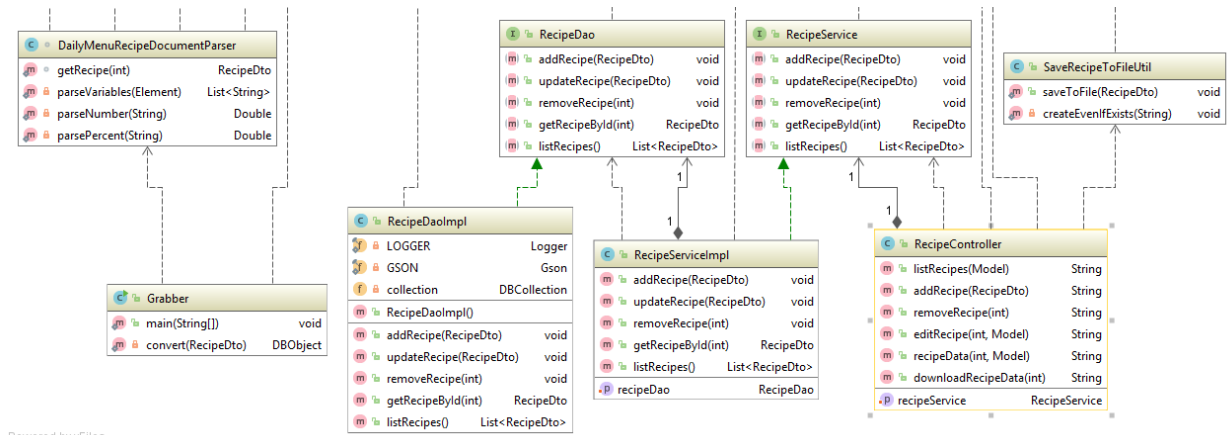


Рисунок 2.3 – Діаграма класів (верхня частина)



Powered by yFiles

Рисунок 2.4 – Діаграма класів (нижня частина)

2.4.2 Методи обробки даних

Доволі значна частина роботи так чи інакше пов'язана з обробкою даних, що отримуються з різних місць, перетворюємо їх у щось інше та передаємо далі.

У проекті присутній модуль `recipe_grabber`, що займається пошуком рецептів з необхідним ID на сайті `daily-menu.ru` та конвертацією інформації про страву з HTML сторінки спочатку у клас-модель `RecipeDto`, а після – модель `RecipeDto` у документ JSON, який відправляється у колекцію `Recipes`.

У модулі `web` дуже часто зустрічається конвертація даних з формату JSON у модель `RecipeDto` та навпаки, особливо у класах DAO, де знаходиться логіка стосовно безпосереднього спілкування серверу з базою даних. Також у модулі `web`, пакеті DAO здійснюється пошук необхідних рецептів, що відповідають заданим користувачем критеріям.

Дані про рецепти відображаються на JSP сторінках у більш-менш зручному для користувача вигляді. Front-end отримує інформацію завдяки запитам на контролери, отримує необхідні атрибути та заповнює отриманими відповідями сторінку.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1 Опис інструментів для розробки

3.1.1 Опис використовуваних мов програмування, розмітки

Мова розробки back-end частини програми – Java, IDE – IntelliJ IDEA.

Java – це мова комп'ютерного програмування, яка є багатопоточною, на основі класів, об'єктно-орієнтованою і спеціально розроблена для того, щоб забезпечити якнайменше залежностей [9-10]. Вона призначена для того, щоб дозволити розробникам додатків "писати один раз, працювати в будь-якому місці" (WORA), що означає, що скомпільований код Java може працювати на всіх платформах, які підтримують Java, без необхідності перекомпіляції. Програми Java, як правило, збираються до байт-коду, який може працювати на будь-якій віртуальній машині Java (JVM) незалежно від архітектури комп'ютера. З 2016 р. Java є однією з найпопулярніших мов програмування, що використовуються, особливо для веб-додатків клієнт-сервер. Java була спочатку розроблена Джеймсом Гослінгом у Sun Microsystems (який з тих пір був придбаний корпорацією Oracle) і випущений в 1995 році як основний компонент платформи Java Sun Microsystems. Мова узяла більшу частину синтаксису з C і C ++, але вона має менше можливостей низького рівня, ніж будь-яка з них [11].

Оригінальна та еталонна реалізація Java-компіляторів, віртуальних машин і бібліотек класу були випущені Sun за власними ліцензіями.

Мови, використовувані для розробки front-end частини – HTML, CSS, JS.

Мова розмітки гіпертексту (HTML) – це стандартна мова розмітки для створення веб-сторінок та веб-програм. Веб-браузери отримують HTML-документи з веб-сервера або з локального сховища та перетворюють

документи на мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично та спочатку включала сигнали для появи документа.

HTML-елементи є будівельними блоками HTML-сторінок. За допомогою конструкцій HTML зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані в рендерингову сторінку. HTML забезпечує засіб для створення структурованих документів шляхом позначення структурної семантики тексту, такого як заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML позначені тегами, написаними за допомогою кутових дужок. Теги, такі як `` та `<input/>`, безпосередньо вводять вміст на сторінку. Інші теги, такі як `<p>` оточують і надають інформацію про текст документа, і можуть містити інші теги. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML може вставляти програми, написані мовою сценаріїв, такими як JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компоновання вмісту. Консорціум World Wide Web, що підтримує як стандарти HTML, так і CSS, заохочував використання CSS над явним презентаційним HTML з 1997 року.

Каскадні таблиці стилів (CSS) – це мова стилів аркушів, що використовується для опису презентації документа, написаного на мові розмітки, як HTML. CSS є наріжним каменем технології Всесвітньої павутини, поряд з HTML та JavaScript.

CSS призначений для того, щоб розділити дизайн та вміст, включаючи макет, кольори та шрифти. Це розділення може поліпшити доступ до контенту, забезпечити більшу гнучкість та контроль у специфікації характеристик презентації, дозволяти декільком веб-сторінкам ділитися форматкуванням, вказавши відповідний CSS в окремому файлі .css та зменшуючи складність і повторення у структурному вмісті.

Розділення форматкування та вмісту також дає змогу презентувати таку ж сторінку розмітки в різних стилях для різних методів візуалізації,

наприклад, на екрані, у друкованому вигляді, голосом (через браузер на основі мовлення або зчитувач екрана), а також на основі брайлівського шрифту тактильні пристрої. CSS також має правила для альтернативного форматування, якщо вміст доступний на мобільному пристрої.

Частина ім'я «каскадні» пішла від схеми із зазначеним пріоритетом, щоб визначити, яке правило стилю застосовується, якщо декілька правил відповідають певному елементу. Ця каскадна пріоритетна схема є передбачуваною.

Специфікації CSS підтримує консорціум World Wide Web.

JavaScript (часто скорочується як JS) – це інтерпретована на високому рівні мова програмування. Це мова, яка також характеризується як динамічна, слабко типізована, прототип-базована і багатопарадигменна.

Поряд з HTML та CSS, JavaScript є однією з трьох основних технологій Всесвітньої павутини. JavaScript надає інтерактивні веб-сторінки та, таким чином, є невід'ємною частиною веб-додатків. Переважна більшість веб-сайтів використовують її, і всі основні веб-переглядачі мають спеціальний движок JavaScript для її виконання.

Як мову мультипарадигми, JavaScript підтримує стилі програмування, керовані подіями, функціональні та імперативні (включаючи об'єктно-орієнтовані та прототипи). Вона має API для роботи з текстом, масивами, датами, регулярними виразами та базовими маніпулюваннями DOM, але сама мова не включає в себе будь-які введення/виведення, такі як мережі, сховища та графічні об'єкти, на які спираються ці дані, середовище хоста, в якому вона вбудована.

Спочатку лише реалізована сторона клієнта в веб-переглядачах, движки JavaScript тепер вбудовані в багато інших типів програмного забезпечення хосту, включаючи серверну сторону в веб-серверах та базах даних, а також в не веб-програмах, таких як текстові процесори та програмне забезпечення PDF, а також у режимі виконання середовища, що робить

JavaScript доступною для написання мобільних і настільних програм, включаючи віртуальні робочі столи.

Незважаючи на те, що між JavaScripts та Java є сильні зовнішні схожості, включаючи назву мови, синтаксис та відповідні стандартні бібліотеки, ці дві мови відрізняються і дуже відрізняються у дизайні. На JavaScript базуються такі мови програмування, як Self та Scheme.

3.1.2 Опис використаних технологій

Використовуються різні бібліотеки, контейнери і фреймворки для спрощення роботи з даними, більш красивої візуалізації інтерфейсу, забезпечення коду хорошим покриттям тестами.

Gradle – це інструмент автоматизації побудови проекту з відкритим вихідним кодом, орієнтований на гнучкість та продуктивність. Скрипти Gradle створюються за допомогою Groovy або Kotlin DSL. Інструмент дуже гнучкий та моделюється таким чином, що можна налаштовувати та розширювати проект будь-яким чином. Gradle виконує завдання швидко за рахунок кешу, який дозволяє не запускати одне и те ж завдання двічі, реагуючи лише на реальні зміни у скрипті та проекті [8].

Jsoup – Java-бібліотека для роботи з реальним HTML-сайтом. Вона надає дуже зручний API для вилучення та обробки даних, використовуючи найкращі методи DOM, CSS та jQuery.

Spring Framework (або Spring) – універсальний фреймворк з відкритим вихідним кодом для Java-платформи. Незважаючи на те, що Spring не забезпечує якусь конкретну модель програмування, він став широко поширеним в Java-співтоваристві головним чином як альтернатива і заміна моделі Enterprise JavaBeans. Spring надає велику свободу Java-розробникам в проектуванні; крім того, він надає добре документовані і легкі у використанні засоби вирішення проблем, що виникають при створенні

додатків корпоративного масштабу. У даному веб-додатку використовуються Spring Data, Spring Boot, Spring WEB, Spring MVC [12].

Програмне забезпечення Apache Tomcat – це програма з відкритим кодом Java Servlet, JavaServer Pages, Java Expression Language та Java WebSocket. Програми Java Servlet, JavaServer Pages, Java Expression Language та Java WebSocket. Програмне забезпечення Apache Tomcat розроблено у відкритому та спільному середовищі і випущено під ліцензію Apache версії 2.

Технологія JavaServer Pages (JSP) забезпечує спрощений, швидкий спосіб створення динамічного веб-контенту. Технологія JSP дозволяє швидко розвивати веб-додатки, які не залежать від серверів і платформ.

Бібліотека стандартних тегів JSP (JSTL) являє собою набір тегів бібліотек, які реалізують загальноприйняті функції, загальні для багатьох веб-додатків.

Servlet API використовують для створення сервлетів. Сервлет є інтерфейсом Java, реалізація якого розширює функціональні можливості сервера. Сервлет взаємодіє з клієнтами за допомогою принципу запит-відповідь. Пакети `javax.servlet` і `javax.servlet.http` забезпечують інтерфейси і класи для створення сервлетів [13].

Gson – це бібліотека Java, яка може бути використана для перетворення об'єктів Java у їх представлення JSON. Вона також може бути використана для перетворення JSON, зберігаємого у вигляді String, на еквівалентний об'єкт Java. Gson – це проект з відкритим кодом. Gson може працювати з будь-якими об'єктами Java.

Фасад Simple Logging For Java (SLF4J) служить простою фасадною або абстрактною структурою для різних систем журналювання (наприклад, `java.util.logging`, `logback`, `log4j`), що дозволяє кінцевому користувачеві підключати потрібну рамку журналювання під час розгортання.

Apache Commons – це проект Apache, орієнтований на всі аспекти компонентів Java, що використовуються багаторазово. Проект присвячений одній головній меті: створення та підтримка таких компонентів. Бібліотеки,

створені у рамках цього проекту, як правило, мають мінімальні залежності від інших бібліотек, щоб ці компоненти могли легко бути розгорнуті. Крім того, компоненти Commons будуть максимально стабільними для своїх інтерфейсів, так що користувачі можуть реалізовувати ці компоненти, не турбуючись про зміни в майбутньому.

Проект Lombok дозволяє сильно спростити Java-код, генеруючи за користувача методи установки значень, генерації значень, hashCode, equals і ще багато чого, замінюючи рутинний та позбавлений бізнес-логіки код на короткі та зрозумілі анотації.

JUnit – це модульна схема тестування для мови програмування Java, проста структура для написання повторюваних тестів. JUnit має важливе значення для розробки тестової розробки та є однією з сімейств структур юніт-тестування, яка спільно називається xUnit, що виникла з SUnit.

Mockito – це система, яка дозволяє писати тести з чистим та простим API. Бібліотека надає можливість використовувати так звані заглушки (mock) у разі, якщо не маємо можливості використовувати реальний об'єкт (наприклад, не хочемо залежати від реальних даних у базі і від її можливості працювати у кожен момент, коли треба запустити тести), перевірити, які методи були викликані з певними параметрами, а які ні, або перевірити, як змінився аргумент після роботи функцій.

PowerMock – це система, що поширює інші моделі, такі як Mockito, з більш потужними можливостями. PowerMock використовує спеціальне маскування класів і байтове кодування, що дозволяє тестувати статичні методи, конструктори, фінальні класи та методи, приватні методи, статичні ініціатори та багато іншого. PowerMock має на меті розширити існуючі API з невеликою кількістю методів та анотацій, щоб увімкнути додаткові функції. В даний час PowerMock підтримує EasyMock і Mockito.

При написанні юніт-тестів часто виникає необхідність обійти інкапсуляцію, і тому PowerMock включає в себе кілька функцій, що спрощує відображення, особливо корисне для тестування. Це дає легкий доступ до

внутрішнього стану, а також спрощує створення часткових та приватних заглушок.

jQuery – це швидка, мала та багатofункціональна бібліотека JavaScript. Вона робить такі речі, як переміщення HTML-документів та маніпулювання, обробка подій, анімація та Ajax, набагато простіше за допомогою простого API, який працює в безлічі браузерів. З поєднанням універсальності та розширюваності, jQuery змінило спосіб, у який мільйони людей пишуть JavaScript.

Bootstrap – це відкритий набір інструментів для розробки HTML, CSS та JS. Він дозволяє швидше реалізувати певні частини інтерфейсу свої ідеї або створювати щось нове за допомогою його змінних і Sass, адаптивної системи сітки, великих попередньо встановлених компонентів та потужних плагінів, побудованих на jQuery.

3.2 Аналіз функцій системи регулювання харчування

В умовах обмеженого часу дуже важливим є постановка акцентів на найважливіші задачі, тобто ті, що необхідно розв'язати у першу чергу. Можливо, що деякі завдання слід розбити на підзавдання. Така практика широко застосовується у методології розробки Scrum, де велике значення має відносна оцінка складності проблеми та часу, що потрібен на її вирішення, який оцінюється у Story Points (що, до речі, є абстрактною оцінкою, бо не прив'язується до якихось конкретних одиниць часу). Якщо охоплення завдання стає надто великим, задачу необхідно розбити на декілька таким чином, щоб людина, що виконує її, ясно уявляла собі рамки завдання та могла впоратися з ним до кінця так званого Sprint.

Тому, у рамках роботи було сформовано та виконано такі вимоги:

1. Як користувач, я хочу мати сторінку, на якій будуть відображені усі рецепти з загальною інформацією: назва рецепту, калорійність, загальна кількість БЖВ, процентне співвідношення БЖВ.
2. Як користувач, я хочу мати можливість, знаходячись на сторінці з рецептами, після натиску на його назву бути перенаправлений на сторінку з повною інформацією про рецепт.
3. Як користувач, я хочу мати можливість, знаходячись на сторінці з обраним мною рецептом, завантажити документ Word у форматі .docx на свій комп'ютер.
4. Як користувач, я хочу мати можливість, знаходячись на сторінці з усіма рецептами, видаляти будь-який з них з бази даних після натиску кнопки «Видалити».
5. Як користувач, я хочу мати можливість, знаходячись на сторінці з усіма рецептами, редагувати будь-який з них після натиску кнопки «Редагувати».
6. Як користувач, я хочу мати можливість, знаходячись на сторінці з усіма рецептами, додати свій рецепт до бази даних через форму.
7. Як користувач, я хочу мати можливість, знаходячись на сторінці з усіма рецептами, відшукати потрібний мені, користуючись такими параметрами пошуку, як бажана кількість калорій, БЖВ, бажані та небажані продукти.
8. Як адміністратор, я хочу мати можливість швидко поповнювати базу даних рецептами з сайту <http://daily-menu.ru>, указуючи необхідні мені ID рецептів.

Кожна із задач потребує окремого пояснення.

3.2.1 Опис задачі 1. Відображення загальної інформації про всі рецепти

Користувач повинен мати можливість бачити узагальнену інформацію про усі рецепти, відображені у вигляді таблиці. У таблиці повинна бути найважливіша інформація, без подробиць, бо інакше вона стане надто перевантаженою та буде незручною для сприйняття на веб-сторінці.

Повинні бути присутніми такі колонки:

- ID – ID рецепту, ціле число;
- Назва – назва рецепту, набір букв;
- Вага, г – вага страви у грамах;
- Енергетична цінність, ккал – енергетична цінність страви у ккал;
- Жири, г – жири страви у грамах;
- Білки, г – білки страви у грамах;
- Вуглеводи, г – вуглеводи страви у грамах;
- Час приготування – час приготування, ціле число з позначенням часу.

Назва рецепту повинна відображатися як посилання, по натисканні на котре здійснюється перенаправлення на сторінку з детальною інформацією про страву ([домен сайту]/recipedata/{id рецепту}). Сторінка з рецептом повинна відкриватися на іншій вкладці браузера.

Усі колонки повинні бути присутні та коректно відображені. Таблиця повинна поміщуватися по ширині (без горизонтальної полоси прокрутки), якщо користувач відкриває сайт з пристрою, діагональ якого є більшою, ніж 8 дюймів, та з роздільною здатністю екрану більшою, ніж 1280x720 пікселів.

Користувач може знайти таблицю за адресою [домен сайту]/recipes.

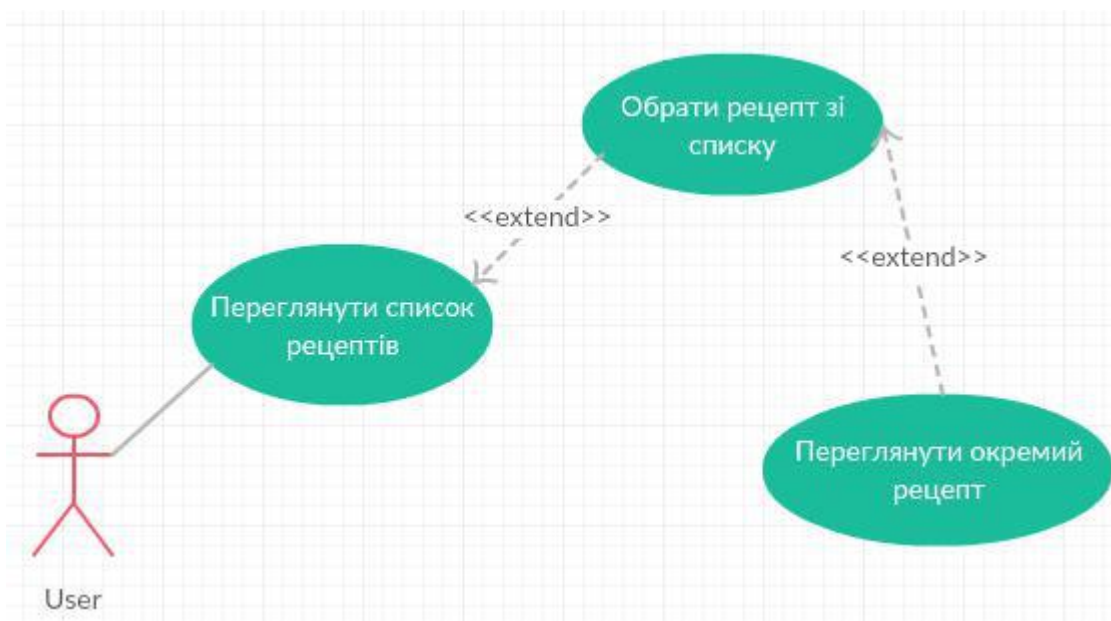


Рисунок 3.1 – Use-case діаграма «Перегляд списку рецептів»

Список рецептов

ID	Название	Общий вес, г	Энергетическая ценность, ккал	Жиры, г	Белки, г	Углеводы, г	Время приготовления	Редактировать	Удалить
119	Блины классические тонкие	586.0	1142.0	36.0	34.0	173.0	40 мин.	Изменить	Удалить
120	Грибной суп с кабачками	1660.0	535.0	16.0	27.0	69.0	30 мин.	Изменить	Удалить
121	Банановые булочки	906.0	1499.0	25.0	55.0	260.0	10 мин.	Изменить	Удалить
122	Овсяные коржики	475.0	831.0	18.0	52.0	121.0	10 мин.	Изменить	Удалить
123	Куриное филе «в вакууме»	620.0	858.0	26.0	138.0	0.0	20 мин.	Изменить	Удалить
124	Клюквенно-манный мусс	950.0	492.0	1.0	12.0	103.0	20 мин.	Изменить	Удалить
125	Айлазан	1875.0	834.0	24.0	28.0	129.0	20 мин.	Изменить	Удалить
126	Запеченные овощи с сыром	420.0	200.0	9.0	12.0	19.0	10 мин.	Изменить	Удалить
127	Рататуй	2430.0	778.0	24.0	22.0	121.0	30 мин.	Изменить	Удалить
128	Зеленый чай с лимоном	263.0	8.0	0.0	1.0	1.0	5 мин.	Изменить	Удалить
129	Кофе в турке	207.0	14.0	1.0	1.0	0.0	10 мин.	Изменить	Удалить
130	Гарнир из пшена	623.0	490.0	14.0	14.0	78.0	15 мин.	Изменить	Удалить
131	Шоколадная курица	140.0	185.0	2.0	29.0	10.0	10 мин.	Изменить	Удалить
132	Печеный картофель	1020.0	950.0	24.0	20.0	163.0	5 мин.	Изменить	Удалить

Рисунок 3.2 – Реалізація виводу загальної інформації про страви у базі даних

3.2.2 Опис задачі 2. Повна інформація про певний рецепт

Користувач повинен мати змогу переглядати повну інформацію про страву на окремій веб-сторінці. Потрапити до цієї сторінки він може, обравши певний рецепт з переліку, що знаходиться за адресою [домен сайту]/recipes, або самостійно ввести до адресної строки браузера адресу [домен сайту]/recipedata/{id рецепту}.

На сторінці повинні бути відображені ID рецепту, його назва, посилання на вихідний матеріал, час приготування, час на додаткові дії, зображення готової страви, список продуктів (кожен продукт повинен мати назву, вагу у грамах, енергетичну цінність у ккал, параметри БЖВ у грамах), оформлений у таблицю, узагальнена інформація про показники страви, на 100 грам та на одну порцію (вага у грамах, енергетична цінність у ккал, параметри БЖВ у грамах), пропорції БЖВ у процентах та алгоритм приготування страви.

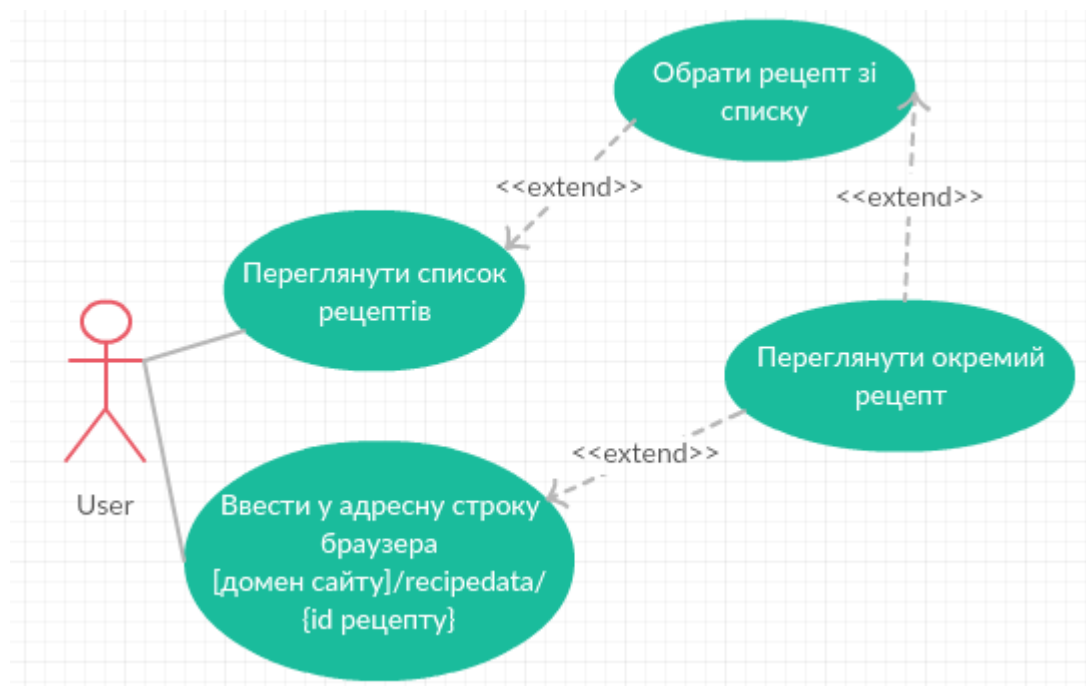


Рисунок 3.3 – Use-case діаграма «Перегляд окремого рецепту»

Фриттата с тунцом

ID:	179
Исходный материал:	http://daily-menu.ru/dailymenu/recipes/view/179
Время приготовления:	10 мин.
+ 20 минут в духовке	

[Скачать файл](#)



Продукт	Вес (г)	Энергетическая ценность (ккал)	Состав		
			Белки (г)	Жиры (г)	Углеводы (г)
Яйцо куриное	180.0	283.0	23.0	21.0	1.0
Горошек замороженный	100.0	72.0	6.0	0.0	16.0
Тунец натуральный Консервы	90.0	86.0	20.0	1.0	0.0
Укроп и другие пряные травы	20.0	8.0	1.0	0.0	1.0
Масло оливковое	10.0	90.0	0.0	10.0	0.0
Помидоры	80.0	16.0	1.0	0.0	3.0
Сыр твердый (маасдам)	20.0	71.0	5.0	5.0	1.0
Всего:	500.0	626.0	56.0	37.0	23.0
На 1 порцию:	250.0	313.0	27.9	18.7	11.5
На 100 г:		147.0	11.9	8.2	6.3
Пропорции, %:			34.0%	52.0%	14.0%

Приготовление:

Фриттата - это омлет с начинкой. Начинка может быть великое множество, каждая хозяйка может придумать свое неповторимое сочетание. Мы предлагаем сделать это блюдо с тунцом и зеленым горошком. 1. Включить духовку на температуру 200 градусов. 2. Взбить яйца, посолить, добавить специй по вкусу. 3. Добавить зеленый горошек, тунец, аккуратно перемешать. 4. Смазать форму маслом и вылить яичную смесь. 5. Выпекать 20 минут, до появления золотистой корочки. Можно сверху украсить кружочками томатов и немного присыпать сыром. 6. Посыпать свежим укропом, подавать с зерновым хлебом.

Рисунок 3.4 – Зовнішній вигляд сторінки з окремим рецептом

3.2.3 Опис задачі 3. Збереження інформації про рецепт у форматі .doc

Користувач повинен мати можливість зберегти окремий рецепт у форматі .doc, якщо побажає. Для цього на сторінці з окремим рецептом повинне бути посилання з найменуванням «Завантажити файл». Після цього користувач зможе обрати назву файлу, шлях збереження. Збережений файл можна відкрити та переглянути. У ньому повинна бути інформація, ідентична той, що присутня на сторінці.

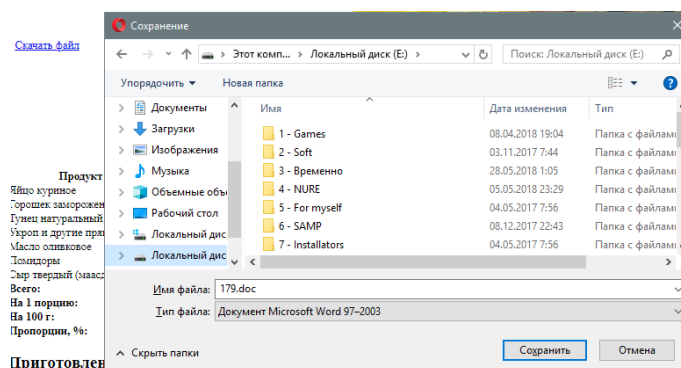


Рисунок 3.6 – Посилання на завантаження файлу та діалогове вікно

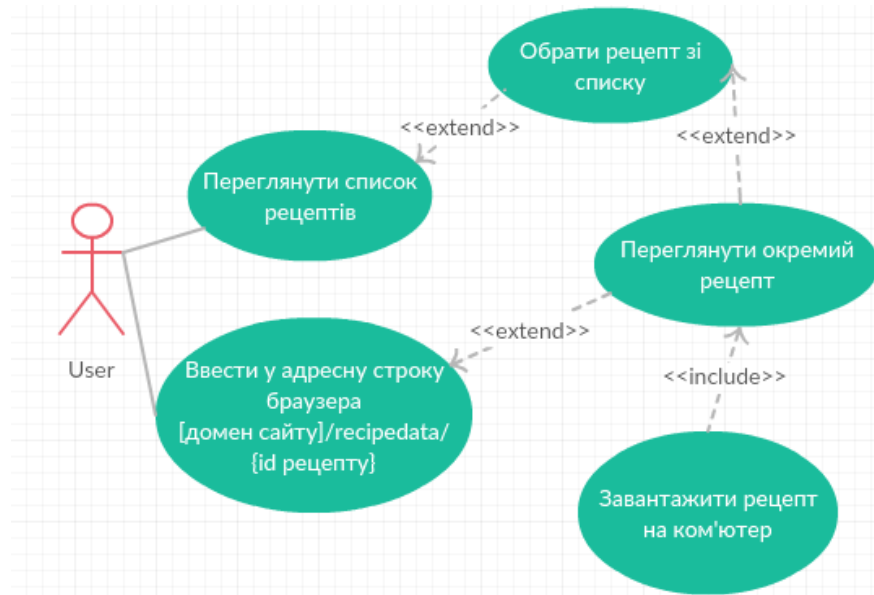


Рисунок 3.6 – Use-case діаграма «Завантаження рецепту»

Состав						
Продукт	Вес (г)	Енергетична цінність (ккал)	Білки (г)	Жири (г)	Углеводи (г)	
Яйцо куриное	180.0	283.0	23.0	21.0	1.0	
Горошек замороженный	100.0	72.0	6.0	0.0	16.0	
Тунец натуральный Консерв	90.0	86.0	20.0	1.0	0.0	
Укроп и другие пряные травы	20.0	8.0	1.0	0.0	1.0	
Масло оливковое	10.0	90.0	0.0	10.0	0.0	
Помидоры	80.0	16.0	1.0	0.0	3.0	
Сыр твердый (маасдам)	20.0	71.0	5.0	5.0	1.0	
Всего:	500.0	626.0	56.0	37.0	23.0	
На 1 порцию:	250.0	313.0	27.9	18.7	11.5	
На 100 г:		147.0	11.9	8.2	6.3	
Пропорции, %:			34.0%	52.0%	14.0%	

Приготовление:

Фриттата – это омлет с начинкой. Начинка может быть великое множество, каждая хозяйка может придумать свое неповторимое сочетание. Мы предлагаем сделать это блюдо с тунцом и зеленым горошком. 1. Включить духовку на температуру 200 градусов. 2. Взбить яйца, посолить, добавить специи по вкусу. 3. Добавить зеленый горошек, тунец, аккуратно перемешать. 4. Смазать форму маслом и вылить яичную смесь. 5. Выпекать 20 минут, до появления золотистой корочки. Можно сверху украсить кружочками томатов и немного присыпать сыром. 6. Посыпать свежим укропом, подавать с зерновым хлебом.

Рисунок 3.7 – Зовнішній вигляд збереженого документу

3.2.4 Опис задачі 4. Видалення рецептів

Користувач повинен мати можливість видаляти будь-які рецепти з бази даних. Зробити це він зможе через сторінку [домен сайту]/recipes, натиснувши на посилання «Видалити» напроти рецепту, якого потрібно позбутися. Після видалення сторінка повинна оновитися, і користувач повинен побачити зміни у таблиці.



Рисунок 3.8 – Use-case діаграма «Видалення рецепту»

119	Блины классические тонкие	586.0	1142.0	36.0	34.0	173.0	40 мин.	Изменить	Удалить
-----	---	-------	--------	------	------	-------	---------	--------------------------	-------------------------

Рисунок 3.9 – Посилання «Видалити» напроти рецепту

3.2.5 Опис задачі 5. Редагування рецептів

Користувач повинен мати право редагувати будь-які рецепти. Зробити це він зможе через сторінку [домен сайту]/recipes, натиснувши на посилання «Редагувати» напроти рецепту, який потрібно змінити. Після цього сторінка оновиться (посилання у адресній стрічці при цьому зміниться на [домен сайту]/edit/{id рецепту}, та користувач побачить внизу форму, за допомогою якої можна вносити зміни до страви. Усі наявні на момент зміни

дані повинні бути занесені до форми заздалегідь (наприклад, якщо користувач забажає змінити лише кількість калорій, йому ні до чого заносити самому інформацію, що не зазнала змін). Після внесення змін користувач натисне кнопку «Редагувати», сторінка повинна оновитися ще раз. Зміни повинні бути видимими у таблиці.

Усі надані до форми дані повинні підлягати валідації. У разі помилки користувача потрібно повідомити про це, а саму операцію відмінити.



Рисунок 3.10 – Use-case діаграма «Редагування рецепту»

119	Блины классические тонкие	586.0	1142.0	36.0	34.0	173.0	40 мин.	Изменить	Удалить
-----	---	-------	--------	------	------	-------	---------	--------------------------	-------------------------

Рисунок 3.9 – Посилання «Редагувати» напроти рецепту

Добавить или изменить рецепт

ID	<input type="text" value="178"/>
Имя рецепта	<input type="text" value="Творожный пирожок"/>
Вес, г	<input type="text" value="360.0"/>
Энергетическая ценность, ккал	<input type="text" value="512.0"/>
Жиры, г	<input type="text" value="15.0"/>
Белки, г	<input type="text" value="44.0"/>
Углеводы, г	<input type="text" value="47.0"/>
Время приготовления	<input type="text" value="15 мин."/>
<input type="button" value="Редактировать"/>	

Рисунок 3.10 – Форма редагування рецепту

3.2.6 Опис задачі 6. Додання нового рецепту до бази даних

Користувач повинен мати право редагувати будь-які рецепти. Зробити це він зможе через сторінку [домен сайту]/recipes, прокрутивши сторінку до самого низу, до форми додавання рецепту до бази даних. Після внесення даних користувач натисне кнопку «Додати», сторінка повинна оновитися. Зміни повинні бути видимими у таблиці.

Усі надані до форми дані повинні підлягати валідації. У разі помилки користувача потрібно повідомити про це, а саму операцію відмінити.

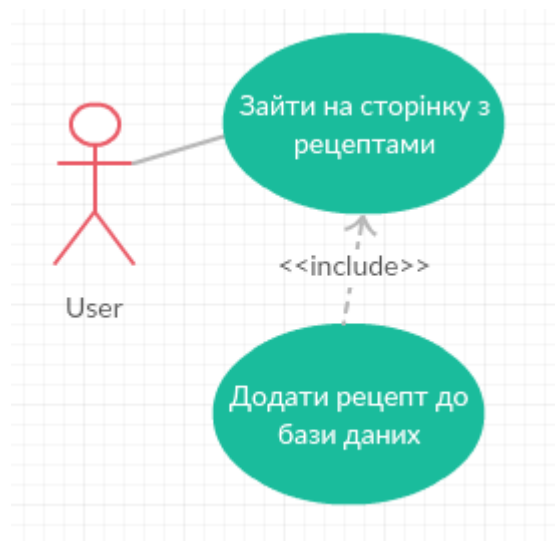


Рисунок 3.11 – Use-case діаграма «Додавання рецепту до бази даних»

Добавить или изменить рецепт

Имя рецепта	<input type="text" value="рецепт"/>
Вес, г	<input type="text" value="100"/>
Энергетическая ценность, ккал	<input type="text" value="2000"/>
Жиры, г	<input type="text" value="10"/>
Белки, г	<input type="text" value="20"/>
Углеводы, г	<input type="text" value="30"/>
Время приготовления	<input type="text" value="10 мин."/>
<input type="button" value="Добавить"/>	

Рисунок 3.12 – Форма додавання рецепту до бази даних

3.2.7 Опис задачі 7. Пошук рецептів

Користувач повинен мати можливість шукати рецепти. Зробити це він зможе через сторінку [домен сайту]/search. Він повинен бачити фільтри пошуку, які включають у себе обмеження по вазі, параметрах БЖВ, бажаних та небажаних продуктах. Якщо поле пусте, параметр пошуку не враховується. Після внесення даних користувач натисне кнопку «Шукати», сторінка повинна оновитися. Фільтри пошуку повинні зберегтися у полях, а під формою пошуку повинні з'явитися результати або надпис «Вибачте, по вашому запиту нічого не було знайдено».

Результат пошуку повинен бути узагальнений у вигляді таблиці. У таблиці повинна бути найважливіша інформація:

- ID – ID рецепту, ціле число;
- Назва – назва рецепту, набір букв;
- Вага, г – вага страви у грамах;
- Енергетична цінність, ккал – енергетична цінність страви у ккал;
- Жири, г – жири страви у грамах;
- Білки, г – білки страви у грамах;
- Вуглеводи, г – вуглеводи страви у грамах;
- Час приготування – час приготування, ціле число з позначенням часу.

Назва повинна бути посиланням на сторінку з повним описом рецепту.



Рисунок 3.13 – Use-case діаграма «Пошук рецептів»

Поиск рецептов

<input type="text" value="800"/>	<input type="text" value="835"/>
Мин. кол-во белков	Макс. кол-во белков
<input type="text" value=""/>	<input type="text" value=""/>
Мин. кол-во жиров	Макс. кол-во жиров
<input type="text" value=""/>	<input type="text" value=""/>
Мин. кол-во углеводов	Макс. кол-во углеводов
<input type="text" value="Банан"/>	<input type="button" value="+"/>
<input type="text" value="Ванилин"/>	<input type="button" value="+"/> <input type="button" value="-"/>
<input type="text" value="Нежелаемый продукт..."/>	<input type="button" value="+"/>
<input type="button" value="Искать"/>	

ID	Название	Общий вес, г	Энергетическая ценность, ккал	Жиры, г	Белки, г	Углеводы, г	Время приготовления	Редактировать	Удалить
121	Банановые булочки	906.0	1499.0	25.0	55.0	260.0	10 мин.	Изменить	Удалить

Рисунок 3.14 – Форма пошуку та результат запиту

3.2.8 Опис задачі 8. Наповнення бази даних рецептами через адміністратора

Адміністратор повинен мати інструмент для швидкого наповнення бази даних рецептами з daily-menu.ru. Інструмент представляє собою пакет jar, що запускається командою `java -jar`, та потребує двох параметрів – початковий та кінцевий ID бажаних рецептів. Якщо він хоче завантажити лише один рецепт, то ID повинен бути один. Якщо ID не вдалося призвести до цілого числа, програма повинна згенерувати виключення та заборонити подальшу роботу алгоритму.

Інструмент повинен парсувати сторінки HTML, забирати всю інформацію, що стосується рецепту у документ JSON та відправляти його до бази даних. Парсер повинен розпізнавати ID рецепту, його назву, посилання на вихідний матеріал, час приготування, час на додаткові дії, зображення готової страви, список продуктів (кожен продукт повинен мати назву, вагу у грамах, енергетичну цінність у ккал, параметри БЖВ у грамах), оформлений у таблицю, узагальнену інформацію про показники страви, на 100 грам та на одну порцію (вага у грамах, енергетична цінність у ккал, параметри БЖВ у грамах), пропорції БЖВ у процентах, та алгоритм приготування страви.

У разі помилок на стороні third-party сервісу (daily-menu.ru) інструмент не повинен пошкодити базу даних, бажано згенерувати у лог повідомлення про недоступність сайту.

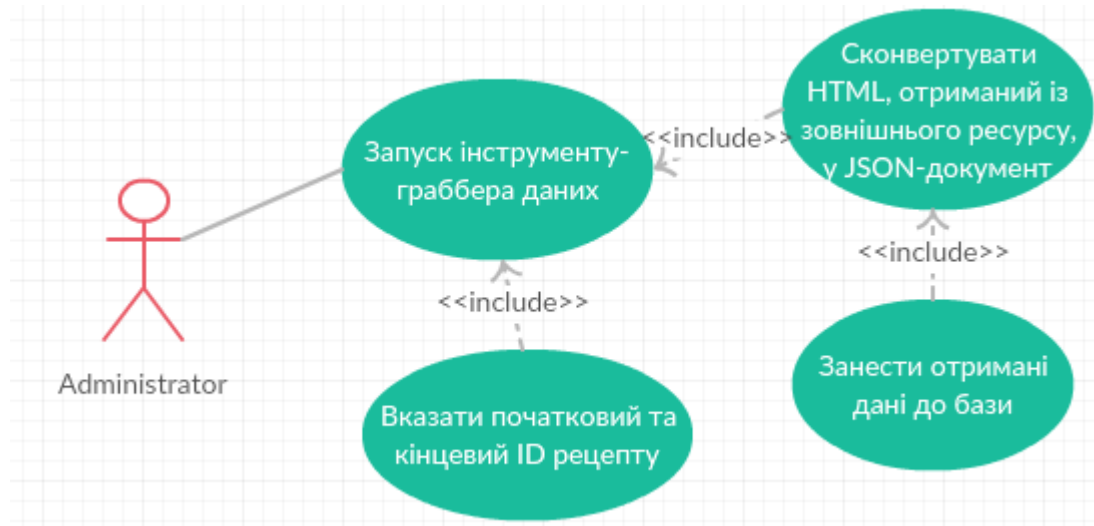


Рисунок 3.15 – Use-case діаграма «Заповнення бази даних рецептами»

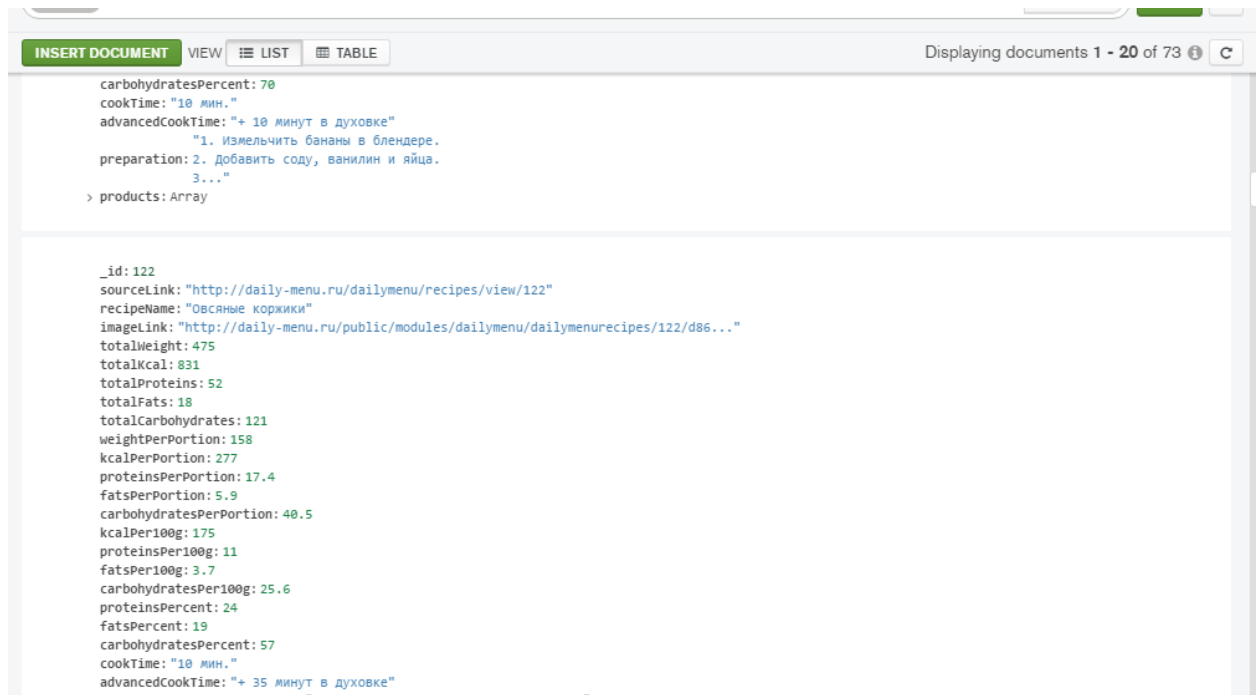


Рисунок 3.16 – База даних, заповнена за допомогою інструменту збору даних

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра була розробка програмної системи регулювання харчування. Аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера, на якому буде виконуватися розробка, так як в процесі проектування використовувалося комп'ютерне обладнання.

4.1 Загальні питання з охорони праці

Санітарно-побутові умови, умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [1] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

4.1.1 Правові та організаційні основи охорони праці

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

4.1.2 Організаційно-технічні заходи з безпеки праці

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [2].

Обов'язковими вимогами враховане наступне:

– ознайомлення з правилами безпеки праці, одержання відповідних інструктажів засвідчується у журналі інструктажів.

– перед допуском до самостійної роботи кожен працівник має право на навчання з питань охорони праці і роботодавець зобов'язаний, і проводить таке навчання у вигляді двох інструктажів з питань охорони праці:

1) вступного, який проводять працівники служби охорони праці об'єкта господарювання з усіма працівниками, яких приймають на роботу незалежно від їхньої освіти та стажу роботи за програмою, в якій подають загальні питання охорони праці із врахуванням її особливостей на об'єкті господарювання;

2) первинного, який проводять керівники структурних підрозділів на місці праці з кожним працівником до початку їхньої роботи на цьому робочому місці.

Проходження працівником цих інструктажів з питань охорони праці підтверджується записами у відповідних журналах обліку інструктажів і

скріплюється підписами осіб, які проводили інструктажі та осіб, які отримали інструктажі.

- 3) Повторний (не рідше одного разу в 6 місяців);
- 4) Позаплановий (при зміні правил охорони праці);
- 5) Поточний (проводять з працівниками перед виконанням робіт, на яких оформляється наряд-допуск)

– обов’язкові організаційні заходи перед початком, під час і після завершення роботи повинні включати перевірку (візуально) наявності і справності електрообладнання та його заземлення, а під час виконання роботи вимогу «не залишати без нагляду обладнання, яке працює». Після закінчення роботи - вимагається прибирання робочого місця, відключення всіх електроприладів від електромережі.

4.2 Аналіз стану умов праці

Робота над створенням програмної системи регулювання харчування проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп’ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	4
Висота, м	2.7
Площа, м ²	20
Об’єм, м ³	54

Згідно з [3] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [4] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	650	не менше 600
Ширина простору для ніг, мм	550	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	430	400 ÷ 500
Ширина сидіння, мм	450	не менше 400
Глибина сидіння, мм	350	не менше 400
Висота поверхні спинки, мм	400	не менше 300
Ширина опорної поверхні спинки, мм	400	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	750	700 ÷ 800

Приміщення кабінету знаходиться на четвертому поверсі чотирьох поверхової будівлі і має об'єм 54 м³, площу – 20 м². У цьому кабінеті обладнано два місця праці, обидва укомплектовані ПК.

Температура в приміщенні протягом року коливається у межах 17–26°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум в лабораторії знаходиться на рівні 50 дБА. Система

вентилювання приміщення — природна неорганізована, а опалення — централізоване.

4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням виконання випускної роботи бакалавра відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту.

Роботу над дипломним проектом визнано, такою, що займає 50% часу робочого дня та при восьмигодинній робочій зміні рекомендовано встановити додаткові регламентовані перерви - для розробників програм тривалістю 15 хв. через кожну годину роботи.

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [7] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U=+220\text{В} \pm 5\%$;
- робочий струм $I=2\text{А}$;
- споживана потужність $P=350\text{ Вт}$.

Робоче місце мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [4].

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
Фізичні			
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[5]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[6]

Продовження таблиці 4.3

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[6]
<i>психофізіологічні:</i>			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[7] [4]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці- сидіння користувача,) та організації робочого часу - безпервна робота)	2	[7] [4]

4.3.2 Пожежна безпека

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання, таких як:

- 1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонам відповідно до ПУЕ;
- 2) застосування в конструкції швидкодійних засобів захисного

відключення можливих джерел запалення;

3) виключення можливості появи іскрового розряду в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення.

Згідно ДБН В.2.5-28:2018 [6] таке приміщення, площею 20 м², відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні

контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають ДСН 3.3.6.042-99 [3] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату:

температури, відносної вологості й рухливості повітря. Рівні позитивних і негативних іонів у повітрі мають відповідати ДСН 3.3.6.042-99 [3].

4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2018 [6]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових – $1/10$:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n – площа підлоги, m^2 .

$$S_n = a \cdot b = 5 \cdot 4 = 20 \text{ м}^2,$$

$$S = 1/8 \cdot 20 = 2,5 \text{ м}^2.$$

Приймаємо 2 вікна площею $S=1,6 \text{ м}^2$ кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, m^2 ; $S = 20 m^2$;

Z – поправочний коефіцієнт світильника ($Z = 1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 20 \cdot 1.1 \cdot 1.5}{5400 \cdot 0.575 \cdot 2} \approx 2$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.4.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, що супроводжує роботу користувачів персональних комп'ютерів (зумовлений як роботою системних блоків, клавіатури, так і

друкуванням на принтерах, а також зовнішніми чинниками), коливається у межах 50–65 дБА ДСН 3.3.6.042-99 [3]. У залах опрацювання інформації та комп'ютерного набору рівні шуму не повинні перевищувати 65 дБА.

Віброізоляція можливо здійснювати за допомогою спеціальної прокладки під системний блок, який послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам ДСН 3.3.6.042-99 [3].

4.4.4 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти).

Також має здійснюватися провітрювання приміщення в залежності від погодних умов. Тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;

- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря.

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;
- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;
- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

Вимоги безпеки при надзвичайних ситуаціях:

- 1) При раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності: периферійні пристрої, ВДТ, системний блок, стабілізатор (або блок безперервного живлення). Витягнути вилки з

розеток. При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК, знайти місце загоряння і виконати всі можливі заходи для його ліквідації, попередивши терміново про це керівництво.

2) При замиканні, перевантаженні електричного струму на електричному обладнанні, внаслідок ураження грозової блискавки та ймовірної небезпеки ураженням електричним струмом, приймають наступне:

- попередження замикання здійснюється правильним вибором, монтажем експлуатації мереж;
- застосування захисту схем у вигляді швидкодіючих реле, а також вимикачів, плавких запобіжників.

Також застосовують різні електричні захисні засоби від ураження струмом:

а) Ізолюючі - ізолюють людини від струмоведучих або заземлених частин, а так-же від землі. Вони діляться на основні та додаткові.

б) Основні - володіють ізоляцією, здатної довго витримувати робоче напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, знаходячи-трудящих під напругою.

в) Запобіжні - володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом під цим напругою.

4.6 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом, приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт використання вертикальних заземлювачів η_v в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги)

враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.3)$$

де $R_{пр.з.}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho = 40$ Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $\rho_{розр.}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$ і горизонтальних $\rho_{розр.г.}$, Ом·м за формулою:

$$\rho_{розр.} = \psi \cdot \rho, \quad (4.4)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів І кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{розр.в.} = 1,7$ і горизонтальних $\rho_{розр.г.} = 5,5$ Ом·м.

$$\rho_{розр.в.} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м}$$

$$\rho_{розр.г.} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача R_v , Ом, за формулою (4.5).

$$R_B = \frac{\rho_{\text{розр.В}}}{2 \cdot \pi \cdot l_B} \cdot \left(\ln \frac{2 \cdot l_B}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_B}{4 \cdot t - l_B} \right), \quad (4.5)$$

де l_B – довжина вертикального заземлювача (для труб - 2–3 м; $l_B=3$ м);

$d_{\text{ст}}$ – діаметр стержня (для труб - 0,03–0,05 м; $d_{\text{ст}}=0,05$ м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.6):

$$t = h_B + \frac{l_B}{2}, \quad (4.6)$$

де h_B – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_B = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_B :

$$n = \frac{2 \cdot R_B}{R_d} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (4.7)$$

Γ визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_B = 0,57$ (табличне значення).

б) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_B , шт:

$$n_B = \frac{2 \cdot R_B}{R_d \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.9)$$

де L_B – відстань між вертикальними заземлювачами, (прийняти за $L_B = 3\text{ м}$);

n_B – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_Γ , Ом:

$$R_\Gamma = \frac{\rho_{\text{розр.}\Gamma}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_\Gamma}, \quad (4.10)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15 \text{ м}$;

h_Γ – глибина закладання горизонтальних заземлювачів ($0,5 \text{ м}$);

l_c – довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_\Gamma = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача η_c відповідно до необхідної кількості вертикальних заземлювачів n_B .

Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг}} = \frac{R_B \cdot R_\Gamma}{R_B \cdot \eta_c + R_\Gamma \cdot n_B \cdot \eta_B} \leq R_d. \quad (4.18)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{заг}} < 4 \text{ Ом}$, а саме:

$$R_{\text{заг}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d$$

3) При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаженія та наявність перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом; описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведено рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

Перелік корисних посилань до розділу 4

1. [Закон України «Про охорону праці»](https://zakon.rada.gov.ua/laws/show/2694-12). Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12> - 10.14.1992 р.

2. [Нормативно-правовий акт з охорони праці. НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці»](https://zakon.rada.gov.ua/laws/show/z0231-05) - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0231-05>

3. Державні санітарні норми. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/va042282-99> - 01.02.1999 р.

4. Державні санітарні правила і норми. ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> - 10.12.1998 р.

5. Державний стандарт України. ДСТУ Б В.2.5-82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом» - Режим доступу: <http://epicentre.co.ua/dstu/doc28522.html> - 01.07.2016 р.

6. Державні будівельні норми. ДБН В.2.5-28:2018 «Природне і штучне освітлення» - Режим доступу: <http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf> - 03.10.2018

7. Нормативно-правовий акт з охорони праці. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18> - 14.02.2018 р.

ВИСНОВКИ

На жаль, в рамках дипломної роботи не вдалося реалізувати більшу частину запланованого функціоналу. Багато в чому на результат вплинула відсутність достатньої кількості часу.

Однак же, вже реалізована частина функціоналу дає зрозуміти, що робота з багатомодульною розподіленою архітектурою має досить вигод в порівнянні зі своїми аналогами. Це дозволяє зменшити повторюваність коду, логічно розподілити обов'язки кожного із модулів та час виконання окремих модулів стає швидшим, бо для роботи, наприклад, модулю `web` використовуються лише необхідний йому модуль `models`, не збираючи у свій `war`-файл непотрібні класи з модулю `recipe_grabber`.

Також використання документо-орієнтованої бази даних MongoDB дозволила спростити представлення даних рецептів. Замість складної архітектури, наприклад, реляційної бази даних маємо документ у вигляді JSON. Так як нам не потрібна цілісність і нам не так важливо, що дані повторюються, це дало змогу якісно підвищити швидкість роботи запитів до бази даних у порівнянні з використанням для такої мети реляційної СУБД.

Під час розробки програмного продукту були використані: теорія проектування `noSQL` баз даних на прикладі MongoDB; основні принципи побудови веб-додатків з використанням Gradle; принципи контролю зміни файлів і їх версіонування за допомогою системи контролю версій Git; середовище розробки програмних систем IntelliJ IDEA, мови HTML та CSS, мова програмування Java і бібліотеки (Jsoup, Spring, JSTL, Servlet API, JSP, SLF4J, Gson, Apache Commons, Jacob).

Робота над дипломним проектом сприяла покращенню навичок, поглибленню знань та набуттю досвіду у розробці багатомодульних веб-додатків.

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також інформація щодо пожежної та електробезпеки. Виконано розрахунок захисного заземлення та освітлення. Наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Основы и принципы правильного питания спортсменов, спортсменов-любителей и всех-всех-всех. [Электронный ресурс] / И. Самарский // Блог Игоря Самарского. Здоровье тела и сила духа. – 2018. – Режим доступа: [www/ URL: http://sportlif.ru/osnovy-i-printsipy-pravilnogo-pitaniya.html](http://www.sportlif.ru/osnovy-i-printsipy-pravilnogo-pitaniya.html).
2. 22-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Збірник матеріалів форуму. Т.7 [Текст] : тез. докл. наук.-практ. конф. (квітень 2018 р.) / відп. ред. В.О. Дорошенко. – Харків : ХНУРЕ, 2018. – 123 с.
3. О проекте МОЙ ДИЕТОЛОГ [Электронный ресурс] / – Режим доступа: [www/ URL: https://moydietolog.ru/about/](https://moydietolog.ru/about/)
4. MyFitnessPal [Electronic resource] / – Access mode: [www/ URL: https://www.myfitnesspal.com](https://www.myfitnesspal.com)
5. Борисенко А. А. Моделирование, разработка и оптимизация продуктов здорового питания: монография. [Текст] / А. А. Борисенко, Л.А. Сарычева, А.А. Борисенко (ст.). – Ставрополь : СевКавГТУ, 2012. – 197 с.
6. MongoDB Documentation. [Electronic resource] / – Access mode: [www/ URL: https://docs.mongodb.com/](https://docs.mongodb.com/)
7. Gradle User Guide [Electronic resource] / – Access mode: [www/ URL: https://docs.gradle.org/4.2.1/userguide/userguide.html](https://docs.gradle.org/4.2.1/userguide/userguide.html)
8. Lombok features [Electronic resource] / – Access mode: [www/ URL: https://projectlombok.org/features/all](https://projectlombok.org/features/all)
9. Шилдт, Г. Java 8. Полное руководство [Текст] / Г. Шилдт – М.: Вильямс-М., 2015. – 1337 с.
10. Bloch, J. Effective Java (2nd edition). [Text] / J. Bloch. – Massachusetts: Courier in Stoughton, 2008. – 369 с.

11. Eckel, B. Thinking in Java (4th edition). [Text] / B. Eckel. – Massachusetts: Courier in Stoughton, 2006. – 1079 p.
12. Spring Documentation. [Electronic resource] / – Access mode: www/
URL: <https://docs.spring.io/spring/docs/current/spring-framework-reference/>
13. Блинов, И.Н. Java. Промышленное программирование : практ. пособие [Текст] / И.Н. Блинов, В.С. Романчик. – Минск : УниверсалПресс, 2007. – 704 с.

Додаток А

Комп'ютерна презентація

ІКС оптимізації харчування



Виконав:

студентка 4 курсу гр. КІ-15бд

Старовойтова Аліна Вікторівна

Керівник :

Сафонова С.О., к.т.н., доцент

Рисунок А.1 – Слайд №1

Актуальність

Тип борщу	Белки	Жиры	Гли	Углеводы	Ккал
Борщ	0.8	1	30	4.1	34
Борщ домашний	1.1	1.9	30	4.2	36
Борщ зеленый на говяжьем бульоне	2.7	2.5	30	2.2	42
Борщ зеленый со щавелем	1.3	2.8	30	3.1	43
Борщ из свежей капусты и карт	1	1.1	30	5.4	36
Борщ летний	1.1	1.2	30		

Меню импульсной диеты

Приблизительное меню для первого этапа

Завтрак: порция гречневой или овсяной каши, приготовленной на воде; стакан йогурта домашнего приготовления или кефира (1% жирности или обезжиренного); груша.

Второй завтрак: 100 г обезжиренного творога и небольшое яблоко.

Обед: овощной суп (рекомендуется не зажаривать); куриная котлета, приготовленная на пару, или просто небольшой кусок куриного филе; несколько ложек пустого отварного коричневого риса или гречневой крупы.

Полдник: несколько штук курятины плюс йогурт домашнего приготовления (или 2-3 ст. л. обезжиренного творога).

Ужин: порция запеченной или тушеной рыбы; овощной салат, заправленный несколькими каплями растительного масла и свежевыжатым лимонным соком.

Приблизительное меню для второго этапа

Завтрак: коктейль доктор Слим; чашка кофе без сахара, в который можно добавить немного обезжиренного молока.

Второй завтрак: коктейль доктор Слим.

Обед: примерно 150 г куриной грудки, приготовленной на пару; овощной суп без заправки.

Полдник: коктейль доктор Слим.

Ужин: порция нежирной рыбы, приготовленной без добавления масла, либо немного морепродуктов до 300 г овощей в любом виде.

Примечание. Во всех приемах пищи рекомендованный профессиональный коктейль можно заменять 50 г нежирного творога и несколькими ломтиками любимого разрешенного фрукта.

Рисунок А.2 - Слайд №2

Порівняння існуючих рішень

Система "МІЙ ДІЕТОЛОГ"

- + Всі матеріали розроблені дієтологами і нутриціологами;
- + Можливість створювати безліч профілів;
- + Широкий функціонал готових раціонів;
- + Можливість придбати необхідні для раціону продукти;
- + Персональний календар контролю раціону;
- + Тест для розрахунку середньої втрати калорій в день;
- Мало фільтрів для пошуку дієт;
- Не можна організувати пошук окремо за рецептами або продуктами;
- Магазин доступний тільки для громадян РФ.

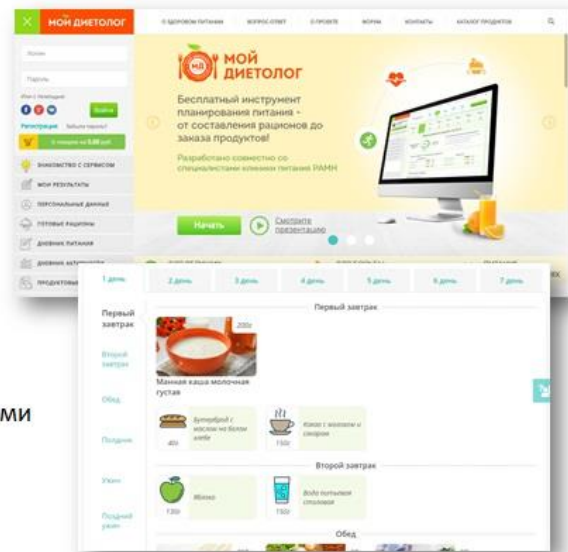


Рисунок А.3 - Слайд №3

Порівняння існуючих рішень

MyFitnessPal (веб-версія)

- + має синхронізацію з власними мобільними додатками;
- + контроль не тільки харчування, але і фіз. активності;
- + наявність в інформації про продукти досить специфічних параметрів;
- + широкі можливості в самоконтролі (свої рецепти, вправи);
- + можливість вести свій щоденник, мати друзів;
- велика база продуктів і вправ на ділі абсолютно марні, так як вільно додаються спільноту, без модерації;
- локалізація на російську мову сумнівної якості;
- частина функцій програми платні.

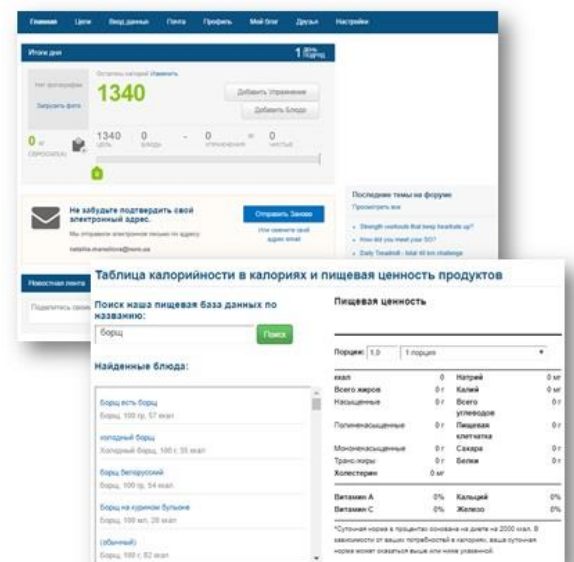


Рисунок А.4 - Слайд №4

Постановка задачі



Рисунок А.5 - Слайд №5

Математична модель

Цільова функція

$$K = |\bar{C} - \bar{B}| - Q = \sum_{i=1}^p (c_i - b_i)^2 - \sum_{j=1}^n q_j x_j \longrightarrow \min_{\bar{X}}$$

Виходячи з фізіологічних норм, на напрямних векторах \bar{X} можливе накладення обмежень:

$$0 \leq x_j \leq k_j,$$

де константи k_j відомі і задаються.

Рисунок А.6 - Слайд №6

Структура даних

```
_id: 119
sourceLink: "http://daily-menu.ru/dailymenu/recipes/view/119"
recipeName: "Блины классические тонкие"
imageLink: "http://daily-menu.ru/public/modules/dailymenu/dailymenurecipes/119/ec1..."
totalWeight: 586
totalKcal: 1142
totalProteins: 34
totalFats: 36
totalCarbohydrates: 173
weightPerPortion: 147
kcalPerPortion: 286
proteinsPerPortion: 8.4
fatsPerPortion: 9
carbohydratesPerPortion: 43.4
kcalPer100g: 198
proteinsPer100g: 5.2
fatsPer100g: 5.1
carbohydratesPer100g: 32
proteinsPercent: 12
fatsPercent: 28
carbohydratesPercent: 60
cookTime: "40 мин."
advancedCookTime: ""
preparation: "1. Молоко немного подогреть, смешать с кефиром 1:1.
2. Добавить яйцо,..."
products: Array
  - Object
    productName: "Яйцо куриное"
    weight: 50
    kcal: 79
    proteins: 6
    fats: 6
    carbohydrates: 0
```



Дані про рецепт зберігаються у вигляді документа JSON, в якому вказані всі необхідні параметри страви

Рисунок А.7 - Слайд №7

Компоненти системи

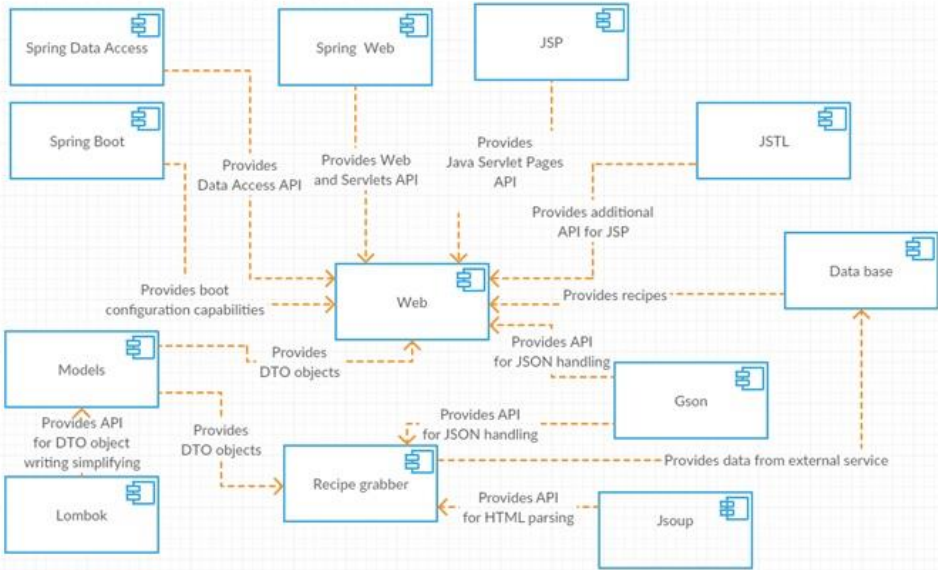


Рисунок А.8 - Слайд №8

Результати роботи

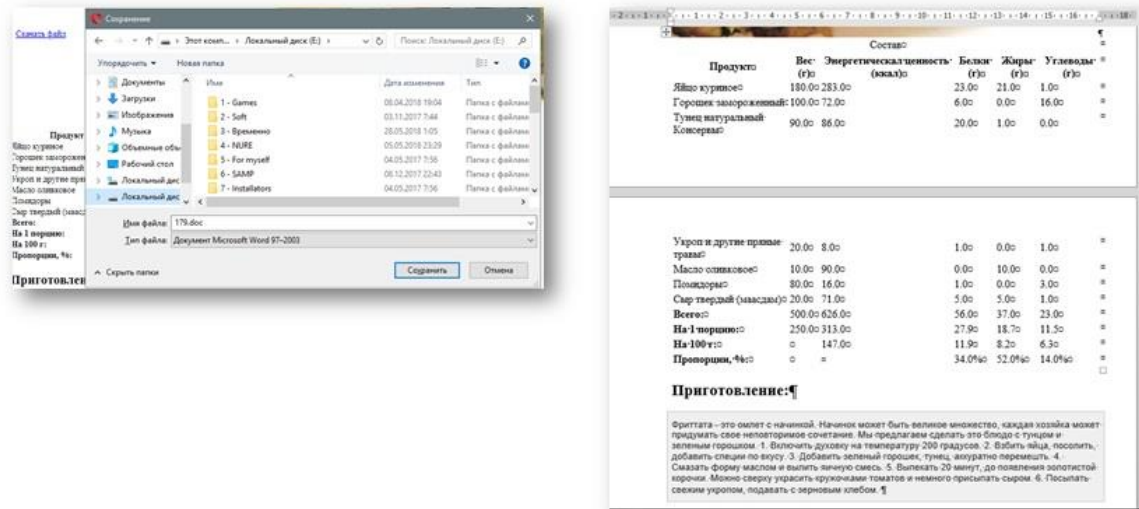


Рисунок А.11 - Слайд №11

Результати роботи

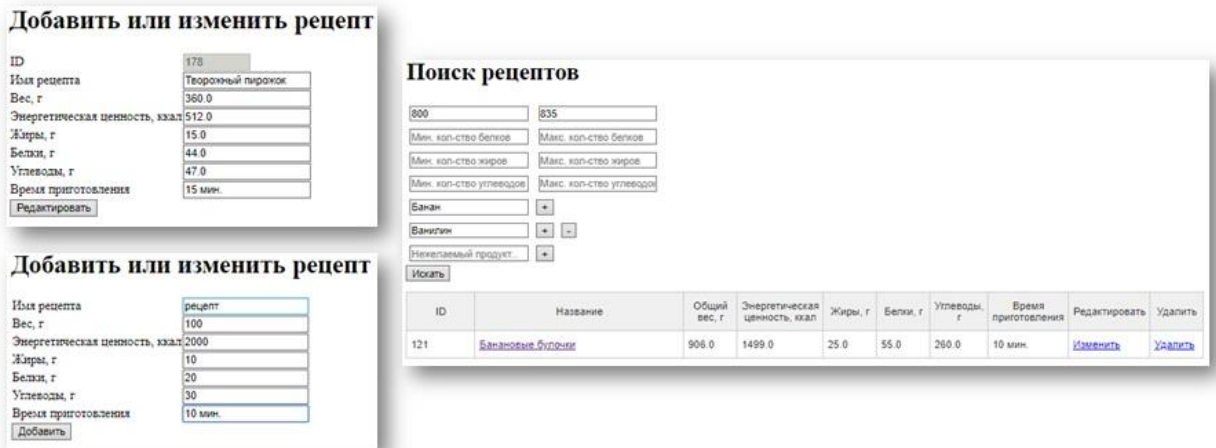


Рисунок А.12 - Слайд №12

Результати роботи

```

carbohydratesPercent: 70
cookTime: "10 мин."
advancedCookTime: "+ 10 минут в духовке"
  "1. Измельчить бананы в блендере.
preparation: 2. Добавить соду, ванилин и яйца.
  3..."
> products: Array

_id: 122
sourceLink: "http://daily-menu.ru/dailymenu/recipes/view/122"
recipeName: "Овсяные коржики"
imageLink: "http://daily-menu.ru/public/modules/dailymenu/dailymenurecipes/122/d86..."
totalWeight: 475
totalKcal: 831
totalProteins: 52
totalFats: 18
totalCarbohydrates: 121
weightPerPortion: 158
kcalPerPortion: 277
proteinsPerPortion: 17.4
fatsPerPortion: 5.9
carbohydratesPerPortion: 48.5
kcalPer100g: 175
proteinsPer100g: 11
fatsPer100g: 3.7
carbohydratesPer100g: 25.6
proteinsPercent: 24
fatsPercent: 19
carbohydratesPercent: 57
cookTime: "10 мин."
advancedCookTime: "+ 35 минут в духовке"
  "1. Включить духовку, разогреть до 180°C, выложить на нее противень.

```

Рисунок А.13 - Слайд №13

Висновки

- розглянута проблема регулювання харчування;
- розглянуті існуючі рішення на ринку та виявлено недоліки;
- розглянута математична модель регулювання харчування;
- розроблений прототип додатка, частково реалізує запланований функціонал;
- в подальшому планується реалізувати особистий кабінет з можливістю сформувати власні переваги і план харчування;

Рисунок А.14 - Слайд №14



Дякуємо за
увагу



Рисунок А.15 - Слайд 15

Додаток Б

DailyMenuRecipeDocumentParser.java

```

1 package ua.nure.manuilova.grabber;
2
3 import ua.nure.manuilova.models.RecipeDto;
4 import ua.nure.manuilova.models.ProductDto;
5
6 import org.jsoup.HttpStatusException;
7 import org.jsoup.Jsoup;
8 import org.jsoup.nodes.Document;
9 import org.jsoup.nodes.Element;
10 import org.jsoup.select.Elements;
11
12 import java.io.IOException;
13 import java.util.ArrayList;
14 import java.util.List;
15 import java.util.stream.Collectors;
16
17 class DailyMenuRecipeDocumentParser {
18     static RecipeDto getRecipe(int id) throws IOException {
19         String url = "http://daily-menu.ru/dailymenu/recipes/view/" +
20 id;
21         Document doc = null;
22         try {
23             doc = Jsoup.connect(url).get();
24         } catch (HttpStatusException e) {
25             if (e.getStatusCode() == 404) {
26                 return null;
27             }
28         }
29
30         RecipeDto recipeDto = new RecipeDto();
31         recipeDto.set_id(id);
32         recipeDto.setSourceLink(url);
33         Elements recipeTitle = doc.getElementsByClass("recipe_title");
34         recipeDto.setRecipeName(recipeTitle.first().ownText());
35
36         Elements image = doc.select(".orange_block > a");
37         recipeDto.setImageLink("http://daily-menu.ru" +
38 image.first().attributes().get("href"));
39
40         Elements recipeCalculationBody =
41 doc.select(".recipe_calculation > tbody");
42
43         Elements ingredients =
44 recipeCalculationBody.first().getElementsByTag("tr");
45
46         for (Element ingredient : ingredients) {
47             ProductDto productDto = new ProductDto();
48             Elements productName =
49 ingredient.getElementsByAttribute("itemprop");
50             productDto.setProductName(productName.text());
51             Elements variables =
52 ingredient.getElementsByClass("variable");

```

```

53         List<String> variablesList = new ArrayList<>();
54         for (Element variable : variables) {
55             variablesList.add(variable.ownText());
56         }
57         productDto.setWeight(parseNumber(variablesList.get(0)));
58         productDto.setKcal(parseNumber(variablesList.get(1)));
59         productDto.setProteins(parseNumber(variablesList.get(2)));
60         productDto.setFats(parseNumber(variablesList.get(3)));
61
62     productDto.setCarbohydrates(parseNumber(variablesList.get(4)));
63     recipeDto.addProduct(productDto);
64 }
65
66     Elements footValues = doc.select(".orange_block >
67 .recipe_calculation_container > .recipe_calculation > tfoot > tr");
68
69     List<String> variablesTotal =
70 parseVariables(footValues.get(0));
71
72     recipeDto.setTotalWeight(parseNumber(variablesTotal.get(0)));
73     recipeDto.setTotalKcal(parseNumber(variablesTotal.get(1)));
74
75     recipeDto.setTotalProteins(parseNumber(variablesTotal.get(2)));
76     recipeDto.setTotalFats(parseNumber(variablesTotal.get(3)));
77
78     recipeDto.setTotalCarbohydrates(parseNumber(variablesTotal.get(4)));
79
80     List<String> dataForOnePortion =
81 parseVariables(footValues.get(1));
82
83
84     recipeDto.setWeightPerPortion(parseNumber(dataForOnePortion.get(0)));
85
86     recipeDto.setKcalPerPortion(parseNumber(dataForOnePortion.get(1)));
87
88     recipeDto.setProteinsPerPortion(parseNumber(dataForOnePortion.get(2)))
89 ;
90
91     recipeDto.setFatsPerPortion(parseNumber(dataForOnePortion.get(3)));
92
93     recipeDto.setCarbohydratesPerPortion(parseNumber(dataForOnePortion.get
94 (4)));
95
96     List<String> dataFor100g = parseVariables(footValues.get(2));
97
98     recipeDto.setKcalPer100g(parseNumber(dataFor100g.get(1)));
99     recipeDto.setProteinsPer100g(parseNumber(dataFor100g.get(2)));
100    recipeDto.setFatsPer100g(parseNumber(dataFor100g.get(3)));
101
102    recipeDto.setCarbohydratesPer100g(parseNumber(dataFor100g.get(4)));
103
104    List<String> percents = parseVariables(footValues.get(3));
105
106    recipeDto.setProteinsPercent(parsePercent(percents.get(2)));
107    recipeDto.setFatsPercent(parsePercent(percents.get(3)));
108
109    recipeDto.setCarbohydratesPercent(parsePercent(percents.get(4)));

```

```

110
111     Elements infoBlocks = doc.select(".info_block > div[style] >
112 div[itemprop]");
113     recipeDto.setCookTime(infoBlocks.get(0).text());
114     recipeDto.setAdvancedCookTime(infoBlocks.get(1).text());
115
116     Element preparationElement =
117 doc.getElementById("recipe_content_block");
118     Elements prepSteps = preparationElement.getElementsByTag("p");
119     StringBuilder preparation = new StringBuilder();
120     for (Element step : prepSteps) {
121         if (!step.text().contains("Смотреть рецепты других ")) {
122
123 preparation.append(step.text()).append(System.lineSeparator());
124             }
125         }
126     recipeDto.setPreparation(preparation.toString());
127
128     return recipeDto;
129 }
130
131 private static List<String> parseVariables(Element element) {
132     return element.getElementsByClass("variable").stream()
133         .map(Element::text)
134         .collect(Collectors.toList());
135 }
136
137 private static Double parseNumber(String number) {
138     return number.contains("-") ? 0 : Double.parseDouble(number);
139 }
140
141 private static Double parsePercent(String number) {
142     return Double.parseDouble(number.substring(0, number.length()
143 - 1));
144 }
145 }

```

Додаток В

Лістинг коду Grabber.java

```

1 package ua.nure.manuilova.grabber;
2
3 import ua.nure.manuilova.models.RecipeDto;
4
5 import com.google.gson.Gson;
6 import com.mongodb.DB;
7 import com.mongodb.DBCollection;
8 import com.mongodb.DBObject;
9 import com.mongodb.MongoClient;
10 import com.mongodb.util.JSON;
11
12 import java.io.IOException;
13 import java.util.ArrayList;
14 import java.util.List;
15 import java.util.Objects;
16 import java.util.stream.Collectors;
17
18 public class Grabber {
19     public static void main(String[] args) throws IOException {
20         MongoClient mongoClient = new MongoClient();
21         DB database = mongoClient.getDB("Akira");
22         DBCollection collection =
23 database.getCollection("Recipies");
24         List<DBObject> recipeDtos = new ArrayList<>();
25         Integer firstNumber = Integer.getInteger(args[0]);
26         Integer secondNumber = Integer.getInteger(args[1]);
27
28         for (int i = firstNumber; i <= secondNumber; ++i) {
29             RecipeDto dto =
30 DailyMenuRecipeDocumentParser.getRecipe(i);
31             recipeDtos.add(convert(dto));
32         }
33
34         collection.insert(recipeDtos.stream()
35             .filter(Objects::nonNull)
36             .collect(Collectors.toList()));
37     }
38
39     private static DBObject convert(RecipeDto recipeDto) {
40         Gson gson = new Gson();
41         String json = gson.toJson(recipeDto);
42         return (DBObject) JSON.parse(json);
43     }

```

