

Державний торговельно-економічний університет
Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка мобільного додатку «Дієтолог»»

Студента 4 курсу, 9 групи,
спеціальності
122 «Комп'ютерні науки»

Соловей
Олександр
Олександрович

_____ *підпис студента*

Науковий керівник
Кандидат технічних наук, доцент

Козлов Валерій
Володимирович

_____ *підпис керівника*

Гарант освітньої програми
кандидат технічних наук, доцент

Демідов Павло
Георгійович

_____ *підпис керівника*

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук та інформаційних систем

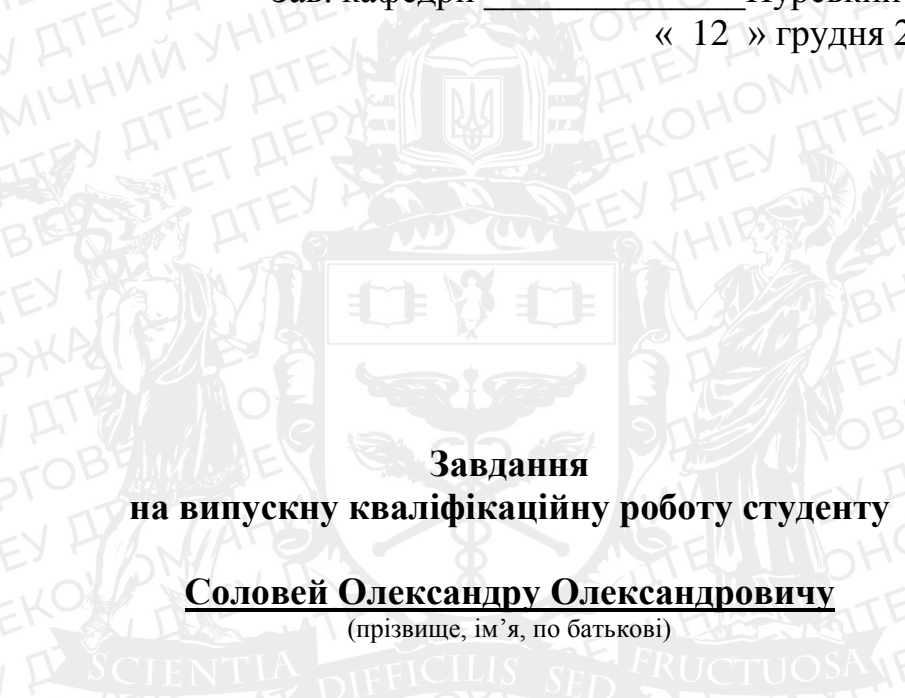
Спеціальність 122 «Комп'ютерні науки»

Затверджую

Зав. кафедри _____

Пурський О. І.

« 12 » грудня 2022р.



Завдання на випускню кваліфікаційну роботу студенту

Соловей Олександр Олександровичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи

«Розробка мобільного додатку «Дієтолог»»

Затверджена наказом ректора від «09» грудня 2022 р. № 3332

2. Строк здачі студентом закінченої роботи 30 травня 2023 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробка мобільного додатку для оптимізації харчування плану харчування клієнта.

Об'єкт дослідження: процеси розробки мобільних додатків.

Предмет дослідження: мобільний додаток для організації правильного харчування.

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Козлов В. В.	15.12.2022 р.	15.12.2022 р.
2	Козлов В. В.	15.12.2022 р.	15.12.2022 р.
3	Козлов В. В.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1. Поняття мобільного додатку та його види

1.2. Переваги та недоліки використання мобільних додатків в області дієтології

1.3. Формування вимог до створення мобільного додатку

Висновки до розділу

РОЗДІЛ 2. ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

2.1. Концепція мобільного додатку "Дієтолог"

2.2. Проектування архітектури мобільного додатку "Дієтолог"

2.3. Вибір інструментів для реалізації мобільного додатку

Висновки до розділу

РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ «ДІЄТОЛОГ»

3.1. Розробка інтерфейсу мобільного додатку

3.2. Розробка алгоритму роботи мобільного додатку

3.3. Тестування мобільного додатку

Висновки до розділу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	04.10.2022	04.10.2022
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	15.12.2022	15.12.2022
3	<i>Вступ</i>	03.02.2023	03.02.2023
4	<i>Розділ 1. Теоретичні основи розробки мобільних додатків</i>	28.02.2023	28.02.2023
5	<i>Розділ 2. Проектування мобільного додатку</i>	06.04.2023	06.04.2023
6	<i>Розділ 3. Розробка мобільного додатку «дієтолог»</i>	12.05.2023	12.05.2023
7	<i>Висновки</i>	15.05.2023	15.05.2023
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	30.05.2023	30.05.2023
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	05-06. 06.2023	05-06. 06.2023
10	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	07.06.2023	07.06.2023
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	15.06.2023	15.06.2023
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.

9. Керівник випускної кваліфікаційної роботи

Козлов В. В.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П. Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Соловей О. О.

(прізвище, ініціали, підпис)

Анотація

Метою цього проекту є розробка мобільного додатку «Дієтолог», який надасть користувачам інструменти для планування та керування їх харчуванням з метою поліпшення здоров'я та досягнення цілей щодо харчування. В додатку можна буде ввести свої дані про вагу, вік, зріст, стать. Після чого користувачам потрібно буде відповісти на запитання стосовно свого харчування. Розробка цього мобільного додатку дасть змогу користувачам узнати коли їм потрібно харчуватись та в якій кількості.

Ключові слова: користувач, мобільний додаток, харчування.

Anotation

The goal of this project is to develop a mobile application "Dietologist", which will provide users with tools to plan and manage their nutrition in order to improve health and achieve nutrition goals. In the application, you can enter your data about weight, age, height, gender. After that, users will need to answer questions about their wish. The development of this mobile application will allow users to know when they need to eat and in what quantity.

Keywords: user, mobile application, nutrition.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ	10
1.1. Поняття мобільного додатку та його види.....	10
1.2. Переваги та недоліки використання мобільних додатків в області дієтології.....	13
1.3. Формування вимог до створення мобільного додатку.....	15
Висновки до розділу.....	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ	21
2.1. Концепція мобільного додатку "Дієтолог".....	21
2.2. Проектування архітектури мобільного додатку "Дієтолог".....	26
2.3. Вибір інструментів для реалізації мобільного додатку.....	30
Висновки до розділу.....	32
РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ «ДІЄТОЛОГ»	33
3.1. Розробка інтерфейсу мобільного додатку.....	33
3.2. Розробка алгоритму роботи мобільного додатку.....	38
3.3. Тестування мобільного додатку.....	45
Висновки до розділу.....	47
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ДОДАТОК	52

ВСТУП

Розробка мобільного додатку "Дієтолог" є актуальним, особливо зростає інтерес до здорового способу життя, правильного харчування та контролю власного здоров'я. Мобільні додатки дієтологів можуть надавати користувачам корисні інструменти для ведення щоденного журналу харчування, розрахунку калорій, планування прийому їжі, надання рекомендацій щодо здорового харчування та інших функцій.

Застосування мобільних додатків у сфері дієтології може допомагати користувачу змінити свої харчові звички, слідкувати за раціоном та досягати своїх здорових цілей. Такі додатки можуть також містити базу даних з інформації про продукти, рецепти, поради щодо вибору продуктів та інші корисні матеріали.

Метою цієї роботи є розробка мобільного додатку для оптимізації харчування та плану харчування клієнта.

Об'єктом розробки мобільного додатку "Дієтолог" є мобільний додаток "Дієтолог". Розробка додатку включає створення програмного забезпечення для мобільних пристроїв таких як смартфон або планшет, яке надає користувачам інструмент та функціональні можливості, пов'язані з дієтологією та здоровим харчуванням. Об'єкт може включати в себе такі складові, як інтерфейс користувача, функціональні можливості, база даних, аналітика та звіт та інтеграція з іншими сервісами.

Предметом розробки мобільного додатку "Дієтолог" є сам процес розробки мобільного додатку, який спрямований на надання користувачам інструментів та функціональності пов'язані з дієтологією і здоровим харчуванням.

Це включає у себе створення програмного забезпечення, визначення функцій, розробку інтерфейсу, розробку бази даних, аналітику та інші аспекти необхідні для створення функціонального додатку.

Такий додаток може надавати користувачам можливість вести щоденник харчування, розраховувати калорії, отримувати рекомендації щодо здорового харчування, планувати прийом їжі, отримувати інформацію про продукти та їх харчові цінності, аналізувати систему свого прогресу та багато іншого. Предмет розробки полягає в створення додатку, який буде корисним інструментом для користувачів у досягненні їх здорових цілей через правильне харчування та дієтологією.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел із 20 найменувань, додатків і містить 42 сторінки основного тексту, 9 рисунків.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1. Поняття мобільного додатку та його види

Мобільний додаток або застосунок - програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних застосунків встановлені на самому пристрої або можуть бути завантажені на нього з онлайн-магазинів мобільних застосунків, таких як App Store, Google Play, Windows Phone Store та інших, безкоштовно або за плату.

Спочатку мобільні застосунки використовувалися для швидкої перевірки електронної пошти, але їх високий попит призвів до розширення їх призначень і в інших областях, таких як ігри для мобільних телефонів, GPS, спілкування, перегляд відео та користування Інтернетом.

Мобільні пристрої можуть виконувати різні функції, такі як доступ до Інтернету, ігри, соціальні мережі, підтримка бізнес-процесів, зберігання даних і багато іншого. Він може бути розроблений для різних операційних систем, таких як Android або iOS, і може працювати в автономному режимі та вимагати підключення до Інтернету.

Приклади функція, які можуть мати мобільні додатки:

1. Соціальні мережі: мобільні програми соціальних мереж дозволяють користувачам реєструватися та створювати профіль, додавати друзів, ділитися інформацією та фотографіями, ділитися статусом тощо.
2. Графічні редактори: мобільні доповнення для графічних редакторів дозволяють користувачам редагувати фотографії та відео, додавати фільтри та ефекти, малювати та створювати колажі.
3. Фітнес-додатки: мобільний додаток до фітнесу не дозволяє

користувачам стежити за своїм фізичним станом, записувати тренування та дієту, встановлювати цілі та стежити за своїми досягненнями.

4. Фінансові додатки: мобільні додатки фінансових послуг дозволяють користувачам керувати своїми фінансами, переказувати гроші, оплачувати рахунки та відстежувати свої транзакції.

5. Ігри: мобільні додатки-ігри не дозволяють користувачам грати в різні ігри, включаючи стратегії, головоломки, аркади, спортивні ігри та інші.

6. Подорожі: мобільні додатки подорожей користувачам бронювати готелі та авіаквитки, розташовувати ресторани та визначні місця, будувати маршрути та планувати дороги.

7. Освіта: мобільний додаток для освіти дає можливість вивчати різні матеріали, або навчатися.

На сьогодні мобільні додатки є важливою складовою нашого заданого життя. Вони дозволяють легко і швидко виконувати різноманітні завдання та отримувати доступ до різноманітних послуг і функцій. Наприклад, ми можемо робити покупки в Інтернеті, спілкуватися з друзями, слухати музику, дивитися фільми та серіали, замовляти їжу доставкою, вивчати інші мови світу, планувати подорожі, стежити за своїм здоров'ям і фізичною активністю і багато іншого. Мобільні додатки дозволяють нам ефективно використовувати час і ресурси, бути більш продуктивними та організованими та покращувати якість життя. Завдяки швидкості та простоті використання мобільні додатки стали невід'ємною частиною нашого життя і з кожним роком продовжують розвиватися та вдосконалюватися.

Існує багато різних видів мобільних додатків, що класифікуються за різними критеріями. За функціональність мобільні додатки можна розділити на такі види:

1. Прикладні додатки - додатки, які допомагають виконувати різні завдання, наприклад, додатки для здоров'я та фітнесу,

метеододатки, фінансові додатки та інші.

2. Навчальні додатки - додатки, які допомагають користувачам вивчати нові мови, підвищувати свою кваліфікацію, отримувати нові знання.

3. Медіадодатки - додатки, які дозволяють користувачам споживати мультимедійний контент, такий як музика, відео, фільми та телевізійні шоу.

4. Додатки для знайомств - додатки, які дозволяють користувачам знаходити нових друзів та партнерів.

5. Додатки для електронної комерції - додатки, які дозволяють користувачам здійснювати онлайн покупки, переглядати товари та порівнювати ціни.

6. Додатки для здоров'я та медицини - додатки, які допомагають користувачам відслідковувати свій стан здоров'я, здійснювати моніторинг показників, контролювати дозування ліків тощо.

7. Додатки для фотографії та відео - додатки, які дозволяють користувачам знімати та редагувати фотографії та відео, створювати колажі, додавати фільтри та ефекти.

Мобільні додатки також можуть бути класифіковані за іншими критеріями, що не пов'язані з їх функціональністю. Приклади мобільних додатків які не пов'язані з функціональністю:

1. Платформа: додатки можуть бути призначені для iOS, Android чи іншої платформи.

2. Розмір команди: додатки можуть бути розроблені невеликою командою або бути продуктом великого підприємства із чисельною командою розробників.

3. Версія: додатки можуть бути оновлюваними і мати різні версії з новими функціями та інтерфейсами.

4. Мова: додатки можуть бути розроблені для певних мов або

мати підтримку багатьох мов.

5. Цільова аудиторія: додатки можуть бути розроблені для конкретної групи користувачів, таких як діти, підлітки, дорослі чи старші люди.

6. Жанр: додатки можуть належати до певного жанру, такого як ігри, соціальні мережі, музика, книги чи новини.

7. Мета: додатки можуть мати різні цілі, такі як розваги, розвиток, освіта, діловість, комунікація тощо.

Ці класифікації мобільних додатків допомагають розуміти різноманітність та складність ринку мобільних додатків. Розуміння різних продуктів і пов'язаних з ними характеристик може допомогти їм краще зрозуміти потреби споживачів і збільшити довіру, на яку вони заслуговують своїми сильними сторонами. Категорії дозволяють користувачам швидко знаходити потрібні додатки та знаходити найкраще рішення для своїх потреб. Ринок мобільних додатків постійно зростає та розвивається, і розуміння типів додатків має вирішальне значення для розробників додатків, маркетологів і користувачів.

1.2. Переваги та недоліки використання мобільних додатків в області дієтології

Мобільні додатки в області дієтології можуть бути корисними для людей, які займаються дієтологією, оскільки вони дозволяють вести облік їжі, калорійності, водного балансу. Переваг використання мобільних додатків у галузі дієтології:

Ефективність і доступність: мобільні пристрої забезпечують легкий і зручний доступ до інформації про їжу та харчування. Користувачі можуть у будь-який час отримати доступ до всієї інформації на своєму смартфоні чи планшеті, що робить процес доступу простішим та зручнішим.

Персоналізація: деякі програми для мобільних пристроїв харчування дозволяють користувачам персоналізувати свій профіль із такою особистою інформацією, як вік, стать, зріст, вага та фізична активність. Щоб вибрати ці дані, додаток може рекомендувати хороші початкові калорії, а також може згадати дієти, які можуть вам підійти у вашому випадку.

Відстеження: мобільні додатки можуть допомогти користувачам вести журнал харчування та відстежувати споживання їжі. Деякі програми пропонують поради щодо здорового харчування та відстежують прогрес користувача.

Освіта: багато програм харчування надають цінну інформацію про здорове харчування та здоровий спосіб життя. Завдяки цим програмам ви можете розробити втручання, які мають відношення до здоров'я, і те, як їх можна використовувати для програм покращення.

А також саме головне в цій області це мотивація багато мобільних додатків в області дієтології мотивують людей, щоб вони вели здоровий спосіб життя та здорове харчування.

Що до недоліків мобільних додатків в області дієтології:

Деякі додатки мають проблему з невідповідністю індивідуальних потреб: мобільні додатки зазвичай не можуть забезпечити індивідуальний підхід до харчування і дієт. Ці додатки можуть надати загальні поради щодо здорового харчування, але не врахувати особливості конкретної людини, такі як алергія та стан здоров'я.

Неналежний контроль якості: існує ризик того, що інформація, надіслана мобільним додатком, є неправильною або неточною. Не всі додатки мають належні процедури забезпечення якості, тому робота з неперевіреними даними може бути проблематичною.

Залежність від технологій: мобільні додатки можуть призвести до технологічної залежності, особливо якщо хтось відмовляється від практики здорового харчування, щоб використовувати програму.

Вартість: деякі мобільні програми вимагають оплати за повну функціональність або преміум-версії. Це може бути фінансовим тягарем для користувачів.

Безпека даних: Мобільні програми можуть збирати та зберігати особисту інформацію про користувачів, таку як вік, стать, стан здоров'я та фізична активність. Якщо ця інформація не захищена належним чином, це може призвести до вторгнення в приватне життя та крадіжки особистих даних. Тому важливо вибирати програми, які забезпечують захист персональних даних.

1.3. Формування вимог до створення мобільного додатку

Однією з найважливіших вимог до створення мобільного додатку є функціональність. Це означає, що додаток повинен працювати без помилок та збоїв і має виконувати всі функції ефективно та швидко. Ще, додаток повинен відповідати потребам користувача та задовольняти його очікування по функціональності.

Для забезпечення правильної функціональності мобільного додатку необхідно провести аналіз вимог користувача та розробити відповідну архітектуру додатку. Та ще необхідно використовувати тестування додатку для вивчення можливих помилок та збоїв їх подальшого виправлення. Існує багато методів тестування мобільних додатків, які допомагають забезпечити високу якість, функціональність та зменшити виникнення помилок та збоїв. Методи тестування мобільних додатків:

Функціональне тестування – цей метод передбачає тестування функціональності програми, щоб перевірити, чи відповідає вона потребам і очікуванням користувачів. Тест проводиться шляхом попереднього введення різних даних і перевірки правильності відповідей.

Навантажувальний тест - цей метод дозволяє визначити роботу в умовах великого навантаження. Це може бути корисно для програм, які

аналізують великі обсяги даних, наприклад для агентів або соціальних мереж.

Перевірка сумісності – у цьому методі перевіряється сумісність програми з різними типами операційних систем і різними типами мобільних пристроїв.

Тестування безпеки даних – цей метод охоплює безпеку програми, її захист від злому та можливість викрадення даних.

Щоб забезпечити високу якість функціональності мобільного додатку вам потрібно використовувати не одну, кілька методів тестування. Також важливо регулярно тестування після випуску додатку на ринок тому що це може вам допомогти виявити та виправити можливі помилки та збої.

Наступна вимога щодо створення мобільного додатку користувацький інтерфейс є також важливим аспектом в створенні мобільних додатків. Мета цього аспекту полягає в тому, щоб забезпечити користувачам легкий доступ до мобільного додатку та допомогти їм зрозуміти, як користуватися додатком. Для досягнення цієї мети потрібно враховувати наступні вимоги:

Привабливість тобто додаток повинен мати привабливий дизайн та зручність у використанні, щоб з'являлися нові користувачі та зберігалась їх зацікавленість.

Простота ця вимога каже що інтерфейс повинен бути простим та зрозумілим для користувачів, щоб він міг легко знаходити необхідні функції та використовувати їх.

Контекстно-залежність цей інтерфейс повинен залежати від контексту використання та показування тільки необхідну інформацією в конкретний момент.

Сумісність тобто мобільний додаток повинен працювати на різних платформах та різних розмірах екрану, щоб забезпечити доступність для якомога більшої кількості користувачів.

Консистентність ця вимога полягає в тому що інтерфейс повинен бути консистентним та стабільним на всіх платформах, щоб користувачі могли

швидко звикнути до його функціональності та використовувати додаток без зайвих зусиль.

Щоб досягти вищої якості інтерфейсу, можна використовувати різні методи тестування, такі як тестування користувачького досвіду, що дозволяє виміряти чи задоволений користувач додатком та інші методи оцінки ефективності та зручності інтерфейсу.

Ще однією з найважливіших вимог до створення мобільного додатку є безпека. Мобільний додаток повинен забезпечувати безпеку користувачів, а також захищати їх конфіденційність даних. Для захисту можуть бути використані різноманітні методи, такі як шифрування даних – це процес перетворення звичайного тексту або даних у форму, яка не може бути зрозуміла без використання спеціального ключа або пароля; двофакторна аутентифікація – це процес перевірки ідентифікації користувача, що вимагає введення двох різних факторів, щоб підтвердити його особу;

Використання безпечних протоколів зв'язку є важливою складовою для забезпечення безпеки мобільного додатку. Безпечні протоколи забезпечують шифрування трафіку між додатком та сервісом, що запобігає несанкціонованому доступу до даних користувача.

Крім того, мобільний додаток повинен відповідати стандартам безпеки даних, які передбачені законодавством у країні, де він використовується. Також, розробники повинні забезпечувати регулярні оновлення додатку для виправлення виявлених уразливостей та підвищити рівень безпеки. При створенні мобільного додатку також необхідно дотримуватись законодавства з питань захисту персональних даних, зокрема забезпечувати можливість користувачам контролювати свої дані та відмовлятися від їх збору та використання.

Ще одна вимога, що до створення мобільного додатку сумісність з різними операційними системами, додаток може бути розроблений з використанням кросплатформових технологій, які дозволяють запускати додаток на різних операційних системах. Крім того, мобільний додаток має

бути протестований на різних пристроях з різними версіями операційних систем, щоб забезпечити його правильність роботи на всіх пристроях.

Наступна вимога оновлення додатку є важливою складовою його розробки. Це дозволяє розробникам додавати нові функції та виправляти помилки, що забезпечує підтримку додатку на довгий термін. Крім того, оновлення можуть виключати в себе покращення безпеки та виправлення вразливостей, що є важливими для захисту даних користувачів. Також система оновлення дозволяє розробникам пристосовувати додаток до змін у операційних системах та апаратному забезпеченні пристроїв. Наприклад, коли випускається нові версії операційної системи, деякі функції додатку можуть бути змінені або взагалі стати несумісними. Оновлення дозволяє розробникам адаптувати додаток до нових умов і зберігати його роботоздатність. Крім того, система оновлення може включати в себе виправлення відомих проблем, що дозволяє підвищити рівень задоволення користувачів і зменшити кількість звернень в службу підтримки.

Ще одним із важливих видів розробки мобільного додатку є аналітика. Вона дозволяє збирати дані про взаємодію користувачів з додатком, такі як час користування, переглянуті сторінки, дії які вони виконують, та багато іншого. Ці дані можуть бути корисними для вивчення того, як користувачі використовувати ці дані для покращення дизайну та функціональності додатку, забезпечення кращого користувацького досвіду та збільшення зацікавленості користувачів. Проте дуже важливо забезпечити захист даних користувачів та дотримання правил конфіденційності при використанні та зборі даних. Для зборів та використання цих даних можуть бути використані різні методи, такі як анонімізація даних – це процес зміни інформації, яка може ідентифікувати користувача на безособові або псевдонімізовані дані, які не містять інформації про конкретного користувача та використання захищених каналів передачі даних воно дозволяє захистити дані користувачів від перехоплення та зловживань з боку зловмисників під час їх передачі через мережу Інтернет.

Наступна вимога до створення мобільного додатку є документація. Вона є дуже важливою складовою додатку, тому що вона допомагає користувачам зрозуміти, як правильно використовувати його функціональність. Документація також може бути у вигляді довідок – це документ, який надає короткий опис функцій та опцій додатку та який може бути корисним для швидкого вирішення найбільш поширених проблем; посібника користувача – це більш розгорнута та додаткова версія довідки, яка зазвичай містить ілюстрації, діаграми та інші зображення, які допомагають користувачам краще зрозуміти як використовувати додаток; тьюторіал – це інструкція або крок-за-кроком навчальний посібник, який демонструє користувачеві як виконувати певні дії в додатку. Також вона повинна бути легкою в доступі та зрозумілою для різних користувачьких груп, включаючи починаючих та досвідчених користувачів. Якщо ви маєте якісну документацію вона сприяє підтримці додатку та зменшенню числа запитів до служби підтримки користувачів.

Технічна підтримка є останньою вимогою до створення мобільного додатку. Розробники мають надавати технічну підтримку користувачам, щоб допомогти вирішувати технічні проблеми та відповідати на запитання. Це може бути забезпечено через електронну пошту, телефонний дзвінок, онлайн-чат або систему квитків для звернень користувачів. Важливо щоб розробники відповідали на запити користувачів як найшвидше та давали якісну технічну підтримку.

Висновок до розділу

Розробка мобільних додатків повинна бути не тільки функціональністю, але й забезпечувати безпеку та зручність використання для користувачів. В цьому розділі було наведено декілька важливих аспектів, які повинні бути враховані при розробці додатку. Серед них безпека даних, захист від зловмисників, сумісність з різними операційними системами та версіями пристроїв,

наявність системи оновлення та аналітики, а також документація та технічної підтримки для користувачів. Врахування вище згаданих аспектів може допомогти зробити мобільний додаток більш зручним та доступним для користувачів, а також забезпечити його безпеку та стабільність у роботі. Проте важливо пам'ятати, що розробка мобільних додатків – це постійний процес, тому потрібно забезпечувати підтримку та оновлення для забезпечення оптимального функціонування та задоволення потреб користувачів



РОЗДІЛ 2

ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

2.1. Концепція мобільного додатку "Дієтолог"

Концепція мобільного додатку "Дієтолог" може бути орієнтована на допомогу користувачам у збалансованому харчуванні та підтримці здорового способу життя. Як правило для створення мобільного додатку використовують основні функції які можуть включати в себе:

Створення персоналізованого харчового плану на основі введеного користувачем профілю може бути досить складним процесом, але він є важливим для забезпечення ефективності та успішності дієти. Щоб створити персоналізований харчовий план, додатку "Дієтолог" потрібно буде враховувати такі фактори:

1. Стать, вік, зріст та вагу користувачів ці дані можуть бути введені користувачем в профілі додатку "Дієтолог" та використовуватись для розрахунку оптимальної кількості калорій та поживних речовин, які потрібні для забезпечення правильного функціонування організму.
2. Рівень активності для користувачів додаток "Дієтолог" може запитувати який рівень фізичної активності користувача та використовувати цю інформацію для врахування витрат калорій на вправи та рух.
3. Здоров'я та медичні показники мобільний додаток може запитувати про наявність хронічних захворювань або медичних показників. Щоб врахувати їх при створенні харчового плану та рекомендаціях.
4. Харчові переваги та обмеження мобільний додаток "Дієтолог" може запитати у користувача про харчові переваги та обмеження, такі як алергія на певні продукти, вегетаріанство, веганство тощо, щоб врахувати ці обмеження в харчовому плані.

Після того як мобільний додаток отримає ці дані, він зможе використовувати розумні алгоритми для створення персоналізованого харчового плану, який враховуватиме вказані фактори та буде оптимальною для досягнення мети користувача.

Можливість вибору типу дієти та урахування її у харчовому плані є важливим функціоналом для додатку "Дієтолог". Кожен тип дієти має свої характеристики та обмеження, тому важливо, щоб харчовий план був створений з урахуванням обраних користувачем налаштувань.

Для цього мобільний додаток може мати вбудовану базу даних різних типів дієти, таких як вегетаріанська, кето, палео, які можуть бути доступні для вибору користувачем у своєму профілі. Після вибору типу дієти, мобільний додаток зможе автоматично змінювати харчовий план, щоб він відповідав вимогам обраної дієти.

Наприклад, якщо користувач вибере вегетаріанську дієту, то додаток може враховувати відсутність м'яса та риби в харчуванні та запропонувати продукти, які містять достатньо білків та інших необхідних для здорового харчування поживних речовин. У разі вибору кето-дієти, додаток може допомагати користувачеві зменшити кількість вуглеводів та збільшити кількість жирів у харчуванні.

Таким чином можливість вибору типу дієти та її урахування у харчовому плані може допомагати користувачам досягати своєї мети з максимальною ефективністю та зручності.

Наступна функція моніторинг харчування є дуже важливою функцією додатку "Дієтолог", оскільки він дозволяє користувачеві контролювати своє харчування та досягти поставлених цілей. Для цього мобільний додаток може мати вбудовану базу даних продуктів, включаючи інформацію про калорії, білки, жири та вуглеводи.

Користувачі матимуть можливість моніторити своє харчування за допомогою сканування штрих-кодів продуктів за допомогою камери свого смартфона або вручну вводячи їх у додаток. Після цього, додаток може

показати користувачу інформацію про кількість калорій, білків, жирів та вуглеводів яку користувач отримає з цього продукту.

Крім того користувач може вести щоденник харчування, додавати в нього продукти, які він спожив, та відстежувати свій раціон на протязі дня, тижня або місяця. Мобільний додаток також може аналізувати дані щодо харчування та запропонувати користувачеві рекомендації щодо досягнення своїх цілей.

Отже, моніторинг харчування за допомогою сканування штрих-кодів продуктів або вводячи їх у додатку "Дієтолог", а також ведення щоденника харчування є корисними функціями мобільного додатку які допоможуть користувачеві здійснювати контроль та досягати своїх цілей щодо харчування.

Ще однією функцією концепції мобільного додатку є розрахунок калорій та поживної цінності до кожного прийому їжі та загального показника на день є важливою функцією для мобільного додатку "Дієтолог". Для цього мобільний додаток може використовувати введені користувачем дані про його стать, вік, зріст, вагу та рівень активності, а також типи дієти, який він обрав.

Додаток може розраховувати кількість калорій, білків, жирів та вуглеводів, яку потрібно спожити за день, та поділяти цю кількість на окремі прийоми їжі такі як сніданок, обід, вечеря, перекус з урахуванням рекомендації дієти. Користувач може ввести інформацію про те, що він з'їв, а додаток автоматично розрахує кількість калорій та поживної цінності кожного прийому їжі.

Додаток може також розраховувати загальний показник калорій та поживної цінності які користувач споживає за день. Це дозволяє користувачеві контролювати своє харчування та вести облік кількості калорій та поживних речовин, які допоможуть користувачеві контролювати своє харчування та досягати своїх цілей щодо здорового способу життя.

Отже, розрахунок калорій та поживної цінності для кожного прийому їжі та загальний показник на день є корисними функціями додатку "Дієтолог" які допоможуть користувачеві контролювати своє харчування та досягати своїх цілей щодо здорового способу життя.

Функція доступу до рецептів та порад від дієтолога з можливістю зберігання улюблених рецептів у додатку є корисною для користувачів які хочуть розширити свій кулінарний досвід та знайти рецепти, що відповідають їхній дієті.

Мобільний додаток може мати базу даних з рецептами які можуть бути відфільтровані за типом дієти та складом. Користувач може знайти рецепт, який відповідає його дієті та вибраним складникам. Крім того, додаток може надати поради від дієтолога щодо здорового харчування та підтримки дієти.

Користувач може зберігати улюблені рецепти у додатку та створювати всій власний кулінарний щоденник. Це дозволяє користувачеві зберігати які він хоче випробувати в майбутньому, та швидко знаходити їх у додатку.

Отже, функція доступу до рецептів та порад від дієтолога з можливістю зберігання улюблених рецептів у додатку є корисною функцією яка дозволяє користувачеві розширити свій кулінарний досвід та допомагає йому досягати своїх цілей щодо здорового способу життя.

Можливість планування та стежити за водним балансом користувача є важливою функцією в мобільному додатку "Дієтолог". Вода відіграє важливу роль у функціонуванні нашого організму, тому важливо підтримувати оптимальний рівень гідратації.

Додаток матиме можливість користувачу вводити кількість води, яку він споживає протягом дня, та відстежувати скільки води було спожито. Крім того, додаток може надати користувачеві інформацію про те, скільки води потрібно випити, в залежності від його статі, віку, ваги та рівня активності.

Мобільний додаток може також нагадувати користувачеві про необхідність пити воду впродовж дня та допомагати встановлювати цілі для водного балансу. Таким чином, користувачеві можна планувати та стежити

за своїм водним балансом, щоб забезпечити оптимальний рівень гідратації та досягти своїх цілей щодо здорового способу життя.

Отже, можливість планування та стеження за водним балансом користувача є корисною функцією в додатку "Дієтолог" яку допомагає користувачеві підтримувати свій гідратаційний баланс та досягати своїх цілей щодо здорового способу життя.

Функція нагадування про час прийому їжі та вживання води є корисною в мобільному додатку "Дієтолог". Ця функція допомагає користувачам створювати здорові звички щодо їжі та гідратації, дотримуватися графіка харчування та забезпечувати оптимальний рівень гідратації.

Додаток може дозволяти користувачам встановлювати нагадування про час прийому їжі та води, щоб допомогти їм дотримуватись режиму харчування та вживання води. Крім того, додаток може надавати рекомендації щодо оптимального інтервалу між прийомами їжі та кількості води, яку потрібно випити.

Ця функція може бути особливою корисною для тих користувачів, які мають нездорові звички щодо харчування та гідратації, або для тих, хто працює в офісі або займається іншою сидячою роботою та може легко забути про їжу та воду.

Отже, нагадування про час прийому їжі та вживання води є корисною функцією в додатку "Дієтолог" яка допомагає користувачам дотримуватись здорового способу життя та створювати здорові звички щодо харчування та гідратації.

Крім того додаток може мати функції спільноти, де користувачі можуть обмінюватися досвідом та порадами щодо здорового способу життя, а також дієти та рецепти.

У майбутньому можуть додаватися нові функції, такі як інтерактивний калькулятор індексу маси тіла, можливість зв'язку з дієтологами онлайн, щоб

користувачі могли отримувати персоналізовані поради та консультації з питань харчування та дієти.

Інтеграція із фітнес-додатками також може бути корисною для користувачів, які хочуть поєднати правильне харчування з регулярними фізичними вправами. Наприклад додаток може інтегруватися із сховищем даних занять, щоб дати користувачеві можливість відстежувати їхні досягнення в області фізичної активності та розвитку.

2.2. Проектування архітектури мобільного додатку "Дієтолог"

Архітектура мобільного додатку "Дієтолог" складається з таких складових частин як:

Клієнтський додаток для мобільних пристроїв може бути розроблений на основі платформ Android та iOS. Функціональні можливості додатку мають бути реалізовані через API, що забезпечить взаємодію з серверною частиною. Також існують основні компоненти клієнтського додатку:

1. Екран входу або реєстрації – ця функція дозволить користувачеві увійти в систему або зареєструватися як новий користувач.
2. Профіль користувача – він дозволить користувачу ввести та зберігати особисту інформацію, таку як стать, вік, зріст, вагу та рівень активності, що будуть використовуватися для створення персонального харчового плану.
3. Створення харчового плану – дозволить користувачу створити персональний харчовий план на основі їх профілю та інших відомостей.
4. Моніторинг харчування – функція дозволяє користувачеві вводити інформацію про їжу та напої, які вони споживають, та відстежувати вміст калорій, білків, жирів, вуглеводів що були спожиті.
5. Рецепти – дозволяє користувачу переглядати та зберігати улюблені рецепти в додатку.
6. Планування водного балансу – дозволяє користувачеві вводити

кількість води яку вони вживають та відстежувати її споживання.

7. Нагадування про час прийому їжі і води – ця функція дозволить користувачеві становити нагадування про час прийому їжі і води.

8. Інтерактивний калькулятор – він дозволяє користувачам обчислити свою масу тіла на основі введення даних про вагу та зріст.

9. Зв'язок з дієтологом онлайн – дозволяє користувачеві отримувати підтримку та поради від дієтологів онлайн.

10. Аналіз даних – ця функція дозволить користувачу отримувати аналітичні звіти про харчування, здоров'я і фізичну активність, що допомагає користувачеві розуміти свої потреби та змінювати свої звички на кращі.

Наступною складовою архітектури мобільного додатку є серверна складова яка може бути розроблена з використанням різноманітних технологій, залежно від вимог до швидкості, масштабності, безпеки. Існують основні функції щодо сервісної складової які можуть включати в себе.

Бази даних ця функція забезпечує зберігання та організацію даних, таких як профіль користувача, харчові плани, рецепти та інформацію про їжу та напої.

API ця функція забезпечує взаємодію клієнтського додатку та серверної складової дозволяю клієнтам виконувати запити до сервера та отримувати відповіді.

Система аутентифікації та авторизації ця функція забезпечує безпеку та конфіденційність даних, дозволяє користувачеві входити в систему та отримувати доступ до своїх персональних даних.

Алгоритм створення харчових планів ця функція використовується для створення персональних харчових планів враховуючи інформацію про профіль користувача його цілі та переваги.

Система моніторингу ця функція забезпечує відстеження та аналіз даних про харчування користувачів дозволяє дієтологам та іншим фахівцям відслідковувати прогрес користувачів, надавати поради та рекомендації.

Взаємодія з дієтологами онлайн ця функція дозволить користувачу зв'язуватись з дієтологом та отримувати консультацію та поради в реальному часі.

Однією з ключових складових архітектури мобільного додатку є база даних яка потрібна для функціонування мобільного додатку "Дієтолог". База даних забезпечує зберігання інформації про користувачів їх профілі, харчові рецепти, продукти, створені харчові плани, що використовуються в додатку. Звичайна база даних розгортається на сервері та використовується для зберігання та організації доступу до даних з клієнтського додатку. В залежності від обсягу даних та вимог щодо продуктивності може використовуватись різноманітні бази даних, такі як MySQL, PostgreSQL, MongoDB та інші.

Наступною складовою архітектури мобільного додатку є API (інтерфейс програмування додатків) він забезпечує взаємодію між різними компонентами програмного забезпечення, в тому числі між клієнтським додатком та серверною складовою. API зазвичай використовують для передачі даних між різними системами та програмами що дозволить їм взаємодіяти та обмінюватись інформацією. API може бути реалізований різними способами включаючи REST, SOAP, GraphQL та інші.

Дуже важливою складовою архітектури мобільного додатку є модуль аутентифікації яка потрібна для забезпечення безпеки даних та аутентифікації користувачів в мобільному додатку "Дієтолог". Цей модуль зазвичай включає в себе функції, такі як перевірка ідентифікаційних даних користувачів, зберігання та перевірка токенів авторизації, обробка запитів на доступ до конкретних ресурсів додатку в залежності від прав користувачів та інші функції які забезпечують безпеку даних та захист від несанкціонованого доступу до них.

Наступною ключовою складовою архітектури мобільного додатку "Дієтолог" є модуль обробки даних. Його завдання полягає в тому, щоб обробляти дані введені користувачами, а також наданні інформації про їхнє харчування та дієти. Модуль може використовувати алгоритми машинного навчання, щоб надавати рекомендації щодо харчування та дієти, на основі введення користувачем даних та даних з бази даних. Наприклад, модуль може розраховувати кількість калорій, які користувач повинен споживати в день, та рекомендувати продукти та страви, що містять необхідну кількість харчових речовин. Крім того модуль може забезпечувати функції, такі як розрахунок калорійності страви та складання харчових планів на основі потреби користувача.

Наступною складовою архітектури мобільного додатку є модуль сканування продуктів. Він є важливою складовою додатку "Дієтолог", так як він дозволяє користувачеві легко та швидко вводити дані про продукти які він споживає, та отримувати точні рекомендації щодо дієти. Крім того, модуль може бути інтегрований з базою даних продуктів щоб допомогти користувачам відстежувати їхнє харчування та отримувати більш точні рекомендації щодо дієти. Модуль сканування продуктів може також допомагати користувачам відстежувати їхню калорійність та отримувати статистику про споживання продуктів на їхньому мобільному пристрої.

Наступною складовою архітектури мобільного додатку "Дієтолог" є модуль нагадувань. Його завдання полягає в наданні користувачам нагадувань про час прийому їжі та вживання води. Це допомагає користувачам відстежувати свої раціони та вживання рідини, а також підтримувати регулярний розпорядок дня. Модуль може бути налаштований користувачем на основі їхніх індивідуальних потреб та графіку дня. Він також може надавати користувачам рекомендації щодо частоти та кількості прийому їжі на основі їхньої дієти та інших факторів.

Наступною важливою складовою архітектури мобільного додатку "Дієтолог" є модуль рецептів. Він надає користувачам доступ до рецептів та

порад від дієтолога, які допомагають їм збалансувати свою дієту та досягти своїх цілей з харчування. Модуль може містити рецепти для різних типів дієти, включаючи вегетаріанську, веганську, безглютенову. Крім того, може надавати поради щодо приготування здорових страв та варіанти заміни несантажних інгредієнтів у рецептах.

Наступною складовою архітектури мобільного додатку "Дієтолог" є модуль планування харчування. Він дозволяє користувачам створювати персоналізовані харчові плани на день, тиждень чи місяць з урахуванням їхніх харчових звичок та обмежень. Модуль може надавати рекомендації щодо оптимального розподілу калорій на прийоми їжі, враховуючи потреби користувача в залежності від його статі, віку, зросту, ваги та рівня фізичної активності. Крім того, модуль може містити базу даних рецептів та інгредієнтів що дозволяє користувачам складати плани харчування на основі певних страв та продуктів, які є в наявності. Це допомагає користувачам планувати своє харчування на певний період часу та забезпечувати своє тіло необхідними поживними речовинами.

2.3. Вибір інструментів для реалізації мобільного додатку

Вибір інструментів для реалізації мобільного додатку залежить від деяких факторів таких як бюджет, потреби продуктивності, масштабованості, складності та інші. Далі будуть наведені деякі інструменти які можуть використовуватись для створення мобільного додатку:

Одним із способів реалізації мобільного додатку є Native Development, який передбачає розробку окремого додатку для кожної мобільної платформи (IOS та Android) з використанням мов програмування, що специфічні для кожної платформи таких як Swift, Objective-C, Java та Kotlin для Android. Цей підхід дозволяє розробникам максимально використовувати можливості та функціональність кожної платформи, а також підтримувати найвищий рівень продуктивності та швидкодії додатку. Однак цей підхід може вимагати

більше часу та зусиль від розробників, а також вищих витрат на розробку та підтримку додатку для кожної платформи окремо.

Ще одним із способів реалізації мобільного додатку є Cross-platform development. Цей підхід дозволяє розробляти додаток один раз, використовуючи спільний код, і після цього запускати його на різних платформах, таких як IOS та Android, За допомогою інструментів таких як React Native, Xamarin, Flutter та інші, розробники можуть швидко створювати мобільні додатки які працюють на різних платформах зберігаючи при цьому однаковий вигляд та функціональність. Такий підхід дозволяє економити час та зусилля, що зазвичай потрібно витратити на розробку двох окремих додатків для різних платформ. Однак є деякі обмеження та компроміси які пов'язані з використанням Cross-platform development. Наприклад можуть виникнути питання з продуктивністю та обмеженнями функціональності на деяких платформах.

Ще одним зі способів реалізації мобільного додатку є Hybrid development. Фреймворки такі як Cordova та Ionic, дозволяють розробникам створювати мобільні додатки з використанням веб-технологій таких як HTML, CSS, JavaScript і збирати їх в пакети для різних платформ. Такий підхід може бути корисним для команди з невеликою кількістю розробників або для тих хто хоче скоротити час на розробку для кількох платформ. Однак є деякі обмеження в гібридній розробці такий як менша продуктивність порівняно з нативною розробкою та більш обмежені можливості для доступу до функцій платформ.

Ще одним варіантом розробки мобільного додатку це використання No-code або Low-code development, які дозволяють створювати додатки без потреби у глибоких знаннях програмування. Ці інструменти зазвичай мають графічний інтерфейс та дозволяють використовувати вбудовані шаблони функції та компоненти для швидкої розробки додатку. Вони можуть бути корисні для швидкої розробки прототипів або простих додатків, але можуть бути обмежені в можливості порівняно з традиційними методами розробки.

Але вибір інструментів залежить від конкретних потреб та обмежень проекту, а також від вміння розробника та команди. Для мобільного додатку "Дієтолог" можна використати React Native або Flutter для забезпечення крос-платформенності та високої продуктивності додатку.

Висновки до розділу

Концепція мобільного додатку "Дієтолог" має включати такі функції, як модуль сканування продуктів, модуль нагадування, модуль рецептів, модуль планування харчування та інші, які допоможуть користувачам відстежувати своє харчування та отримувати рекомендації щодо дієти. Крім того, дизайн додатку повинен бути простим та зрозумілим для користувачів.

Проектування архітектури мобільного додатку "Дієтолог" має бути детально продумане та забезпечувати високу продуктивність та ефективність додатку. Розробники повинні враховувати функціональні та технічні вимоги додатку, такі як швидкість та безпека, та вибирати оптимальні інструменти для реалізації.

Вибір інструментів для реалізації мобільного додатку може включати різні варіанти, такі як native development, cross-platform development, hybrid development, no-code/low-code development та інші. Кожен із цих варіантів має свої переваги та недоліки, які потрібно враховувати при виборі. Розробники можуть використовувати різні інструменти залежно від своїх потреб та вимог клієнта.

Отже, для успішної реалізації проекту мобільного додатку "Дієтолог" необхідно враховувати вимоги та потреби користувачів, обрати оптимальні інструменти для реалізації та детально продумати архітектуру додатку.

РОЗДІЛ 3

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ «ДІЕТОЛОГ»

3.1. Розробка інтерфейсу мобільного додатку

Для початку розглянемо загальні поради для розробки мобільного додатку. Аналіз потреб користувачів цей етап є критичним етапом розробки інтерфейсу додатку "Дієтолог". Для цього варто провести дослідження зібрати інформацію про цільову аудиторію. Кілька кроків які можна виконати:

1. Визначити цільову аудиторію: розглянути для кого призначений наш додаток "Дієтолог". Можливо він призначений для людей які хочуть схуднути, покращити своє здоров'я або вести здоровий спосіб життя. Зробити профіль нашої цільової аудиторії враховуючи вік, стать, місцезнаходження, освіта, професія та стиль життя.

2. Провести анкетування: Розробити опитування або анкету щоб отримати детальнішу інформацію про потреби користувачів. Поставити питання про їхні харчові звички, цілі що до харчування, проблеми, з якими вони зіштовхуються, та що користувачі очікують від додатку "Дієтолог".

3. Вивчити конкуренцію: Дослідити існуючі додатки та платформи які пропонують послуги з дієтології. Проаналізувати їх переваги та недоліки, а також відгуки від користувачів. Це допоможе нам зрозуміти як можна покращити свій додаток та чим саме привернути увагу користувачів.

4. Провести спостереження та інтерв'ю: Спостерігати за потенційними користувачами або провести інтерв'ю, щоб отримати більш детальні відомості про їхні звички, проблеми та очікування від додатку.

Структура та навігація при розробці мобільного додатку "Дієтолог" важливо забезпечити логіну та зручну інформаційну архітектуру, щоб

користувачі могли швидко та легко знаходити необхідну інформацію.

Наведем кілька кроків які можна використати:

1. Визначення головного екрану: Визначим головні екрани які будуть присутні в додатку "Дієтолог". Це можуть бути "Головна сторінка", "Дієти", "Рецепти", "Трекер харчування", "Поради та рекомендації". Подумати про основні функції та інформацію яка повинна бути доступна на кожному екрані.

2. Організація меню та навігаційних елементів: Розглянути варіанти організації меню та навігаційних елементів, щоб забезпечити зручний доступ до всіх функцій та розділів додатку. Для цього можна використовувати бічне меню, нижню навігацію або випадкові списки, Важливо, щоб навігація була логічною та зрозумілою для користувачів.

3. Категоризація дієт та рецептів: Якщо у додатку доступні різні типи дієти або рецептів, можна розглянути їх категоризацію. Наприклад можна розділити дієти на вегетаріанські, кето та для вагітних. Також важливо мати можливість швидко шукати або фільтрувати за категоріями такими як інгредієнти, тип, харчування, калорійність.

Дизайн інтерфейсу для додатку "Дієтолог" важливо створити привабливий та функціональний дизайн який буде залучати користувачів та забезпечувати їх зручність використання. Кілька рекомендацій щодо розробки дизайну інтерфейсу:

1. Простота та зрозумілість: Забезпечити простоту використання додатку розташовуючи елементи інтерфейсу логічно та зрозуміло. Уникати перевантаження екрану зайвими елементами, але впевнитись що всі необхідні функції доступні зручними способом.

2. Інтуїтивність: створити інтуїтивний дизайн який дозволяє користувачам швидко зорієнтуватись у додатку. Використовувати звичні символи та піктограми для кнопок та функцій щоб користувачі могли легко розпізнати їх.

3. Кольорова палітра: Використовувати кольорову палітру що

відповідає темі дієтології та сприяє здоровому способу життя. Наприклад можна використовувати природні та свіжі кольори які асоціюються зі здоровою їжею такі як зелений, оранжевий та жовтий. Також важливо забезпечити достатню контрастність між текстом і фоном щоб поліпшити читабельність.

4. Розташування елементів: Розмістити елементи інтерфейсу зручно та логічно. Основні функції можна розмістити на головному екрані або на видимій панелі навігації щоб користувач міг швидко отримувати доступ до них. Пункти меню та підпункти також слід розташовувати логічно та ієрархічно.

Функціональність при розробці мобільного додатку "Дієтолог" варто враховувати потреби користувачів у галузі харчування та дієти. Кілька основних функцій:

1. Розділи з дієтами: Додавати розділи де користувач зможе дізнатись про різні види дієт їх особливості. Забезпечити короткий опис кожної дієти, список рекомендованих та заборонених продуктів, можливі меню на тиждень.

2. Трекер харчування: Розробити функцію трекера харчування де користувачі зможуть записувати їжу яку вони споживають щодня. Надайте можливість введення кількості та типу харчових продуктів, а також відстежувати калорійність та поживні речовини.

3. Рецепти : Включити розділ з рецептами, де користувачі зможуть знайти інформацію про здорові та смачні страви, Додати можливість пошуку рецептів за категоріями, інгредієнтами або типом дієти. Забезпечити кожний рецепт зображенням та детальними інструкціями.

4. Поради що до здорового способу життя: Надавати користувачам корисні поради та рекомендації стосовно здорового способу життя. Це можуть бути статі про фізичну активність, психологічний стан, засоби боротьби зі стресом та інші аспекти здорового жаття.

Тестування прототипу додатку "Дієтолог" з реальними користувачами та отримання їхнього зворотного зв'язку дуже важливі для поліпшення функціональності та дизайну. Кілька кроків які можна виконати:

1. Внутрішнє тестування: Потати потрібно з внутрішнього тестування додатку зі своєї команди. Попросити колег використовувати додаток і надати свої враження виявлені проблеми та пропозиції щодо поліпшень. Це може допомагати виявити очевидні проблеми та зробити початкові корективи.

2. Зовнішнє тестування з фокус-групами: Залучити малу групу реальних користувачів які відповідають нашій цільовій аудиторії. Запросити їх протестувати прототип додатку та завдавати запитання пов'язані з їхнім досвідом використання. Спостерігати їхні дії та слухати їхні відгуки реагувати на них.

3. Бета-тестування: Розглянути можливість запуску бета-версію додатку для більшої аудиторії користувачів. Запросити зацікавлених користувачів щоб вони випробували додаток та надали свої відгуки. Забезпечити зручний спосіб зворотного зв'язку, через форму зворотного зв'язку або електрону пошту.

4. Аналіз зворотного зв'язку: Потрібно ретельно проаналізувати всі зворотні відгуки та пропозиції від користувачів. Визначити загальні проблеми які потребують уваги та пріоритетності виправлень.

Чіткість та простота є важливим аспектом дизайну інтерфейсу додатку "Дієтолог". Потрібно використовувати мінімальну кількість елементів на екрані, щоб уникати перевантаження. Зосередитись на основних функціях та інформації яка є найважливішою для користувачів. Використовувати чітку ієрархію для організації інформації на екрані. Визначити головні екрани та підекрани розмістити їх в логічному порядку та зробити доступ до них простим та зручним. Розташувати елементи таким чином щоб вони були логічними та легко доступними для користувачів. Використовувати зрозумілі іконки, підписи і кнопки для ясного відображення функціональності.

Постаратися зробити форми взаємодії з додатком простими і зрозумілими. Використовувати зрозумілі мовні повідомлення та інтуїтивні жести. Мінімізувати кількість кроків для виконання завдань. Перевірити додаток на чіткість та простоту використовуючи тестування з реальними користувачами. Спостерігати їхні дії та реакції слухати їхні фідбеки і вносити необхідні виправлення. Потрібно зробити все можливе щоб спростити досвід користувача забезпечити легку навігацію та зрозумілість.

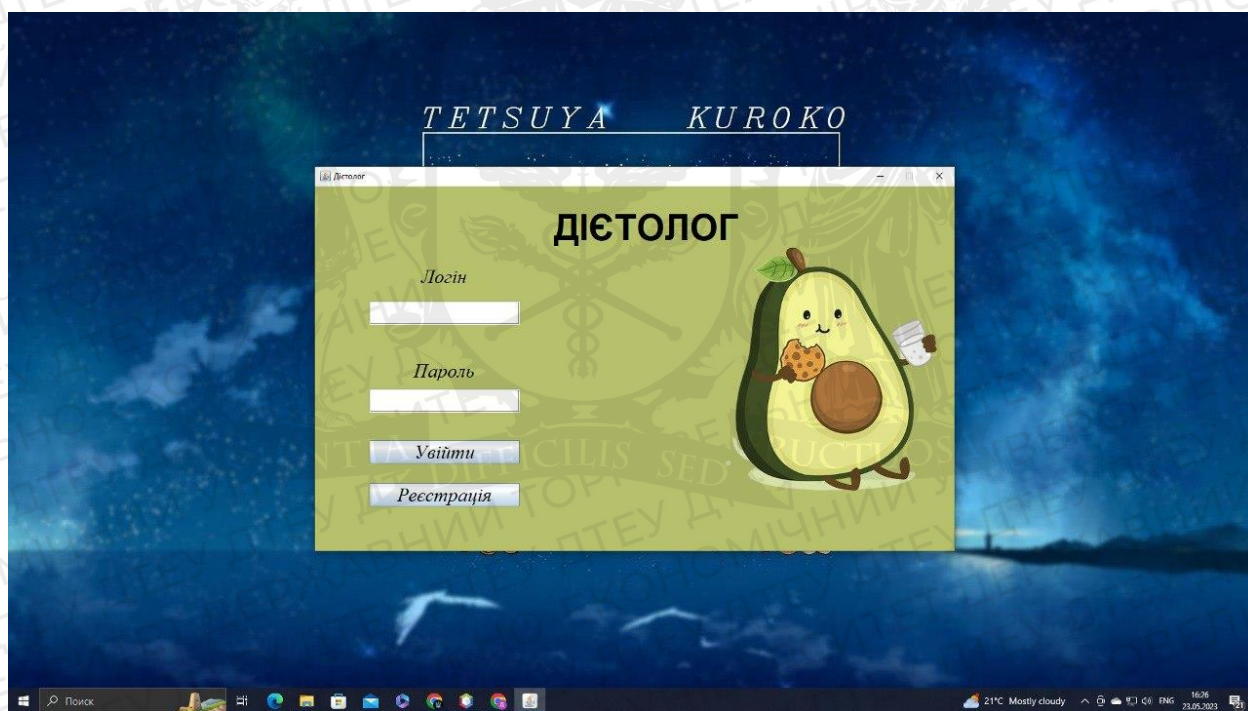
Використання візуальних елементів може значно покращити привабливість мобільного додатку "Дієтолог". Потрібно вибрати кольорову палітру яка відображає концепцію здорового способу життя та харчування. Використовувати світлі та природні відтінки які стимулюють апетит та асоціюються зі здоровими продуктами. Уникати яскравих агресивних кольорів. Використовувати читабельні шрифти які підходять для відображення тексту на мобільних пристроях. Розглянути використання комбінації шрифтів для виділення заголовків, підзаголовків та основного тексту. Використовувати графіку, іконки та ілюстрації які підкреслюють тему дієтології та здорового способу життя. Вони можуть служити для ілюстрації рецептів, продуктів харчування та вправ. Впевнитись, що графіка не заважає сприйняттю інформації та не перевантажує екран. Підтримувати простоту та консистентність використання візуальних елементів. Використовувати однакові стилі, розміри, пропорції та кольори для створення єдиного вигляду додатку. Пам'ятати про адаптивний дизайн який забезпечує оптимальний вигляд додатку на різних розмірах екрану. Перевірити як елементи виглядають на різних мобільних пристроях.

3.2. Розробка алгоритму мобільного додатку

Розробка алгоритму мобільного додатку є важливим етапом у розробці мобільного додатку. Алгоритм – це набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв’язання задачі за скінчену кількість дій.

Для початку ми визначимо вимоги щодо розробки мобільного додатку. Під час спілкування із зацікавленими сторонами, такими як користувачі та замовники важливо збирати і аналізувати їх потреби і очікування щодо функціональності, дизайну, перформансу та інших аспектів додатку. Виконали декілька кроків для визначення вимог: Провели опитування з потенційними користувачами додатку щоб з’ясувати їх потреби, проблеми, цілі та очікування. Запитали як вони бачать функціональність мобільного додатку які функції є най важливішими для них і як вони очікують, що додаток буде взаємодіяти з ними. Після проведення опитування користувачами які користувались додатком були знайдені деякі проблеми з реєстрацією та входом після чого вони були виправлені та користувачі змогли підключитись. Вивчали існуючі мобільні додатки які присутні на ринку та пропонують подібну функціональність. Проаналізували їх переваги та недоліки, щоб зрозуміти як можна вдосконалити свій додаток і забезпечити конкурентні переваги. На підставі зібраних вимог і аналізу конкурентів визначили основні функції які має містити наш додаток. Розробили загальний дизайн інтерфейсу враховуючі зібрані вимоги та потреби користувачів.

При проектуванні інтерфейсу мобільного додатку ми врахували деякі ключові аспекти, щоб забезпечити логічну структуру зручну навігацію та правильне розташування елементів. Виконані етапи проектування інтерфейсу: дослідили цільову аудиторію їхні потреби та очікування щодо додатку. Розуміння цілей та обмежень користувачів допоможе визначити які функції і взаємодії їм потрібно. Створили логічну структуру головного екрану та їх зв'язків. Розробили зручну систему навігації яка допоможе користувачам легко переходити між екранами і функціями додатку. Врахували принцип дизайну, щоб розмістити елементи на екран додатку логічно та зручно для користувачів. Забезпечили належний візуальний



баланс.

Рис. 3.1 Вікно входу

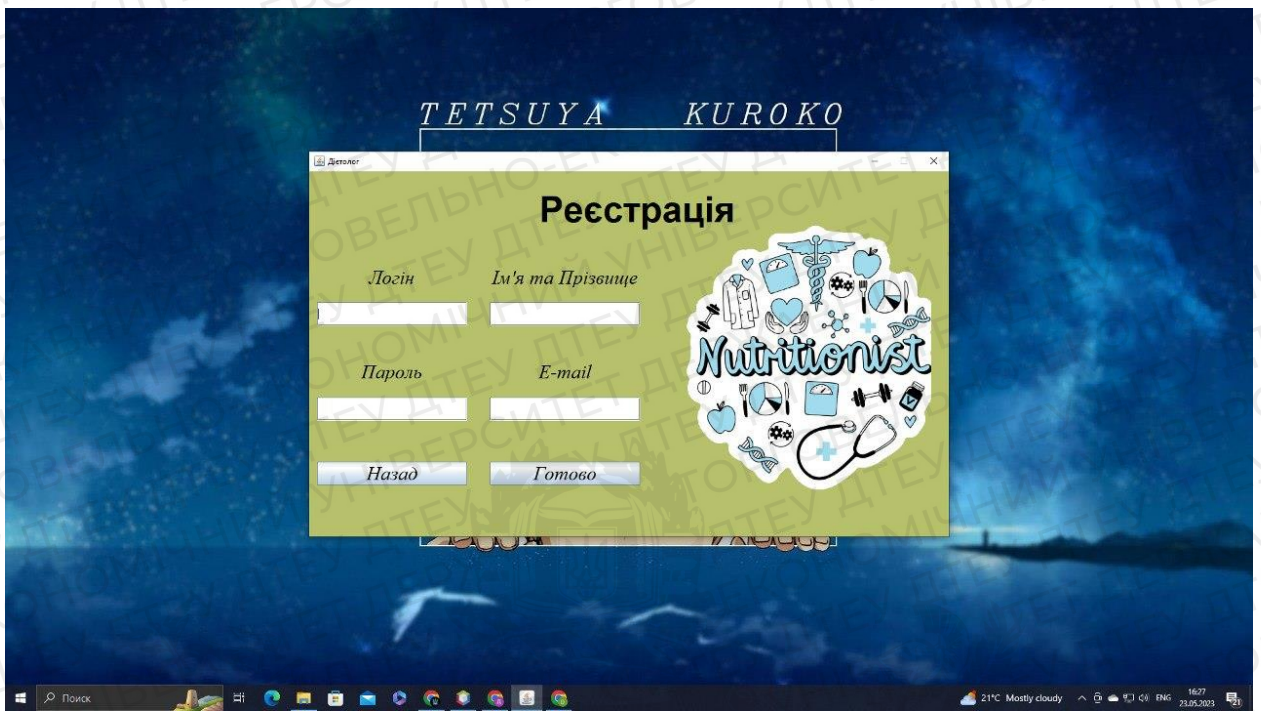
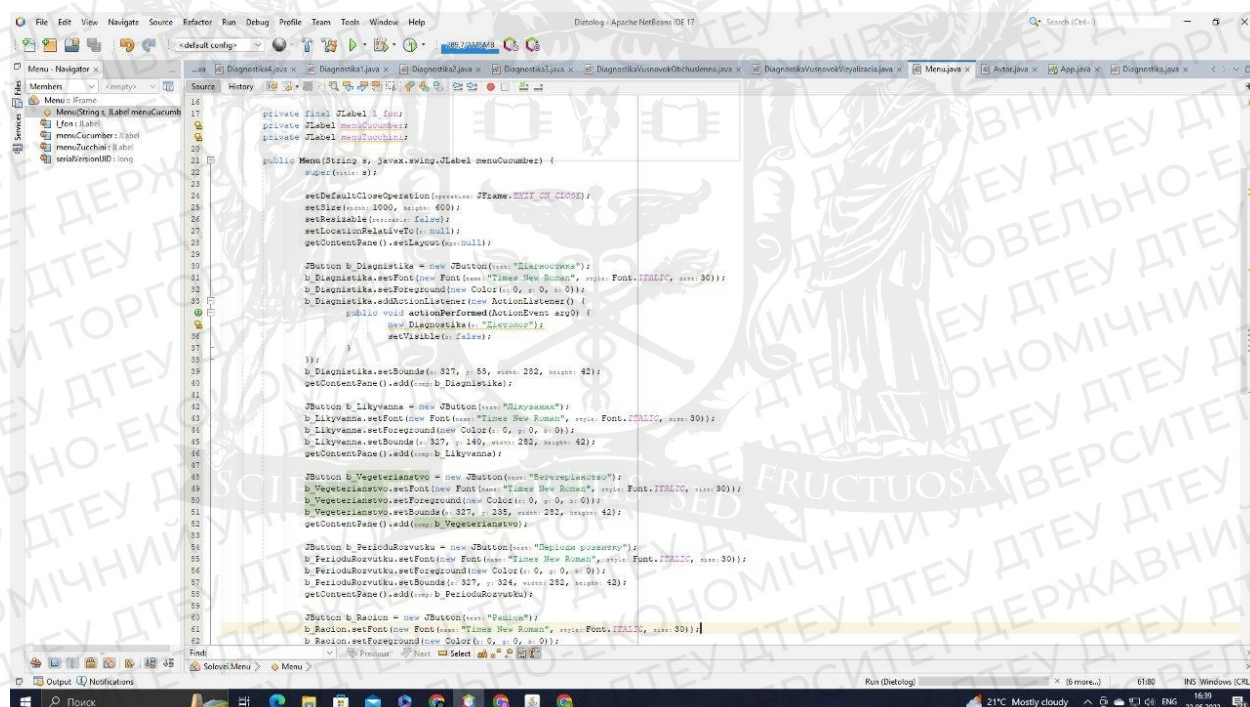


Рис. 3.2 Вікно реєстрації



Рис. 3.3 Вікно головного меню

Розробка функцій мобільного додатку: визначили вимоги до додатку зокрема функції які користувачі очікують від додатку. Розробили детальний опис функціональності кожної функції включаючи вхідні дані взаємодії з користувачами обробку даних та вихідні результати. Написали код, що реалізує функціональність додатку. Що включає в себе розробку класів, методів, функцій та модулів необхідних для виконання запланованих функцій. Вдосконалили функціональність на основі отриманого зв'язку та виявлених помилок. Відлагодили проблеми та виправили помилки у програмному коді.



```
16 private final JLabel l_menu;
17 private JLabel menuDiagnostika;
18 private JLabel menuLikyanna;
19
20 public Menu(String s, Javax.swing.JLabel menuDiagnostika) {
21     super(s);
22
23
24     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25     setSize(1000, 600);
26     setResizable(false);
27     setLocationRelativeTo(null);
28     getContentPane().setLayout(new BorderLayout());
29
30     JButton b_Diagnostika = new JButton("Діагностика");
31     b_Diagnostika.setFont(new Font("Times New Roman", style: Font.PLAIN, size: 30));
32     b_Diagnostika.setForeground(new Color(0, 0, 0));
33     b_Diagnostika.addActionListener(new ActionListener() {
34         public void actionPerformed(ActionEvent e) {
35             new Diagnostika("Diagnostika");
36             setVisible(false);
37         }
38     });
39     b_Diagnostika.setBounds(327, 55, 282, 42);
40     getContentPane().add(b_Diagnostika);
41
42     JButton b_Likyanna = new JButton("Лікьянна");
43     b_Likyanna.setFont(new Font("Times New Roman", style: Font.PLAIN, size: 30));
44     b_Likyanna.setForeground(new Color(0, 0, 0));
45     b_Likyanna.setBounds(327, 140, 282, 42);
46     getContentPane().add(b_Likyanna);
47
48     JButton b_Vegeterianstvo = new JButton("Вегетеріанство");
49     b_Vegeterianstvo.setFont(new Font("Times New Roman", style: Font.PLAIN, size: 30));
50     b_Vegeterianstvo.setForeground(new Color(0, 0, 0));
51     b_Vegeterianstvo.setBounds(327, 235, 282, 42);
52     getContentPane().add(b_Vegeterianstvo);
53
54     JButton b_PerioduRozvutku = new JButton("Період розвитку");
55     b_PerioduRozvutku.setFont(new Font("Times New Roman", style: Font.PLAIN, size: 30));
56     b_PerioduRozvutku.setForeground(new Color(0, 0, 0));
57     b_PerioduRozvutku.setBounds(327, 324, 282, 42);
58     getContentPane().add(b_PerioduRozvutku);
59
60     JButton b_Racion = new JButton("Рацион");
61     b_Racion.setFont(new Font("Times New Roman", style: Font.PLAIN, size: 30));
62     b_Racion.setForeground(new Color(0, 0, 0));
```

Рис.3.4. Частина програмного коду меню

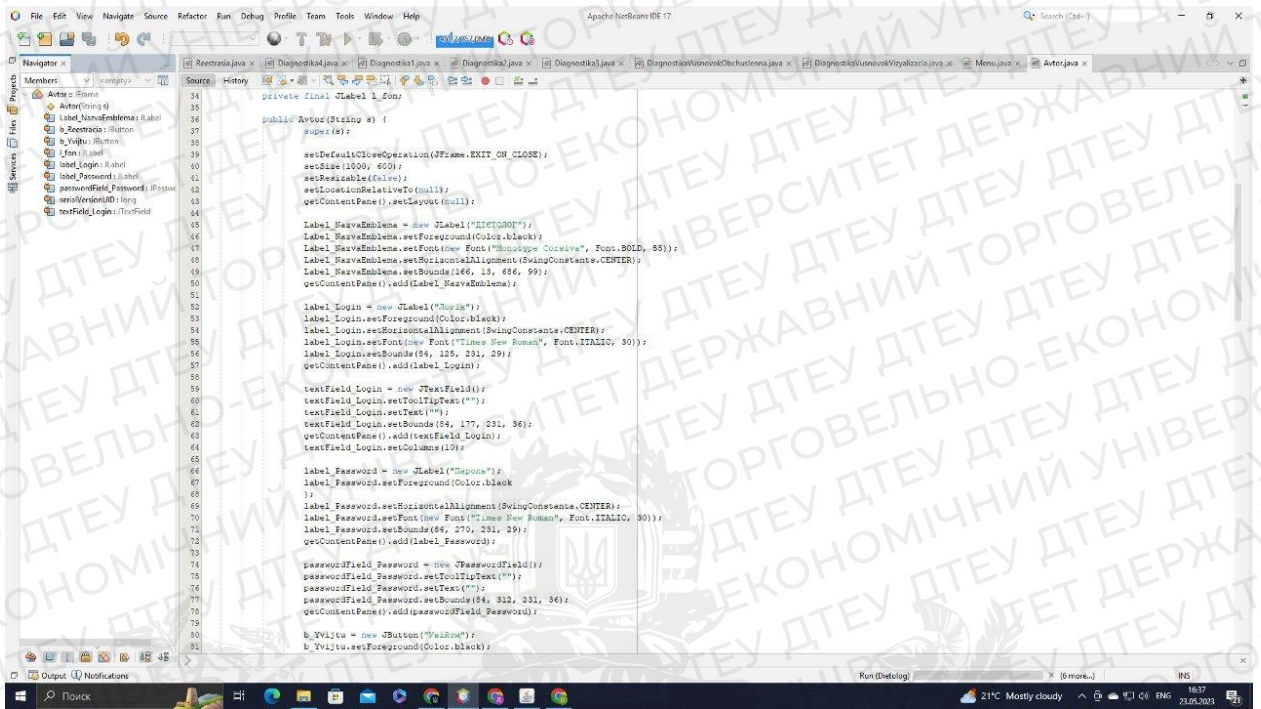


Рис.3.5 Частина програмного коду авторизації

Інтеграція з базою даних є важливим аспектом в розробці мобільного додатку.

Тестування та налагодження є важливим аспектом у розробці мобільного додатку. Для перевірки правильності роботи додатку та виявлення помилок та недоліків. При введенні правильних даних у розділі реєстрація вам буде вибиватись віконце з написом.

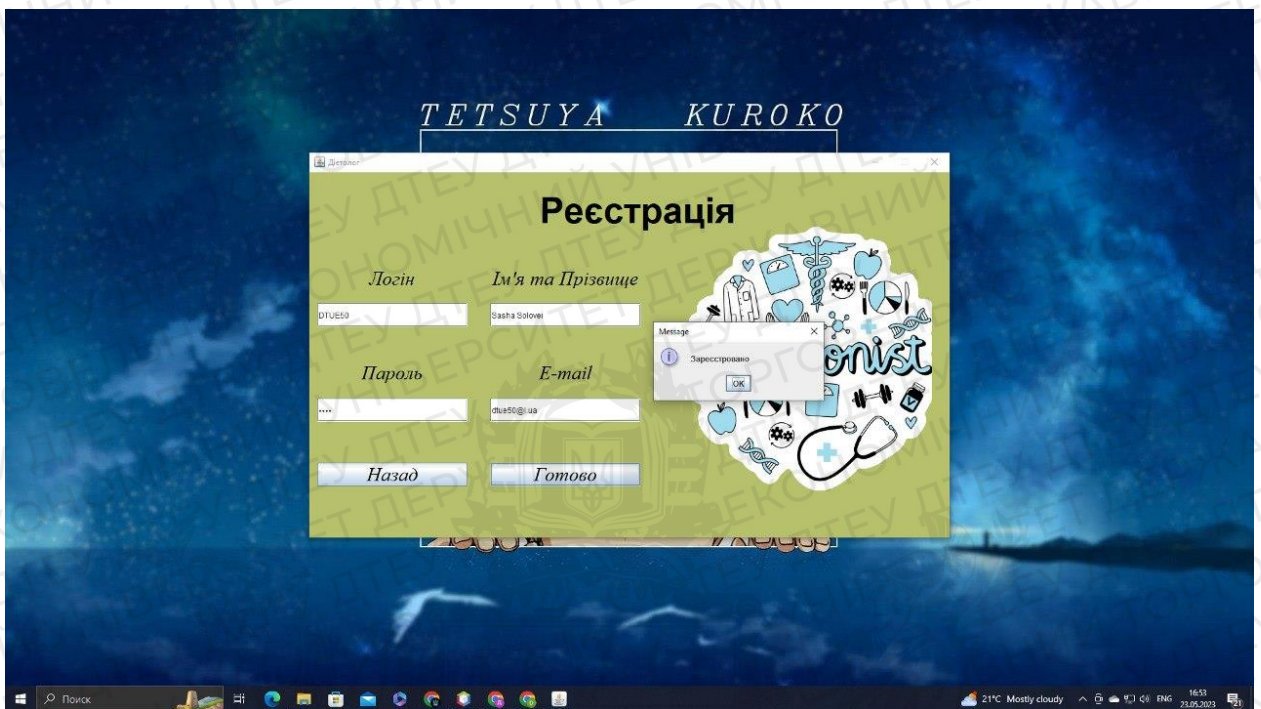


Рис.3.6. «Зареєстровано»

Якщо дані будуть введені некоректно то буде вибиватись віконце з написом.

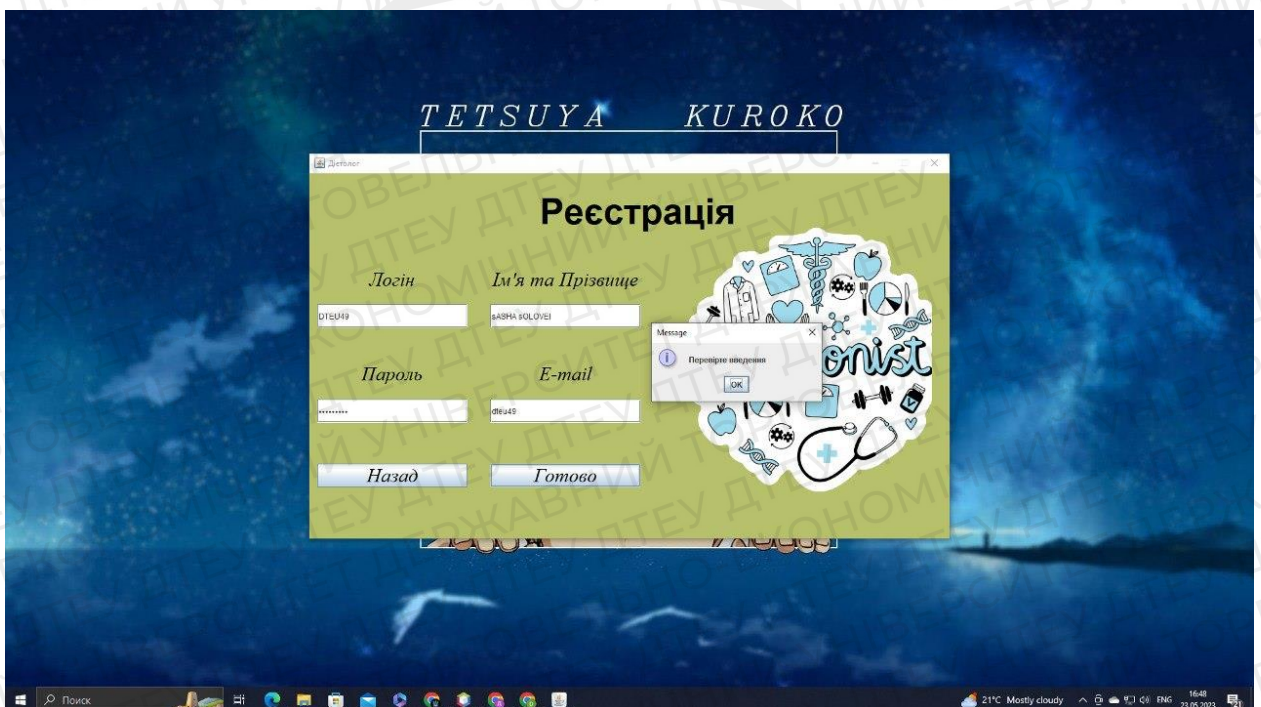


Рис.3.7. « перевірте введення»

При вході якщо дані будуть зареєстровані та коректно введені то вам виб'ється віконце з написом

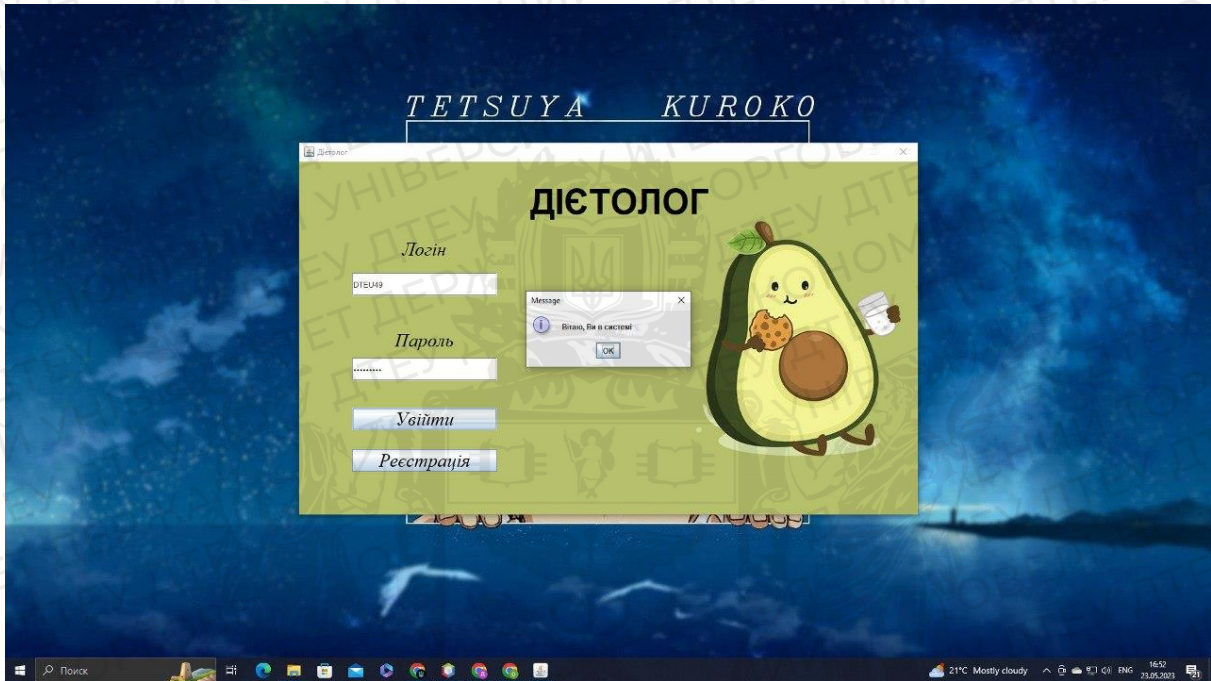


Рис. 3.8. « Вітаєм, Ви в системі»

Після чого вас перекине в головне меню. Якщо дані будуть незареєстровані або введені некоректно то вам виб'ється віконце з написом.

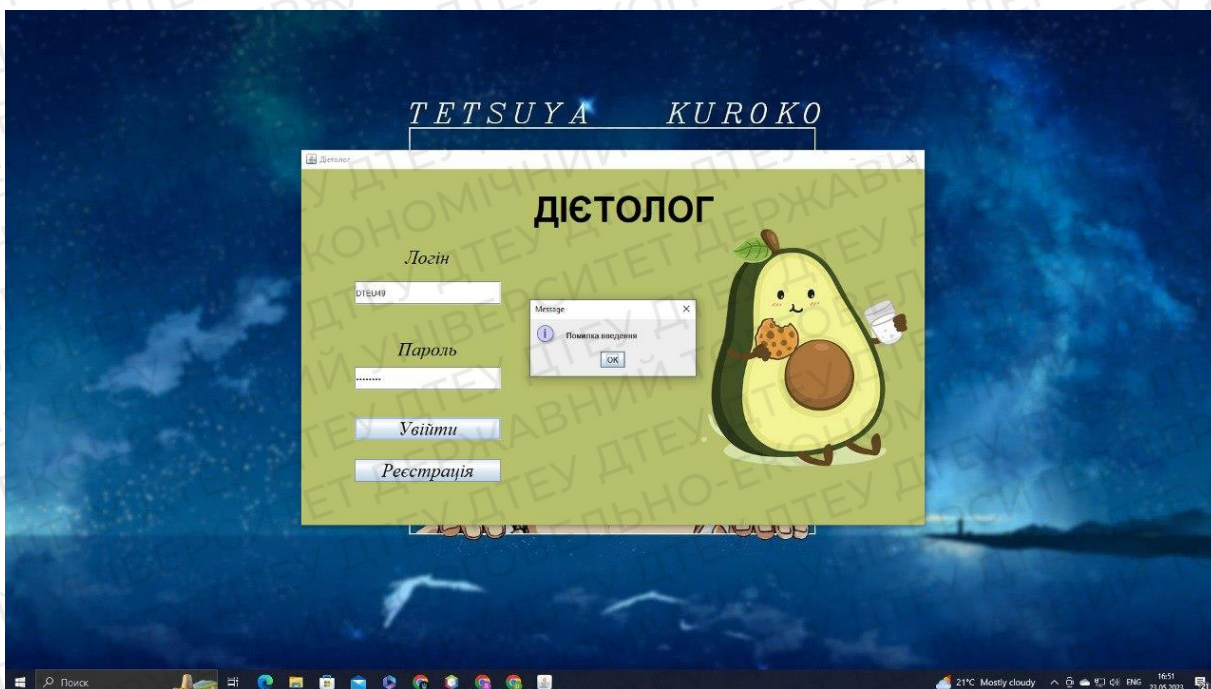


Рис.3.9. « Помилка введення»

Потім вам потрібно буде ввести коректні дані або зареєструватися.

3.3. Тестування мобільного додатку

Тестування мобільного додатку є важливим етапом в розробці оскільки воно допомагає виявити помилки перевірити функціональність та забезпечити високу якість додатку перед його випуском. Є кілька тестувань якими ми протестуємо наш додаток.

Функціонального тестування (Functional Testing) – виявлення невідповідностей між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і вимог. Функціональні тести повинні охоплювати всі реалізовані функції з урахуванням найбільш ймовірних типів помилок. Тестові сценарії, що поєднують окремі тести, орієнтовані на перевірку якості розв'язку функціональних задач.

Функціональні тести створюються за зовнішніми специфікаціями функцій, проєктної інформації і за текстом на МП, що стосуються його функціональних характеристик і застосовуються на процесі комплексного тестування й іспитів для визначення повноти реалізації функціональних задач і їхньої відповідності вхідним вимогам.

Ми протестували основні функції додатку. Переконались що вони працюють правильно і відповідають очікуванням користувачів. Перевірили чи можна додавати, редагувати та видаляти дані які пов'язані з харчуванням та дієтами. Перевірили як користувачі взаємодіють з додатком. Переконались що елементи інтерфейсу працюють правильно і виконують очікувані дії. Перевірили що інформація відображається правильно на екрані. Перевірили чи коректно відображаються тексти, зображення та інші елементи. Перевірили чи відповідають дані які показуються очікування користувачів. Спробували спеціально ввести некоректні дані або виконати неправильні дії

щоб перевірити як додаток оброблює помилки. Переконались що користувачам надаються зрозумілі повідомлення про помилки та інструкції щодо виправлення ситуації. Та перевірили додаток на різних пристроях та операційних системах.

Модульне тестування — це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яка може бути протестованою. У процедурному програмуванні модулем вважають окрему функцію або процедуру. В об'єктно-орієнтованому програмуванні — метод.

Модульні тести, або unit-тести, розробляють в процесі розробки програмісти та, іноді, тестувальники білої скриньки (white-box testers).

Зазвичай unit-тести застосовують для того, щоб упевнитися, що код відповідає вимогам архітектури та має очікувану поведінку.

Перевірили як додаток відображається та працює на різних пристроях, таких як смартфон та планшет з різними екранами. Переконались що елементи інтерфейсу не перекривають тексти читабельність, а функції доступні для використання на різних пристроях. Перевірили як додаток працює на різних операційних системах. Впевнились що додаток працює стабільно і функціонально підтримуваних версіях операційних систем. Перевірили як додаток відображається на пристроях з різною роздільною здатністю екрану. Переконались що інтерфейс адаптується до різних роздільних здатностей і забезпечує зручну навігацію та використання. Переконались що додаток правильно адаптується до різних орієнтацій екрану, як горизонтальної, так і вертикальної. Перевірте, що елементи інтерфейсу зберігають своє положення та читабельність незалежно від орієнтації екрану. Переконались що додаток працює на різних версіях операційних систем які підтримуються.

Тестування інтерфейсу користувача — це процес тестування продукту інтерфейсу користувача, для забезпечення його відповідності до даної специфікації.

Оцінка з користувачем та зручності використання додатку виконали такі кроки. Запросили декілька користувачів які відповідають нашій цільовій аудиторії, протестували додаток. Спостерігали, як вони взаємодіють з інтерфейсом, як швидко вони знаходять потрібну інформацію та виконують необхідні дії. Звертали увагу на будь-які проблеми, затримки або незручності, з якими стикаються користувачі під час використання додатку. Склали список завдань які користувачі можуть виконувати за допомогою додатку. Попросили користувачів виконати ці завдання спостерігаючи їхню взаємодію та реакції. Звертали увагу на будь-які труднощі, незрозумілості або помилки, які виникали під час виконання завдань. Перевірили наскільки легко користувачі можуть переходити між різними екранами та розділами додатку. Переконались, що навігаційні елементи чіткі та доступні з будь-якого місця додатку. Запитали у користувачів чи вони легко орієнтуються в структурі додатку та знаходять потрібні функції.

Для тестування реальних сценаріїв ми використовували такий сценарій. Спробували зареєструватись як новий користувач і увійти в систему зареєстрованим користувачем. Перевірили чи працює реєстрація, чи коректно зберігаються дані користувача і чи успішно відбувається вхід в систему. Переглянули доступні дієти їх опис, складові та рекомендації. Переконались, що інформація про дієти відображається коректно та зрозуміло. Вибрали певну дієту і переконались що вона вибрана правильно і відображається в профілі користувача.

Висновки до розділу

Розробка інтерфейсу мобільного додатку Для початку розглянемо загальні поради для розробки мобільного додатку.

Використовувати звичні символи та піктограми для кнопок та функцій, щоб користувачі могли легко розпізнати їх.

Використання візуальних елементів може значно покращити привабливість мобільного додатку "Дієтолог".

При проектуванні інтерфейсу мобільного додатку ми врахували деякі ключові аспекти, щоб забезпечити логічну структуру зручну навігацію та правильне розташування елементів.

Переконалися що додаток правильно адаптується до різних орієнтацій екрану, як горизонтальної, так і вертикальної.

Запитали у користувачів чи вони легко орієнтуються в структурі додатку та знаходять потрібні функції.



ВИСНОВОК

В ході розробки мобільного додатку "Дієтолог" було виконано багато важливих етапів які включали в себе визначення вимог, проектування інтерфейсу, розробку функціональності, інтеграцію з базою даних, тестування та налагодження.

Проектування інтерфейсу було зосереджено на створенні логічної структури додатку. Зрозумілої навігації та зручному розташуванні елементів. Були визначенні потрібні функції, взаємодії з користувачами відповідні дії, що дозволять користувачам зручно використовувати додаток.

Розробка функціональності включала в себе створення модулів для дієт. Було написано програмний код для реалізації цих функцій забезпечуючи коректну роботу додатку.

Інтеграція з базою даних була успішно виконана дозволяючи зберігати необхідну інформацію про дієти, користувачів та інші дані. Застосування відповідних методів безпеки даних було забезпечено для захисту інформації користувачів.

Тестування та налагодження були проведенні для перевірки правильності роботи додатку виявлення та виправлення помилок та недоліків. Були виконані функціональні тести, тести взаємодії з користувачами та інші види тестів. Це тестування допомогло перевірити правильність роботи додатку у різних сценаріях.

У підсумку розробка мобільного додатку "Дієтолог" була успішною забезпечуючи користувачам зручну та ефективну інтеракцію з додатком. Додаток відповідає поставленим вимогам і функціонує належним чином. Інтерфейс є зрозумілим і добре організованим що дозволяє користувачам легко зорієнтуватись та використовувати його.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Абраменко А. І. Розробка дизайну мобільного додатку за допомогою інструментів Figma: thesis. 2021.
URL: <http://ir.stu.cn.ua/123456789/22995> (дата звернення: 23.05.2023).
2. Бандурович К. Дієтологія- професія здоров'я. *Країна знань*. 2017. № 1/2 (122). С. 54–56.
3. Безпека мобільних додатків : thesis / О. Ю. Бреславець та ін. 2017.
URL: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/45812> (дата звернення: 23.05.2023).
4. Кригін В. Р., Олешко І. В. Проектування та розробка захищеного мобільного додатку оповіщення важливою інформацією : thesis. 2021.
URL: <https://openarchive.nure.ua/handle/document/15745> (дата звернення: 23.05.2023).
5. Леник А. Розробка алгоритму автономного пересування мобільного робота в обмеженому просторі : thesis. 2017.
URL: <http://elartu.tntu.edu.ua/handle/123456789/19378> (дата звернення: 23.05.2023).
6. Наку К., Шикуча О. Розробка мобільного додатку «система логічного розвитку на основі комп'ютерних ігор “evolve your brain”». *Vodnij transport*. 2020. № 2(30). С. 128–135. URL: <https://doi.org/10.33298/2226-8553.2020.2.30.15> (дата звернення: 23.05.2023).
7. Овчаренко О. І. Мобільний додаток “Фітнес контроль” : thesis. 2020. URL: <http://ir.stu.cn.ua/123456789/23428> (дата звернення: 23.05.2023).
8. Савка М. І.-В. Розробка мобільного клієнта сховища файлів реального часу : магістерська робота. 2020.
URL: <https://dspace.znu.edu.ua/jspui/handle/12345/1713> (дата звернення: 23.05.2023).

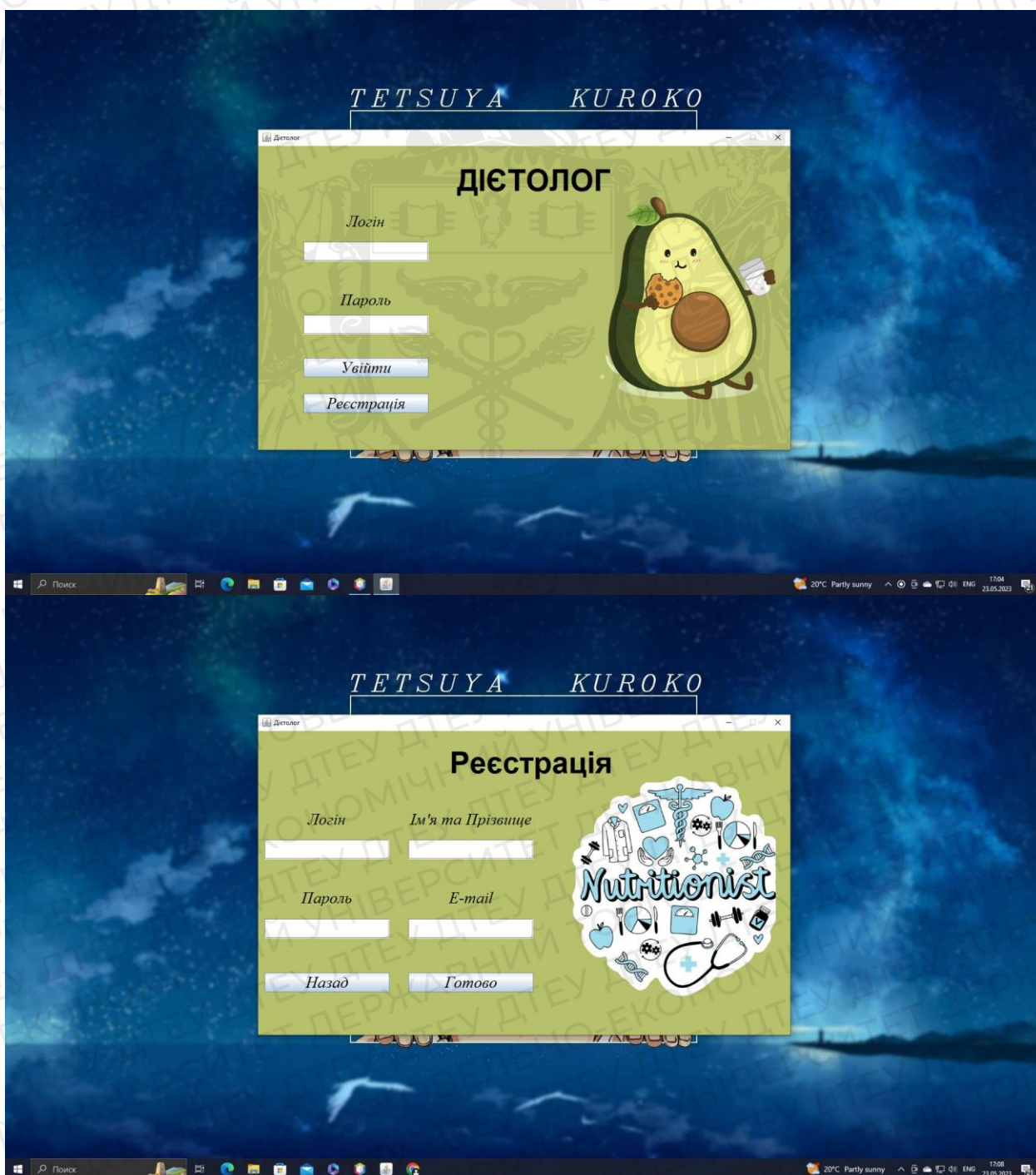
9. App Developer App Developer Notebooks. Mobile UI/UX sketchbook: wireframing and prototyping notebook for UI/UX designers, students, mobile app developers, and hobbyists. Independently Published, 2019.
10. Chen L.-d., Skelton G. W. Mobile application development I. *Mobile commerce application development*. 2005. P. 228–269. URL: <https://doi.org/10.4018/978-1-59140-806-2.ch009> (date of access: 23.05.2023).
11. David Tchilingirian N. Alive and kicking. *SecEd*. 2012. Vol. 2012, no. 9. URL: <https://doi.org/10.12968/sece.2012.9.405> (date of access: 23.05.2023).
12. Funny Nutritionist Funny Nutritionist Gifts. How nutritionists swear coloring book: nutritionist coloring book for adults. Independently Published, 2019.
13. Joseph Correa (Certified Sports Nutritionist). Advanced nutrition for recreational racquetball players: using your resting metabolic rate to improve performance, reduce cramps, and last longer. CreateSpace Independent Publishing Platform, 2016. 204 p.
14. Krah E.-S. Mobile app. *Bankmagazin*. 2012. Vol. 61, no. 12. P. 43. URL: <https://doi.org/10.1365/s35127-012-0527-4> (date of access: 23.05.2023).
15. McWherter J., Gowell S. Professional mobile application development. Wiley & Sons, Incorporated, John, 2012. 432 p.
16. Mobile APP development. dmsinfosystem, 2022.
17. Notebooks N. P. J. Nutritionist : I am a nutritionist, what's your superpower ? Unique customized journal gift for nutritionist - blue journal , thoughtful cool present for nutritionist: lined blank notebook for nutritionist. Independently Published, 2020. 120 p.
18. Pal N. Mobile application development. Independently Published, 2021.
19. Woolwich N. G. O. Latest blue zone diet cookbook: complete guide to nourishing and healthy recipes for eating and living like world's healthiest people. Independently Published, 2022.

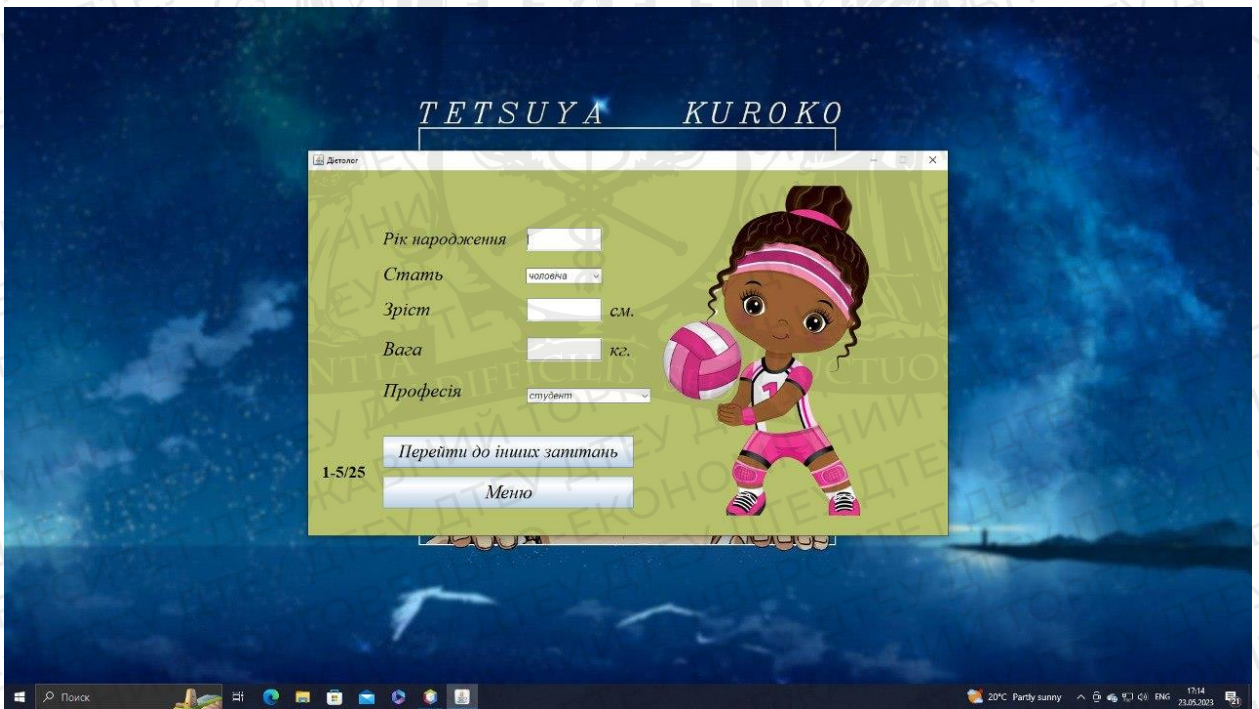
20. Woolwich N. G. O. Latest blue zone diet cookbook: complete guide to nourishing and healthy recipes for eating and living like world's healthiest people. Independently Published, 2022.

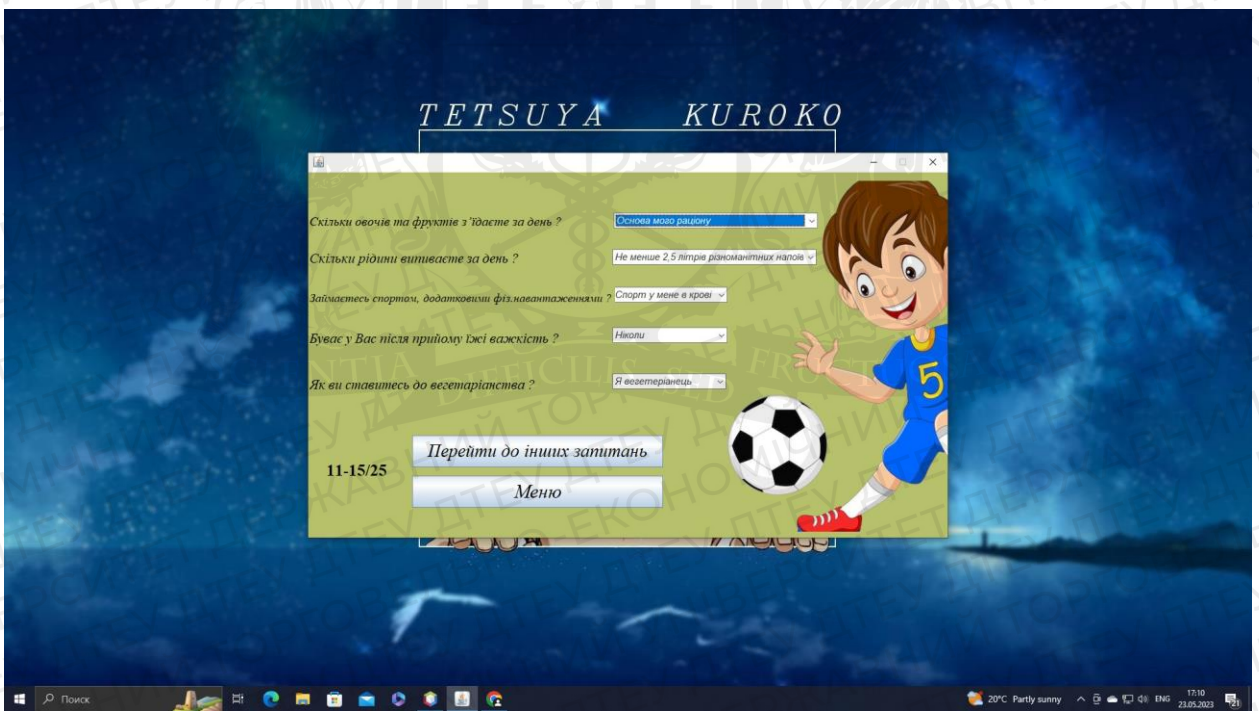
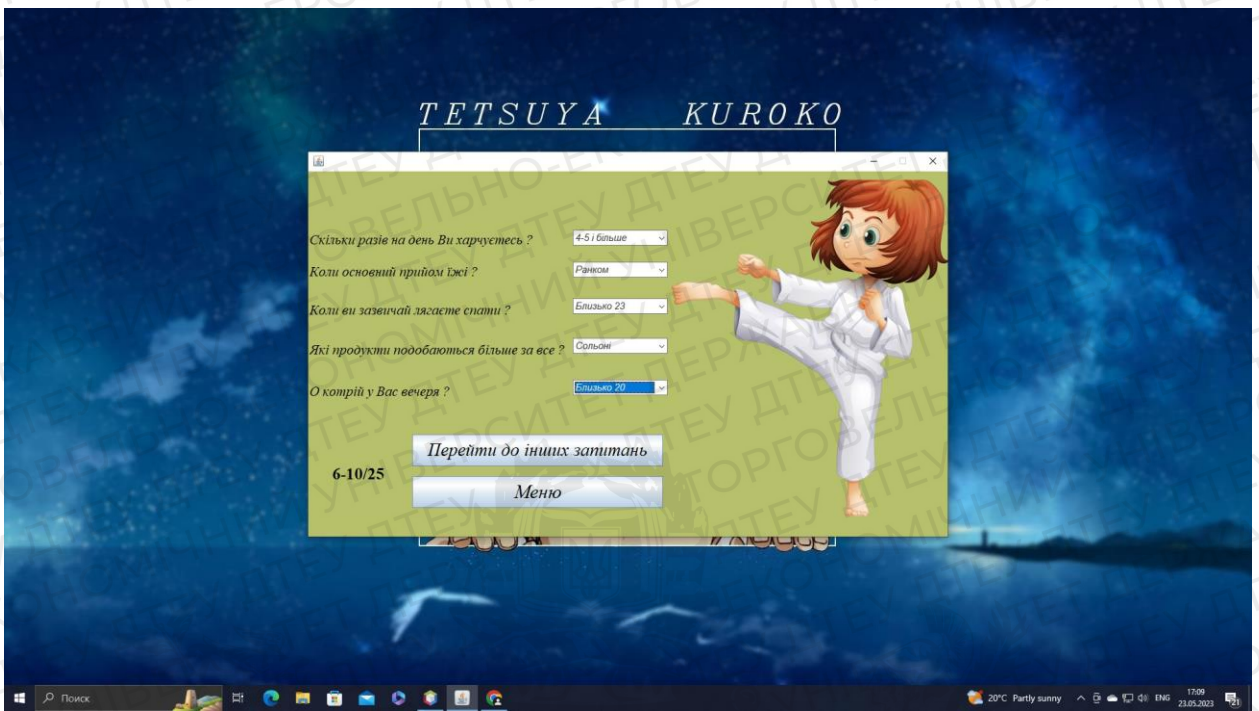
ДОДАТКИ

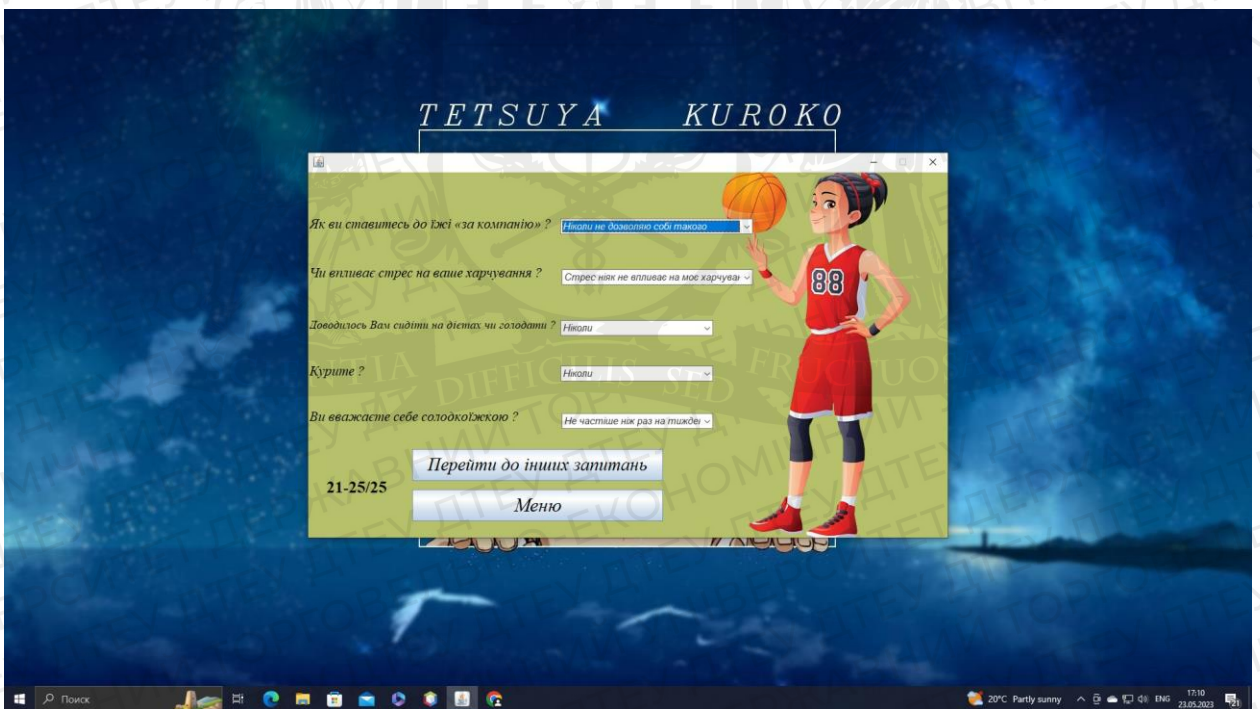
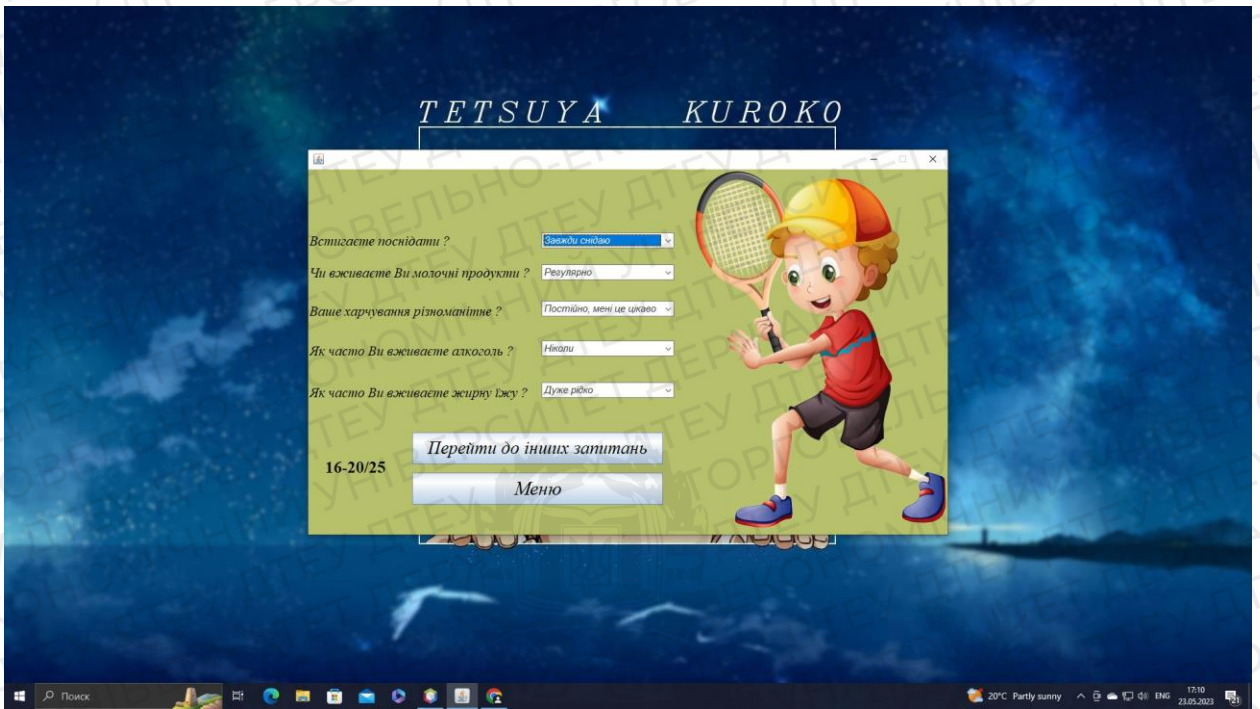
Додаток А

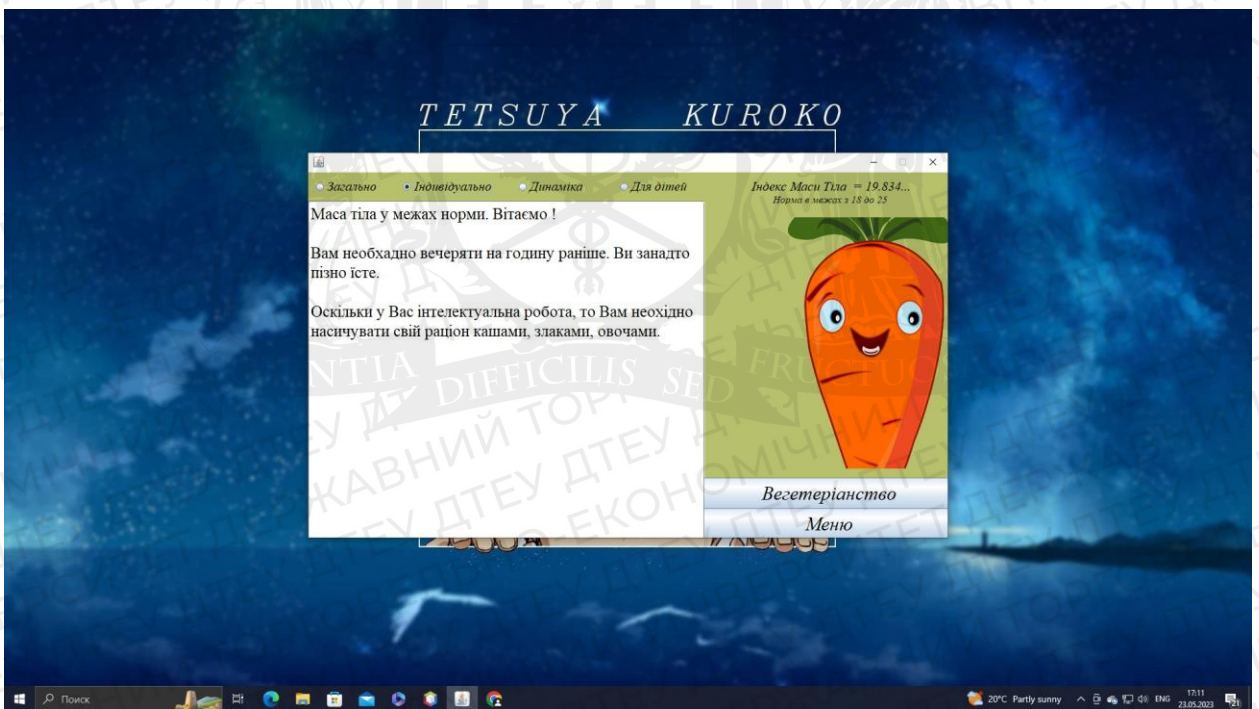
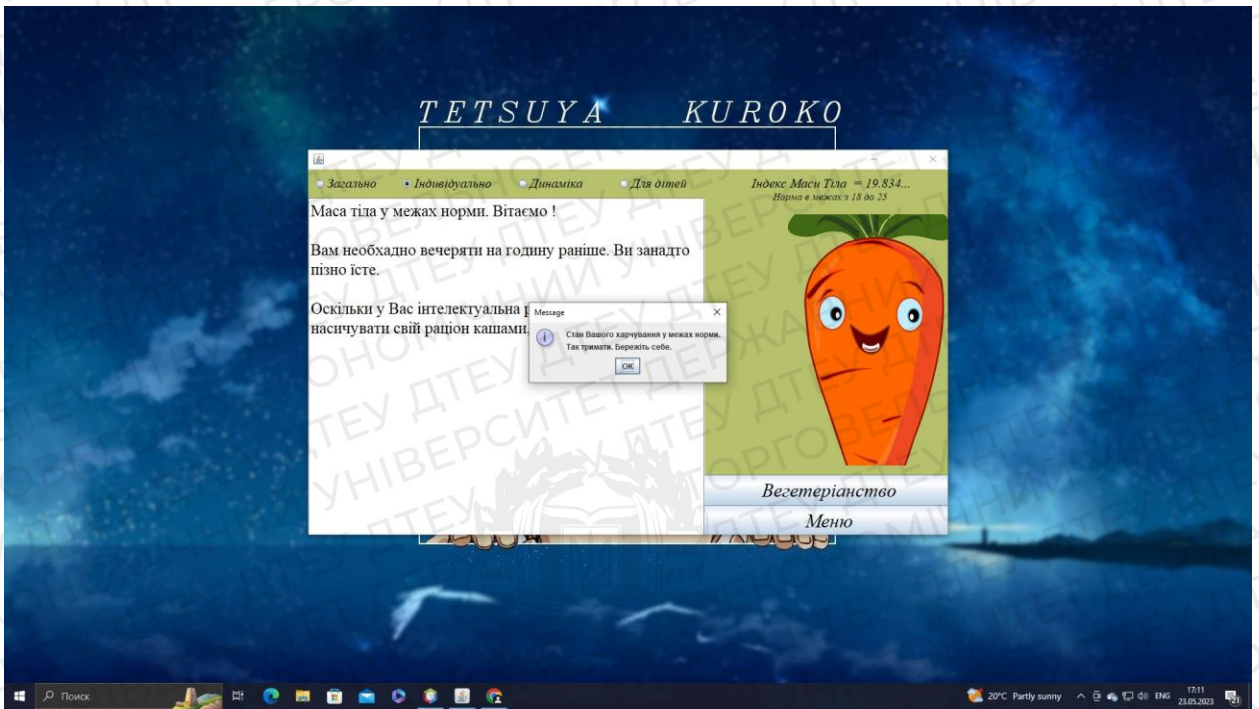
Інтерфейс додатку «Дієтолог»












TETSUYA KUROKO

Засадно • Індивідуально • Динаміка • Для дітей

Індекс Маси Тіла = 19.834...
Норма в межах з 18 до 25



Їсти бажано в один і той же час, який ви повинні встановити собі самі (у зв'язку із характером праці).

Склянка води без газу за півгодини до їжі зменшує апетит.

Їжу потрібно починати з нежирного супу або салату.

Слідкуйте за розміром порцій і не беріть добавки.

Їжте повільно, ретельно пережовуючи. При швидкому поглинанні їжі відчуття насичення "відстає" від кількості поглинутої їжі. Внаслідок чого з'являється зайва кількість жиру.


Купуйте продукти за попередньо складеним списком.

Вегетеріанство
Меню

TETSUYA KUROKO

Засадно • Індивідуально • Динаміка • Для дітей

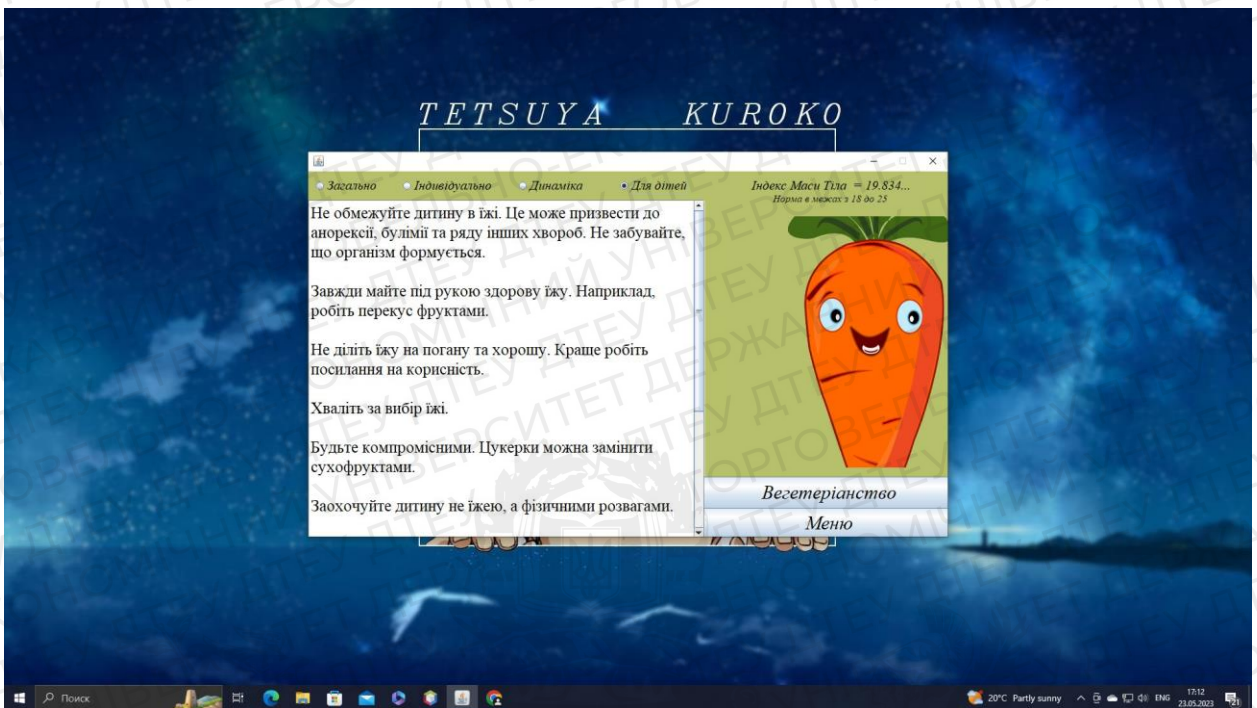
Індекс Маси Тіла = 19.834...
Норма в межах з 18 до 25



Індекс маси Вашого тіла став кращим. Ви наберете форму.

Ви так і не почали раніше харчуватись. Вам необхідно вечеряти на годину раніше. Ви занадто пізно їсте.

Вегетеріанство
Меню



Додаток Б

Код для авторизації

```
public class Avtor extends JFrame {
    private static final long serialVersionUID = 1L;
    private final JLabel Label_NazvaEmblema;
    private final JLabel label_Login;
    private JTextField textField_Login;
    private final JLabel label_Password;
    private JPasswordField passwordField_Password;
    private final JButton b_Yvijtu;
    private final JButton b_Reestracia;
    private final JLabel l_fon;
    public Avtor(String s) {
        super(s);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(1000, 600);
    }
}
```

```
setResizable(false);
setLocationRelativeTo(null);
getContentPane().setLayout(null);
Label_NazvaEmblema = new JLabel("ДІСТОЛІОГ");
Label_NazvaEmblema.setForeground(Color.black);
Label_NazvaEmblema.setFont(new Font("Monotype Corsiva",
Font.BOLD, 55));
Label_NazvaEmblema.setHorizontalAlignment(SwingConstants.CENTER);
Label_NazvaEmblema.setBounds(166, 13, 686, 99);
getContentPane().add(Label_NazvaEmblema);
label_Login = new JLabel("Логін");
label_Login.setForeground(Color.black);
label_Login.setHorizontalAlignment(SwingConstants.CENTER);
label_Login.setFont(new Font("Times New Roman", Font.ITALIC,
30));
label_Login.setBounds(84, 125, 231, 29);
getContentPane().add(label_Login);
textField_Login = new JTextField();
textField_Login.setToolTipText("");
textField_Login.setText("");
textField_Login.setBounds(84, 177, 231, 36);
getContentPane().add(textField_Login);
textField_Login.setColumns(10);
label_Password = new JLabel("Пароль");
label_Password.setForeground(Color.black
);
label_Password.setHorizontalAlignment(SwingConstants.CENTER);
label_Password.setFont(new Font("Times New Roman",
Font.ITALIC, 30));
label_Password.setBounds(84, 270, 231, 29);
```

```

getContentPane().add(label_Password);
passwordField_Password = new JPasswordField();
passwordField_Password.setToolTipText("");
passwordField_Password.setText("");
passwordField_Password.setBounds(84, 312, 231, 36);
getContentPane().add(passwordField_Password);
b_Yvijtu = new JButton("Увійти");
b_Yvijtu.setForeground(Color.black);
b_Yvijtu.addActionListener(new ActionListener() {
    private String s_Login;
    private String s_Password;
    private Scanner scanner_Avtoruzacia;
    private String s_Avtoruzacia;
    String[][] Reading = new String[1][4];
    private String Reading_Password;
    private String Reading_Name;
    private String Reading_Prizvusko;
    private Formatter formatter_RobocuyProfil;
    @SuppressWarnings("deprecation")
    public void actionPerformed(ActionEvent arg0) {
        s_Login = textField_Login.getText();
        s_Password = passwordField_Password.getText();
        try {
            scanner_Avtoruzacia = new Scanner(new
File("res/Avtoruzacia/" + s_Login + ".txt"));
            while (scanner_Avtoruzacia.hasNext()) {
                for (String[] Reading1 : Reading) {
                    Reading_Password = " ";
                    Reading_Name = " ";
                    Reading_Prizvusko = " ";

```

```

for (int col = 0; col < Reading1.length; col++) {
    Reading1[col] = scanner_Avtoruzacia.next();
    if (col == 0) {
        Reading_Password = Reading1[col];
    }
    if (col == 1) {
        Reading_Name = Reading1[col];
    }
    if (col == 2) {
        Reading_Prizvusko = Reading1[col];
    }
}

s_Avtoruzacia = Reading_Password;
scanner_Avtoruzacia.close();

if (s_Password.equals(s_Avtoruzacia)) {
    JOptionPane.showMessageDialog(null,
        "Вітаю, Ви в системі");
    String s_Korustuvac = Reading_Name + " "
        + Reading_Prizvusko;
    try {
        formatter_RobocuyProfil = new
        Formatter("res/TumcasoviFaylu/Korustuvac.txt");
    } catch (FileNotFoundException e) {
        JOptionPane.showMessageDialog(null,
            "System Error");
    }
}

```

```

formatter_RobocuyProfil.format(s_Korustuvac);
formatter_RobocuyProfil.close();
new Menu("Дієтолог", null);
setVisible(false);
} else {
JOptionPane.showMessageDialog(null,
"Помилка введення");
passwordField_Password.setText("");
}
} catch (HeadlessException e) {
JOptionPane.showMessageDialog(null, "Помилка
введення");
} catch (FileNotFoundException e) {
JOptionPane.showMessageDialog(null, "Помилка
введення");
}
});
b_Yvijtu.setFont(new Font("Times New Roman", Font.ITALIC, 30));
b_Yvijtu.setBounds(84, 391, 231, 36);
getContentPane().add(b_Yvijtu);
b_Reestracia = new JButton("Реєстрація");
b_Reestracia.setForeground(Color.black);
b_Reestracia.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
new Reestracia("Дієтолог");
setVisible(false);
}
});
b_Reestracia.setFont(new Font("Times New Roman", Font.ITALIC,

```

30));

```
b_Reestracia.setBounds(84, 457, 231, 36);
```

```
getContentPane().add(b_Reestracia);
```

```
JLabel label = new JLabel("");
```

```
label.setIcon(new ImageIcon("res/fon/Avocado.png"));
```

```
label.setBounds(527, 34, 430, 505);
```

```
getContentPane().add(label);
```

```
l_fon = new JLabel("");
```

```
l_fon.setIcon(new ImageIcon("res/fon/Avtor.png"));
```

```
l_fon.setBounds(-1, 0, 995, 572);
```

```
getContentPane().add(l_fon);
```

```
setVisible(true);
```

```
}
```

```
}
```

Додаток В

Код для реєстрації

```
public class ReestrAsia extends JFrame {  
    private static final long serialVersionUID = 1L;  
    private final JLabel label_NazvaVikna;  
    private final JLabel label_Login;  
    private JTextField textField_Login;  
    private final JLabel label_Password;  
    private JPasswordField passwordField_Password;  
    private final JButton button_Nazad;  
    private final JLabel label_PIP;  
    private JTextField textField_PIP;  
    private final JLabel label_email;  
    private JTextField textField_mail;  
    private final JButton button_Gotovo;  
    private final JLabel l_fon;  
    public ReestrAsia(String s) {  
        super(s);
```

```

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(1000, 600);
setResizable(false);
setLocationRelativeTo(null);
getContentPane().setLayout(null);
label_NazvaVikna = new JLabel("Реєстрація");
label_NazvaVikna.setForeground(new Color(0, 0, 0));
label_NazvaVikna.setBounds(256, 13, 497, 88);
label_NazvaVikna.setFont(new Font("Monotype Corsiva",
Font.BOLD, 55));
label_NazvaVikna.setHorizontalAlignment(SwingConstants.CENTER);
getContentPane().add(label_NazvaVikna);
label_Login = new JLabel("Логін");
label_Login.setForeground(new Color(0, 0, 0));
label_Login.setBounds(12, 149, 231, 29);
label_Login.setHorizontalAlignment(SwingConstants.CENTER);
label_Login.setFont(new Font("Times New Roman", Font.ITALIC,
30));
getContentPane().add(label_Login);
textField_Login = new JTextField();
textField_Login.setBounds(12, 201, 231, 36);
textField_Login.setToolTipText("");
textField_Login.setText("");
textField_Login.setColumns(10);
getContentPane().add(textField_Login);
label_Password = new JLabel("Пароль");
label_Password.setForeground(new Color(0, 0, 0));
label_Password.setBounds(12, 294, 231, 29);
label_Password.setHorizontalAlignment(SwingConstants.CENTER);
label_Password.setFont(new Font("Times New Roman",
Font.ITALIC, 30));
getContentPane().add(label_Password);
passwordField_Password = new JPasswordField();
passwordField_Password.setBounds(12, 347, 231, 36);
passwordField_Password.setToolTipText("");
passwordField_Password.setText("");
getContentPane().add(passwordField_Password);
button_Nazad = new JButton("Назад");
button_Nazad.setForeground(new Color(0, 0, 0));

```



```

button_Nazad.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new Avtor("Дієтолог");
        setVisible(false);
    }
});
button_Nazad.setBounds(12, 446, 231, 36);
button_Nazad.setFont(new Font("Times New Roman", Font.ITALIC,
30));
getContentPane().add(button_Nazad);
label_PIP = new JLabel("Ім'я та Прізвище");
label_PIP.setForeground(new Color(0, 0, 0));
label_PIP.setBounds(278, 149, 231, 29);
label_PIP.setHorizontalAlignment(SwingConstants.CENTER);
label_PIP.setFont(new Font("Times New Roman", Font.ITALIC,
30));
getContentPane().add(label_PIP);
textField_PIP = new JTextField();
textField_PIP.setBounds(278, 201, 231, 36);
textField_PIP.setToolTipText("");
textField_PIP.setText("");
textField_PIP.setColumns(10);
getContentPane().add(textField_PIP);
label_email = new JLabel("E-mail");
label_email.setForeground(new Color(0, 0, 0));
label_email.setBounds(278, 294, 231, 29);
label_email.setHorizontalAlignment(SwingConstants.CENTER);
label_email.setFont(new Font("Times New Roman", Font.ITALIC,
30));
getContentPane().add(label_email);
textField_mail = new JTextField();
textField_mail.setBounds(278, 347, 231, 36);
textField_mail.setToolTipText("");
textField_mail.setText("");
getContentPane().add(textField_mail);
button_Gotovo = new JButton("Готово");
button_Gotovo.setForeground(new Color(0, 0, 0));
button_Gotovo.setBounds(278, 446, 231, 36);
button_Gotovo.addActionListener(new ActionListener() {

```

```

private String s_Login;
private String s_Password;
private String s_PIP;
private String s_mail;

private Formatter formatter_reestracia;
private String s_reestracia;
private int i_korektnist = 0;
private int i_korektnist1 = 0;
@SuppressWarnings("deprecation")
public void actionPerformed(ActionEvent arg0) {
    s_Login = textField_Login.getText();
    s_Password = passwordField_Password.getText();
    s_PIP = textField_PIP.getText();
    s_mail = textField_mail.getText();
    File file = new File("res/Avtoruzacia/" + s_Login +
".txt");
    if (file.exists()) {
        JOptionPane.showMessageDialog(null, "Уже
зарєєстрований користувач під даним логіном");
    } else {
        int blockCount = s_PIP.split(" ").length;
        if (blockCount != 2) {
            JOptionPane.showMessageDialog(null,
"Некоректне ім'я чи прізвище");
        } else {
            char[] chars = s_PIP.toCharArray();
            for (int i = 0; i < chars.length; i++) {
                try {
                    if (chars[i] == '1' || chars[i] ==
'0' || chars[i] == '(' || chars[i] == ')'
|| chars[i] == '+' ||
chars[i] == '=' || chars[i] == '/' || chars[i] == '.'
|| chars[i] == ',') {
                        i_korektnist++;
                    }
                } catch (Exception e) {
                }
            }
        }
    }
}

```

```

char[] chars1 = s_mail.toCharArray();
for (int i = 0; i < chars1.length; i++) {
    try {
        if (chars1[i] == '@') {
            i_korektnist1 = 1;
        }
    } catch (Exception e) {
    }
}
if (i_korektnist == 0 && i_korektnist1 == 1)
{
    try {
        formatter_reestracia = new
Formatter("res/Avtoruzacia/" + s_Login + ".txt");
    } catch (FileNotFoundException e) {
        JOptionPane.showMessageDialog(null, "System Error");
    }
    s_reestracia = s_Password + " " +
s_PIP + " " + s_mail;
    formatter_reestracia.format(s_reestracia);
    formatter_reestracia.close();
    JOptionPane.showMessageDialog(null,
"Зареєстровано");
    new Avtor("Дієтолог");
    setVisible(false);
} else {
    JOptionPane.showMessageDialog(null, "Перевірте введення");
}
i_korektnist = 0;
}
}
});
button_Gotovo.setFont(new Font("Times New Roman",
Font.ITALIC, 30));
getContentPane().add(button_Gotovo);
JLabel label = new JLabel("");

```

```

        label.setIcon(new ImageIcon("res/fon/Reestracia1.png"));
        label.setBounds(527, 34, 430, 505);
        getContentPane().add(label);
        l_fon = new JLabel("");
        l_fon.setForeground(Color.WHITE);
        l_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));
        l_fon.setBounds(-1, 0, 995, 572);
        getContentPane().add(l_fon);
        setVisible(true);
    }
}

```

Додаток Г

Код для головного меню

```

public class Menu extends JFrame {
    private static final long serialVersionUID = 1L;
    private final JLabel l_fon;
    private JLabel menuCucumber;
    private JLabel menuZucchini;
    public Menu(String s, javax.swing.JLabel menuCucumber) {
        super(s);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(1000, 600);
        setResizable(false);
        setLocationRelativeTo(null);
        getContentPane().setLayout(null);
        JButton b_Diagnostika = new JButton("Діагностика");
        b_Diagnostika.setFont(new Font("Times New Roman", Font.ITALIC,
30));
        b_Diagnostika.setForeground(new Color(0, 0, 0));
        b_Diagnostika.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                new Diagnostika("Дієтолог");
                setVisible(false);
            }
        });
        b_Diagnostika.setBounds(327, 55, 282, 42);
        getContentPane().add(b_Diagnostika);
    }
}

```

```

JButton b_Likyvanna = new JButton("Лікування");
b_Likyvanna.setFont(new Font("Times New Roman", Font.ITALIC,
30));
b_Likyvanna.setForeground(new Color(0, 0, 0));
b_Likyvanna.setBounds(327, 140, 282, 42);
getContentPane().add(b_Likyvanna);
JButton b_Vegeterianstvo = new JButton("Вегетеріанство");
b_Vegeterianstvo.setFont(new Font("Times New Roman",
Font.ITALIC, 30));
b_Vegeterianstvo.setForeground(new Color(0, 0, 0));
b_Vegeterianstvo.setBounds(327, 235, 282, 42);
getContentPane().add(b_Vegeterianstvo);
JButton b_PerioduRozvutku = new JButton("Періоди розвитку");
b_PerioduRozvutku.setFont(new Font("Times New Roman",
Font.ITALIC, 30));
b_PerioduRozvutku.setForeground(new Color(0, 0, 0));
b_PerioduRozvutku.setBounds(327, 324, 282, 42);
getContentPane().add(b_PerioduRozvutku);
JButton b_Racion = new JButton("Раціон");
b_Racion.setFont(new Font("Times New Roman", Font.ITALIC,
30));
b_Racion.setForeground(new Color(0, 0, 0));
b_Racion.setBounds(327, 407, 282, 42);
getContentPane().add(b_Racion);
JButton b_ZvernennaDoLakara = new JButton("Звернення до
лікаря");
b_ZvernennaDoLakara.setFont(new Font("Times New Roman",
Font.ITALIC, 29));
b_ZvernennaDoLakara.setForeground(new Color(0, 0, 0));
b_ZvernennaDoLakara.setBounds(327, 486, 282, 42);
getContentPane().add(b_ZvernennaDoLakara);

JLabel label = new JLabel("");
label.setIcon(new ImageIcon("res/fon/menuAvocado.png"));
label.setBounds(676, 25, 256, 244);
getContentPane().add(label);
menuCucumber = new JLabel("");
menuCucumber.setIcon(new

```

```

ImageIcon("res/fon/menuCucumber.png"));
    menuCucumber.setBounds(70, 154, 256, 244);
    getContentPane().add(menuCucumber);
    menuZucchini = new JLabel("");
    menuZucchini.setIcon(new ImageIcon("res/fon/menuZucchini.png"));
    menuZucchini.setBounds(676, 304, 256, 244);
    getContentPane().add(menuZucchini);
    l_fon = new JLabel("");
    l_fon.setForeground(Color.WHITE);
    l_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));
    l_fon.setBounds(-1, 0, 995, 572);
    getContentPane().add(l_fon);
    setVisible(true);
    this.menuCucumber = menuCucumber;
}
}

```

Додаток Д

Код діагностики

```

public class Diagnostika extends JFrame {
    private static final long serialVersionUID = 1L;
    private JTextField t_RikNarodgenna;
    private JTextField t_Zrist;
    private JTextField t_Vaga;
    private Choice choice_Stat;
    private final JLabel next;
    private Choice choice_Profesia;
    private final JLabel Stat;
    private final JLabel Zrist;
    private final JLabel RikNarodgenna;
    private final JLabel Profesia;
    private final JLabel Vaga;
    private final JLabel cm;
    private final JLabel kg;
    private final JButton btnNewButton;
    private final JLabel l_fon;
    public Diagnostika(String s) {
        super(s);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

```
setSize(1000, 600);
setResizable(false);
setLocationRelativeTo(null);
getContentPane().setLayout(null);
Stat = new JLabel("Стать");
Stat.setForeground(new Color(0, 0, 0));
Stat.setFont(new Font("Times New Roman", Font.ITALIC, 30));
Stat.setBounds(115, 141, 151, 36);
getContentPane().add(Stat);
RikNarodgenna = new JLabel("Рік народження");
RikNarodgenna.setForeground(new Color(0, 0, 0));
RikNarodgenna.setFont(new Font("Times New Roman",
Font.ITALIC, 27));
RikNarodgenna.setBounds(115, 87, 205, 36);
getContentPane().add(RikNarodgenna);
Zrist = new JLabel("Зріст");
Zrist.setForeground(new Color(0, 0, 0));
Zrist.setFont(new Font("Times New Roman", Font.ITALIC, 30));
Zrist.setBounds(115, 195, 151, 36);
getContentPane().add(Zrist);
Vaga = new JLabel("Вага");
Vaga.setForeground(new Color(0, 0, 0));
Vaga.setFont(new Font("Times New Roman", Font.ITALIC, 30));
Vaga.setBounds(115, 256, 151, 36);
getContentPane().add(Vaga);
Profesia = new JLabel("Професія");
Profesia.setForeground(new Color(0, 0, 0));
Profesia.setFont(new Font("Times New Roman", Font.ITALIC, 30));
Profesia.setBounds(115, 320, 151, 36);
getContentPane().add(Profesia);
t_RikNarodgenna = new JTextField();
t_RikNarodgenna.setBounds(336, 88, 116, 36);
getContentPane().add(t_RikNarodgenna);
t_RikNarodgenna.setColumns(10);
t_Zrist = new JTextField();
t_Zrist.setColumns(10);
t_Zrist.setBounds(336, 196, 116, 36);
getContentPane().add(t_Zrist);
t_Vaga = new JTextField();
```

```

t_Vaga.setColumns(10);
t_Vaga.setBounds(336, 257, 116, 36);
getContentPane().add(t_Vaga);
choice_Profesia = new Choice();
choice_Profesia.setFont(new Font("Times New Roman",
Font.ITALIC, 15));
choice_Profesia.setBounds(336, 334, 191, 24);
getContentPane().add(choice_Profesia);
choice_Profesia.add("студент");
choice_Profesia.add("фізична робота");
choice_Profesia.add("інтелектуальна робота");
choice_Stat = new Choice();
choice_Stat.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_Stat.setBounds(335, 150, 117, 54);
getContentPane().add(choice_Stat);
choice_Stat.add("чоловіча");
choice_Stat.add("жіноча");
next = new JLabel("1-5/25");
next.setForeground(Color.BLACK);
next.setFont(new Font("Times New Roman", Font.BOLD, 25));
next.setHorizontalAlignment(SwingConstants.CENTER);
next.setBounds(12, 440, 84, 49);
getContentPane().add(next);
cm = new JLabel("см.");
cm.setForeground(new Color(0, 0, 0));
cm.setFont(new Font("Times New Roman", Font.ITALIC, 30));
cm.setBounds(464, 196, 75, 36);
getContentPane().add(cm);
kg = new JLabel("кг.");
kg.setForeground(new Color(0, 0, 0));
kg.setFont(new Font("Times New Roman", Font.ITALIC, 30));
kg.setBounds(464, 257, 75, 36);
getContentPane().add(kg);
JButton b_Menu = new JButton("Меню");
b_Menu.setForeground(new Color(0, 0, 0));
b_Menu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        Object[] options = { "Так, перейти", "Ні, залишитись"

```



```

    };
        int i_menu = JOptionPane.showOptionDialog(null,
"Бажаєте перейти в меню ? Введені дані не збережуться.",
        "Попередження",
JOptionPane.YES_NO_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, options,
        null);
        if (i_menu == 0) { /*menu*/ /*menu*/
            setVisible(false);
        }
    }
});
b_Menu.setFont(new Font("Times New Roman", Font.ITALIC, 30));
b_Menu.setBounds(115, 470, 386, 49);
getContentPane().add(b_Menu);
btnNewButton = new JButton("Перейти до інших запитань");
btnNewButton.setForeground(new Color(0, 0, 0));
btnNewButton.setFont(new Font("Times New Roman", Font.ITALIC,
28));
btnNewButton.addActionListener(new ActionListener() {
    private String s_RikNarodgenna;
    private String s_Zrist;
    private String s_Vaga;
    private int i_RikNarodgenna;
    private int i_Zrist;
    private int i_Vaga;
    public void actionPerformed(ActionEvent arg0) {
        s_RikNarodgenna = t_RikNarodgenna.getText();
        s_Zrist = t_Zrist.getText();
        s_Vaga = t_Vaga.getText();
        if (s_RikNarodgenna.equals("") ||
s_RikNarodgenna.equals(" ") || s_RikNarodgenna.equals(" ")
            || s_RikNarodgenna.equals(" ")) {
            JOptionPane.showMessageDialog(null, "Введіть
рік народження");
        } else {
            if (s_Zrist.equals("") || s_Zrist.equals(" ") ||
s_Zrist.equals(" ") || s_Zrist.equals(" ")) {

```

```

JOptionPane.showMessageDialog(null,
    "Введіть зріст");
    } else {
        if (s_Vaga.equals("") || s_Vaga.equals(" ") ||
s_Vaga.equals(" ") || s_Vaga.equals(" ")) {
            JOptionPane.showMessageDialog(null, "Введіть вагу");
            } else {
                try {
                    i_RikNarodgenna =
Integer.parseInt(s_RikNarodgenna);
                    i_Zrist =
Integer.parseInt(s_Zrist);
                    i_Vaga =
Integer.parseInt(s_Vaga);
                } catch (NumberFormatException e)
            JOptionPane.showMessageDialog(null, "Перевірте введення");
            }
            if (i_RikNarodgenna > 1900 &&
i_RikNarodgenna <= 2017) {
                if (i_Zrist > 30 && i_Zrist <=
250) {
                    if (i_Vaga > 2 &&
i_Vaga <= 800) {
                        String s_Stat =
choice_Stat.getSelectedItem();
                        String s_Profesia =
choice_Profesia.getSelectedItem();
                        int i_Stat = 0;
                        int i_Profesia = 0;
                        switch (s_Stat) {
                            case "чоловіча":
                                break;
                            case "жіноча":
                                break;

```

```

    }
    switch
    case "студент":
    case "фізична робота":
    case "інтелектуальна робота"
    break;
    }
    new
    Diagnostikal(i_RikNarodgenna, i_Stat, i_Zrist, i_Vaga, i_Profesia);
    setVisible(false);
    } else {
        JOptionPane.showMessageDialog(null,
        "Помилкова вага");
    } else {
        JOptionPane.showMessageDialog(null, "Помилковий зріст");
    } else {
        JOptionPane.showMessageDialog(null, "Помилковий рік народження");
    }
    }
    });
    btnNewButton.setBounds(115, 408, 386, 49);
    getContentPane().add(btnNewButton);

    JLabel label = new JLabel("");
    label.setIcon(new ImageIcon("res/Diagnostika.png"));
    label.setBounds(515, 24, 479, 506);
    getContentPane().add(label);
    l_fon = new JLabel("");
    l_fon.setForeground(Color.WHITE);
    l_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));

```

```

l_fon.setBounds(-1, 0, 995, 572);
getContentPane().add(l_fon);
setVisible(true);
    }
}

```

Додаток Е

Код діагностики 1

```

public class Diagnostika1 extends JFrame {
    private static final long serialVersionUID = 1L;
    private final JLabel l_6;
    private Choice choice_6;
    private Choice choice_7;
    private Choice choice_8;
    private Choice choice_9;
    private Choice choice_10;
    private final JLabel l_8;
    private final JLabel l_9;
    private final JLabel l_10;
    private final JLabel l_7;
    private final JLabel next1;
    private final JButton button_1;
    private final JLabel label;
    private final JButton button_2;
    private final JLabel l_fon;
    public Diagnostika1(final int i_RikNarodgenna, final int i_Stat, final int
i_Zrist, final int i_Vaga,
        final int i_Profesja) {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(1000, 600);
        setResizable(false);
        setLocationRelativeTo(null);
        getContentPane().setLayout(null);
        l_6 = new JLabel("Скільки разів на день Ви харчуєтесь ? ");
        l_6.setForeground(new Color(0, 0, 0));
        l_6.setFont(new Font("Times New Roman", Font.ITALIC, 21));
        l_6.setBounds(2, 87, 401, 36);
        getContentPane().add(l_6);
        choice_6 = new Choice();

```

```
choice_6.setFont(new Font("Times New Roman", Font.ITALIC, 15));
choice_6.setBounds(407, 90, 146, 22);
getContentPane().add(choice_6);
choice_6.add("4-5 і більше");
choice_6.add("3 рази");
choice_6.add("2 рази і менше");
l_7 = new JLabel("Коли основний прийом їжі ?");
l_7.setForeground(new Color(0, 0, 0));
l_7.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_7.setBounds(2, 136, 399, 36);
getContentPane().add(l_7);
choice_7 = new Choice();
choice_7.setFont(new Font("Times New Roman", Font.ITALIC, 15));
choice_7.setBounds(407, 139, 146, 22);
getContentPane().add(choice_7);
choice_7.add("Вобід");
choice_7.add("Ранком");
choice_7.add("Ввечері");
choice_7.add("Вночі");

l_8 = new JLabel("Коли ви зазвичай лягаєте спати ? ");
l_8.setForeground(new Color(0, 0, 0));
l_8.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_8.setBounds(2, 195, 399, 36);
getContentPane().add(l_8);
choice_8 = new Choice();
choice_8.setFont(new Font("Times New Roman", Font.ITALIC, 15));
choice_8.setBounds(407, 195, 146, 22);
getContentPane().add(choice_8);
choice_8.add("Після першої");
choice_8.add("Близько опівночі");
choice_8.add("Близько 23");
choice_8.add("До 22");

l_9 = new JLabel("Які продукти подобаються більше за все ?");
l_9.setForeground(new Color(0, 0, 0));
l_9.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_9.setBounds(2, 256, 399, 36);
getContentPane().add(l_9);
choice_9 = new Choice();
```

```

choice_9.setFont(new Font("Times New Roman", Font.ITALIC, 15));
choice_9.setBounds(407, 256, 146, 22);
getContentPane().add(choice_9);
choice_9.add("Кислі");
choice_9.add("Сольоні");
choice_9.add("Солодкі");
l_10 = new JLabel("О котрій у Вас вечеря ?");
l_10.setForeground(new Color(0, 0, 0));
l_10.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_10.setBounds(2, 320, 399, 36);
getContentPane().add(l_10);
choice_10 = new Choice();
choice_10.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_10.setBounds(407, 320, 146, 22);
getContentPane().add(choice_10);
choice_10.add("Після 24");
choice_10.add("Близько 22");
choice_10.add("Близько 20");
choice_10.add("До 18");
next1 = new JLabel("6-10/25");
next1.setForeground(Color.BLACK);
next1.setHorizontalAlignment(SwingConstants.CENTER);
next1.setFont(new Font("Times New Roman", Font.BOLD, 25));
next1.setBounds(31, 441, 93, 49);
getContentPane().add(next1);
button_1 = new JButton("Перейти до інших запитань");
button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String s_choice_6 = choice_6.getSelectedItem();
        String s_choice_7 = choice_7.getSelectedItem();
        String s_choice_8 = choice_8.getSelectedItem();
        String s_choice_9 = choice_9.getSelectedItem();
        String s_choice_10 = choice_10.getSelectedItem();
        int i_choice_6 = 0;
        int i_choice_7 = 0;
        int i_choice_8 = 0;
        int i_choice_9 = 0;
        int i_choice_10 = 0;

```

```

switch (s_choice_6) {
case "4-5 і більше":
break;
case "3 рази":
break;
case "2 рази і менше":
break;
}
switch (s_choice_7) {
case "Вобід":
break;
case "Ранком":
break;
case "Ввечері":
break;
case "Вночі":
break;
}
switch (s_choice_8) {
case "Після першої":
break;
case "Близько опівночі":
break;
case "Близько 23":
break;
case "До 22":
break;
}
switch (s_choice_9) {
case "Кислі":
break;
case "Сольоні":
break;
case "Солодкі":
break;
}
switch (s_choice_10) {
case "Після 24":
break;
case "Близько 22":
break;
case "Близько 20":
break;
case "До 18":
break;
}
new Diagnostika2(i_RikNarodgenna, i_Stat, i_Zrist, i_Vaga,
i_Profesia, i_choice_6, i_choice_7,
i_choice_8, i_choice_9, i_choice_10);
setVisible(false);
}
});
button_1.setForeground(new Color(0, 0, 0));

```

```

button_1.setFont(new Font("Times New Roman", Font.ITALIC, 28));
button_1.setBounds(160, 404, 386, 49);
getContentPane().add(button_1);
button_2 = new JButton("Меню");
button_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Object[] options = { "Так, перейти", "Ні, залишитись"
};
        int i_menu = JOptionPane.showOptionDialog(null,
"Бажаєте перейти в меню ? Введені дані не збережуться.",
        "Попередження",
JOptionPane.YES_NO_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, options,
        null);
        if (i_menu == 0) {
            setVisible(false);
        }
    }
});
button_2.setForeground(new Color(0, 0, 0));
button_2.setFont(new Font("Times New Roman", Font.ITALIC, 30));
button_2.setBounds(160, 468, 386, 49);
getContentPane().add(button_2);
label = new JLabel("");
label.setIcon(new ImageIcon("res/Diagnostika1.png"));
label.setBounds(559, 13, 435, 539);
getContentPane().add(label);
l_fon = new JLabel("");
l_fon.setForeground(Color.WHITE);
l_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));
l_fon.setBounds(-1, 0, 995, 572);
getContentPane().add(l_fon);
setVisible(true);
}
}

```

Додаток Ж

Код діагностики 2

```
public class Diagnostika2 extends JFrame {
```



```

private static final long serialVersionUID = 1L;
private final JLabel l_11;
private Choice choice_11;
private Choice choice_12;
private Choice choice_13;
private Choice choice_14;
private Choice choice_15;
private final JLabel l_13;
private final JLabel l_14;
private final JLabel l_15;
private final JLabel l_12;
private final JLabel next2;
private final JButton button;
private final JButton button_1;
private final JLabel label;
private final JLabel l_fon;
public Diagnostika2(final int i_RikNarodgenna, final int i_Stat, final int
i_Zrist, final int i_Vaga,
                    final int i_Profesja, final int i_choice_6, final int i_choice_7,
                    final int i_choice_8,
                    final int i_choice_9, final int i_choice_10) {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(1000, 600);
    setResizable(false);
    setLocationRelativeTo(null);
    getContentPane().setLayout(null);
    l_11 = new JLabel("Скільки овочів та фруктів з'їдаєте за день ?");
    l_11.setForeground(new Color(0, 0, 0));
    l_11.setFont(new Font("Times New Roman", Font.ITALIC, 20));
    l_11.setBounds(2, 58, 462, 36);
    getContentPane().add(l_11);
    choice_11 = new Choice();
    choice_11.setFont(new Font("Times New Roman", Font.ITALIC,
15));
    choice_11.setBounds(470, 62, 314, 24);
    getContentPane().add(choice_11);
    choice_11.add("Основа мого раціону");
    choice_11.add("Сумарно десь тарілочку на тиждень");
    choice_11.add("Майже не їм");

```

```
l_12 = new JLabel("Скільки рідини випиваєте за день ?");
l_12.setForeground(new Color(0, 0, 0));
l_12.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_12.setBounds(2, 114, 460, 36);
getContentPane().add(l_12);
choice_12 = new Choice();
choice_12.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_12.setBounds(468, 118, 314, 24);
getContentPane().add(choice_12);
choice_12.add("Не менше 2,5 літрів різноманітних напоїв");
choice_12.add("Кілька кружок чаю чи кави");
choice_12.add("Чашечку кави ранком та чашечку ввечері");
l_13 = new JLabel("Буває у Вас після прийому їжі важкість ?");
l_13.setForeground(new Color(0, 0, 0));
l_13.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_13.setBounds(2, 237, 460, 36);
getContentPane().add(l_13);

choice_13 = new Choice();
choice_13.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_13.setBounds(468, 238, 177, 22);
getContentPane().add(choice_13);
choice_13.add("Ніколи");
choice_13.add("Дуже рідко");
choice_13.add("Іноді після вечері");
choice_13.add("Постійно");
l_14 = new JLabel("Займаєтесь спортом, додатковими
фіз.навантаженнями ?");
l_14.setForeground(new Color(0, 0, 0));
l_14.setFont(new Font("Times New Roman", Font.ITALIC, 18));
l_14.setBounds(2, 175, 470, 36);
getContentPane().add(l_14);
choice_14 = new Choice();
choice_14.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_14.setBounds(468, 176, 177, 22);
getContentPane().add(choice_14);
```

```

choice_14.add("Спорт у мене в крові");
choice_14.add("1-2 рази на таждень");
choice_14.add("Дуже рідко");
choice_14.add("Ніколи, це не для мене");
l_15 = new JLabel("Як ви ставитесь до вегетаріанства ?");
l_15.setForeground(new Color(0, 0, 0));
l_15.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_15.setBounds(2, 308, 460, 36);
getContentPane().add(l_15);
choice_15 = new Choice();
choice_15.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_15.setBounds(468, 309, 175, 28);
getContentPane().add(choice_15);
choice_15.add("Я вегетеріанець");
choice_15.add("Цікавлюсь цим");
choice_15.add("Це не для мене");
next2 = new JLabel("11-15/25");
next2.setForeground(Color.BLACK);
next2.setHorizontalAlignment(SwingConstants.CENTER);
next2.setFont(new Font("Times New Roman", Font.BOLD, 25));
next2.setBounds(12, 434, 125, 49);
getContentPane().add(next2);
button = new JButton("Перейти до інших запитань");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String s_choice_11 = choice_11.getSelectedItem();
        String s_choice_12 = choice_12.getSelectedItem();
        String s_choice_13 = choice_13.getSelectedItem();
        String s_choice_14 = choice_14.getSelectedItem();
        String s_choice_15 = choice_15.getSelectedItem();
        int i_choice_11 = 0;
        int i_choice_12 = 0;
        int i_choice_13 = 0;
        int i_choice_14 = 0;
        int i_choice_15 = 0;
        switch (s_choice_11) {
            case "Основа мого раціону":
                break;

```

```

case "Сумарно десь тарілочку на тиждень":
break;
case "Майже не їм":
break;
}

switch (s_choice_12) {
case "Не менше 2 літрів різноманітних напоїв":
break;
case "Кілька кружок чаю чи кави":
break;
case "Чашечку кави ранком та чашечку ввечері":
break;
}
switch (s_choice_13) {
case "Ніколи":
break;
case "Дуже рідко":
break;
case "Іноді після вечері":
break;
case "Постійно":
break;
}
switch (s_choice_14) {
case "Спорт у мене в крові":
break;
case "1-2 рази на таждень":
break;
case "Дуже рідко":
break;
case "Ніколи, це не для мене":
break;
}
switch (s_choice_15) {
case "Я вегетеріанець":
break;
case "Цікавлюсь цим":
break;
case "Це не для мене":
break;
}
new Diagnostika3(i_RikNarodgenna, i_Stat, i_Zrist,
i_Vaga, i_Profesia, i_choice_6, i_choice_7,
i_choice_8, i_choice_9, i_choice_10,
i_choice_11, i_choice_12, i_choice_13, i_choice_14,
i_choice_15);

```

```

        setVisible(false);
    }
});
button.setForeground(new Color(0, 0, 0));
button.setFont(new Font("Times New Roman", Font.ITALIC, 28));
button.setBounds(160, 404, 386, 49);
getContentPane().add(button);
button_1 = new JButton("Меню");
button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Object[] options = { "Так, перейти", "Ні, залишитись"
};
        int i_menu = JOptionPane.showOptionDialog(null,
"Бажаєте перейти в меню ? Введені дані не збережуться.",
        "Попередження",
        JOptionPane.YES_NO_CANCEL_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, options,
        null);
        if (i_menu == 0) {
            setVisible(false);
        }
    }
});
button_1.setForeground(new Color(0, 0, 0));
button_1.setFont(new Font("Times New Roman", Font.ITALIC, 30));
button_1.setBounds(160, 466, 386, 49);
getContentPane().add(button_1);
label = new JLabel("");
label.setIcon(new ImageIcon("res/Diagnostika2.png"));
label.setBounds(645, 13, 349, 539);
getContentPane().add(label);
l_fon = new JLabel("");
l_fon.setForeground(Color.WHITE);
l_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));
l_fon.setBounds(-1, 0, 995, 572);
getContentPane().add(l_fon);
setVisible(true);
}
}

```

Код діагностики 3

```
public class Diagnostika3 extends JFrame {
    private static final long serialVersionUID = 1L;
    private JLabel l_16;
    private Choice choice_16;
    private Choice choice_17;
    private Choice choice_18;
    private Choice choice_19;
    private Choice choice_20;
    private JLabel l_18;
    private JLabel l_19;
    private JLabel l_20;
    private JLabel l_17;
    private final JLabel next3;
    private JButton button;
    private JButton button_1;

    private JLabel l_fon;
    public Diagnostika3(final int i_RikNarodgenna, final int i_Stat, final int
i_Zrist, final int i_Vaga,
        final int i_Profesia, final int i_choice_6, final int i_choice_7,
        final int i_choice_8,
        final int i_choice_9, final int i_choice_10, final int i_choice_11,
        final int i_choice_12,
        final int i_choice_13, final int i_choice_14, final int
i_choice_15) {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(1000, 600);
        setResizable(false);
        setLocationRelativeTo(null);
        getContentPane().setLayout(null);
        l_16 = new JLabel("Встигаєте поснідати ?");
        l_16.setForeground(new Color(0, 0, 0));
        l_16.setFont(new Font("Times New Roman", Font.ITALIC, 21));
        l_16.setBounds(2, 92, 349, 36);
        getContentPane().add(l_16);
        choice_16 = new Choice();
```

```
choice_16.setFont(new Font("Times New Roman", Font.ITALIC,
15));
```

```
choice_16.setBounds(359, 97, 204, 31);
getContentPane().add(choice_16);
choice_16.add("Завжди снідаю");
choice_16.add("Іноді");
choice_16.add("Ніколи");
l_17 = new JLabel("Чи вживаєте Ви молочні продукти ?");
l_17.setForeground(new Color(0, 0, 0));
l_17.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_17.setBounds(2, 141, 347, 36);
getContentPane().add(l_17);
choice_17 = new Choice();
choice_17.setFont(new Font("Times New Roman", Font.ITALIC,
15));
```

```
choice_17.setBounds(359, 146, 204, 31);
getContentPane().add(choice_17);
choice_17.add("Регулярно");
choice_17.add("Іноді");
choice_17.add("Ніколи не вживаю");
l_18 = new JLabel("Ваше харчування різноманітне ?");
l_18.setForeground(new Color(0, 0, 0));
l_18.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_18.setBounds(2, 200, 347, 36);
getContentPane().add(l_18);
choice_18 = new Choice();
choice_18.setFont(new Font("Times New Roman", Font.ITALIC,
15));
```

```
choice_18.setBounds(359, 202, 204, 31);
getContentPane().add(choice_18);
choice_18.add("Постійно, мені це цікаво");
choice_18.add("Якщо є час");
choice_18.add("Ніколи");
l_19 = new JLabel("Як часто Ви вживаєте алкоголь ?");
l_19.setForeground(new Color(0, 0, 0));
l_19.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_19.setBounds(2, 261, 347, 36);
getContentPane().add(l_19);
choice_19 = new Choice();
```

```
choice_19.setFont(new Font("Times New Roman", Font.ITALIC,
15));
```

```
choice_19.setBounds(359, 263, 204, 31);
```

```
getContentPane().add(choice_19);
```

```
choice_19.add("Ніколи");
```

```
choice_19.add("Раз на 3 місяці");
```

```
choice_19.add("Раз на місяць");
```

```
choice_19.add("Частіше ніж раз на тиждень");
```

```
l_20 = new JLabel("Як часто Ви вживаєте жирну їжу ?");
```

```
l_20.setForeground(new Color(0, 0, 0));
```

```
l_20.setFont(new Font("Times New Roman", Font.ITALIC, 21));
```

```
l_20.setBounds(2, 325, 120, 36);
```

```
getContentPane().add(l_20);
```

```
choice_20 = new Choice();
```

```
choice_20.setFont(new Font("Times New Roman", Font.ITALIC,
15));
```

```
choice_20.setBounds(359, 327, 204, 31);
```

```
getContentPane().add(choice_20);
```

```
choice_20.add("Дуже рідко");
```

```
choice_20.add("Раз на тиждень");
```

```
choice_20.add("Не уявляю свого життя без неї");
```

```
next3 = new JLabel("16-20/25");
```

```
next3.setForeground(Color.BLACK);
```

```
next3.setHorizontalAlignment(SwingConstants.CENTER);
```

```
next3.setFont(new Font("Times New Roman", Font.BOLD, 25));
```

```
next3.setBounds(12, 434, 120, 49);
```

```
getContentPane().add(next3);
```

```
button = new JButton("Перейти до інших запитань");
```

```
button.addActionListener(new ActionListener() {
```

```
public void actionPerformed(ActionEvent e) {
```

```
String s_choice_16 = choice_16.getSelectedItem();
```

```
String s_choice_17 = choice_17.getSelectedItem();
```

```
String s_choice_18 = choice_18.getSelectedItem();
```

```
String s_choice_19 = choice_19.getSelectedItem();
```

```
String s_choice_20 = choice_20.getSelectedItem();
```

```
int i_choice_16 = 0;
```

```
int i_choice_17 = 0;
```

```
int i_choice_18 = 0;
```



```
int i_choice_19 = 0;
int i_choice_20 = 0;
switch (s_choice_16) {
case "Завжди снідаю": {
    i_choice_16 = 0;
}
    break;
case "Іноді": {
    i_choice_16 = 1;
}
    break;
case "Ніколи": {
    i_choice_16 = 2;
}
    break;
}

switch (s_choice_17) {
case "Регулярно": {
    i_choice_17 = 0;
}
    break;
case "Іноді": {
    i_choice_17 = 1;
}
    break;
case "Ніколи не вживаю": {
    i_choice_17 = 2;
}
    break;
}

switch (s_choice_18) {
case "Постійно, мені це цікаво": {
    i_choice_18 = 0;
}
    break;
case "Якщо є час": {
    i_choice_18 = 1;
}
}
```

```
        break;
    case "Ніколи": {
        i_choice_18 = 2;
    }
    break;
}
switch (s_choice_19) {
    case "Ніколи": {
        i_choice_19 = 0;
    }
    break;
    case "Раз на 3 місяці": {
        i_choice_19 = 1;
    }
    break;
    case "Раз на місяць": {
        i_choice_19 = 2;
    }
    break;
    case "Частіше ніж раз на тиждень": {
        i_choice_19 = 3;
    }
    break;
}

switch (s_choice_20) {
    case "Дуже рідко": {
        i_choice_20 = 0;
    }
    break;
    case "Раз на тиждень": {
        i_choice_20 = 1;
    }
    break;
    case "Не уявляю свого життя без неї": {
        i_choice_20 = 2;
    }
    break;
}
```

```

        new Diagnostika4(i_RikNarodgenna, i_Stat, i_Zrist,
        i_Vaga, i_Profesia, i_choice_6, i_choice_7,
                                i_choice_8, i_choice_9, i_choice_10,
        i_choice_11, i_choice_12, i_choice_13, i_choice_14,
                                i_choice_15, i_choice_16, i_choice_17,
        i_choice_18, i_choice_19, i_choice_20);
        setVisible(false);
    }
});
button.setForeground(new Color(0, 0, 0));
button.setFont(new Font("Times New Roman", Font.ITALIC, 28));
button.setBounds(160, 404, 386, 49);
getContentPane().add(button);
button_1 = new JButton("Меню");
button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Object[] options = { "Так, перейти", "Ні, залишитись"
    };
        int i_menu = JOptionPane.showOptionDialog(null,
        "Бажаєте перейти в меню ? Введені дані не збережуться.",
        "Попередження",
        JOptionPane.YES_NO_CANCEL_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, options,
        null);
        if (i_menu == 0) {
            new Menu("Дієтолог", null);
            setVisible(false);
        }
    }
});
button_1.setForeground(new Color(0, 0, 0));
button_1.setFont(new Font("Times New Roman", Font.ITALIC, 30));
button_1.setBounds(160, 466, 386, 49);
getContentPane().add(button_1);
JLabel label = new JLabel("");
label.setBounds(569, 0, 425, 565);
label.setIcon(new ImageIcon("res/Diagnostika3.png"));
getContentPane().add(label);

```

```

l_fon = new JLabel("");
l_fon.setForeground(Color.WHITE);
l_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));
l_fon.setBounds(-1, 0, 995, 572);
getContentPane().add(l_fon);
setVisible(true);
}
}

```

Додаток I

Код діагностики 4

```

public class Diagnostika4 extends JFrame {
    private static final long serialVersionUID = 1L;
    private final JLabel l_21;
    private Choice choice_21;
    private Choice choice_22;
    private Choice choice_23;
    private Choice choice_24;
    private Choice choice_25;
    private final JLabel l_23;
    private final JLabel l_24;
    private final JLabel l_25;
    private final JLabel l_22;

    private final JLabel next4;
    private final JButton button;
    private final JButton button_1;
    private final JLabel label;
    private final JLabel l_fon;

    public Diagnostika4(final int i_RikNarodgenna, final int i_Stat, final int
i_Zrist, final int i_Vaga,
        final int i_Profesia, final int i_choice_6, final int i_choice_7,
        final int i_choice_8,
        final int i_choice_9, final int i_choice_10, final int i_choice_11,
        final int i_choice_12,
        final int i_choice_13, final int i_choice_14, final int
i_choice_15, final int i_choice_16,
        final int i_choice_17, final int i_choice_18, final int
i_choice_19, final int i_choice_20) {

```

```

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(1000, 600);
setResizable(false);
setLocationRelativeTo(null);
getContentPane().setLayout(null);
l_21 = new JLabel("Ви вважаєте себе солодкоїжкою ?");
l_21.setForeground(new Color(0, 0, 0));
l_21.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_21.setBounds(2, 357, 380, 36);
getContentPane().add(l_21);
choice_21 = new Choice();
choice_21.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_21.setBounds(390, 370, 233, 24);
getContentPane().add(choice_21);
choice_21.add("Не частіше ніж раз на тиждень");
choice_21.add("Раз на кілька днів ласую");
choice_21.add("Не люблю");
choice_21.add("І дня не проживу без солодкого");
l_22 = new JLabel("Чи впливає стрес на ваше харчування ?");
l_22.setForeground(new Color(0, 0, 0));
l_22.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_22.setBounds(2, 136, 378, 36);
getContentPane().add(l_22);
choice_22 = new Choice();
choice_22.setFont(new Font("Times New Roman", Font.ITALIC,
15));
choice_22.setBounds(390, 149, 294, 24);
getContentPane().add(choice_22);
choice_22.add("Стрес ніяк не впливає на моє харчування");
choice_22.add("Їжа допомагає мені зняти стрес");
choice_22.add("Під час стресу забуваю їсти");
l_23 = new JLabel("Доводилось Вам сидіти на дієтах чи голодати
?");
l_23.setForeground(new Color(0, 0, 0));
l_23.setFont(new Font("Times New Roman", Font.ITALIC, 18));
l_23.setBounds(2, 216, 380, 36);
getContentPane().add(l_23);
choice_23 = new Choice();

```

```
choice_23.setFont(new Font("Times New Roman", Font.ITALIC,
15));
```

```
choice_23.setBounds(388, 227, 235, 24);
getContentPane().add(choice_23);
choice_23.add("Ніколи");
choice_23.add("1 раз");
choice_23.add("Кільки разів");
choice_23.add("Постійно сижду на якійсь дієті");
l_24 = new JLabel("Курите ?");
l_24.setForeground(new Color(0, 0, 0));
l_24.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_24.setBounds(2, 287, 380, 36);
getContentPane().add(l_24);
choice_24 = new Choice();
choice_24.setFont(new Font("Times New Roman", Font.ITALIC,
```

```
15));
```

```
choice_24.setBounds(388, 297, 235, 24);
getContentPane().add(choice_24);
choice_24.add("Ніколи");
choice_24.add("Берусь і відразу кидаю");
choice_24.add("Іноді, пачку на тиждень");
choice_24.add("Так, що най менше пачку в день");
l_25 = new JLabel("Як ви ставитесь до їжі «за компанію» ?");
l_25.setForeground(new Color(0, 0, 0));
l_25.setFont(new Font("Times New Roman", Font.ITALIC, 21));
l_25.setBounds(2, 61, 380, 36);
getContentPane().add(l_25);
choice_25 = new Choice();
choice_25.setFont(new Font("Times New Roman", Font.ITALIC,
```

```
15));
```

```
choice_25.setBounds(388, 71, 296, 24);
getContentPane().add(choice_25);
choice_25.add("Ніколи не дозволяю собі такого");
choice_25.add("Іноді потрапляю на банкет");
choice_25.add("Люблю із друзями так обідати/вечеряти");
choice_25.add("Завжди так харчуюсь");
next4 = new JLabel("21-25/25");
next4.setForeground(Color.BLACK);
next4.setHorizontalAlignment(SwingConstants.CENTER);
```

```
next4.setFont(new Font("Times New Roman", Font.BOLD, 25));
next4.setBounds(12, 460, 124, 49);
getContentPane().add(next4);
button = new JButton("Перейти до інших запитань");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String s_choice_21 = choice_21.getSelectedItem();
        String s_choice_22 = choice_22.getSelectedItem();
        String s_choice_23 = choice_23.getSelectedItem();
        String s_choice_24 = choice_24.getSelectedItem();
        String s_choice_25 = choice_25.getSelectedItem();
        int i_choice_21 = 0;
        int i_choice_22 = 0;
        int i_choice_23 = 0;
        int i_choice_24 = 0;
        int i_choice_25 = 0;

        switch (s_choice_21) {
            case "Не частіше ніж раз на тиждень": {
                i_choice_21 = 0;
            }
            break;
            case "Раз на кілька днів ласую": {
                i_choice_21 = 1;
            }
            break;
            case "Не люблю": {
                i_choice_21 = 2;
            }
            break;
            case "І дня не проживу без солодкого": {
                i_choice_21 = 3;
            }
            break;
        }
        switch (s_choice_22) {
            case "Стрес ніяк не впливає на моє харчування": {
                i_choice_22 = 0;
            }
        }
    }
});
```

```
break;
case "Їжа допомагає мені зняти стрес": {
    i_choice_22 = 1;
}
break;
case "Під час стресу забуваю їсти": {
    i_choice_22 = 2;
}
break;
}
switch (s_choice_23) {
case "Ніколи": {
    i_choice_23 = 0;
}
break;
case "1 раз": {
    i_choice_23 = 1;
}
break;
case "Кільки разів": {
    i_choice_23 = 2;
}
break;
case "Постійно сижду на якійсь дієті": {
    i_choice_23 = 3;
}
break;
}
switch (s_choice_24) {
case "Ніколи": {
    i_choice_24 = 0;
}
break;
case "Берусь і відразу кидаю": {
    i_choice_24 = 1;
}
break;
case "Іноді, пачку на тиждень": {
    i_choice_24 = 2;
}
```



```

    }
    break;
    case "Так, щонайменше пачку в день": {
        i_choice_24 = 3;
    }
    break;
}

switch (s_choice_25) {
    case "Ніколи не дозволяю собі такого": {
        i_choice_25 = 0;
    }
    break;
    case "Іноді потрапляю на банкет": {
        i_choice_25 = 1;
    }
    break;
    case "Люблю із друзями так обідати/вечеряти": {
        i_choice_25 = 2;
    }
    break;
    case "Завжди так харчуюсь": {
        i_choice_25 = 3;
    }
    break;
}
new

```

```

DiagnostikaVusnovokObchuslenna(i_RikNarodgenna, i_Stat, i_Zrist, i_Vaga,
i_Profesia, i_choice_6,

```

```

        i_choice_7, i_choice_8, i_choice_9,
        i_choice_10, i_choice_11, i_choice_12, i_choice_13,
        i_choice_14, i_choice_15, i_choice_16,
        i_choice_17, i_choice_18, i_choice_19, i_choice_20,
        i_choice_21, i_choice_22, i_choice_23,
        i_choice_24, i_choice_25);
    setVisible(false);
}
});
button.setForeground(new Color(0, 0, 0));

```

```

button.setFont(new Font("Times New Roman", Font.ITALIC, 28));
button.setBounds(160, 425, 386, 49);
getContentPane().add(button);
button_1 = new JButton("Меню");
button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Object[] options = { "Так, перейти", "Ні, залишитись"
};
        int i_menu = JOptionPane.showOptionDialog(null,
"Бажаєте перейти в меню ? Введені дані не збережуться.",
        "Попередження",
JOptionPane.YES_NO_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, options,
        null);
        if (i_menu == 0) {
            new Menu("Дієтолог", null);
            setVisible(false);
        }
    }
});
button_1.setForeground(new Color(0, 0, 0));
button_1.setFont(new Font("Times New Roman", Font.ITALIC, 30));
button_1.setBounds(160, 487, 386, 49);
getContentPane().add(button_1);
label = new JLabel("");
label.setBounds(629, 0, 365, 565);
label.setIcon(new ImageIcon("res/Diagnostika4.png"));
getContentPane().add(label);

l_fon = new JLabel("");
l_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));
l_fon.setBounds(-1, 0, 995, 572);
getContentPane().add(l_fon);
setVisible(true);
}
}

```

Додаток Л

Код обчислення діагностики

```
public class DiagnostikaVusnovokObchuslenna {
```

```
    private Formatter formatter_dani;
```

```
    private Scanner scanner_Korustuvac;
```

```
    String[][] Reading_PIP = new String[1][2];
```

```
    private String Reading_Name;
```

```
    private String Reading_Prizvusko;
```

```
    private Scanner scanner_Dani;
```

```
    String[][] Reading_Dani = new String[1][28];
```

```
    private String Reading_1;
```

```
    private String Reading_3;
```

```
    private String Reading_4;
```

```
    private String Reading_6;
```

```
    private String Reading_7;
```

```
    private String Reading_8;
```

```
    private String Reading_9;
```

```
    private String Reading_10;
```

```
    private String Reading_11;
```

```
    private String Reading_12;
```

```
    private String Reading_13;
```

```
    private String Reading_16;
```

```
    private String Reading_17;
```

```
    private String Reading_18;
```

```
    private String Reading_19;
```

```
    private String Reading_20;
```

```
    private String Reading_21;
```

```
    private String Reading_25;
```

```
    private int i_Reading_1;
```

```
    private int i_Reading_3;
```

```
    private int i_Reading_4;
```

```
    private int i_Reading_6;
```

```
    private int i_Reading_7;
```

```
    private int i_Reading_8;
```

```
    private int i_Reading_9;
```

```
    private int i_Reading_10;
```

```
    private int i_Reading_11;
```

```
    private int i_Reading_12;
```

```
    private int i_Reading_13;
```

```
    private int i_Reading_16;
```

```
    private int i_Reading_17;
```

```

private int i_Reading_18;
private int i_Reading_19;
private int i_Reading_20;
private int i_Reading_21;
private int i_Reading_25;
int i_Error = 0;
String s_IMT = "";
String s_RekomendaciiZagalni = "Їсти бажано в один і той же час, який
ви повинні встановити "
+ "собі самі (у зв'язку із характером праці).\n\n"
+ "Склянка води без газу за півгодини до їжі зменшує
апетит.\n\n"
+ "Їжу потрібно починати з нежирного супу або
салату.\n\n"
+ "Слідкуйте за розміром порцій і не беріть добавки.\n\n"
+ "Їжте повільно, ретельно пережовуючи. При швидкому
поглинанні "
+ "їжі відчуття насичення \"відстає\" від кількості
поглинутої їжі."
+ " Внаслідок чого з'їдається зайва кількість їжі.\n\n"
+ "Купуйте продукти за попередньо складеним
списком.\n\n" + "Не купуйте продукти у стані голоду.\n\n"
+ "Купуючи продукти, завжди читати етикетку про склад
(жири, вуглеводи, кілокалорії).\n\n"
+ "Не довіряти словам \"дієтичний\" або
\"низькокалорійний\", дивитися на цифри.\n\n";
String s_RekomendaciiPerconalni = "";
int i_Vegeterianstvo = 0;
int i_Ocinka = 0;
String s_Dunamika = "";
@SuppressWarnings("static-access")
public DiagnostikaVusnovokObchuslenna(final int i_RikNarodgenna, final
int i_Stat, final int i_Zrist,
final int i_Vaga, final int i_Profesia, final int i_choice_6, final
int i_choice_7, final int i_choice_8,
final int i_choice_9, final int i_choice_10, final int i_choice_11,
final int i_choice_12,
final int i_choice_13, final int i_choice_14, final int
i_choice_15, final int i_choice_16,

```

```

        final int i_choice_17, final int i_choice_18, final int
i_choice_19, final int i_choice_20,
        final int i_choice_21, final int i_choice_22, final int
i_choice_23, final int i_choice_24,
        final int i_choice_25) {
    Calendar calendar = Calendar.getInstance();
    int den = calendar.get(calendar.DAY_OF_MONTH);
    int misac = calendar.get(calendar.MONTH) + 1;
    int rik = calendar.get(calendar.YEAR);
    String dani = i_RikNarodgenna + " " + i_Stat + " " + i_Zrist + " " +
i_Vaga + " " + i_Profesia + " "
        + i_choice_6 + " " + i_choice_7 + " " + i_choice_8 + " "
+ i_choice_9 + " " + i_choice_10 + " "
        + i_choice_11 + " " + i_choice_12 + " " + i_choice_13 +
" " + i_choice_14 + " " + i_choice_15 + " "
        + i_choice_16 + " " + i_choice_17 + " " + i_choice_18 +
" " + i_choice_19 + " " + i_choice_20 + " "
        + i_choice_21 + " " + i_choice_22 + " " + i_choice_23 +
" " + i_choice_24 + " " + i_choice_25 + " "
        + den + " " + misac + " " + rik;
    try {
        scanner_Korustuvac = new Scanner(new
File("res/TumcasoviFaylu/Korustuvac.txt"));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Помилка введення");
    }

    while (scanner_Korustuvac.hasNext()) {
        for (int row = 0; row < Reading_PIP.length; row++) {
            for (int col = 0; col < Reading_PIP[row].length; col++) {
                Reading_PIP[row][col] =
scanner_Korustuvac.next();
                if (col == 0) {
                    Reading_Name = Reading_PIP[row][col];
                }
                if (col == 1) {
                    Reading_Prizvusko =
Reading_PIP[row][col];
                }
            }
        }
    }
}

```

```

    }
    }
    scanner_Korustuvac.close();
    String PIP = Reading_Name + " " + Reading_Prizvusko;
    final File file = new File("res/Dani/" + PIP + ".txt");
    if (file.exists()) {
        try {
            scanner_Dani = new Scanner(new File("res/Dani/" + PIP
+ ".txt"));
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Помилка
введення");
        }
        while (scanner_Dani.hasNext()) {
            for (int row = 0; row < Reading_Dani.length; row++) {
                for (int col = 0; col < Reading_Dani[row].length;
col++) {
                    Reading_Dani[row][col] =
scanner_Dani.next();
                    switch (col) {
                        case 0:
                            Reading_1 =
Reading_Dani[row][col];
                            break;
                        case 2:
                            Reading_3 =
Reading_Dani[row][col];
                            break;
                        case 3:
                            Reading_4 =
Reading_Dani[row][col];
                            break;
                        case 5:
                            Reading_6 =
Reading_Dani[row][col];
                            break;
                        case 6:
                            Reading_7 =

```

Reading_Dani[row][col];

break;

case 7:

Reading_8 =

Reading_Dani[row][col];

break;

case 8:

Reading_9 =

Reading_Dani[row][col];

break;

case 9:

Reading_10 =

Reading_Dani[row][col];

break;

case 10:

Reading_11 =

Reading_Dani[row][col];

break;

case 11:

Reading_12 =

Reading_Dani[row][col];

break;

case 12:

Reading_13 =

Reading_Dani[row][col];

break;

case 15:

Reading_16 =

Reading_Dani[row][col];

break;

case 16:

Reading_17 =

Reading_Dani[row][col];

break;

case 17:

Reading_18 =

Reading_Dani[row][col];

break;

case 18:

```
        Reading_Dani[row][col];
        Reading_19 =
            break;
        case 19:
            Reading_20 =
                break;
        case 20:
            Reading_21 =
                break;
        case 24:
            Reading_25 =
                break;
    }
}
scanner_Dani.close();
i_Reading_1 = Integer.parseInt(Reading_1);
i_Reading_3 = Integer.parseInt(Reading_3);
i_Reading_4 = Integer.parseInt(Reading_4);
i_Reading_6 = Integer.parseInt(Reading_6);
i_Reading_7 = Integer.parseInt(Reading_7);
i_Reading_8 = Integer.parseInt(Reading_8);
i_Reading_9 = Integer.parseInt(Reading_9);
i_Reading_10 = Integer.parseInt(Reading_10);
i_Reading_11 = Integer.parseInt(Reading_11);
i_Reading_12 = Integer.parseInt(Reading_12);
i_Reading_13 = Integer.parseInt(Reading_13);
i_Reading_16 = Integer.parseInt(Reading_16);
i_Reading_17 = Integer.parseInt(Reading_17);
i_Reading_18 = Integer.parseInt(Reading_18);
i_Reading_19 = Integer.parseInt(Reading_19);
i_Reading_20 = Integer.parseInt(Reading_20);
i_Reading_21 = Integer.parseInt(Reading_21);
i_Reading_25 = Integer.parseInt(Reading_25);
if (i_Reading_1 != i_RikNarodgenna) {
```



```

        i_Error = 1;
    }
    double d_IMT0 = i_Vaga
        / (((i_Zrist / 100) + ((i_Zrist % 100) * 0.01)) *
        ((i_Zrist / 100) + ((i_Zrist % 100) * 0.01)));
    double d_IMT = d_IMT0;
    if (d_IMT0 >= 18 && d_IMT0 <= 25) {
        s_IMT = s_IMT + "Маса тіла у межах норми. Вітаємо
!\n\n";
    }
    if (d_IMT0 <= 18) {
        s_IMT = s_IMT + "Недостатня маса тіла. Необхідні
термінові заходи.\n\n";
    }
    if (d_IMT0 >= 25) {
        s_IMT = s_IMT + "Надмірна маса тіла. Необхідні
термінові заходи.\n\n";
    }
    double d_IMT1 = i_Reading_4 / (((i_Reading_3 / 100) +
        ((i_Reading_3 % 100) * 0.01))
        * ((i_Reading_3 / 100) + ((i_Reading_3 % 100) *
        0.01)));
    d_IMT1 = d_IMT1 - 22;
    d_IMT0 = d_IMT0 - 22;

    if (d_IMT0 < 0) {
        d_IMT0 = 0 - d_IMT0;
    }
    if (d_IMT1 < 0) {
        d_IMT1 = 0 - d_IMT1;
    }
    if (d_IMT1 == d_IMT0) {
        s_Dunamika = s_Dunamika + "Індекс маси Вашого
тіла стабільний.\n\n";
    } else {
        if (d_IMT1 < d_IMT0) {
            s_Dunamika = s_Dunamika + "Індекс маси
Вашого тіла раніше був кращий. Ви втрачаєте форму.\n\n";
        } else {

```

```

s_Dunamika = s_Dunamika + "Індекс маси
Вашого тіла став кращим. Ви набираєте форму.\n\n";
}
}
int i_PereidannaDunamika = 0;
if (i_choice_6 != 0 && i_choice_13 == 0 || i_choice_6 != 0 &&
i_choice_13 == 1) {
s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Спосперігається недоїдання,
намагайтесь їсти частіше.\n\n";
i_Ocinka = i_Ocinka + 2;
i_PereidannaDunamika = 1;
}
if (i_choice_6 != 0 && i_choice_13 == 2 || i_choice_6 != 0 &&
i_choice_13 == 3) {
s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Недоречні прийоми їжі. Намагайтесь
їсти частіше, але меншими порціями.\n\n";
i_Ocinka = i_Ocinka + 2;
i_PereidannaDunamika = 2;
}
if (i_choice_6 == 0 && i_choice_13 == 2 || i_choice_6 == 0
&& i_choice_13 == 3) {
s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Спосперігається переїдання,
намагайтесь їсти рідше."
+ "Необхідно іноді влаштовувати
розвантажувальний день, наприклад, "
+ "2 літри знежиреного кефіру або
молока, або 1 кг. кабачків і 200 г. "
+ "нежирного м'яса за день. При цьому
воду або чай можна пити без обмежень.\n\n";
i_Ocinka = i_Ocinka + 2;
i_PereidannaDunamika = 3;
}
if (i_Reading_6 != 0 && i_Reading_13 == 0 || i_Reading_6 !=
0 && i_Reading_13 == 1) {
if (i_PereidannaDunamika == 0) {
s_Dunamika = s_Dunamika + "Ви вирішили

```

```

проблему із недоїданням. Вітаю.\n\n";
    }
    if (i_PereidannaDunamika == 1) {
        s_Dunamika = s_Dunamika + "Проблема із
недоїданням у Вас так і залишилась.\n\n";
    }
    if (i_PereidannaDunamika == 2) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із недоїданням. Але тепер у Вас "
        + "Проблема із недоречними
прийоми їжі."
        + " Намагайтесь їсти частіше, але
меншими порціями.\n\n";
    }
    if (i_PereidannaDunamika == 3) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із недоїданням. Але тепер у Вас "
        + "проблема із переїданням у
Вас.\n\n";
    }
}
if (i_Reading_6 != 0 && i_Reading_13 == 2 || i_Reading_6 !=
0 && i_Reading_13 == 3) {
    if (i_PereidannaDunamika == 0) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із недоречними прийоми їжі.\n\n";
    }
    if (i_PereidannaDunamika == 1) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із недоречними прийоми їжі. Але тепер у Вас \n\n"
        + " Проблема із недоїданням.\n\n";
    }
    if (i_PereidannaDunamika == 2) {
        s_Dunamika = s_Dunamika + "Проблема із
недоречними прийоми їжі у Вас так і залишилась.\n\n";
    }
    if (i_PereidannaDunamika == 3) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із недоречними прийоми їжі. Але тепер у Вас \n\n"

```

```

        + "проблема із переїданням.\n\n";
    }
}
if (i_Reading_6 == 0 && i_Reading_13 == 2 || i_Reading_6 ==
0 && i_Reading_13 == 3) {
    if (i_PereidannaDunamika == 0) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із переїданням.\n\n";
    }
    if (i_PereidannaDunamika == 1) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із переїданням. Але тепер у Вас\n\n"
+ " Проблема із недоїданням.\n\n";
    }
    if (i_PereidannaDunamika == 2) {
        s_Dunamika = s_Dunamika + "Ви вирішили
проблему із переїданням. Але тепер у Вас\n\n"
+ " Проблема із недоречним
прийомом їжі.\n\n";
    }
    if (i_PereidannaDunamika == 3) {
        s_Dunamika = s_Dunamika + "Проблема із
переїданням у Вас так і залишилась.\n\n";
    }
}
int i_IjaPeredSnom = i_choice_8 - i_choice_10;
int i_IjaPeredSnomDunamika = 0;
if (i_IjaPeredSnom == 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Вам необхідно вечеряти на годину
раніше. Ви занадто пізно їсте.\n\n";
    i_Ocinka = i_Ocinka + 1;
    i_IjaPeredSnomDunamika = 1;
}
if (i_IjaPeredSnom > 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Вам необхідно вечеряти на 2 години
раніше. Ви занадто пізно їсте.\n\n";
    i_Ocinka = i_Ocinka + 2;
}

```

```

        i_IjaPeredSnomDunamika = 2;
    }
    int i_IjaPeredSnom1 = i_Reading_8 - i_Reading_10;

    if (i_IjaPeredSnom1 < 0) {
        if (i_IjaPeredSnomDunamika == 1) {
            s_Dunamika = s_Dunamika + "Ви почали
занадто пізно вечеряти. "
+ "Вам необхідно вечеряти на
годину раніше.\n\n";
        }
        if (i_IjaPeredSnomDunamika == 2) {
            s_Dunamika = s_Dunamika + "Ви почали дуже
пізно вечеряти. "
+ "Вам необхідно вечеряти на 2
години раніше.\n\n";
        }
    }
    if (i_IjaPeredSnom1 == 0) {
        if (i_IjaPeredSnomDunamika == 0) {
            s_Dunamika = s_Dunamika + "Ви почали
вечеряти значно раніше, Молодець.\n\n";
        }
        if (i_IjaPeredSnomDunamika == 1) {
            s_Dunamika = s_Dunamika + "Ви так і не почали
раніше харчуватись."
+ "Вам необхідно вечеряти на
годину раніше. Ви занадто пізно їсте.\n\n";
        }
        if (i_IjaPeredSnomDunamika == 2) {
            s_Dunamika = s_Dunamika + "Ви почали ще
пізніше харчуватись! "
+ "Вам необхідно вечеряти на дві
години раніше. Ви дуже пізно їсте.\n\n";
        }
    }
    if (i_IjaPeredSnom1 > 0) {
        if (i_IjaPeredSnomDunamika == 0) {
            s_Dunamika = s_Dunamika + "Ви почали

```

```

вечеряти значно раніше, Молодень." + " Ви виправились.\n\n";
    }
    if (i_IjaPeredSnomDunamika == 1) {
        s_Dunamika = s_Dunamika + "Ви почали
вечеряти значно раніше, Молодень."
        + "Але це ще не кінець,
продовжуйте.\n\n";
    }
    if (i_IjaPeredSnomDunamika == 2) {
        s_Dunamika = s_Dunamika + "Ви так і не почали
раніше харчуватись."
        + "Вам необхідно вечеряти на 2
години раніше. Ви занадто пізно їсте.\n\n";
    }
}
int i_Reading_7_Diagnostika = 0;
if (i_choice_7 != 0) {
    s_RekomendaciiPersonalni = s_RekomendaciiPersonalni
+ "Основним прийомом їжі має бути обід.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_7_Diagnostika = 1;
}
if (i_Reading_7 != 0) {
    if (i_Reading_7_Diagnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви зробили
основним прийомом їжі обід, Молодень.\n\n";
    }
    if (i_Reading_7_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не
зробили основним прийомом їжі обід, виправляйтесь.\n\n";
    }
} else {
    if (i_Reading_7_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви почали робити
помилки. Поверніть попереднє ставлення до обіду. "
        + "Основним прийомом їжі має
бути обід.\n\n";
    }
}
}
}

```

```

int i_Reading_9_Diagnostika = 0;
if (i_choice_9 == 2) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Вам варто вживати менше солодкого.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_9_Diagnostika = 1;
}
if (i_Reading_9 == 2) {
    if (i_Reading_9_Diagnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви почали менше
вживати солодкого, Вітаю.\n\n";
    }
    if (i_Reading_9_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не
зменшили вживання солодкого, виправляйтесь.\n\n";
    }
} else {
    if (i_Reading_9_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Поверніть
попереднє ставлення до солодкого. "
+ "Ви почали занадто багато його
вживати.\n\n";
    }
}
int i_Reading_11_Diagnostika = 0;
if (i_choice_11 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Варто більше вживати овочів та фруктів.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_11_Diagnostika = 1;
}
if (i_Reading_11 != 0) {
    if (i_Reading_11_Diagnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви більше стали
вживати овочів та фруктів, так тримати.\n\n";
    }
    if (i_Reading_11_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не стали
більше вживати овочів та фруктів, виправляйтесь.\n\n";
    }
}

```

```

    }
  } else {
    if (i_Reading_11_Diagnostika == 1) {
      s_Dunamika = s_Dunamika + "Поверніть
попереднє ставлення до овочів та фруктів. "
      + "Ви почили занадто мало їх
вживати.\n\n";
    }
  }
  int i_Reading_12_Diagnostika = 0;
  if (i_choice_12 != 0) {
    s_RekomendaciiPersonalni = s_RekomendaciiPersonalni
    + "Людина має пити не менше 2 літрів
води на день.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_12_Diagnostika = 1;
  }
  if (i_Reading_12 != 0) {
    if (i_Reading_12_Diagnostika == 0) {
      s_Dunamika = s_Dunamika + "Ви більше стали
вживати води, Молодець.\n\n";
    }
    if (i_Reading_12_Diagnostika == 1) {
      s_Dunamika = s_Dunamika + "Ви так і не стали
більше вживати води, виправляйтесь.\n\n";
    }
  } else {
    if (i_Reading_12_Diagnostika == 1) {
      s_Dunamika = s_Dunamika + "Ви робите нові
помилки. Поверніть попереднє ставлення до води. "
      + "Ви почили занадто мало її
вживати.\n\n";
    }
  }
  if (i_Profesia != 1) {
    s_RekomendaciiPersonalni = s_RekomendaciiPersonalni
    + "Оскільки у Вас інтелектуальна робота,
то Вам необхідно насичувати свій раціон"
    + " кашами, злаками, овочами.\n\n";
  }

```



```

    }
    if (i_Profesia == 1 && i_choice_14 == 0) {
        s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
            + "Оскільки у Вас фізична робота та Ви
займаєтесь спортом,"
            + " то Вам необхідно насичувати свій
раціон" + " м'ясом, яйцями, рибою, горіхами.\n\n";
    }
    if (i_choice_15 == 2) {
        i_Vegeterianstvo = 1;
    }
    int i_Reading_16_Diadnostika = 0;
    if (i_choice_16 != 0) {
        s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
            + "Завжди необхідно снідати.\n\n";
        i_Ocinka = i_Ocinka + 2;
        i_Reading_16_Diadnostika = 1;
    }
    if (i_Reading_16 != 0) {
        if (i_Reading_16_Diadnostika == 0) {
            s_Dunamika = s_Dunamika + "Ви почали
снідати, так тримати.\n\n";
        }
        if (i_Reading_16_Diadnostika == 1) {
            s_Dunamika = s_Dunamika + "Ви так і не почали
снідати, виправляйтесь.\n\n";
        }
        } else {
            if (i_Reading_16_Diadnostika == 1) {
                s_Dunamika = s_Dunamika + "Терміново знову
почніть снідати.\n\n";
            }
        }
    }
    int i_Reading_17_Diadnostika = 0;
    if (i_choice_17 != 0) {
        s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
            + "Вживайте більше молочних продуктів.\n\n";
        i_Ocinka = i_Ocinka + 2;
        i_Reading_17_Diadnostika = 1;
    }

```

```

}
if (i_Reading_17 != 0) {
    if (i_Reading_17_Diagnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви більше стали
вживати молочних продуктів, Вітаю.\n\n";
    }
    if (i_Reading_17_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не стали
більше вживати молочних продуктів, виправляйтесь.\n\n";
    }
} else {
    if (i_Reading_17_Diagnostika == 1) {
        s_Dunamika = s_Dunamika
+ "Ви робите нові помилки.
Поверніть попереднє ставлення до молочних продуктів.\n\n";
    }
}
int i_Reading_18_Diagnostika = 0;
if (i_choice_18 == 2) {
    s_RekomendaciiPersonalni = s_RekomendaciiPersonalni
+ "Намагайтесь урізноманітнювати Ваш раціон.\n\n";
    i_Ocinka = i_Ocinka + 1;
    i_Reading_18_Diagnostika = 1;
}
if (i_Reading_18 == 2) {
    if (i_Reading_18_Diagnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви
урізноманітнили свій регіон, так тримати.\n\n";
    }
    if (i_Reading_18_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не стали
урізньоманітнювати свій регіон, виправляйтесь.\n\n";
    }
} else {
    if (i_Reading_18_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Поверніть
попередній різноманітний раціон.\n\n";
    }
}
}

```

```

int i_Reading_19_Diagnostika = 0;
if (i_choice_19 == 2) {
    s_RekomendaciiPersonalni = s_RekomendaciiPersonalni
        + "Алкогoль убиває Вас і Ваш раціон.
Позбудьтесь його.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_19_Diagnostika = 1;
}
if (i_Reading_19 == 2) {
    if (i_Reading_19_Diagnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви позбулись
алкоголю, Вітаю.\n\n";
    }
    if (i_Reading_19_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не стали
унікати алкоголю, виправляйтесь.\n\n";
    }
    } else {
        if (i_Reading_19_Diagnostika == 1) {
            s_Dunamika = s_Dunamika + "Вам варто
позбутись алкоголю.\n\n";
        }
    }
}
int i_Reading_20_Diagnostika = 0;
if (i_choice_20 == 2) {
    s_RekomendaciiPersonalni = s_RekomendaciiPersonalni
        + "Старайтесь мінімізувати вживання
жирної їжі.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_20_Diagnostika = 1;
}
if (i_Reading_20 == 2) {
    if (i_Reading_20_Diagnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви позбулись
жирної їжі, Молодець.\n\n";
    }
    if (i_Reading_20_Diagnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не стали
унікати жирної їжі, виправляйтесь.\n\n";
    }
}

```

```

    }
  } else {
    if (i_Reading_20_Diadnostika == 1) {
      s_Dunamika = s_Dunamika + "Вам варто
      позбутись жирної їжі.\n\n";
    }
  }
  int i_Reading_21_Diadnostika = 0;
  if (i_choice_21 == 3) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
    + "Солодке псує ваше здоров'я.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_21_Diadnostika = 1;
  }
  if (i_Reading_21 == 3) {
    if (i_Reading_21_Diadnostika == 0) {
      s_Dunamika = s_Dunamika + "Ви позбулись
      солодкого, Вітаю.\n\n";
    }
    if (i_Reading_21_Diadnostika == 1) {
      s_Dunamika = s_Dunamika + "Ви так і не
      позбулись солодкого, виправляйтесь.\n\n";
    }
  } else {
    if (i_Reading_21_Diadnostika == 1) {
      s_Dunamika = s_Dunamika + "Вам варто
      зменшити вживання солодкого.\n\n";
    }
  }
  if (i_choice_22 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
    + "Пам'ятайте, що стрес дуже негативно
    впливає на Ваше харчування та здоров'я.\n\n";
    i_Ocinka = i_Ocinka + 2;
  }
  if (i_choice_23 == 2 && i_choice_23 == 3) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
    + "Вірте тільки тим дістам, які
    порекомендував Вам лікар. "

```

```

+ "Подрузі, книги чи Інтернет не є
надійним джерелом.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
int i_Reading_25_Diadnostika = 0;
if (i_choice_25 == 2 && i_choice_25 == 3) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Вам варто відмовитись від харчування
із друзями, колегами. Це шкодить вашому раціону.\n\n";
    i_Ocinka = i_Ocinka + 2;
    i_Reading_25_Diadnostika = 1;
}
if (i_Reading_25 == 2 && i_Reading_25 == 3) {
    if (i_Reading_25_Diadnostika == 0) {
        s_Dunamika = s_Dunamika + "Ви перестали
харчуватись за компанію, так тримати.\n\n";
    }
    if (i_Reading_25_Diadnostika == 1) {
        s_Dunamika = s_Dunamika + "Ви так і не
перестали харчуватись за компанію, виправляйтесь.\n\n";
    }
} else {
    if (i_Reading_25_Diadnostika == 1) {
        s_Dunamika = s_Dunamika + "Вам варто
перестати харчуватись за компанію.\n\n";
    }
}
if (i_choice_24 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Вам необхідно кидати курити.
Прогнозується, що це супроводжуватиметься підвищенням апетиту"
+ ", тому дотримуйтесь раціону. Іноді
курці замінюють сигаретою прийом їжі, недопускайте цього.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
}
try {
    formatter_dani = new Formatter("res/Dani/" + PIP +
".txt");

```

```

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "System Error");
    }
    formatter_dani.format(dani);
    formatter_dani.close();
    new DiagnostikaVusnovokVizyalizacia(s_IMT, d_IMT,
s_RekomendaciiZagalni, s_RekomendaciiPerconalni,
        i_Vegeterianstvo, i_Ocinka, i_Error, s_Dunamika);
    } else {
        // System.out.println("Файл не существует.");
        try {
            formatter_dani = new Formatter("res/Dani/" + PIP +
".txt");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "System Error");
        }
        formatter_dani.format(dani);
        formatter_dani.close();
        s_Dunamika = s_Dunamika + "Оскільки Ви вперше
користуєтесь даним функціоналом, "
            + "то система не може відобразити динаміку.
Зверніться, будь ласка, пізніше.";
        double d_IMT0 = i_Vaga
            / (((i_Zrist / 100) + ((i_Zrist % 100) * 0.01)) *
((i_Zrist / 100) + ((i_Zrist % 100) * 0.01)));
        double d_IMT = d_IMT0;
        if (d_IMT0 >= 18 && d_IMT0 <= 25) {
            s_IMT = s_IMT + "Маса тіла у межах норми. Вітаємо
!\n\n";
        }
        if (d_IMT0 <= 18) {
            s_IMT = s_IMT + "Недостатня маса тіла. Необхідні
термінові заходи.\n\n";
        }
        if (d_IMT0 >= 25) {
            s_IMT = s_IMT + "Надмірна маса тіла. Необхідні
термінові заходи.\n\n";
        }
        if (i_choice_6 != 0 && i_choice_13 == 0 || i_choice_6 != 0 &&

```

```

i_choice_13 == 1) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
        + "Спосперігається недоїдання,
намагайтесь їсти частіше.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_6 != 0 && i_choice_13 == 2 || i_choice_6 != 0 &&
i_choice_13 == 3) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
        + "Недоречні прийоми їжі. Намагайтесь
їсти частіше, але меншими порціями.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_6 == 0 && i_choice_13 == 2 || i_choice_6 == 0
&& i_choice_13 == 3) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
        + "Спосперігається переїдання,
намагайтесь їсти рідше."
        + "Необхідно іноді влаштовувати
розвантажувальний день, наприклад, "
        + "2 літри знежиреного кефіру або
молока, або 1 кг. кабачків і 200 г. "
        + "нежирного м'яса за день. При цьому
воду або чай можна пити без обмежень.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
int i_IjaPeredSnom = i_choice_8 - i_choice_10;
if (i_IjaPeredSnom == 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
        + "Вам необхідно вечеряти на годину
раніше. Ви занадто пізно їсте.\n\n";
    i_Ocinka = i_Ocinka + 1;
}
if (i_IjaPeredSnom > 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
        + "Вам необхідно вечеряти на 2 години
раніше. Ви занадто пізно їсте.\n\n";
    i_Ocinka = i_Ocinka + 2;
}

```

```

if (i_choice_7 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Основним прийомом їжі має бути обід.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_9 == 2) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Вам варто вживати менше солодкого.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_11 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Варто більше вживати овочів та фруктів.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_12 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Людина має пити не менше 2 літнів
води на день.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_Profesia != 1) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Оскільки у Вас інтелектуальна робота,
то Вам необхідно насичувати свій раціон"
+ " кашами, злаками, овочами.\n\n";
}
if (i_Profesia == 1 && i_choice_14 == 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Оскільки у Вас фізична робота та Ви
займаєтесь спортом,"
+ " то Вам необхідно насичувати свій
раціон" + " м'ясом, яйцями, рибою, горіхами.\n\n";
}
if (i_choice_15 == 2) {
    i_Vegeterianstvo = 1;
}
if (i_choice_16 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni

```



```

+ "Завжди необхідно снідати.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_17 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Вживайте більше молочних продуктів.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_18 == 2) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Намагайтесь урізноманітнювати Ваш раціон.\n\n";
    i_Ocinka = i_Ocinka + 1;
}
if (i_choice_19 == 2) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Алкоголь убиває Вас і Ваш раціон.
Позбудьтесь його.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_20 == 2) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Старайтесь мінімізувати вживання
жирної їжі.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_21 == 3) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Солодке псує ваше здоров'я.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_22 != 0) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
+ "Пам'ятайте, що стрес дуже негативно
впливає на Ваше харчування та здоров'я.\n\n";
    i_Ocinka = i_Ocinka + 2;
}
if (i_choice_23 == 2 && i_choice_23 == 3) {
    s_RekomendaciiPerconalni = s_RekomendaciiPerconalni

```

```

        + "Вірте тільки тим дістам, які
порекомендував Вам лікар. "
        + "Подрузі, книги чи Інтернет не є
надійним джерелом.\n\n";
        i_Ocinka = i_Ocinka + 2;
    }
    if (i_choice_25 == 2 && i_choice_25 == 3) {
        s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
        + "Вам варто відмовитись від харчування
із друзями, колегами. Це шкодить вашому раціону.\n\n";
        i_Ocinka = i_Ocinka + 2;
    }
    if (i_choice_24 != 0) {
        s_RekomendaciiPerconalni = s_RekomendaciiPerconalni
        + "Вам необхідно кидати курити.
Прогнозується, що це супроводжуватиметься підвищенням апетиту"
        + ", тому дотримуйтеся раціону. Іноді
курці замінюють сигаретою прийом їжі, недопускайте цього.\n\n";
        i_Ocinka = i_Ocinka + 2;
    }
    new DiagnostikaVusnovokVizyalizacia(s_IMT, d_IMT,
s_RekomendaciiZagalni, s_RekomendaciiPerconalni,
        i_Vegeterianstvo, i_Ocinka, i_Error, s_Dunamika);
    }
}
}

```

Додаток М

Код візуалізації обчислення діагностики

```

public class DiagnostikaVusnovokVizyalizacia extends JFrame {
    private static final long serialVersionUID = 1L;
    private final ButtonGroup group_Rezultatu;
    private JTextPane textPane_Rezultatu;
    private final JRadioButton rB_Zagalno;
    private final JRadioButton rB_Induividualno;
    private final JRadioButton rB_Dunamika;
    private final JButton b_Vegan;
    private final JRadioButton rB_Ditu;
    String s_RekomendaciiDlaDitey = "Не обмежуйте дитину в їжі. Це може

```

призвести до анорексії, булімії "

+ "та ряду інших хвороб. Не забувайте, що організм формується.\n\n"

+ "Завжди майте під рукою здорову їжу. Наприклад, робіть перекус фруктами.\n\n"

+ "Не діліть їжу на погану та хорошу. Краще робіть посилання на корисність.\n\n"

+ "Хваліть за вибір їжі.\n\n" + "Будьте компромісними.

Цукерки можна замінити сухофруктами.\n\n"

+ "Заохочуйте дитину не їжею, а фізичними розвагами.\n\n"

+ "Сімейна вечеря - це традиція.\n\n"

+ "Радьтесь із дітьми, щоб знати, що їм подобається.\n\n"

+ "Не забувайте радитись із експертами у галузі

харчування.\n\n"

+ "Навчайте дітей правильно харчуватись.\n\n";

```
private final JLabel l_fon;  
boolean d_Ocinka = false;  
boolean d_Error = false;  
String s_Error = "";  
private final JLabel l_IMT_komentar;  
public DiagnostikaVusnovokVizyalizacia(final String s_IMT, double  
d_IMT0, final String s_RekomendaciiZagalni,  
final String s_RekomendaciiPerconalni, int i_Vegeterianstvo,  
final int i_Ocinka, final int i_Error,  
final String s_Dunamika) {  
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
setSize(1000, 600);  
setResizable(false);  
setLocationRelativeTo(null);  
getContentPane().setLayout(null);  
JButton b_Menu = new JButton("Меню");  
b_Menu.setForeground(new Color(0, 0, 0));  
b_Menu.addActionListener(new ActionListener() {  
public void actionPerformed(ActionEvent arg0) {  
new Menu("Дієтолог", null);  
setVisible(false);  
}  
});
```

```

b_Menu.setFont(new Font("Times New Roman", Font.ITALIC, 30));
b_Menu.setBounds(608, 516, 386, 49);
getContentPane().add(b_Menu);
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(0, 44, 610, 521);
getContentPane().add(scrollPane);
textPane_Rezultatu = new JTextPane();
scrollPane.setViewportViewView(textPane_Rezultatu);
textPane_Rezultatu.setText(s_IMT + s_RekomendaciiPerconalni);
textPane_Rezultatu.setForeground(new Color(183, 192, 107));
textPane_Rezultatu.setFont(new Font("Times New Roman",
Font.PLAIN, 25));
textPane_Rezultatu.setForeground(Color.BLACK);
textPane_Rezultatu.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent arg0) {
        if (d_Ocinka == false) {
            if (i_Ocinka <= 9) {
                JOptionPane.showMessageDialog(null,
                    "Стан Вашого харчування у
межах норми.\n" + "Так тримати. Бережіть себе.");
            }
            if (i_Ocinka >= 9 && i_Ocinka <= 18) {
                JOptionPane.showMessageDialog(null,
                    "Стан Вашого харчування - небезпечний.\n"
                    + "Детально перегляньте своє
харчування. Бережіть себе.");
            }
            if (i_Ocinka >= 18) {
                JOptionPane.showMessageDialog(null,
                    "Стан Вашого харчування
критичний.\n" + "ТЕРМІНОВО займіться собою. Бережіть себе.");
            }
            d_Ocinka = true;
        }
        if (i_Error == 1 && d_Error == false) {
            s_Error = s_Error
                + "Помилкове введення,
рекомендовано пройти опитування ще раз.\nЗверніть увагу на рік

```

народження.";

```
JOptionPane.showMessageDialog(null, s_Error);
d_Error = true;
    }
    });
rB_Zagalno = new JRadioButton("Загально");
rB_Zagalno.setOpaque(false);
rB_Zagalno.setFont(new Font("Times New Roman", Font.ITALIC,
20));
rB_Zagalno.setForeground(Color.black);
rB_Zagalno.setBounds(8, 10, 120, 25);
getContentPane().add(rB_Zagalno);
rB_Zagalno.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textPane_Rezultatu.setText(s_RekomendaciiZagalni);
    }
});
rB_Induividualno = new JRadioButton("Індивідуально");
rB_Induividualno.setOpaque(false);
rB_Induividualno.setSelected(true);
rB_Induividualno.setFont(new Font("Times New Roman",
Font.ITALIC, 20));
rB_Induividualno.setForeground(Color.black);
rB_Induividualno.setBounds(142, 10, 152, 25);
getContentPane().add(rB_Induividualno);
rB_Induividualno.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textPane_Rezultatu.setText(s_IMT +
s_RekomendaciiPerconalni);
    }
});
rB_Dunamika = new JRadioButton("Динаміка");
rB_Dunamika.setOpaque(false);
rB_Dunamika.setFont(new Font("Times New Roman", Font.ITALIC,
20));
rB_Dunamika.setForeground(Color.black);
rB_Dunamika.setBounds(320, 10, 152, 25);
getContentPane().add(rB_Dunamika);
```

```

rB_Dunamika.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textPane_Rezultatu.setText(s_Dunamika);
    }
});
b_Vegan = new JButton("Вегетеріанство");
b_Vegan.setForeground(new Color(0, 0, 0));
b_Vegan.setFont(new Font("Times New Roman", Font.ITALIC, 30));
getContentPane().add(b_Vegan);
if (i_Vegeterianstvo == 0) {
    b_Vegan.setBounds(608, 469, 386, 49);
}
rB_Ditu = new JRadioButton("Для дітей");
rB_Ditu.setOpaque(false);
rB_Ditu.setForeground(Color.black);
rB_Ditu.setFont(new Font("Times New Roman", Font.ITALIC, 20));
rB_Ditu.setBounds(476, 10, 152, 25);
getContentPane().add(rB_Ditu);
rB_Ditu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textPane_Rezultatu.setText(s_RekomendaciiDlaDitey);
    }
});
group_Rezultatu = new ButtonGroup();
group_Rezultatu.add(rB_Zagalno);
group_Rezultatu.add(rB_Induividualno);
group_Rezultatu.add(rB_Dunamika);
group_Rezultatu.add(rB_Ditu);
JLabel l_Kartunka = new JLabel("");
l_Kartunka.setIcon(new
ImageIcon("res/DiagnostukaRezultatu.png"));
l_Kartunka.setBounds(608, 69, 386, 386);
getContentPane().add(l_Kartunka);
JLabel l_IMT = new JLabel("Індекс Маси Тіла = " + d_IMT0);
l_IMT.setForeground(Color.black);
l_IMT.setHorizontalAlignment(SwingConstants.CENTER);
l_IMT.setFont(new Font("Times New Roman", Font.ITALIC, 20));
l_IMT.setBounds(679, 10, 251, 25);
getContentPane().add(l_IMT);

```

```
1_IMT_komentar = new JLabel("Норма в межах з 18 до 25");
1_IMT_komentar.setHorizontalAlignment(SwingConstants.CENTER);
1_IMT_komentar.setForeground(Color.black);
1_IMT_komentar.setFont(new Font("Times New Roman",
Font.ITALIC, 15));
1_IMT_komentar.setBounds(622, 35, 362, 14);
getContentPane().add(1_IMT_komentar);
1_fon = new JLabel("");
1_fon.setIcon(new ImageIcon("res/fon/Reestracia.png"));
1_fon.setBounds(-1, 0, 995, 572);
getContentPane().add(1_fon);
setVisible(true);
}
}
```

