

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

Дипломний проект

**здобувача освіти денної форми навчання
РП.06.11.000.ДП**

**ІЩЕНКО АНДРІЙ
ВІТАЛІЙОВИЧ**

**м. Одеса
2023 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

**Розробка програмного забезпечення
для підбору дієтичного харчування**

Проектний матеріал складається з пояснювальної записки на 66 сторінках та графічного (презентаційного) матеріалу на 7 аркушах (слайдах).

Дипломник  (Іщенко А.В.)

Керівник  (Селіванова А. В.)

Консультанти:

з економічної частини  (Копайгородська Т. Г.)

з охорони праці  (Чорновол Н. І.)

з дотримання вимог ЄСКД  (Петрашова В. І.)

старший консультант  (Кунуп Т. В.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю. В.)

Завідувач відділення  (Скорнякова О. В.)

Захист «26» 06 2023 р. Протокол ДКК № 3

Оцінка ДКК 5 (Відмінно)

Секретар ДКК 

Позначення

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВП Ігор БЕРКАНЬ

“ ” 2023 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Іщенко Андрій Віталійович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка програмного забезпечення для підбору дієтичного харчування

затверджена наказом по коледжу від “ 17 ” жовтня 2022 р. № 235-А2-ОД

2. Термін здачі закінченого проекту (роботи) 9 червня 2023

3. Вихідні дані до проекту (роботи) Технічне завдання, тестові питання для формування завдань, СУБД Firebase, додаток.

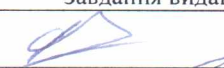
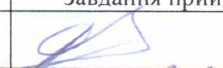


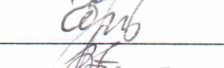
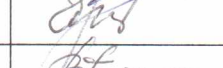
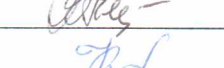

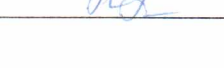

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Огляд предметної галузі, аналіз технологій; Мета розробки, постановка задачі, призначення та основні параметри, вибір функціональних елементів реалізації системи; вибір технологій та інструментів реалізації системи; основні етапи проектування; опис етапів реалізації програмного продукту; тестування, впровадження та експлуатація програмного продукту

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Мультимедійна презентація 14 слайдів: тема, виконавець; актуальність; мета, задачі; аналіз аналогів; засоби реалізації; схема БД; схема інтерфейсу; особливості розробки (2 сл.); інтерфейс (4 слайдів); висновки.

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технологічний	Селіванова А. В.		
Економічна частина	Копайгородська Т. Г.		
Охорона праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кунуп Т. В.		

7. Дата видачі завдання

15.05.2023

Керівник


(підпис)

Завдання прийняв до виконання


(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Аналітичний огляд предметної галузі, обґрунтування актуальності теми	16.05.2023	<i>Виконав</i>
2	Аналіз технологій та аналогічних систем	16.05.2023	<i>Виконав</i>
3	Визначення мети, постановка задачі,	16.05.2023	<i>Виконав</i>
4	Вибір технологій та інструментів реалізації системи	21.05.2023	<i>Виконав</i>
5	Моделювання програмного забезпечення	31.05.2023	<i>Виконав</i>
6	Проектування програмного забезпечення	31.05.2023	<i>Виконав</i>
7	Реалізація програмного забезпечення	31.05.2023	<i>Виконав</i>
8	Доопрацювання та отримання основних результатів проектування	31.05.2023	<i>Виконав</i>
9	Робота над розділом «Економічна частина»	05.06.2023	<i>Виконав</i>
10	Робота над розділом «Охорона праці»	05.06.2023	<i>Виконав</i>
11	Робота над графічною частиною дипломного проекту. Створення мультимедійної презентації проекту.	08.06.2023	<i>Виконав</i>

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

ВСТУП.....	7
1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	9
1.1 Огляд предметної галузі, аналіз технологій	9
1.1.1 Актуальність тематики роботи	11
1.1.2 Аналіз аналогів.....	14
Аналог 1.....	14
Аналог 2.....	15
Аналог 3.....	15
1.2 Мета розробки, постановка задачі, вибір функціональних елементів реалізації системи.....	17
1.3 Вибір технологій та інструментів реалізації.....	19
1.3.1 Обрання СУБД.....	19
1.3.2 Обрання мови програмування та середовища розробки.....	21
1.4 Основні етапи проектування та реалізація.....	22
1.4.1 Визначення вимог до системи	22
1.4.2 Моделювання системи та її реалізація.....	23
1.4.3 Проектування інтерфейсу.....	28
1.4.4 Опис реалізації бази даних.....	31
1.4.5 Опис реалізації інтерфейсу	32
1.5 Тестування, впровадження та експлуатація програмного продукту.....	39
1.5.1 Перевірка коректної роботи реєстрації та аутентифікація	39
1.5.2 Перевірка правильності роботи розділів	41
2. ЕКОНОМІЧНА ЧАСТИНА.....	43
2.1 Розрахунок ціни програмного продукту.....	46

					РП 06. 11 000. ДП ПЗ	Арк.
						5
Зм.	Арк	№ докум.	Підпис	Дата		

3. ОХОРОНА ПРАЦІ	48
3.1 Вступ.....	48
3.2 Розробка заходів з охорони праці.....	48
3.2.1 Виробничі приміщення.....	48
3.2.2 Шуми.....	49
3.2.3 Освітлення.....	50
3.2.4 Мікроклімат.....	50
3.2.5 Пожежна безпека.....	50
3.2.6 Електробезпека.....	51
ВИСНОВКИ.....	53
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТОК А. Фрагмент лістингу першого розділу ПП для підбору дієтичного харчування	56
ДОДАТОК Б. Слайди мультимедійної презентації	59

					РП 06. 11 000. ДП ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		6

ВСТУП

Правильне харчування відіграє важливу роль у збереженні здоров'я та профілактиці захворювань. Розробка програмного забезпечення для підбору дієтичного харчування може сприяти свідомому вибору продуктів та формуванню здорових харчових звичок у користувачів.

Ця тема дипломної роботи є актуальною з кількох причин. По-перше, існує зростаючий інтерес до здорового способу життя в сучасному суспільстві. Багато людей стають більш усвідомленими про своє здоров'я та харчування і шукають способи покращити свої харчові звички. Це створює попит на дієтичні програми та додатки, які можуть допомогти в плануванні та контролі харчування.

По-друге, проблема з харчуванням та ожирінням стає все більш поширеною. За останні роки кількість людей з ожирінням значно зросла, що має серйозні наслідки для громадського здоров'я. Розробка програмного забезпечення, яке допомагає контролювати раціон та підбирати дієтичні страви, може бути ефективним інструментом у боротьбі з цією проблемою.

Крім того, кожна людина має унікальні харчові потреби та обмеження, пов'язані зі станом здоров'я, фізичною активністю, алергіями, хворобами тощо. Розробка програмного забезпечення, яке враховує ці індивідуальні фактори, може допомогти кожній людині підібрати оптимальний раціон з урахуванням її потреб і обмежень.

Завдяки технологічному розвитку та доступності, програмне забезпечення для підбору дієтичного харчування стає доступним широкій аудиторії. Люди зможуть легко отримувати необхідну інформацію про свої дієтичні потреби та отримувати персоналізовані рекомендації.

Це може допомогти зменшити ризик розвитку серцево-судинних захворювань, діабету, ожиріння та інших хронічних захворювань.

Розробка програмного забезпечення для підбору дієтичного харчування також може зробити процес планування їжі більш зручним та ефективним для користувачів. Це дозволить їм швидко отримувати рекомендації щодо раціону,

					<i>РП 06. 11 000. ДП ПЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		7

переглядати список продуктів, калорійність та поживну цінність страв, а також вести облік споживання їжі.

Загалом, розробка програмного забезпечення для підбору дієтичного харчування є важливою і актуальною темою. Враховуючи зростання інтересу до здорового способу життя, проблеми з харчуванням та ожирінням, індивідуалізацію підходу до харчування, технологічний розвиток та доступність, сприяння здоров'ю та профілактику захворювань, а також зручність та ефективність, розробка такого програмного забезпечення може мати значний вплив на поліпшення харчових звичок та здоров'я користувачів [пі]

					<i>РП 06. 11 000. ДП ПЗ</i>	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Огляд предметної галузі, аналіз технологій

Огляд предметної галузі:

Під час огляду предметної галузі дієтичного харчування та здорового способу життя було проведено глибоке дослідження, щоб зрозуміти основні принципи та тенденції, що панують у цій галузі. За останні роки спостерігається значний ріст інтересу до здорового способу життя та правильного харчування, оскільки люди все більше усвідомлюють важливість збалансованого харчування для здоров'я та добробуту.

Огляд включає вивчення рекомендацій щодо харчування, основних компонентів здорового харчування, різних дієт, які можуть бути використані для досягнення різних цілей, таких як зниження ваги, покращення спортивних показників або підтримка загального здоров'я. Були проаналізовані різні підходи до харчування, такі як вегетаріанство, веганство, кетогенна дієта та інші, з метою з'ясування їх особливостей та ефективності.

Також були досліджені існуючі додатки та сервіси, які надають підбір дієтичного харчування. Було проаналізовано їх функціональність, особливості та підхід до підбору харчування для користувачів. Було враховано, які алгоритми та методи використовуються для аналізу даних користувача, рекомендацій харчування та створення персоналізованих планів харчування.

Огляд технологій:

Під час аналізу технологій для розробки програмного забезпечення для підбору дієтичного харчування було розглянуто різні можливості та інструменти, які можуть бути використані для створення необхідного функціоналу. Один з найважливіших виборів полягає в виборі платформи розробки, яка забезпечить оптимальну продуктивність та зручність для розробників.

Android Studio, як інтегроване середовище розробки для платформи Android, виявилось найбільш підходящим вибором для розробки програмного забезпечення для мобільних пристроїв. Це потужне середовище надає розширені

					РП 06. 11 001. ДП ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

можливості розробки, включаючи редактор коду, інструменти для проектування користувацького інтерфейсу, інструменти для тестування та налагодження додатків. Крім того, Android Studio має велику спільноту розробників та підтримується Google, що забезпечує доступ до актуальних інструментів та ресурсів для розробки програмного забезпечення на Android.

У контексті використання машинного навчання та штучного інтелекту для покращення функціональності підбору дієтичного харчування, Android Studio може бути інтегрована з різними бібліотеками машинного навчання, такими як TensorFlow або ML Kit. Ці бібліотеки надають інструменти та алгоритми для створення та навчання моделей машинного навчання, які можуть аналізувати дані про харчування користувачів і надавати рекомендації на основі цих даних. Таким чином, можна створити модель, яка аналізує харчування користувача, враховує його потреби та мету (наприклад, зниження ваги чи поліпшення спортивної продуктивності) і надає персоналізовані рекомендації стосовно харчування.

У додаток до машинного навчання, Android Studio також забезпечує можливість інтеграції з іншими сторонніми сервісами та API, які можуть бути корисними для підбору дієтичного харчування. Наприклад, можна інтегрувати сервіси для отримання харчової інформації про продукти, дієтичних рекомендацій або рецептів. Це дозволить користувачам отримувати швидкий доступ до актуальної інформації та ресурсів, що сприятимуть їхньому здоровому способу життя.

Загалом, використання Android Studio та інтеграція з машинним навчанням, штучним інтелектом та сторонніми сервісами дозволяють створити потужний та інноваційний інструмент для підбору дієтичного харчування. Це дозволяє користувачам отримувати персоналізовані рекомендації, сприяючи поліпшенню їхнього здоров'я та досягненню поставлених цілей [1].

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

1.1.1 Актуальність тематики роботи

У важкій вазі

Частка населення, яке страждає на ожиріння, %, 2016 рік

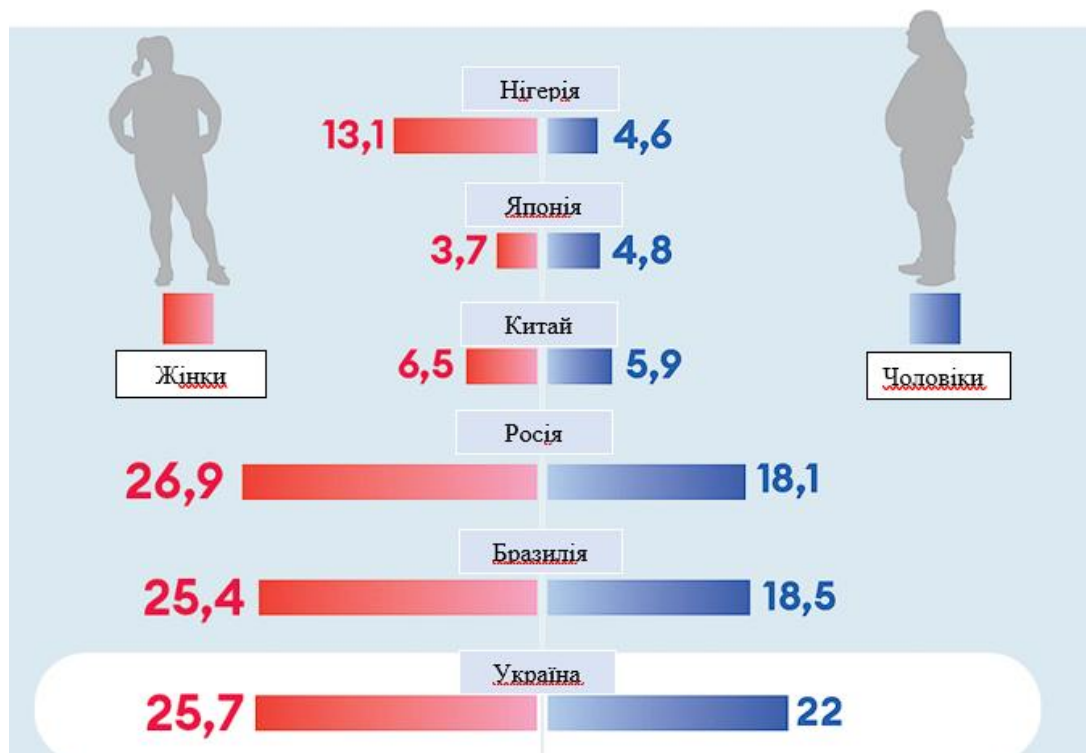


Рисунок 1.1 Частка населення яка страждає на ожиріння^[1]

На мою думку актуальність цієї теми підтверджують ці пункти:

1. Зростання інтересу до здорового способу життя: У сучасному суспільстві все більше людей проявляє інтерес до свого здоров'я та харчування. Здоровий спосіб життя, включаючи раціональне харчування, став важливою тенденцією, що спричиняє зростання популярності дієтичних програм та додатків.
2. Проблеми з харчуванням та ожиріння: Ожиріння є однією з найпоширеніших проблем здоров'я у багатьох країнах світу. За останні роки кількість людей з ожирінням значно зросла, що має серйозні наслідки для громадського здоров'я. Розробка програмного забезпечення, яке

допомагає контролювати раціон та підбирати дієтичні страви, може стати ефективним інструментом у боротьбі з цією проблемою.

3. Індивідуалізація підходу до харчування: Кожна людина має унікальні харчові потреби та обмеження, які пов'язані зі станом здоров'я, фізичною активністю, алергіями, хворобами тощо. Розробка програмного забезпечення, яке враховує ці індивідуальні фактори, може допомогти кожній людині підібрати оптимальний раціон з урахуванням її потреб і обмежень.
4. Технологічний розвиток та доступність: Завдяки зростанню технологічних можливостей і поширенню мобільних пристроїв, доступ до програмного забезпечення став значно легшим. Розробка мобільних додатків та веб-платформ для підбору дієтичного харчування може зробити цю інформацію доступною для широкої аудиторії. Люди зможуть з легкістю отримати необхідну інформацію про свої дієтичні потреби та отримувати персоналізовані рекомендації.
5. Сприяння здоров'ю та профілактика захворювань: Правильне харчування відіграє важливу роль у збереженні здоров'я та профілактиці захворювань. Розробка програмного забезпечення для підбору дієтичного харчування може сприяти свідомому вибору продуктів та формуванню здорових харчових звичок у користувачів. Це може допомогти зменшити ризик розвитку серцево-судинних захворювань, діабету, ожиріння та інших хронічних захворювань.
6. Зручність та ефективність: Розробка програмного забезпечення для підбору дієтичного харчування може зробити процес планування їжі більш зручним та ефективним. Користувачі зможуть швидко отримувати рекомендації щодо раціону, переглядати список продуктів, калорійність та поживну цінність страв, а також вести облік споживання їжі. Це може заощадити їм час і зусилля при плануванні їхнього харчування підібрати оптимальний раціон, враховуючи її індивідуальні потреби, переваги та обмеження.

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

Враховуючи зазначені аргументи, можна побачити, що розробка програмного забезпечення для підбору дієтичного харчування має велику актуальність. Зростання інтересу до здорового способу життя, збільшення кількості людей, які бажають контролювати своє харчування та здоров'я, технологічний розвиток і доступність програмного забезпечення, сприяння здоров'ю та профілактика захворювань, зручність та ефективність - всі ці фактори підтверджують необхідність розробки таких програм.

Подібне програмне забезпечення може стати незамінним помічником у формуванні здорового способу харчування, допомагати здобувати правильні знання про продукти харчування, складати персоналізовані раціони, контролювати споживання калорій, вітамінів та мінералів, а також вести статистику і аналізувати результати.

Далі у дипломній роботі можна розглянути різні підходи до розробки такого програмного забезпечення, аналіз існуючих ринкових рішень, вибір технологій та інструментів для розробки, створення архітектури програми, розробку алгоритмів підбору раціонів, дизайн інтерфейсу та його користувацьку зручність, тестування та оцінку ефективності програми.

У підсумку, розробка програмного забезпечення для підбору дієтичного харчування є актуальним та значущим напрямом. Це дозволяє відповідати на зростаючий попит на інструменти, які допомагають покращувати харчування та здоров'я. Такі програми сприяють популяризації здорового способу життя, допомагають користувачам усвідомлено вибирати продукти харчування та складати збалансовані раціони, а також слідкувати за своїм прогресом.

Розробка такої програми вимагає широкого спектру компетенцій, включаючи знання в галузі дієтології, харчової науки, програмування та інформаційних технологій. Крім того, важливим елементом є партнерство з дієтологами та фахівцями з медичної галузі для забезпечення достовірної та науково обґрунтованої інформації.

Розробка програмного забезпечення для підбору дієтичного харчування має потенціал допомогти багатьом людям покращити своє здоров'я та якість

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

життя. Подальші дослідження та розвиток в цій галузі можуть призвести до вдосконалення інструментів та розширення їх функціональності, що зробить їх ще більш ефективними та корисними для користувачів^[1].

1.1.2 Аналіз аналогів

Перш ніж приступити до розробки системи управління харчуванням, важливо проаналізувати існуючі аналогічні продукти, які пропонують схожий функціонал. Це дозволить нам зрозуміти потенційні переваги та недоліки існуючих рішень і використати цю інформацію для покращення та вдосконалення системи, яка розробляється. Нижче наведено три приклади проаналізованих подібних продуктів:

Аналог 1

MyFitnessPal: Це популярний додаток для моніторингу харчування та фітнесу. Він надає можливість вводити щоденний раціон їжі, отримувати статистику про спожиті калорії, білки, жири та вуглеводи, а також отримувати рекомендації щодо здорового харчування. Однак, в деяких випадках можуть бути обмеження у персоналізації плану харчування та аналізі індивідуальних потреб користувача [1].

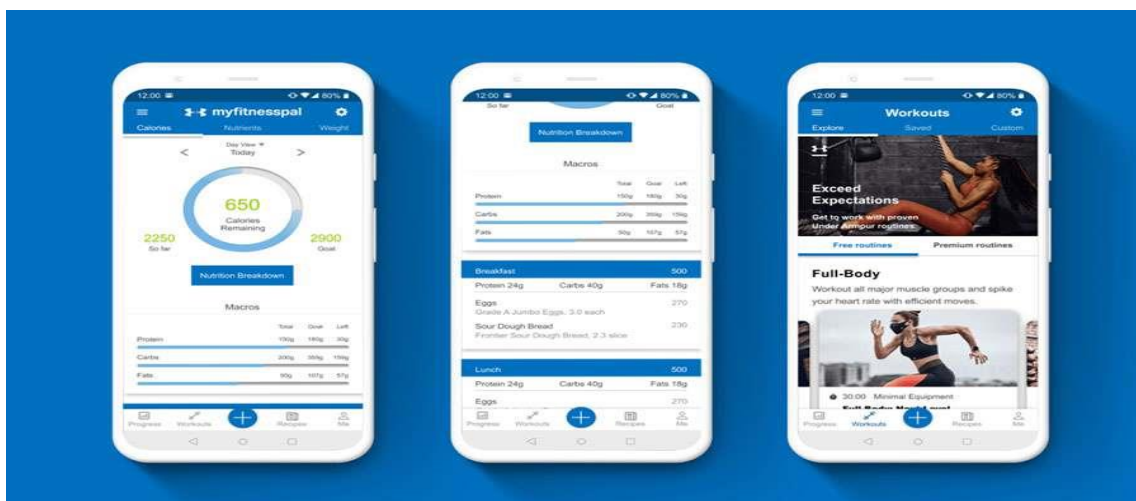


Рисунок 1.2. Застосунок MyFitnessPal

Аналог 2

Lifesum: Це ще один популярний додаток для ведення здорового способу життя, який надає функціонал з моніторингу харчування, тренувань та інших аспектів здоров'я. Він має вбудовану базу даних продуктів та рецептів, а також можливість створювати персоналізовані плани харчування. Однак, інтерфейс додатку може бути складним для деяких користувачів, а також можуть відсутні деякі функціональні можливості, які були бажані для нашої системи^[1]



Рисунок 1.3. Меню застосунка LifeSum

Аналог 3

Yazio: Цей додаток також пропонує моніторинг харчування та фітнесу, а також створення персоналізованих планів харчування. Він має зручний інтерфейс та добре розроблену базу даних з продуктами харчування.

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

Однак, у нього можуть бути обмеження у функціоналі машинного навчання та інтеграції зі сторонніми сервісами [1]

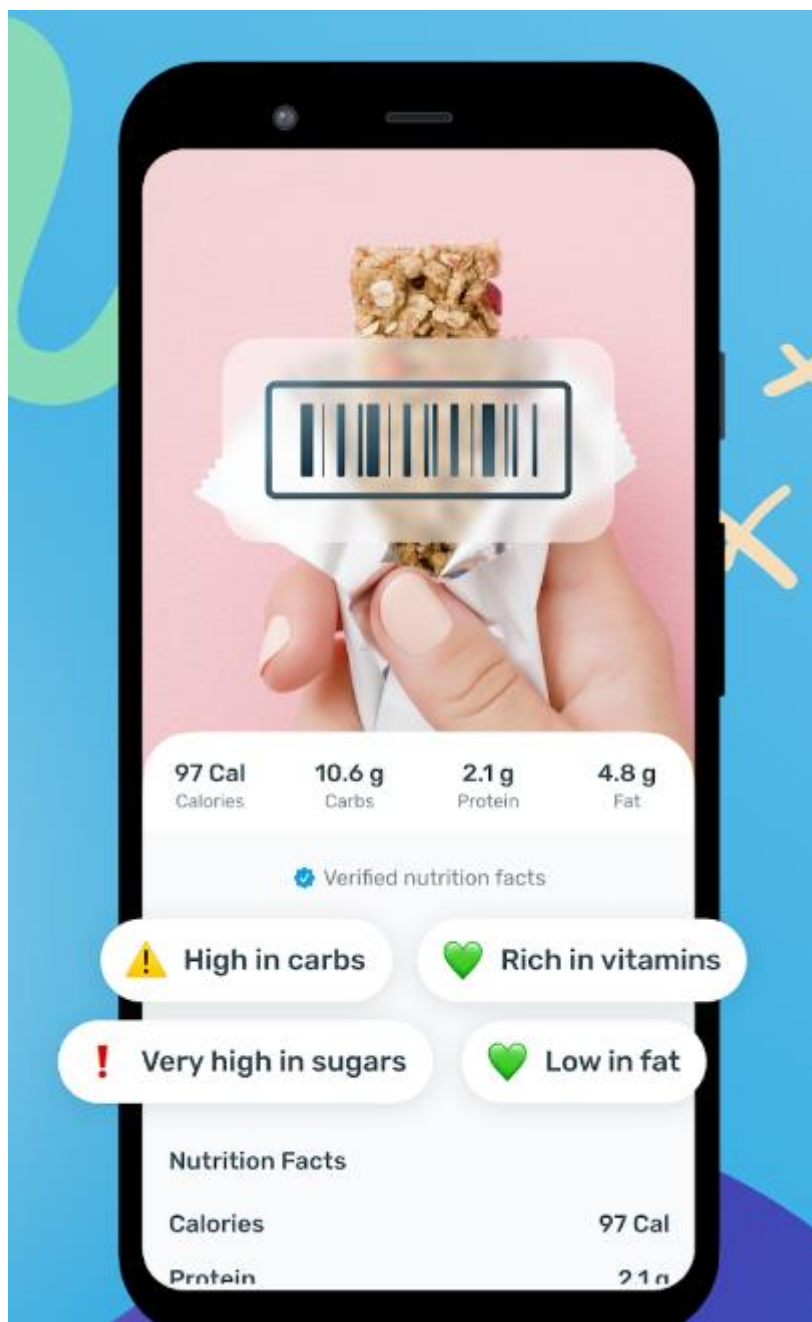


Рисунок 1.4. Меню застосунка Yazio

Таблиця 1.1 Порівняльна таблиця аналогів.

Параметри/Аналоги	MyFitnessPal	Lifesum	Yazio
Функціональність	Моніторинг харчування, статистика, рекомендації	Моніторинг харчування, тренування, здоров'я	Моніторинг харчування, персоналізовані плани
База даних продуктів	Так	Так	Так
Персоналізація	Обмежена	Деяка	Є
Машинне навчання	Обмежений	Обмежений	Невідомо
Інтеграція зі сторонніми сервісами	Так	Деяка	Невідомо
Інноваційні можливості	Залежно від версії та платного плану	Деякі інноваційні можливості	Обмежені інноваційні можливості

Аналіз існуючих аналогів дав змогу виявити певні переваги та недоліки, які можуть бути враховані під час розробки нашої системи підбору дієтичного харчування. Таким чином, ми зможемо створити рішення, яке краще відповідає потребам користувачів та пропонує більш унікальні та інноваційні можливості.

1.2 Мета розробки, постановка задачі, вибір функціональних елементів реалізації системи

Метою розробки є створення системи, яка надасть користувачам зручний і ефективний інструмент для підбору дієтичного харчування, що відповідає їхнім потребам та меті. Головною метою є створення ПЗ для поліпшення здоров'я та досягнення поставлених цілей з харчування.

					РП 06. 11 001. ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Постановка задачі включає наступні етапи:

1. Збір вихідних даних: це включає збір інформації про користувачів, їхні пріоритети, потреби та обмеження в харчуванні. Також потрібно зібрати дані про харчування, продукти, рецепти та іншу відповідну інформацію.

2. Розробка алгоритму підбору: на основі зібраних даних потрібно розробити алгоритм, який зможе аналізувати цю інформацію та надавати персоналізовані рекомендації щодо харчування. Алгоритм може використовувати методи машинного навчання та штучного інтелекту для аналізу та прогнозування.

3. Розробка користувацького інтерфейсу: система повинна мати зручний та інтуїтивно зрозумілий інтерфейс, який дозволить користувачам вводити свої дані, переглядати рекомендації та отримувати доступ до інших функціональних можливостей.

4. Інтеграція зі сторонніми сервісами: система повинна бути інтегрована зі сторонніми сервісами, які надають інформацію про продукти, дієтичні рекомендації та рецепти. Це забезпечить доступ до актуальної інформації та розширить функціонал системи.

5. Тестування та валідація: розроблена система повинна бути піддана ретельному тестуванню для перевірки її працездатності, надійності та коректності роботи. Також потрібно провести валідацію, порівнявши результати системи зі звичайними методами підбору харчування.

6. Розгортання та підтримка: розроблена система повинна бути розгорнута на платформі Android та готова до використання користувачами. Також потрібно забезпечити підтримку та оновлення системи в майбутньому.

Вибір функціональних елементів реалізації системи залежить від потреб та мети користувачів. Можливі функціональні елементи включають:

1. Реєстрація та вхід в систему: користувачі повинні мати можливість створити обліковий запис і увійти в систему для збереження своїх налаштувань та персоналізованих рекомендацій.

					РП 06. 11 001. ДП ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

2. Введення даних користувача: користувачі повинні мати можливість ввести свої параметри, цілі, обмеження та іншу відповідну інформацію.

3. Аналіз даних користувача: система повинна здійснювати аналіз введених даних користувача для створення персоналізованих рекомендацій та планів харчування.

4. Перегляд рекомендацій та планів харчування: користувачі повинні мати можливість переглядати рекомендації щодо харчування, персоналізовані плани харчування та інші відповідні дані.

5. Пошук продуктів та рецептів: система повинна надавати можливість пошуку продуктів, дієтичних рецептів та інших релевантних ресурсів.

6. Відстеження прогресу: користувачі повинні мати можливість відстежувати свій прогрес в досягненні цілей харчування та здоров'я.

7. Нагадування та сповіщення: система може надавати нагадування та сповіщення користувачам щодо прийому їжі, досягнення цілей та інших важливих подій.

Вибір конкретних функціональних елементів повинен бути підтриманий дослідженнями та аналізом потреб користувачів, а також можливостями реалізації системи з використанням Android Studio та інших необхідних інструментів.

1.3 Вибір технологій та інструментів реалізації

1.3.1 Обрання СУБД

Один з найважливіших аспектів розробки системи дієтичного харчування - обрання системи управління базами даних (СУБД). У нашому випадку, я звернув увагу на СУБД, яка надає надійність, ефективність та гнучкість для зберігання і обробки даних про харчування користувачів.

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.5 Логотип Firebase

Система управління базами даних Firebase є популярним вибором для розробки мобільних додатків та веб-проектів. Вона є хмарною платформою, розробленою Google, і надає низку функцій та послуг, які спрощують роботу з базами даних і забезпечують широкі можливості для реалізації проектів.

Однією з переваг Firebase є його простота використання. Вона має дружній інтерфейс та зрозумілу документацію, що дозволяє розробникам швидко оволодіти платформою. Firebase також надає можливість роботи в режимі реального часу, що дозволяє миттєво оновлювати дані на всіх підключених пристроях.

Іншою перевагою Firebase є його масштабованість. Платформа автоматично масштабується в залежності від потреб проекту, що дозволяє легко впоратися зі зростаючим обсягом даних та трафіку. Firebase також забезпечує високу доступність і надійність, оскільки він працює на серверах Google зі стійкими каналами зв'язку.

Ще однією важливою особливістю Firebase є його широкий набір інструментів і послуг. Вона підтримує автентифікацію користувачів, зберігання файлів, статистику використання, розсилку повідомлень та багато іншого. Ці інструменти дозволяють розробникам швидко додавати необхідну функціональність до своїх додатків.

Однак, слід зазначити деякі потенційні недоліки Firebase.

Оскільки це хмарна платформа, вона вимагає підключення до Інтернету для доступу до бази даних. Це може створювати обмеження для додатків, які потребують офлайн-режиму або працюють в умовах зі слабким Інтернетом.

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

з'єднанням. Крім того, хоча Firebase має безкоштовний рівень обслуговування, деякі розширені функції можуть бути платними, що може вплинути на бюджет проекту. Так як Firebase є NoSQL СУБД то необхідність захисту від аномалій відновлення покладаються на розробника

Обравши Firebase як СУБД для проекту, я зможу реалізувати її, використовуючи спеціальні бібліотеки та SDK, які надаються Firebase. Також зможу здійснювати підключення до бази даних, створювати схему даних, виконувати операції читання/запису та здійснювати інші необхідні дії. Firebase також надає можливість реалізації аутентифікації користувачів, що дозволить забезпечити безпеку та контроль доступу до даних.

Загалом, Firebase є потужною та гнучкою СУБД, яка може бути використана для реалізації проекту дієтичного харчування. Її простота використання, масштабованість та широкий набір інструментів роблять її привабливим вибором для розробників.

1.3.2 Обрання мови програмування та середовища розробки

Вибір мови програмування та середовища розробки є важливим етапом при реалізації будь-якого проекту. У моєму випадку, я обрав мову програмування Kotlin та середовище розробки Android Studio для створення моєї програми. Такий вибір має кілька переваг, які сприятимуть успішній реалізації цього проекту.

Kotlin є сучасною мовою програмування, розробленою компанією JetBrains спеціально для розробки на платформі Android. Вона базується на мові Java, але пропонує більш простий та елегантний синтаксис, що сприяє покращенню продуктивності розробників. Kotlin підтримує всі функціональності Java, а також має деякі унікальні можливості, такі як нульова безпека (null safety) та розширені функціональні можливості^[1].

Kotlin чудово пасує до Android застосувань^[1], привносячи всі переваги сучасної мови на Android платформу, без введення нових обмежень:

					РП 06. 11 001. ДП ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Сумісність: Kotlin повністю сумісний з JDK 6, гарантуючи, що Kotlin застосування можуть робити на старіших пристроях Android без жодних проблем. Інструментарій Kotlin повністю підтримується в Android Studio, та сумісний з системою побудови Android.

2. Продуктивність: Застосування Kotlin роблять так само швидко, як еквівалентні Java застосування, дякуючи дуже подібному байткоду. З підтримкою в Kotlin інлайн функцій, код з використанням лямбда часто виконується навіть швидше, ніж той самий код на Java.

3. Взаємодія: Kotlin є на 100% взаємодіючим з Java, що дозволяє використовувати всі існуючі бібліотеки Android в застосуванні Kotlin. Це включає процес анотацій, так що прикріплення даних та Dagger також працюють.

4. Місце: Kotlin має дуже компактну бібліотеку часу виконання, що може бути ще зменшеним завдяки використанню ProGuard. В реальному застосуванні, рантайм Kotlin додає тільки декілька сотен методів, та менше ніж 100Кб до розміру .apk файлу.

5. Час компіляції: Kotlin підтримує ефективну інкрементальну компіляцію, так що хоча для чистих побудов існує деяке додаткове навантаження, інкрементальні побудови звичайно так сами швидкі, або ще кращі, ніж Java.

1.4 Основні етапи проектування та реалізація

1.4.1 Визначення вимог до системи

Наразі у більшості користувачів смартфонів вже виконано мінімальні потреби а саме:

1. Процесор: Багатоядерний процесор з тактовою частотою не менше 1,5 ГГц.
2. Оперативна пам'ять (ОЗП): Зазвичай 2 Гб або більше.
3. Внутрішня пам'ять: Зазвичай 16 Гб або більше.

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

4. Дисплей: Роздільна здатність екрана залежить від конкретного смартфона, але зазвичай рекомендується HD (1280x720 пікселів) або вище.

5. Операційна система: Android.

Програма працює навіть на пристроях з мінімальними технічними характеристиками. Вона не потребує великих обсягів пам'яті та процесорної потужності, що дозволяє користуватись нею без зайвих обмежень. Незважаючи на скромні вимоги до системи, програма працює швидко та ефективно, надаючи вам необхідні функції і інструменти для зручного використання. Користувач зможе насолоджуватись всіма її можливостями без необхідності в дорогому апаратному забезпеченні.

1.4.2 Моделювання системи та її реалізація

Так як для користування даною програмою потрібно мати аккаунт у застосунку, потрібно вот така не складна схема.

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

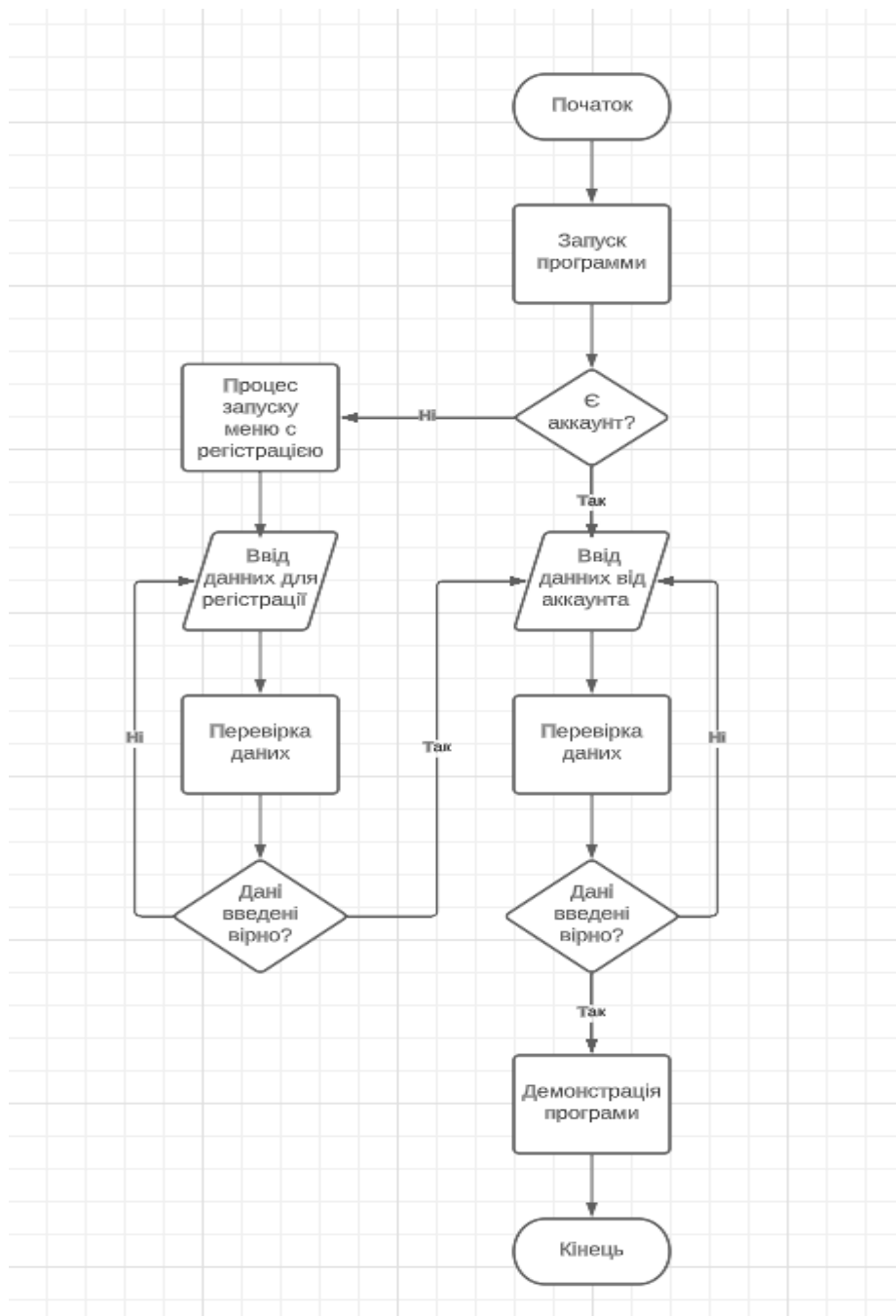


Рисунок 1.6 Алгоритм роботи інтерфейсу

Зм.	Арк.	№ докум.	Підпис	Дата

Для того щоб ця схема працювала я написав такий не складний код:

```
1 package com.amidomaru.dietapp.registration
2
3 import ...
4
5 class PrivacyPolicyDialog (context: Context, email: String, password: String) : BottomSheetDialogFrag
6     private val context : Context
7     private val email : String
8     private val password : String
9
10    private lateinit var binding: RegisterDialogBinding
11
12    override fun onCreateView(
13        inflater: LayoutInflater,
14        container: ViewGroup?,
15        savedInstanceState: Bundle?
16    ): View? {
17        binding = RegisterDialogBinding.inflate(inflater, container, attachToParent: false)
18
19        binding.completeRegisterBtn.setOnClickListener { it: View!
20            //если нажали на кнопку подтверждения, то регистрируем пользователя
21            FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, password)
22                .addOnCompleteListener { it: Task<AuthResult!>
23                    if (it.isSuccessful){
24                        //если регистрация прошла успешно, то переходим на главный экран
25                        startActivity(Intent(context, MainActivity::class.java))
26                    } else {
27                        //если регистрация не прошла, то выводим сообщение об ошибке
28                        Toast.makeText(context, it.exception?.message, Toast.LENGTH_SHORT).show()
29                    }
30                }
31        }
32
33        binding.cancelRegistrationBtn.setOnClickListener { it: View!
34            dismiss()
35        }
36
37        return binding.root
38    }
39
40    init{
41        this.context = context
42        this.email = email
43        this.password = password
```

Рисунок 1.7 Скріншот коду меню реєстрації

```

package com.amidomaru.dietapp.registration

import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.amidomaru.dietapp.databinding.ActivityRegisterBinding

class RegisterActivity : AppCompatActivity() {

    private lateinit var binding: ActivityRegisterBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityRegisterBinding.inflate(LayoutInflater)
        setContentView(binding.root)

        //обрабатываем нажатие на кнопку регистрации
        binding.signUpBtn.setOnClickListener { @View()
            //проверяем, что поля не пустые
            if (binding.emailEdit.text.toString().isEmpty() || binding.passwordSignUp.text.toString().isEmpty()){
                //если пустые, то выводим сообщение об ошибке
                Toast.makeText(context, this, text: "Будь ласка, введіть усі поля", Toast.LENGTH_SHORT).show()
            } else {
                val privacyPolicyDialog = PrivacyPolicyDialog(applicationContext, binding.emailEdit.text.toString(), binding.passwordSignUp.text.toString())
                privacyPolicyDialog.show(supportFragmentManager, tag: "privacyPolicyDialog")
            }
        }
    }
}

```

Рисунок 1.8 Скріншот коду меню реєстрації

За допомогою цього коду ми можемо отримати ось таке меню:

зареєструватися

електронною поштою

пароль

зареєструватися

Рисунок 1.9 Скріншот вікна реєстрації

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

Для авторизації використовується інша частина коду а саме:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="#fff"
    tools:context=".LoginActivity">

    <TextView
        android:id="@+id/sign_in_txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="36dp"
        android:text="увійти"
        android:textSize="26sp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:orientation="vertical"
        android:layout_centerInParent="true">

        <EditText
            android:id="@+id/email_edit"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minHeight="50dp"
            android:hint="електронною поштою"
            android:textColorHint="#000"
            android:paddingStart="20dp"
            android:paddingEnd="20dp"
            android:paddingTop="10dp"
            android:paddingBottom="10dp"
            android:textColor="#000"
            android:background="@drawable/edit_txt_shape"
            android:layout_marginTop="20dp"
            android:inputType="textEmailAddress" />

    </LinearLayout>

</RelativeLayout>
```

Рисунок 1.10 Скріншот частини коду для авторизації

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Результатом цього коду є ось таке меню:

увійти

The image shows a login interface. At the top, the text 'увійти' (login) is centered. Below it are two input fields: the first is labeled 'електронною поштою' (by email) and the second is labeled 'пароль' (password). At the bottom of the form is a purple rounded button labeled 'увійти' (login).

Рисунок 1.11 Скріншот меню входу до акаунта

1.4.3 Проектування інтерфейсу

Інтерфейс поділений на 3 розділи, в кожному з яких буде своя інформація та функціонал

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

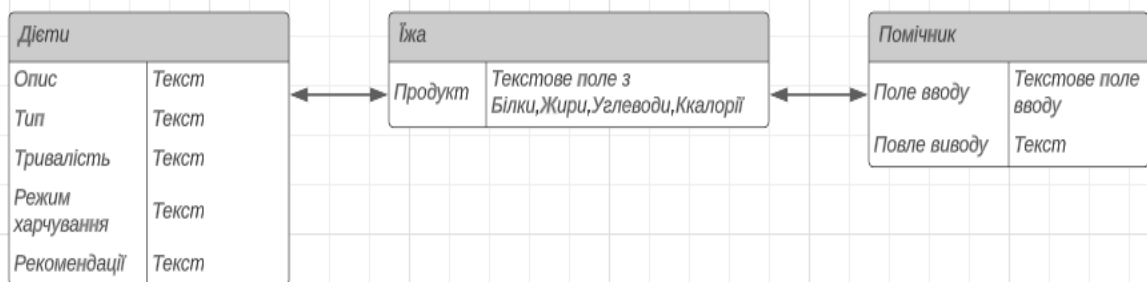


Рисунок 1.12 Блок схема інтерфейса

В середовищі розробки ці дані виглядають ось так:

```
private fun loadCorrectionDiets() {
    binding.loadingPB.visibility = View.VISIBLE
    binding.root.isClickable = false
    FirebaseDatabase.getInstance().reference.child("Diets").addListenerForSingleValueEvent(object: ValueEventListener{
        override fun onDataChange(snapshot: DataSnapshot) {
            var diets = ArrayList<Diet>()
            for (snapshot1 in snapshot.children){
                if (snapshot1.key.equals( other: "t16")||snapshot1.key.equals( other: "t17")
                    ||snapshot1.key.equals( other: "t18")) {
                    val name = snapshot1.child( path: "name").value.toString()
                    val type = snapshot1.child( path: "type").value.toString()
                    val duration = snapshot1.child( path: "duration").value.toString()
                    val recommend = snapshot1.child( path: "recommend").value.toString()
                    val desc = snapshot1.child( path: "desc").value.toString()
                    val diet = snapshot1.child( path: "diet").value.toString()

                    diets.add(Diet(desc, diet, duration, name, recommend, type))
                }
            }
        }
    })
}
```

Рисунок 1.13 Скріншот частини коду першого розділу “Дієти”

```

private fun parseJson(json: String) {
    try {
        val jsonArray = JSONArray(json)
        val dataList = mutableListOf<Product>()

        for (i in 0 until jsonArray.length()) {
            val jsonObject = jsonArray.getJSONObject(i)
            // Parse the JSON data and create DataModel objects
            val data = Product(
                jsonObject.getString( name: "Продукт"),
                jsonObject.getDouble( name: "Жири, г"),
                jsonObject.getDouble( name: "Білки, г"),
                jsonObject.getDouble( name: "Вуглеводи, г"),
                jsonObject.getDouble( name: "Калорійність, Ккал"),
                // Parse other keys as needed
            )
            dataList.add(data)
        }
    }
}

```

Рисунок 1.14 Скріншот частини коду розділу “Їжа”

```

binding.sendRequestBtn.setOnClickListener { it: View!
    if (binding.requestEt.text.isNotEmpty()) {
        ukrainianEnglishTranslator.translate(binding.requestEt.text.toString()).addOnSuccessListener { it: String!
            onResponse(it) { response ->
                Handler(Looper.getMainLooper()).post {
                    englishUkrainianTranslator.translate(response).addOnSuccessListener { it: String!
                        binding.txtResponse.text = it
                    }
                }
            }
        }
    }
}

return root
}

fun onResponse(question: String, callback: (String) -> Unit){
    //установка текста для вопроса в нижней строке.
    binding.idTVQuestion.text = binding.requestEt.text
    binding.txtResponse.setText("Typing...")
    binding.requestEt.setText("")
}

```

Рисунок 1.15 Скріншот частини коду розділу “Помічник”

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

1.4.4 Опис реалізації бази даних

В якості бази даних продовжується використовуватися додаток Firebase, так як у ньому дуже зручно зберігати, редагувати, додавати або знищувати потрібні данні

Для того щоб взаємодіяти з додатком у коді використовується невеличкий код а саме

```
package com.amidomaru.dietapp

import android.app.Application
import com.google.firebase.FirebaseApp

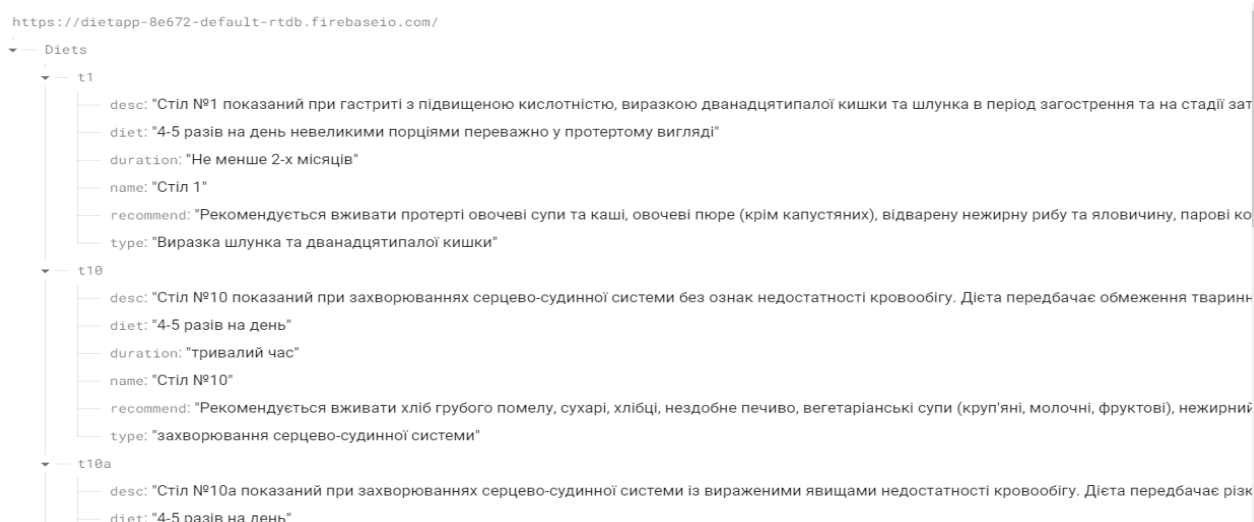
class App : Application() {

    override fun onCreate() {
        super.onCreate()

        FirebaseApp.initializeApp(context: this)
    }
}
```

Рисунок 1.16 Скріншот вікно кода ініціалізації бази даних

Після звернення данні напряму потрапляють у виділений для них розділ, звісно данні зберігаються лише в хмарному сервісі, тому застосунок буде працювати лише при використанні інтернету. У хмарному просторі ми бачимо дані ось так:



The screenshot shows a web interface for a Firebase Realtime Database. The URL is <https://dietapp-8e672-default-rtdb.firebaseio.com/>. The database structure is as follows:

- Diets
 - t1
 - desc: "Стіл №1 показаний при гастриті з підвищеною кислотністю, виразкою дванадцятипалої кишки та шлунка в період загострення та на стадії зат
 - diet: "4-5 разів на день невеликими порціями переважно у протертому вигляді"
 - duration: "Не менше 2-х місяців"
 - name: "Стіл 1"
 - recommen: "Рекомендується вживати протерті овочеві супи та каші, овочеві пюре (крім капустяних), відварену нежирну рибу та яловичину, парові ко
 - type: "Виразка шлунка та дванадцятипалої кишки"
 - t10
 - desc: "Стіл №10 показаний при захворюваннях серцево-судинної системи без ознак недостатності кровообігу. Дієта передбачає обмеження тваринн
 - diet: "4-5 разів на день"
 - duration: "тривалий час"
 - name: "Стіл №10"
 - recommen: "Рекомендується вживати хліб грубого помелу, сухарі, хлібці, нездобне печиво, вегетаріанські супи (круп'яні, молочні, фруктові), нежирний
 - type: "захворювання серцево-судинної системи"
 - t10a
 - desc: "Стіл №10a показаний при захворюваннях серцево-судинної системи із вираженими явищами недостатності кровообігу. Дієта передбачає різк
 - diet: "4-5 разів на день"

Рисунок 1.17 Скріншот меню бази даних у Firebase

					РП 06. 11 001. ДП ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

Дуже зручно меню та спосіб додавання нової інформації

1.4.5 Опис реалізації інтерфейсу

Для того щоб реалізувати 3 розділи в додатку я створив три Kotlin Class

File

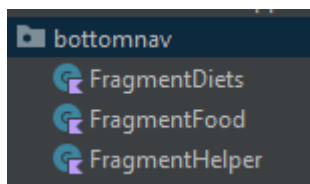


Рисунок 1.18 Скріншот фрагменти розділів

Кожних с них має свою функцію та данні, інтерфейс дуже простий та інтуїтивно зрозумілий:



Рисунок 1.19 Скріншот навігаційні кнопок меню

Перший розділ має список дієт які поділяються на лікувальні та корекційні

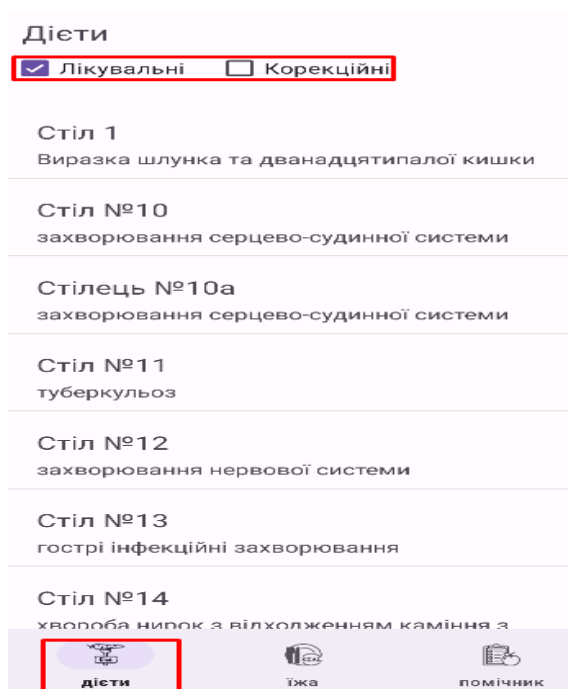


Рисунок 1.20 Скріншот головне меню першого розділу

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

У коді ми можемо побачити що реалізовано воно таким чином:

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Дієти"
    android:id="@+id/diets_txtv"
    android:textSize="22sp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginTop="15dp"
    android:layout_marginStart="10dp"/>

<LinearLayout
    android:id="@+id/diets_lin"
    android:layout_width="wrap_content"
    android:layout_height="60dp"
    android:layout_marginStart="3dp"
    app:layout_constraintTop_toBottomOf="@id/diets_txtv"
    app:layout_constraintStart_toStartOf="parent"
    android:orientation="horizontal"
    >

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Лікувальні"
        android:textSize="16sp"
        android:id="@+id/therapeutic_diets_cb"
        />

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Корекційні"
        android:textSize="16sp"
        android:layout_marginStart="20dp"
        android:id="@+id/correction_diets_cb"
        />
```

Рисунок 1.21 Скріншот коду який розділяє лікувальні та корекційні дієти

					РП 06. 11 001. ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

При обиранні дієти вилізе вікно знизу додатку де ми можемо побачити такі дані як: Опис, Тип,Тривалість,Режим харчування та рекомендації.

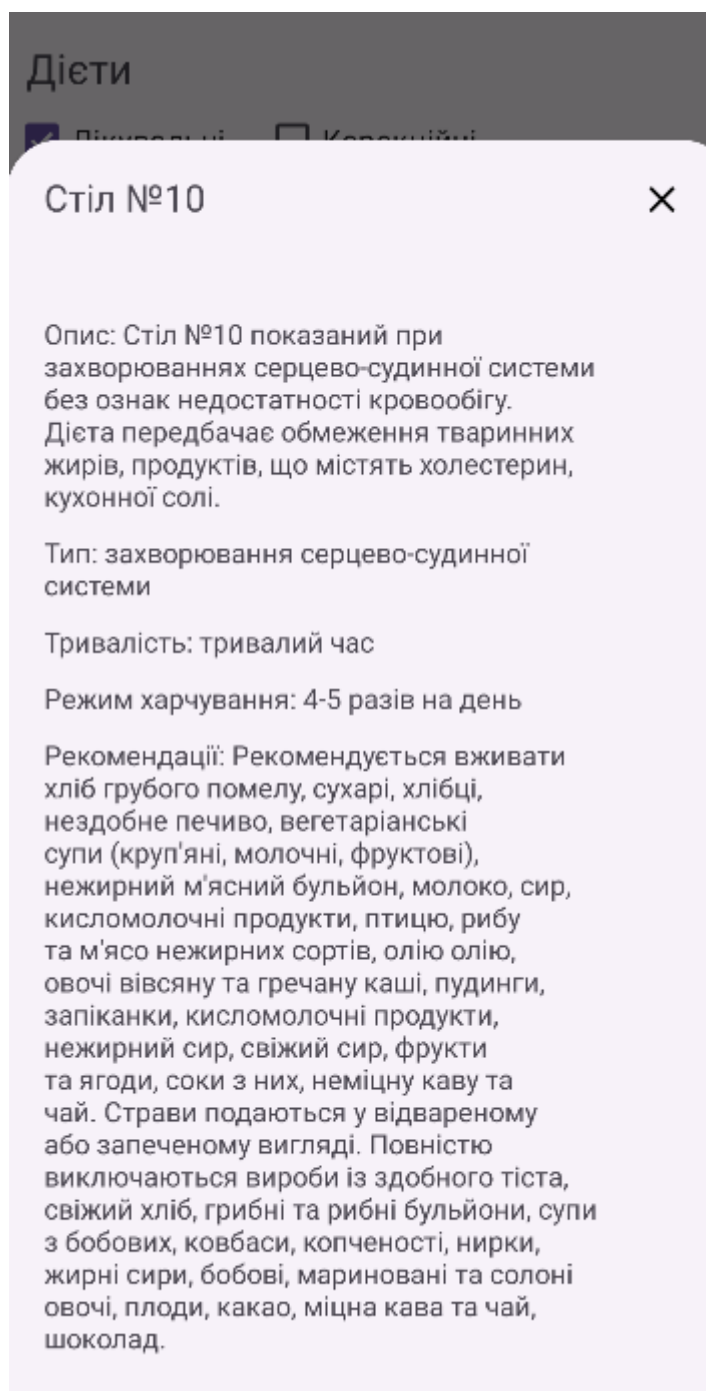


Рисунок 1.22 Скріншот дані до дієти в першому розділі

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

Ось так це виглядає в вигляду даних в облачному сервісі

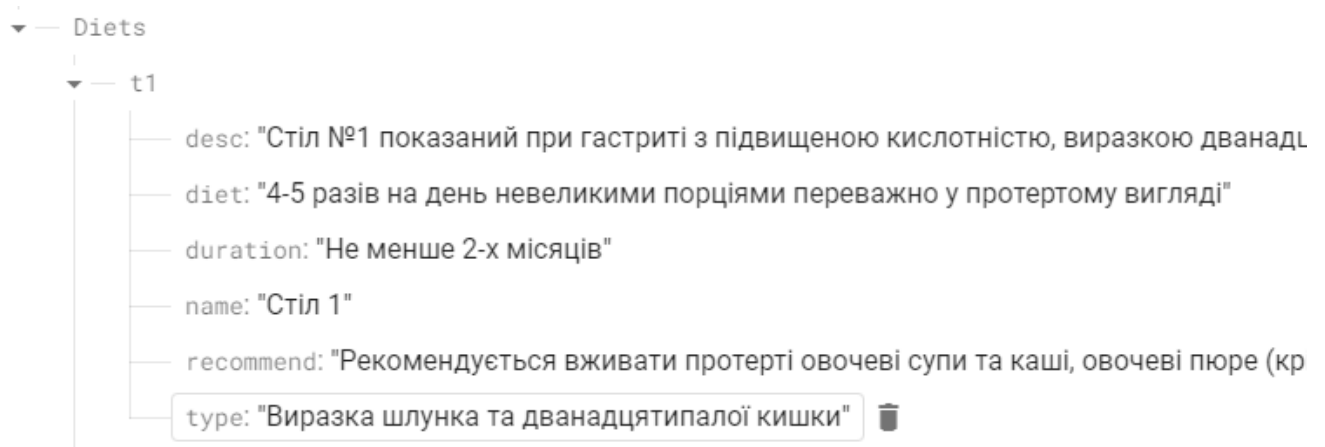


Рисунок 1.23 Скріншот з даними та форматом заповнення їх

На другому розділі ми можемо отримати дані о їжі, а саме Білок, Жири, Вуглеводи та Калорії в продукті. Це дуже допомагає дізнатися скільки БЖУ та Калорій міститься у продукті.

Ці дані я знайшов на відкритому просторі в інтернеті, вони містять тисячу продуктів що дуже багато та в більшості випадків буде достатньо цього розділу для того щоб дізнатись про продукт

Ось так вони виглядають у коді

```
{
  "Продукт": "Арахіс",
  "Жири, г": 45.2,
  "Білки, г": 26.3,
  "Вуглеводи, г": 9.9,
  "Калорійність, Ккал": 552
},
{
  "Продукт": "Арахіс смажений",
  "Жири, г": 52,
  "Білки, г": 26,
  "Вуглеводи, г": 13.4,
  "Калорійність, Ккал": 626
},
```

Рисунок 1.24 Скріншот заповнених даних у коді

<div style="display: flex; justify-content: space-between;"> 🔍 ✕ </div>				
Арахіс	Б:	Ж:	У:	Кал:
	26.3	45.2	9.9	552.0
Арахіс смажений	Б:	Ж:	У:	Кал:
	26.0	52.0	13.4	626.0
Горох відварений	Б:	Ж:	У:	Кал:
	10.5	0.8	20.4	130.0
Горошок зелений	Б:	Ж:	У:	Кал:
	5.0	0.2	8.3	55.0
Горошок зелений, консерви	Б:	Ж:	У:	Кал:
	3.1	0.2	6.5	40.0
Гірчиця	Б:	Ж:	У:	Кал:
	25.8	30.8	23.4	474.0
Волоський горіх	Б:	Ж:	У:	Кал:
	16.2	60.8	11.1	656.0
Каштани	Б:	Ж:	У:	Кал:
	2.0	2.0	46.0	213.0
Кедрові горіхи	Б:	Ж:	У:	Кал:
	14.0	68.0	13.0	673.0

Рисунок 1.25 Скріншот меню другого розділу

Також в цьому розділі для зручності можна використовувати пошук.

Третій розділ – це помічник Штучний Інтелект який може використовуватися за допомогою API ChatGPT, він може відповідати на всі питання які цікавлять користувача.

Також при першому відкритті ми можемо побачити повідомлення для більш коректної праці онлайн помічника

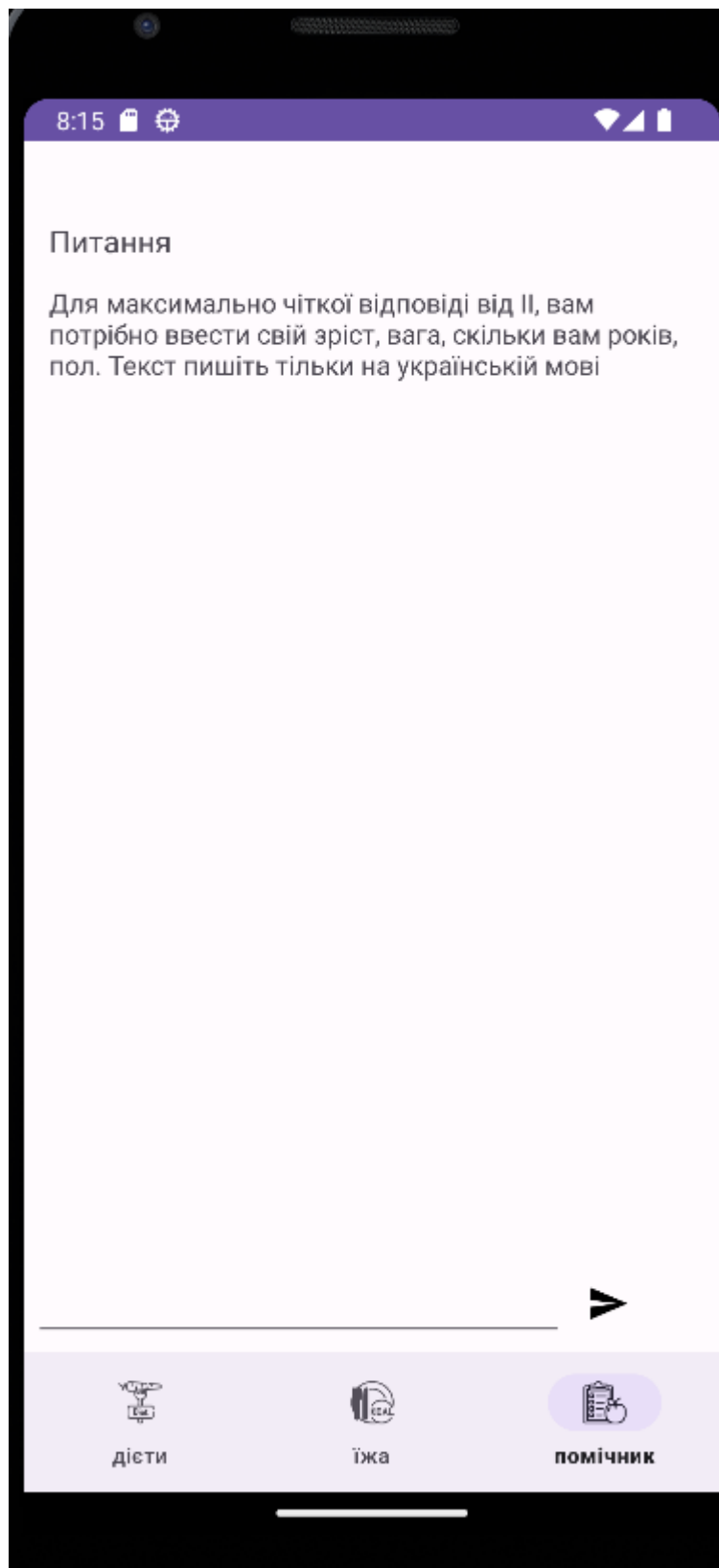


Рисунок 1.26 Скріншот головне меню третього розділу

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

Ось так виглядає сам код помічника:

```
package com.amidomaru.dietapp.bottomnav

import ...

class FragmentHelper : Fragment() {

    private lateinit var binding: FragmentHelperBinding
    private lateinit var ukrainianEnglishTranslator: Translator
    private lateinit var englishUkrainianTranslator: Translator
    private val client = OkHttpClient()

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
        binding = FragmentHelperBinding.inflate(inflater, container, attachToParent: false)
        val root = binding.root

        //перевод с украинского на английский
        val optionsFromRussianToEnglish = TranslatorOptions.Builder()
            .setSourceLanguage(TranslateLanguage.UKRAINIAN)
            .setTargetLanguage(TranslateLanguage.ENGLISH)
            .build()

        //перевод с английского на украинский
        val optionsFromEnglishToUkrainian = TranslatorOptions.Builder()
            .setSourceLanguage(TranslateLanguage.ENGLISH)
            .setTargetLanguage(TranslateLanguage.UKRAINIAN)
            .build()

        ukrainianEnglishTranslator = Translation.getClient(optionsFromRussianToEnglish)
        englishUkrainianTranslator = Translation.getClient(optionsFromEnglishToUkrainian)

        binding.loadingPB.visibility = View.VISIBLE
        ukrainianEnglishTranslator.downloadModelIfNeeded().addOnSuccessListener { it: Void!
            englishUkrainianTranslator.downloadModelIfNeeded().addOnSuccessListener { it: Void!
                }.addOnFailureListener { it: Exception
                    Toast.makeText(requireContext(), text: "Failed to download model", Toast.LENGTH_SHORT).show()
                }
            }.addOnFailureListener { it: Exception
                Toast.makeText(requireContext(), text: "Failed to download model", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Рисунок 1.27 Скріншот коду помічника з пакетами перекладів

Для того щоб отримувати максимально коректну та чітку відповідь у додатку реалізовано встановлення двох мовних пакетів українсько – англійська та англійсько – українська.

Працює це дуже просто, дані ми вводимо на українській, програма переводить на англійська та відправляє це питання на сервіс з штучним інтелектом який в свою чергу дає відповідь на англійській мову, програма переводить с англійської на українську.

Потім за допомогою цієї частини коду ми відправляємо сам текст:

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

```
binding.sendRequestBtn.setOnClickListener { it: View!
    if (binding.requestEt.text.isNotEmpty()) {
        ukrainianEnglishTranslator.translate(binding.requestEt.text.toString()).addOnSuccessListener { it: String!
            onResponse(it) { response ->
                Handler(Looper.getMainLooper()).post {
                    englishUkrainianTranslator.translate(response).addOnSuccessListener { it: String!
                        binding.txtResponse.text = it
                    }
                }
            }
        }
    }
}

return root
}

fun onResponse(question: String, callback: (String) -> Unit){
    //установка текста для вопроса в нижней строке.
    binding.idTVQuestion.text = binding.requestEt.text
    binding.txtResponse.setText("Typing...")
    binding.requestEt.setText("")
}
```

Рисунок 1.28 Скріншот частини коду для вводу тексту

Та для зв'язку з самим штучним інтелектом використовується даний відривок коду:

```
val apiKey="sk-km9R0KK1du3CNALkByNMT3B1bkFJx9jJtL7Qitt8qH5dPhA0"
val url="https://api.openai.com/v1/engines/text-davinci-003/completions"
```

Рисунок 1.29 Скріншот частини коду з зверненням до хмарного сервісу

1.5 Тестування, впровадження та експлуатація програмного продукту

1.5.1 Перевірка коректної роботи реєстрації та аутентифікація

Щоб потрапити до головного меню, потрібно пройти реєстрацію та аутентифікацію тому спробуємо зареєструвати пошту яка не може існувати, для цього заходимо до реєстрації та вводимо пошту без коректного адресу:

testgmail@gmilcom

.....

Рисунок 1.30 Скріншот поле ввода даних

На що застосунок нам одразу каже що ця пошта не може існувати:

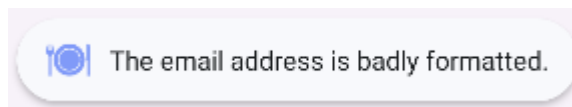


Рисунок 1.31 Скріншот інформацію при неправильному вводу даних

Тому можемо бути впевнені що пошта буде введена коректна.

Провіримо зможемо чи зайти на аккаунт який не існує, для цього переходимо до кнопки “Вхід” та вводим пошту яка ніколи не реєструвалися в системі:

neisnuyacha@gmail.com

.....

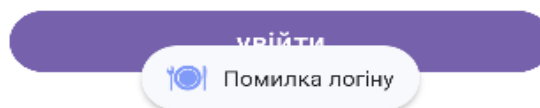


Рисунок 1.32 Неправильний ввід логіну

Нас не пустило до системи, тому аби хто зайти не зможе без реєстрації

Провіriamo можливо чи зайти до аккаунт написавши не правильний пароль.
Для цього вводимо коректну пошту але не правильний пароль та отримуємо:



Рисунок 1.33 Скріншот неправильний ввід логіну або пароля

У нас не вдалося зайти без правильного паролю.

1.5.2 Перевірка правильності роботи розділів

Тепер можна перевірити як працюють розділи, для початку можна перевірити яка себе поведе програма якщо постійно переходити між розділами:

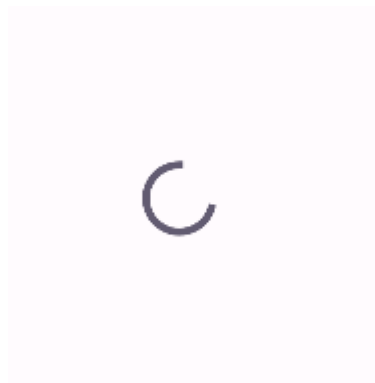


Рисунок 1.34 Скріншот загрузочного екрану

					<i>РП 06. 11 001. ДП ПЗ</i>	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

Скільки би користувач не переходив між розділами, всюди працює спеціальна затримка з візуалізованим колом, що крутиться.

Перевіримо що буде якщо багато разів на один й той же розділ:

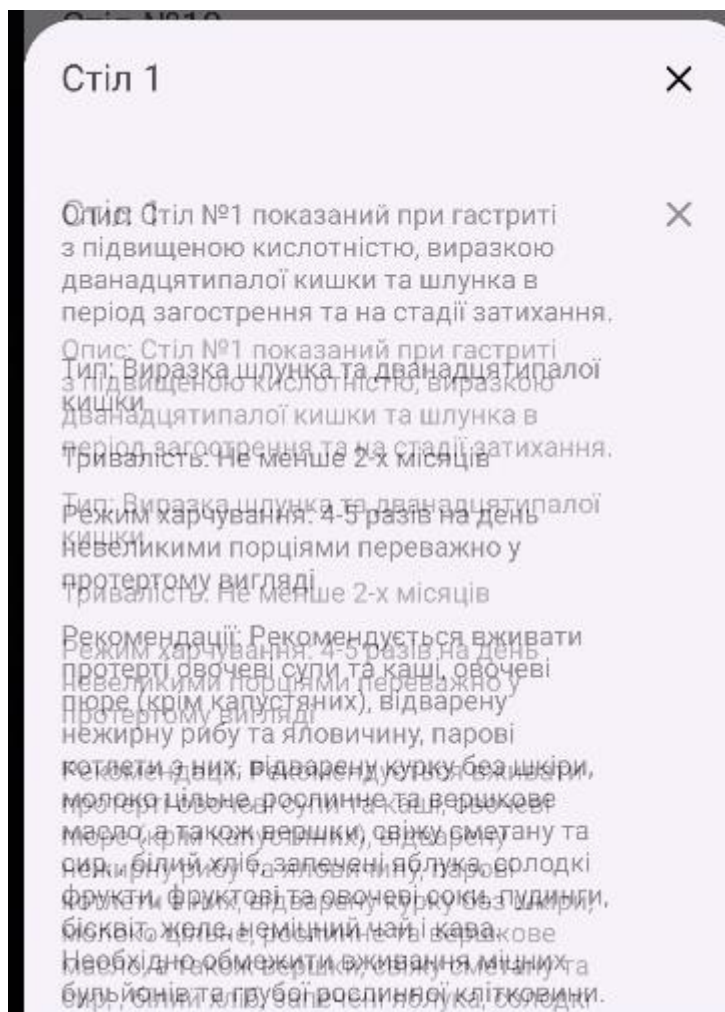


Рисунок 1.35 Скріншот результату багаторазового натискання на дієту

Буде з'являтися 2 меню, але завжди буде залишатися лише одне інформаційне меню, тому програма цей етап пройшла

Протестуємо другий розділ, а саме пошук продукту. Для того щоб перевірити коректно чи працює пошук введемо не існуючі дані наприклад Nuts:

2. ЕКОНОМІЧНА ЧАСТИНА

Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт.

Таблиця 2.1 Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , усл. машинних командах.
1. ПП СУБД	2500 – 9800
2. Комплексні системи ведення БД	950 – 7430
3. ПП введення інформації	1060 – 5750
4. ПП оптимізації розрахунків	1300 – 4200
5. ПП автоматизації засобів по каталогу	680 – 7000
6. ПП автоматизованих розрахунків	1300 – 8600
7. ПП загальної математики і ПП імітаційного моделювання	7800 – 8800
8. ПП організації обчислювального процесу	13000 – 10200
9. ПП оптимізаційних розрахунків	1300 – 4200

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл. 2.2

Таблиця. 2.2 Трудомісткість ПП

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306
6.00	330
7.00	357
8.00	385
9.00	414
10.00	445
12.00	510
14.00	580
16.00	654
18.00	731

На підставі отриманого значення, по таблиці 2.2, визначається укрупнена норма часу на розробку аналога програмного забезпечення, яка коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7 \div 0,8$:

$$T^a = 229 * 0,8 = 183,2 \text{ (люд/годин)}.$$

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

- Розробка технічного завдання

$$T_{ТЗ} = T^a p \times L_1 \times K_H \quad (2.1)$$

- Розробка технічного проекту

$$T_{ТП} = T^a p \times L_2 \times K_H \quad (2.2)$$

- Розробка робочого проекту

$$T_{РП} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.3.);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4.);

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5.).

Таблиця 2.3. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_H
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПП маючий аналог	0,7

Тому що розробка системи є ПП, що має аналоги програмних продуктів, то код ступеня новизни для мого ПП – В, а значення коефіцієнта $K_n = 0,7$. По таблиці 2.4, знаючи код ступеня новизни, тепер можна визначити значення питомих коефіцієнтів трудомісткості (по таблиці 2.3):

$$L_1=0,12 ; L_2=0,11 ; L_3=0,61$$

Значення коефіцієнта ступеня використання в типових програм

Таблиця 2.5. Коефіцієнт типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

У розробленому програмному продукті використовується від 60 та вище відсотків існуючих функцій, це значить, що $K_T = 0,6$ Тепер потрібно розрахувати трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ} = T^a p * L_1 * K_n = 183,2 * 0,12 * 0,7 = 15,38 \text{ (люд/годин)}$$

Трудомісткість розробки технічного проекту

$$T_{ТП} = T^a p * L_2 * K_n = 183,2 * 0,11 * 0,7 = 14,10 \text{ (люд/годин)}$$

Трудомісткість розробки робочого проекту

$$T_{РП} = T^a p * L_3 * K_n * K_T = 183,2 * 0,61 * 0,7 * 0,6 = 46,93 \text{ (люд/годин)}$$

Для подальших розрахунків необхідно визначити кількість папера, витраченого на кожен етап. $N_{ТЗ} = 6$ (стр), $N_{ТП} = 6$ (стр), $N_{РП} = 26$ (стр), $N_{ПЗ} = 35$ (стр) – технічне завдання, розробка технічного проекту, розробка робочого проекту, пояснювальна записка відповідно. Розрахунок зведений у таблицю 2.6

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, години.		
	Розробка ПП	Контроль керівника	Нормоконтроль
1.ТЗ	$T_{РТЗ} = 15,38$	$T_{КК} = 0,7 * N_{ТЗ} = 0,7 * 6 = 4,2$	$T_{НК} = 0,15 * N_{ТЗ} = 0,15 * 6 = 0,9$
2.Розробка ТП	$T_{РТП} = 14,10$	$T_{КК} = 0,7 * N_{ТП} = 0,7 * 6 = 4,2$	$T_{НК} = 0,15 * N_{ТП} = 0,15 * 6 = 0,9$
3.Розробка РП	$T_{РРП} = 46,93$	$T_{КК} = 0,7 * N_{РП} = 0,7 * 26 = 18,2$	$T_{НК} = 0,15 * N_{РП} = 0,15 * 26 = 3,9$

					РП 06. 11 002. ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

Продовження таблиці 2.6 Розрахунок трудомісткості ПП

4.Розробка пояснювальної записки	$T_{рпз}=1,5*N_{рпз}=1,5*35=52,5$	$T_{кк}=0,7*N_{рпз}=0,7*35=24,5$	$T_{нк}=0,15*N_{рпз}=0,15*35=5,25$
Усього, в т.ч.:	$\Sigma T= 190,96$		
- на розробку	$\Sigma T_p= 128,91$		
- контроль керівника		$\Sigma T_{кк}=51,1$	
- нормоконтроль			$\Sigma T_{нк}=10,95$

На основі таблиці 2.6 розрахуємо тривалість розробки в роках:

$$T_{пп}=T/(8.0*0.73*360)= 2102,4 \text{ (р)}, \quad (2.4)$$

8,0 – тривалість робочого дня;

0,73 – коефіцієнт перекладу в календарні дні;

2.2.2 Розрахунок ціни програмного продукту.

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2023» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2023 року – 6700 гривень; мінімальну погодинну тарифну ставку – 40.43 грн.

Таблиця 2.7 Розрахунок основної заробітної плати виконавців.

Найменування робіт	Трудомісткість робіт, роб.години	Годинна тарифна ставка,грн..	Розрахунок, грн.
1.Розробка ПП	$\Sigma T_p= 128,91$	120	15469,2
2.Контроль керівника	$\Sigma T_{кк}= 51,1$	100	5110
3.Нормоконт-ролю	$\Sigma T_{кк}= 10,95$	100	1095
Усього (З _о)	-	-	$\Sigma Z_o= 21674,2$

Розрахунок основної заробітної плати виконуємо по формулі:

$$Z_o = T_{mj} \times Z_{год}, \quad (2.5)$$

T_{mj} – трудомісткість j – того виду робіт, робоч. год;

$Z_{год}$ – погодинна тарифна ставка, грн.

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.8

					РП 06. 11 002. ДП ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.8. Розрахунок матеріальних витрат на розробку ПП

Найменування матеріальних витрат	Тип, модель	Кількість, шт	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	59	3	177
Разом	-	-	-	$V_{M1} = 177$
Транспортно – заготівельні Витрати 10%				$V_{Tr-z} = 0,1 \times V_{M1} = 0,1 \times 177 = 17,7$
Усього				$V_M = V_{M1} + V_{Tr-z} = 122,7$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн	Формула розрахунку
1. Матеріали	177,7	V_M (див. табл. 2.8)
2. Основна заробітна плата	21674,2	Z_o (див. табл. 2.7)
3. Додаткова заробітна плата	2167,4	$Z_d = 0,1 \times Z_o = 21674,2 * 0,1$
4. Відрахування до єдиного фонду соціального внеску	5245,1	$V_{e.c.v.} = 0,22 \times (Z_o + Z_d) = 0,22 * (21674,2 + 2167,4)$
5. Накладні витрати	13004,5	$V_{нак.} = 0,6 \times Z_o = 0,6 * 21674,2$
6. Повна собівартість	42268,9	$C_{пов} = V_M + Z_o + Z_d + V_{e.c.v.} + V_{нак.}$

Розмір прибутку, що включається в ціну, визначається по наступній формулі:

$$П = (C_{пов} * P) / 100 = (42268,9 * 10) / 100 = 4226,8 \text{ (грн)}, \quad (2.6)$$

де p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{пов} + П = 42268,9 + 4226,8 = 46495,7 \text{ (грн)}; \quad (2.7)$$

Податок на додану вартість визначається по наступній формулі:

$$ПДВ = 0,2 * Ц_o = 0,2 * 46495,7 = 9299,1 \text{ (грн)}; \quad (2.8)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + ПДВ = 46495,7 + 9299,1 = 55794,8 \text{ грн}; \quad (2.9)$$

					РП 06. 11 002. ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

3. ОХОРОНА ПРАЦІ

3.1 Вступ

З розвитком сучасного технологічного та наукового прогресу набуває все більшої ваги можливість безпечного виконання працівниками своїх обов'язків. У зв'язку з цим, виникає необхідність вдосконалення системи "Охорона праці".

"Охорона праці" включає комплекс правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та профілактичних заходів і засобів, спрямованих на збереження життя, здоров'я та працездатності людей під час виконання трудових обов'язків.

У даному розділі дипломного проекту будуть розглянуті оптимальні умови праці програмістів з метою забезпечення їхньої безпеки та ефективності праці.

3.2 Розробка заходів з охорони праці

3.2.1 Виробничі приміщення

Вимоги до офісного приміщення, де розміщуються робочі місця з ПК, включають наступне:

- Окреме приміщення: Робочі місця з ПК, як правило, повинні розміщуватися в окремих приміщеннях. У випадку розміщення в спеціальних залах або приміщеннях з небезпечними виробничими факторами, робочі місця повинні бути розміщені у повністю ізольованих кабінетах.
- Природне освітлення: Робочі місця з ПК повинні мати природне освітлення. Це означає, що приміщення повинно мати вікна або інші джерела природного світла.
- Повітрообмін: Робочі місця з ПК у приміщеннях з джерелами небезпечних виробничих факторів повинні мати організований повітрообмін. Це забезпечує належну вентиляцію та зменшує вплив шкідливих речовин на працівників.
- Площа приміщення: Кожне робоче місце з ПК повинно мати площу площадки не менше як 6,0 м², а об'єм приміщення повинен бути не менше як 20,0 м³. Це забезпечує достатню просторову організацію та комфорт працівників.
- Рівна та антистатична підлога: Підлога в офісному приміщенні повинна бути рівною, без вибоїн і неслизькою. Вона також повинна бути зручною

					<i>РП 06. 11 003. ДП ПЗ</i>	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

для очищення та вологого прибирання, а також мати антистатичні властивості для запобігання накопиченню електростатичного заряду.

- Обмеження використання полімерних матеріалів: Забороняється використовувати полімерні матеріали, що виділяють шкідливі хімічні речовини, для оздоблення інтер'єру. Це забезпечує здорове та безпечне робоче середовище.

Ці вимоги спрямовані на забезпечення ергономічних, безпечних та комфортних умов праці для користувачів робочих місць з ПК в офісному середовищі. Вони допомагають запобігати травмам, забезпечують належну вентиляцію та освітлення, а також зменшують вплив шкідливих факторів на здоров'я працівників.

3.2.2 Шуми

Рівень шуму є важливим аспектом охорони праці, оскільки впливає на здоров'я та добробут працівників. У кожній країні встановлені певні стандарти та норми щодо припустимого рівня шуму на робочих місцях. Україна не є винятком і має свої власні нормативні акти, які регулюють це питання.

Україна використовує Державні будівельні норми "Шум та вібрація. Норми безпеки" (ДБН В.2.2-15-2005), які встановлюють максимально допустимі рівні шуму для різних типів промислових та комерційних приміщень.

Згідно з цими нормами, на робочих місцях рівень шуму повинен бути не більше 65 дБ для робочих приміщень, де виконується інтелектуальна робота, така як робота програмістів.

Додатково, Державні санітарні норми та правила "Шум на робочих місцях" (ДСН 3.3.6.037-99) встановлюють конкретні параметри та заходи для контролю рівня шуму, які повинні бути здійснені на робочих місцях для забезпечення безпеки та здоров'я працівників.

3.2.3 Освітлення

У робочих приміщеннях для програмістів, де велика кількість роботи виконується за комп'ютером, рекомендується застосовувати комбіновану

					<i>РП 06. 11 003. ДП ПЗ</i>	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

систему освітлення. Окрім загального освітлення, такі приміщення можуть бути оснащені також світильниками місцевого освітлення.

Необхідна освітленість на поверхні робочого столу в зоні, де розташовано комп'ютер та робочі матеріали, повинна складати 300-500 лк (люксів). При виборі джерел світла для штучного освітлення рекомендується використовувати переважно люмінесцентні лампи типу ЛД, а також допускається використання ламп розжарювання у світильниках місцевого освітлення.

Належне освітлення робочого місця програміста є важливим аспектом забезпечення комфорту та ефективності праці. Застосування комбінованої системи освітлення і дотримання встановлених норм освітленості сприяють зменшенню напруження очей, підвищують зосередженість та продуктивність працівників у процесі роботи з комп'ютером.

3.2.4 Мікроклімат

Мікроклімат на робочому місці програміста має важливе значення. Варто забезпечити оптимальну температуру в діапазоні 20-24°C та вологість повітря на рівні 40-60%. Потрібно запобігати переохолодженню та дискомфорту, а також забезпечити достатній доступ до свіжого повітря. Дотримання правил провадження робіт та технічної експлуатації є обов'язковим. Також варто застосовувати ізоляцію для захисту від електричного струму.

3.2.5 Пожежна безпека

Пожежна безпека в офісному приміщенні є важливою складовою забезпечення безпеки працівників і майна. Деякі засоби безпеки, такі як вогнегасники, мають важливе значення при запобіганні та виявленні пожеж.

Один з найважливіших засобів пожежної безпеки в офісі - це вогнегасник. Вогнегасник є портативним пристроєм, призначеним для тушіння невеликих початкових пожеж. Він має бути розташований у відповідному місці, легкодоступному для всіх працівників. Крім того, працівники повинні бути ознайомлені з розташуванням вогнегасників, їх типом та правилами їх використання.

					<i>РП 06. 11 003. ДП ПЗ</i>	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

У офісному приміщенні з технікою рекомендується використовувати вогнегасник типу АВС (порошковий вогнегасник з універсальним вогасником). Це тип вогнегасника є найбільш універсальним і ефективним для пожеж різного типу, які можуть виникнути в офісному середовищі.

Вогнегасник типу АВС містить суміш спеціальних хімічних речовин, які можуть загасити пожежі на основі твердих речовин (тип А), рідин (тип В) і газів (тип С). У офісному середовищі можуть бути присутні різні матеріали, які можуть спричинити пожежу, такі як папір, дерево, текстиль, електроніка, побутові рідини і гази. Тому вогнегасник типу АВС є найбільш універсальним і здатним справлятися з широким спектром пожежних ситуацій.

Крім того, порошкові вогнегасники типу АВС є ефективними погасником пожеж, вони швидко загасають вогонь і мають досить велику дальність дії. Вони також володіють властивістю попереджувати подальше розповсюдження вогню, що може бути критичним у випадку пожежі з технікою.

Додатковими засобами пожежної безпеки можуть бути димові детектори, автоматичні системи пожежогасіння та евакуаційні плани. Димові детектори виявляють наявність диму і видають звуковий сигнал, щоб сповістити працівників про пожежу. Автоматичні системи пожежогасіння та пожежні спринклерні системи активуються автоматично при виявленні вогню або підвищенні температури. Евакуаційні плани надають працівникам інформацію про безпечний шлях виходу з приміщення у разі пожежі.

3.2.6 Електробезпека

Використання електричної енергії на робочому місці програміста пов'язане з потенційною небезпекою впливу електричного струму на людський організм.

Ризик ураження електричним струмом може виникнути під час роботи під напругою або при несправному стані електроустановок, таких як дотик до оголених проводів, незаземлених металевих корпусів електричного обладнання, відкритих рубильників та інших струмопровідних частин.

					<i>РП 06. 11 003. ДП ПЗ</i>	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

Для запобігання ураженню електричним струмом необхідно дотримуватись таких правил безпеки:

- Суворо дотримуватись правил проведення робіт і правил технічної експлуатації.
- Забезпечити відсутність доступу працівника до частин устаткування, що працюють під небезпечною напругою, неізольованих деталей, які призначені для роботи при низькій напрузі і не підключені до захисного заземлення.
- Використовувати ізоляцію, яка служить для захисту від ураження електричним струмом.
- Заземлені конструкції, що знаходяться у приміщенні з робочими місцями програмістів (батареї опалення, водопровідні труби, кабелі з заземленим відкритим екраном), повинні бути надійно захищені діелектричними щитками або сітками, щоб запобігти доступу працівника до напруги.
- У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, необхідно встановити аварійний резервний вимикач на видному та доступному місці. Цей вимикач здатний повністю відключити електричне живлення приміщення, залишаючи освітлення працездатним.

Дотримання вищезазначених заходів забезпечує безпеку під час роботи програмістів з електричною енергією і зменшує ризик ураження струмом, забезпечуючи безпечні умови праці.

					<i>РП 06. 11 003. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

ВИСНОВКИ

Зважаючи на проведену аналітичну та практичну роботу в рамках даної дипломної роботи, можна зробити наступні висновки:

У процесі дослідження було проведено аналіз актуальних проблем і тенденцій у вибраній області. Це дозволило отримати всебічне розуміння контексту та основних факторів, що впливають на розробку програмного продукту.

Визначено основні цілі та завдання дипломної роботи, які були успішно досягнуті. Розробка програмного продукту була здійснена відповідно до вимог та специфікацій, що були визначені на початковому етапі.

Використано певні методи та технології для реалізації програмного продукту. Досліджено й використано сучасні підходи та кращі практики, що дозволило забезпечити якість та ефективність розробленого продукту.

Здійснено тестування та аналіз результатів. Це дало можливість виявити та виправити помилки та недоліки, а також підтвердити відповідність програмного продукту вимогам та очікуванням користувачів.

Охорона праці та безпека були враховані у процесі розробки та використання програмного продукту. Були прийняті необхідні заходи та рекомендації щодо забезпечення безпеки працюючих, а також запобігання можливим ризикам та аварійним ситуаціям.

Дипломна робота внесла певний внесок у вибрану галузь, доповнила існуючі знання та надала практичні рекомендації для подальшого вдосконалення програмного продукту або подібних проектів.

Отже, в цілому, дипломна робота є важливим кроком у професійному розвитку та внеском у вибрану галузь. Результати дослідження, розробки та використання програмного продукту підтверджують успішність проекту та його потенціал для подальшого розширення та використання.

					<i>РП 06. 11 003. ДП ПЗ</i>	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kotlin Reference. *github*.
URL: <http://ac2epsilon.github.io/TRANS/KOTLIN/KotlinReference.html> (date of access: 28.05.2023).
2. КОМУНАЛЬНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО ЗАКАРПАТСЬКИЙ ОБЛАСНИЙ ЦЕНТР ГРОМАДСЬКОГО ЗДОРОВ'Я ЗАКАРПАТСЬКОЇ ОБЛАДИ. ЗДОРОВЕ ХАРЧУВАННЯ збірник матеріалів для працівників системи охорони здоров'я: автореф. збірник матеріалів. Ужгород, 2020. 64 с.
URL: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/29992/1/інформ.Харч.%20збірник%20нові%20поля.pdf> (дата звернення: 01.06.2023).\
3. Global Nutrion Report. Частка населення яка страждає на ожиріння, 2021. *Forbes*. URL: <https://forbes.ua/ru/news/16-doroslogo-naselennya-ukraini-strazhdae-na-ozhirinnya-yak-biznes-adaptuetsya-do-pogladshannya-ukraintsiv-10092021-2401> (дата звернення: 01.06.2023).
4. Дія. Як штучний інтелект використовується у сфері відкритих даних. *Дія.Відкриті дані*. URL: <https://diia.data.gov.ua/info-center/aiod> (дата звернення: 01.06.2023).
5. КПІ ім. Ігоря Сокорського. Основи здорового способу життя: автореф. Практична робота. Київ, 2022. 419 с.
URL: <https://ela.kpi.ua/bitstream/123456789/50782/1/OZSZh.pdf> (дата звернення: 01.06.2023).
6. Філановський О. Особливості використання програми MyFitnessPal. *ICOOLA*. URL: <https://icoola.ua/blog/my-fitness-pal-app/> (дата звернення: 01.06.2023).
7. Lifesum. Lifesum: Healthy Eating & Diet – Додатки в Google Play. *Android Apps on Google Play*. URL: <https://play.google.com/store/apps/details?id=com.sillens.shapeupclub&hl=uk> (дата звернення: 01.06.2023).

					РП 06. 11 003. ДП ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

8. YAZIO. YAZIO Fasting & Food Tracker. *Android Apps on Google Play*. URL: <https://play.google.com/store/apps/details?id=com.yazio.android&hl=uk> (date of access: 01.06.2023).

9. KOTLIN. *Brander*. URL: <https://brander.ua/technologies/kotlin> (date of access: 01.06.2023).

10. Миронова Н. Тренды здорового питания 2021: как совместить вкусное с полезным. *The Page*. URL: <https://thepage.ua/style/health/zdorovoe-pitanie-2021-ration-produkty-i-trendy> (дата звернення: 05.06.2023).

11. ОППВ. Оснащення офісних приміщень первинними засобами пожежогасіння - Охорона праці і пожежна безпека. *Охорона праці і пожежна безпека*. URL: <https://oppb.com.ua/articles/osnashchennya-ofisnyh-prymishchen-pervynnymy-zasobamy-pozhezhogasinnya> (дата звернення: 07.06.2023).

					<i>РП 06. 11 003. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

ДОДАТОК А. Фрагмент лістингу першого розділу III для підбору дієтичного харчування

```
package com.amidomaru.dietapp.bottomnav

import android.os.Bundle
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.recyclerview.widget.DividerItemDecoration
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.amidomaru.dietapp.databinding.FragmentDietsBinding
import com.amidomaru.dietapp.diets.Diet
import com.amidomaru.dietapp.diets.DietsAdapter
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ValueEventListener

class FragmentDiets : Fragment() {

    private lateinit var binding: FragmentDietsBinding

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
        binding = FragmentDietsBinding.inflate(inflater, container, false)
        val root = binding.root

        binding.therapeuticDietsCb.isChecked = true
        loadTherapeuticDiets()

        binding.correctionDietsCb.setOnCheckedChangeListener { buttonView, isChecked ->
            if (isChecked) {
                binding.therapeuticDietsCb.isChecked = false
                loadCorrectionDiets()
            }
        }

        binding.therapeuticDietsCb.setOnCheckedChangeListener { buttonView, isChecked ->
            if (isChecked) {
                binding.correctionDietsCb.isChecked = false
                loadTherapeuticDiets()
            }
        }

        return root
    }

    private fun loadCorrectionDiets() {
```

```
binding.loadingPB.visibility = View.VISIBLE
binding.root.isClickable = false
```

```
FirebaseDatabase.getInstance().reference.child("Diets").addListenerForSingleValueEvent(object:
 ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        var diets = ArrayList<Diet>()
        for (snapshot1 in snapshot.children){
            if (snapshot1.key.equals("t16")||snapshot1.key.equals("t17")
                ||snapshot1.key.equals("t18")) {
                val name = snapshot1.child("name").value.toString()
                val type = snapshot1.child("type").value.toString()
                val duration = snapshot1.child("duration").value.toString()
                val recommend = snapshot1.child("recommend").value.toString()
                val desc = snapshot1.child("desc").value.toString()
                val diet = snapshot1.child("diet").value.toString()

                diets.add(Diet(desc, diet, duration, name, recommend, type))
            }
        }
    }
}
```

```
//заполняем наш RecyclerView
val layoutManager = LinearLayoutManager(context, RecyclerView.VERTICAL, false)
val dividerItemDecoration = DividerItemDecoration(
    binding.dietsRv.getContext(),
    layoutManager.orientation
)
val adapter = DietsAdapter(diets, parentFragmentManager)
```

```
binding.dietsRv.addItemDecoration(dividerItemDecoration)
binding.dietsRv.layoutManager = layoutManager
binding.dietsRv.adapter = adapter
```

```
binding.root.isClickable = true
binding.loadingPB.visibility = View.GONE
}
```

```
override fun onCancelled(error: DatabaseError) {
    Log.d("TAG", "onCancelled: ${error.message}")
}
```

```
})
}
```

```
private fun loadTherapeuticDiets() {
    binding.loadingPB.visibility = View.VISIBLE
    binding.root.isClickable = false
```

```
FirebaseDatabase.getInstance().reference.child("Diets").addListenerForSingleValueEvent(object:
 ValueEventListener {
```

```

override fun onDataChange(snapshot: DataSnapshot) {
    var diets = ArrayList<Diet>()
    for (snapshot1 in snapshot.children){
        if (!snapshot1.key.equals("t16")&&!snapshot1.key.equals("t17")
            &&!snapshot1.key.equals("t18")) {
            val name = snapshot1.child("name").value.toString()
            val type = snapshot1.child("type").value.toString()
            val duration = snapshot1.child("duration").value.toString()
            val recommend = snapshot1.child("recommend").value.toString()
            val desc = snapshot1.child("desc").value.toString()
            val diet = snapshot1.child("diet").value.toString()

            diets.add(Diet(desc, diet, duration, name, recommend, type))
        }
    }

    //заполняем наш RecyclerView
    val layoutManager = LinearLayoutManager(context, RecyclerView.VERTICAL, false)
    val dividerItemDecoration = DividerItemDecoration(
        binding.dietsRv.getContext(),
        layoutManager.orientation
    )
    val adapter = DietsAdapter(diets, parentFragmentManager)

    binding.dietsRv.addItemDecoration(dividerItemDecoration)
    binding.dietsRv.layoutManager = layoutManager
    binding.dietsRv.adapter = adapter

    binding.root.isClickable = true
    binding.loadingPB.visibility = View.GONE
}

override fun onCancelled(error: DatabaseError) {
    Log.d("TAG", "onCancelled: ${error.message}")
}

})
}
}

```

Розробка програмного забезпечення для підбору дієтичного харчування

Виконав:

Іщенко Андрій Віталійович

Керівник:

к.т.н., доц. Селіванова Алла Віталіївна

- ▶ Зростання інтересу до здорового способу життя
- ▶ Проблеми з харчуванням та ожиріння
- ▶ Індивідуалізація підходу до харчування
- ▶ Сприяння здоров'ю та профілактика захворювань



Таким чином, програма для підбору дієтичного харчування буде актуальною поки люди будуть цікавитись своїм здоров'ям

Актуальність

- ▶ **Метою даної роботи** головною метою є створення ПЗ для поліпшення здоров'я та досягнення поставлених цілей з харчування.
- ▶ **Постановка задачі:**
 - ▶ Збір вихідних даних
 - ▶ Розробка алгоритму підбору
 - ▶ Розробка користувацького інтерфейсу
 - ▶ Інтеграція зі сторонніми сервісами
 - ▶ Тестування та валідація
 - ▶ Розгортання та підтримка

Мета, задачі

Продукти	MyFitnessPal	Lifesum	Yazio
Функціональність	Моніторинг харчування, статистика, рекомендації	Моніторинг харчування, тренування, здоров'я	Моніторинг харчування, персоналізовані плани
База даних продуктів	Так	Так	Так
Персоналізація	Обмежена	Деяка	Є
Машинне навчання	Обмежений	Обмежений	Невідомо
Інтеграція зі сторонніми сервісами	Так	Деяка	Невідомо
Інноваційні можливості	Залежно від версії та платного плану	Деякі інноваційні можливості	Обмежені інноваційні можливості

Аналіз аналогів

- ▶ В якості бази даних Firebase
- ▶ Мова програмування - Kotlin
- ▶ Середа розробки - Android Studio

Засоби реалізації

5

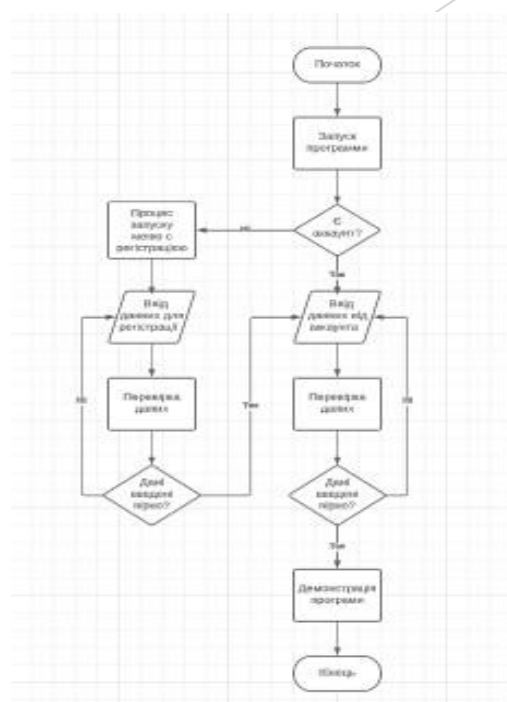


Схема інтерфейсу реєстрації

6

6

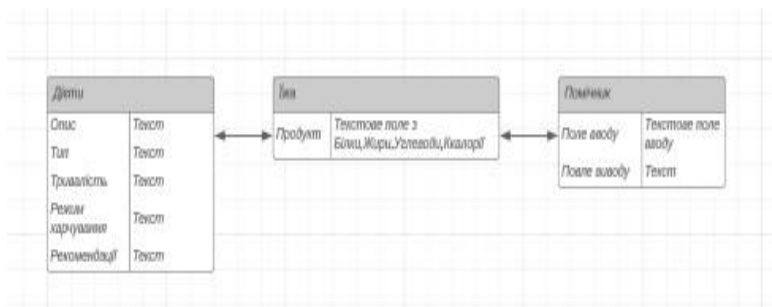


Схема інтерфейсу

7

7

- Для упорядкування та правильної роботи коду все було розділено так:



Особливості розробки

8

8

- ▶ Головна сторінка яка вже буде переправляти на потрібні розділи

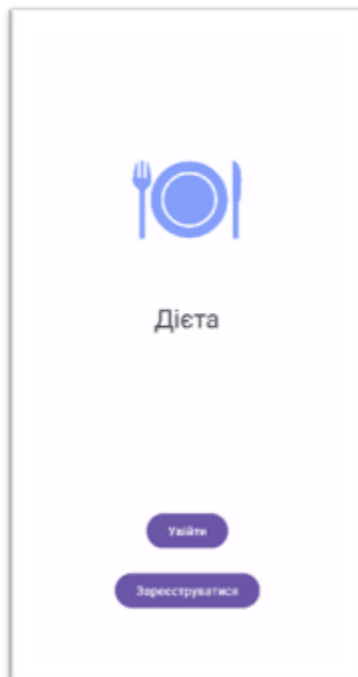
```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        supportFragmentManager.beginTransaction().replace(R.id.home_container, FragmentDiets()).commit ()  
        binding.navView.selectedItemId = R.id.navigation_diets  
  
        //обробка навігації на основі меню  
        binding.navView.setOnItemSelectedListener { item ->  
            when (item.itemId) {  
                R.id.navigation_diets -> {  
                    loadFragment(FragmentDiets())  
                    true //задієт вибрано  
                }  
  
                R.id.navigation_food -> {  
                    loadFragment(FragmentFood())  
                    true //задієт вибрано  
                }  
  
                R.id.navigation_info -> {  
                    loadFragment(FragmentHelper())  
                    true //задієт вибрано  
                }  
            }  
        }  
        else -> {  
            loadFragment(FragmentDiets())  
            true //задієт вибрано  
        }  
    }  
}
```

Особливості розробки

9

9

Робота із програмою розпочинається з реєстрації та авторизації

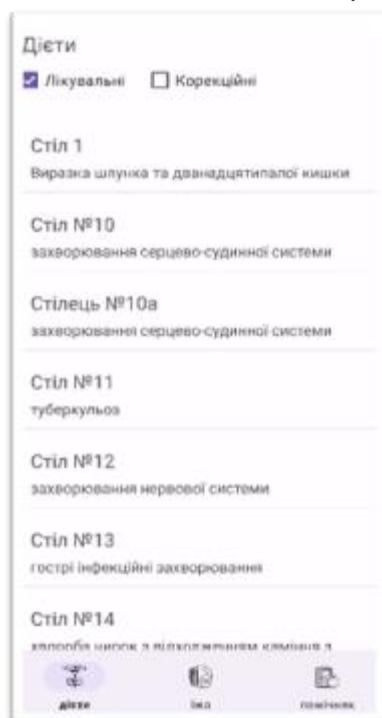


10

10

- ▶ Після реєстрації ми можемо побачити перший розділ з дієтами

Інтерфейс



11

11

- ▶ При переході на розділ “Їжа” ми можемо побачити такий інтерфейс

Інтерфейс

Їжа	Б:	Ж:	У:	Кал:
Арахіс	26.3	45.2	9.9	552.0
Арахіс смажений	26.0	52.0	13.4	626.0
Горох відварений	10.5	0.8	20.4	130.0
Горошок зелений	5.0	0.2	8.3	55.0
Горошок зелений, консерви	3.1	0.2	6.5	40.0
Гірчиця	25.8	30.8	23.4	474.0
Волоський горіх	16.2	60.8	11.1	656.0
Каштани	2.0	2.0	46.0	213.0
Кедрові горіхи	14.0	68.0	13.0	673.0

12

12

- ▶ Третій інтерфейс “Помічник”



Інтерфейс

13

- ▶ У процесі дослідження було проведено аналіз актуальних проблем і тенденцій у вибраній області. Це дозволило отримати всебічне розуміння контексту та основних факторів, що впливають на розробку програмного продукту.
- ▶ Визначено основні цілі та завдання дипломної роботи, які були успішно досягнуті. Розробка програмного продукту була здійснена відповідно до вимог та специфікацій, що були визначені на початковому етапі.
- ▶ Використано певні методи та технології для реалізації програмного продукту. Досліджено й використано сучасні підходи та кращі практики, що дозволило забезпечити якість та ефективність розробленого продукту.

Висновки

14

14

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Іщенко Андрія Віталійовича

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) к.т.н., доц. Селіванова Алла Віталіївна

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) «Розробка програмного забезпечення для підбору дієтичного харчування»

Обсяг розрахунково-пояснювальної записки _____ сторінок

Обсяг графічної (презентаційної) частини _____ аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню Представлений на рецензію дипломний проект повністю відповідає меті проектування та технічному завданню. Тематика дипломного проекту є актуальною і присвячена розробці програмного забезпечення для вивчення підбору дієтичного харчування. Впровадження та використання розробленого програмного забезпечення сприяє розповсюдженню інформації про здорове харчування та допомагає хворим підібрати раціон у відповідності до рекомендацій лікарів.

б) характеристика виконання кожного розділу дипломного проекту (роботи) _____
Дипломний проект складається з вступу, трьох розділів, висновків, переліку використаних джерел. У технологічному розділі виконано огляд і аналіз технологій створення програмного забезпечення для підбору раціону харчування та програмних засобів, що вирішують аналогічні задачі, розроблено інформаційну модель системи, спроектовано базу даних та додаток, здійснено розробку мобільного програмного забезпечення за допомогою Firebase, Kotlin та Android Studio. Другий та третій розділ присвячено економічній частині та охороні праці відповідно. Перелік літературних джерел виконано у відповідності до діючого стандарту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи) Графічна частина виконана на достатньо високому рівні у вигляді презентації із використанням офісного пакету MS PowerPoint. Пояснювальна записка включає всі необхідні

компоненти і виконана у відповідності до норм оформлення документів із використанням офісного пакету MS Word. Має певні фразеологічні недоліки. Загальна якість виконання документації – добра, перевірку на плагіат робота пройшла.

г) перелік позитивних якостей дипломного проекту (роботи)

1. Дружній інтерфейс програмного забезпечення.

2. Можливість використання на мобільному пристрої.

3. Використання сучасних технологій розробки.

д) основні недоліки дипломного проекту (роботи)

1. Текстова частина програмного забезпечення має недоліки.

2. Використання не реляційної СУБД потребує додаткових засобів захисту від аномалій відновлення.

3. У програмне забезпечення закладено не велику кількість дієт.

Оцінка розрахункової частини «відмінно»

Оцінка графічної частини «добре»

Загальна оцінка «добре»

Прізвище, ім'я, по батькові рецензента к.т.н., доцент Сахарова Світлана Валеріївна

Місце роботи і посада рецензента к.т.н., доцент кафедри комп'ютерної інженерії Одеського національного технологічного університету.

Підпис:

« 15 » червня 2023 р.



ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Іщенко Андрія Віталійовича

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: «Розробка програмного забезпечення для підбору дієтичного харчування»

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) дипломний проект Іщенко Андрія Віталійовича виконано згідно завданню і складається з пояснювальної записки і графічного матеріалу у вигляді мультимедійної презентації. Робота виконана у відповідності до вимог у повному обсязі.

б) самостійність роботи над проектом: дипломник Іщенко Андрій Віталійович працював над дипломним проектом самостійно у відповідності із затвердженим календарним планом.

в) теоретична підготовка випускника (випускниці): під час роботи над дипломним проектом Іщенко Андрій Віталійович показав добрий рівень теоретичної підготовки та вміння застосовувати отримані під час навчання знання та навички при вирішенні практичних завдань.

г) вміння розв'язувати виробничі та конструкторські питання в процесі роботи над дипломним проектом Іщенко Андрій Віталійович показав вміння розв'язувати виробничі та конструкторські питання, а саме вміння проектувати програмне забезпечення, реалізовувати функціональні елементи системи із застосуванням відповідних інструментальних засобів.

Оцінка розрахункової частини «відмінно»

Оцінка графічної частини «добре»

Загальна оцінка «відмінно»

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Селіванова Алла Віталіївна

Місце роботи і посада керівника дипломного проекту доцент кафедри

інформаційних технологій та кібербезпеки ОНТУ

Підпис _____



« 08 » _____ червня _____ 2023 р.

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015507150

Дата перевірки:
08.06.2023 14:23:58 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
08.06.2023 14:24:33 EEST

ID користувача:
100011688

Назва документа: 4РП-06 Іщенко Андрій

Кількість сторінок: 65 Кількість слів: 7295 Кількість символів: 54754 Розмір файлу: 2.57 MB ID файлу: 1015162561

4.21% Схожість

Найбільша схожість: 2.15% з Інтернет-джерелом (<https://oppb.com.ua/articles/osnashchennya-ofisnyh-prymishchen-pe..>)

4.21% Джерела з Інтернету

251

Сторінка 67

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

28

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Ищенко Андрій Віталійович,
здобувач освіти гр. 4РП-06, та

Селіванова Алла Віталіївна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи фахового молодшого бакалавра на тему: «Розробка програмного забезпечення для підбору дієтичного харчування» (*автор роботи – Ищенко А.В., керівник роботи – Селіванова А.В.*)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець  / Ищенко А.В. /

Керівник  / Селіванова А.В. /

« _____ » _____ 2023р.