

**Київський національний університет імені Тараса Шевченка**

Факультет інформаційних технологій

Кафедра програмних систем і технологій

УДК \_\_\_\_\_

На правах рукопису

# **ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА**

**Тема: “Фітнес додаток для визначення раціону студента”**

*(назва згідно з наказом ректора)*

**Спеціальність – 121 “Інженерія програмного забезпечення”**

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

**БР.ІПЗ - 22.00.00.000**

### **Студент**

**ІПЗ-41\_\_\_\_\_ /Олександр СКІБЦЬКИЙ/**

(шифр групи) (підпис) (дата) (розшифровка підпису)

### **Науковий керівник**

**к.ф.м.н., доц. \_\_\_\_\_ /Катерина МЕРКУЛОВА/**

(посада) (підпис) (дата) (розшифровка підпису)

**Допускається до захисту з питань  
нормоконтролю**

**фахівець \_\_\_\_\_ /Гамара ЧАПОВСЬКА/**

### **Завідувач кафедри**

**д.т.н., проф. \_\_\_\_\_ /Олексій БИЧКОВ/**

(посада) (підпис) (дата) (розшифровка підпису)

Київ – 2021

**Київський національний університет імені Тараса Шевченка**

Факультет інформаційних технологій

Кафедра програмних систем і технологій

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖЕНО

Зав. кафедри програмних систем і технологій

\_\_\_\_\_ ( Олексій БИЧКОВ )

(підпис)

(прізвище та ініціали)

## **ЗАВДАННЯ**

### **НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

**Скібіцькому Олександровичу**

- 
- 1. Тема бакалаврської роботи “ Фітнес додаток для визначення раціону студента ”, керівник проекту (роботи) Меркулова Катерина Володимирівна, к.т.н., доцент \_\_\_\_\_ затверджені наказом вищого навчального закладу від “ 11 ” листопаду 2020 р. № 736/7 \_\_\_\_\_**
  - 2. Строк подання студентом роботи 18 лютого 2021 р.**
  - 3. Вихідні дані до проекту (роботи) Базові концепції та розуміння проектування та розробки програмних технологій**
  - 4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**
    1. Вибір середі розробки та її основні переваги
    2. Створення програмної частини додатку
    3. Розробка UX дизайну з урахуванням потреб кінцевого користувача
    4. Тестування кінцевої версії продукту

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

1. Дизайн додатку (рис. 2.1, ст. 17)
2. Uml діаграма (рис. 3, ст. 21)
3. Перша сторінка (рис. 3.1, ст. 22)
4. Друга сторінка (рис. 3.6, ст. 26).
5. Інверсний морфізм (рис. 3.16, ст. 34)

## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Проектування додатку	Меркулова Катерина Володимиріна		

7. Дата видачі завдання 13 жовтня 2019 р.

Керівник \_\_\_\_\_ (Катерина МЕРКУЛОВА)

Завдання прийняв до виконання \_\_\_\_\_ (Олександр СКІБЦЬКИЙ)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів бакалаврської роботи	Термін виконання етапів роботи	Примітка
1	Проведення досліджень літератури		Виконано
2	Дослідження можливих алгоритмів		Виконано
3	Вивчення побудови додатку		Виконано
4	Проектування		Виконано
5	Опис розробленого алгоритму		Виконано
6	Затвердження пояснювальної записки роботи завідуючого кафедри		Виконано

Студент – бакалавр \_\_\_\_\_ (Олександр СКІБЦЬКИЙ)

Керівник роботи \_\_\_\_\_ (Катерина МЕРКУЛОВА)

## АНОТАЦІЯ

**Випускна кваліфікаційна бакалаврська робота:** 60 с., 31 рис., 2 табл., 1 додат., 15 джерел.

**Тема:** фітнес додаток для визначення раціону студента

**Об'єкт дослідження:** публікації різних додатків аналогів в мережі Інтернет

**Мета роботи:** розробка додатку спрощення процесу приготування страв студента

**Предмет дослідження:** Потреби споживача в оптимізації процесу приготування їжі з урахуванням особливостей потреб споживача в зменшенні, збільшенні, чи підтриманні маси тіла.

**Результати дослідження:** Методом експертної оцінки було виявлено високий рівень попиту обраного сегменту споживачів(студентів) на програмне забезпечення для приготування страв з мінімальною кількістю операцій задля максимізації скорочення часу на весь процес готування.

### **Висновок**

В ході даної дипломної роботи було розроблено та протестовано програмне забезпечення з приготування їжі з урахування потреб обраного сегменту споживачів.

ДОДАТОК,ПРОГРАМА,ПРОЕКТУВАННЯ,РОЗРОБКА,ПОТРЕБИ,  
ДОСЛІДЖЕННЯ, ПРОГРАМУВАННЯ, ДИЗАЙН.

## SUMMARY

**Final qualifying bachelor's thesis:** 60 pages, 21 figures, 2 tables, 1 appendices, 33 sources.

**Topic:** Fitness app that help students determine their diet

**Object of research:** publications of various analog applications on the Internet

**Purpose:** to develop an application to simplify the process of cooking meals of students

**Subject of research:** Consumer needs in optimizing the cooking process taking into account the characteristics of the consumer's needs to reduce, increase or maintain body weight.

**Research results:** The method of expert evaluation revealed a high level of demand of the selected segment of consumers (students) for software for cooking with a minimum number of operations to maximize the reduction of time for the entire cooking process.

### Conclusion

During working on this diploma project there was developed and tested software for cooking to meet the needs of the selected segment of consumers.

APPLICATION, PROGRAM, DESIGNING, DEVELOPMENT, DEMAND,  
RESEARCH, PROGRAMMING, DESIGN.

## АННОТАЦИЯ

**Выпускная квалификационная бакалаврская работа:** 60 с., 31 рис., 2 табл., 1 доп., 15 источника.

**Тема:** фитнес приложение для определения рациона студента

**Объект исследования:** публикации различных приложений аналогов в сети Интернет.

**Цель работы:** разработка приложения упрощения процесса приготовления блюд студента

**Предмет исследования:** Потребности потребителя в оптимизации процесса приготовления пищи с учетом особенностей потребностей потребителя в уменьшении, увеличении, или поддержании массы тела.

**Результаты исследования:** Методом экспертной оценки было выявлено высокий уровень спроса избранного сегмента потребителей (студентов) на программное обеспечение для приготовления блюд с минимальным количеством операций для максимизации сокращения времени на весь процесс приготовления.

### Вывод

В ходе данной дипломной работы была разработана и протестирована программное обеспечение по приготовлению пищи с учетом потребностей избранного сегмента потребителей.

ПРИЛОЖЕНИЕ, ПРОГРАММА, ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА,  
ПОТРЕБНОСТИ, ИССЛЕДОВАНИЯ, ПРОГРАМИРОВАНИЕ, ДИЗАЙН

## ЗМІСТ

Зміст .....	7
Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	8
Вступ .....	9
<b>РОЗДІЛ 1</b>	
Створення концепції додатку .....	11
1.1 Огляд існуючих додатків аналогів .....	11
<b>РОЗДІЛ 2</b>	
<b>СТВОРЕННЯ ДИЗАЙНУ</b> .....	17
2.1 Дизайн додатку .....	17
2.2 Фото для додатку .....	18
<b>РОЗДІЛ 3</b>	
Розробка програмної частини .....	21
3.1 Архітектура додатку .....	21
3.2 Перша сторінка .....	22
3.3 Друга сторінка .....	26
3.4 Третя сторінка .....	32
3.5 Четверта сторінка .....	34
Висновки .....	35
Список використаних джерел .....	36
Додатки .....	38

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

ПЗ	-	програмне забезпечення
AS	-	android studio
UX	-	user experience
SDK	-	software development kit

## **ВСТУП**

### **Актуальність роботи**

Даний програмний продукт на сьогоднішній день є все більше актуальним в зв'язку з глобальною тенденцією до ведення здорового способу життя суспільством. З кожним роком оптимізують все більше повсякденних речей і темп життя постійно зростає. Відсутність налагодженої системи харчування та сформованого режиму прийому їжі знижує рівень як фізичного так і психоемоційного стану людини. Також в зв'язку з тенденцією до глобалізації, підвищенням інтенсивності життя та скороченням тривалості більшості бізнес процесів виникає потреба в зменшенні часу у виборі страв та приготуванні їжі. Додаток, що створений в ході цієї дипломної роботи допоможе з мінімальною кількістю дій знайти рецепт, відповідно до інгредієнтів які містяться в холодильнику. Використавши дане ПЗ можна мінімізувати витрати часу на приготування їжі та сфокусуватись на своїх цілях.

### **Порівняння роботи з відомими розв'язаннями проблеми**

Схожого додатку який вирішував би дану проблему не існує, тому наше рішення є унікальним оскільки воно поєднує в собі потреби споживача та сучасне технічне рішення загорнуте в сучасний дизайн.

Хоча існує багато програм, написані іншими програмістами, які знаходять рецепти або дають фітнес поради лише в плані фізичних вправ і не містять ніякої інформації про харчування.

В даній дипломній роботі були використані основні переваги проаналізованих додатків які є на ринку, та добавлено те, чого потребує обраний сегмент споживачів (студенти)

### **Мета і задачі дослідження**

Метою бакалаврської роботи є розробка додатку оптимізації процесу приготування їжі з урахуванням особливостей потреб споживача в зменшенні, збільшенні, чи підтриманні маси тіла з мінімальною кількістю операцій задля максимізації скорочення часу на весь процес.

Щоб досягти мети було розв'язано такі задачі:

- ❖ Огляд додатків-аналогів на ринку
- ❖ Визначення недоліків та переваг існуючих додатків
- ❖ Проектування власного покращеного додатку
- ❖ Вибір середовища та мови розробки
- ❖ Створення UX дизайну

**Об'єктом дослідження** є публікації різних додатків аналогів в мережі Інтернет.

**Предметом дослідження** Потреби споживача в оптимізації процесу приготування їжі з урахуванням особливостей потреб споживача в зменшенні, збільшенні, чи підтриманні маси тіла.

### **Методи дослідження**

Для роботи з програмним забезпеченням були використані методи моделювання реляційних баз даних та використання нормальних форм проектування реляційних структур даних.

### **Практичне значення одержаних результатів**

Створений додаток допоможе студентам зменшити витрати часу на приготування їжі та допоможе контролювати свій режим харчування залежно від їх потреб. Особливо даний додаток буде корисний для першокурсників, адже для більшості з них вдома готувала мама і потрібних знань вони не мають. Дане ПЗ спростить та покращить життя.

### **Особистий внесок студента**

Основним результатом є:

1. Творчий та новаторський підхід в створенні додатків
2. Створення власної логіки всередині програмного продукту

### **Структура та обсяг роботи даних**

Робота викладена на 60 сторінках друкованого тексту. Робота містить 2 таблиці, 31 рисунок та 1 додатки, обсягом 23 стор.

## РОЗДІЛ 1

### СТВОРЕННЯ КОНЦЕПЦІЇ ДОДАТКУ

#### 1.1 Огляд існуючих додатків аналогів

Перший додаток який трапився на шляху називається ”Подбери рецепт”.

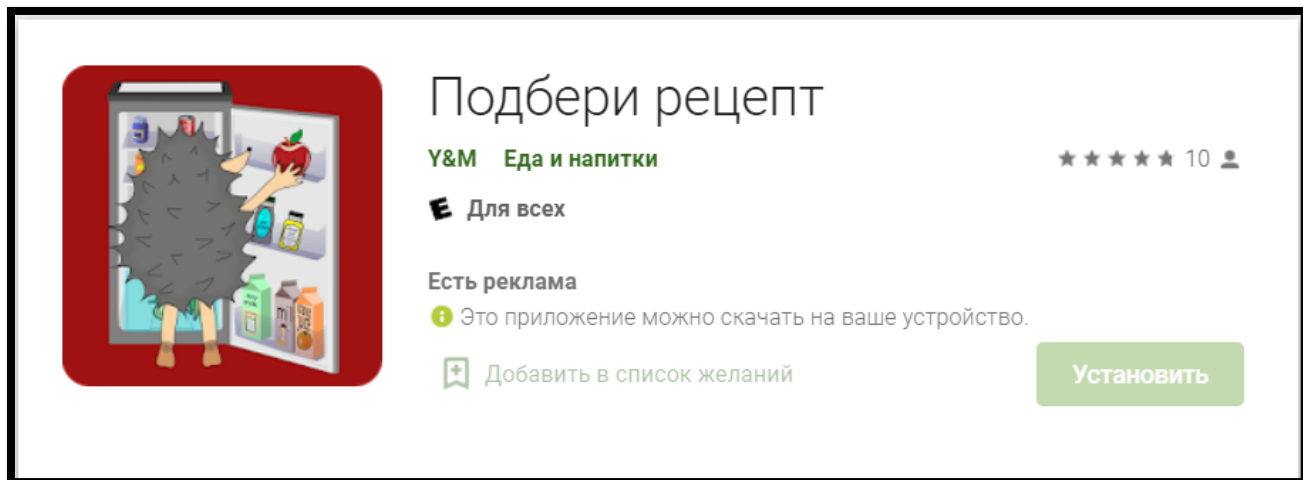


Рис. 1.1 Логотип першого додатку

Як можна бачити з рисунку 1.2 додаток користується попитом, а отже підходить для проведення аналізу.

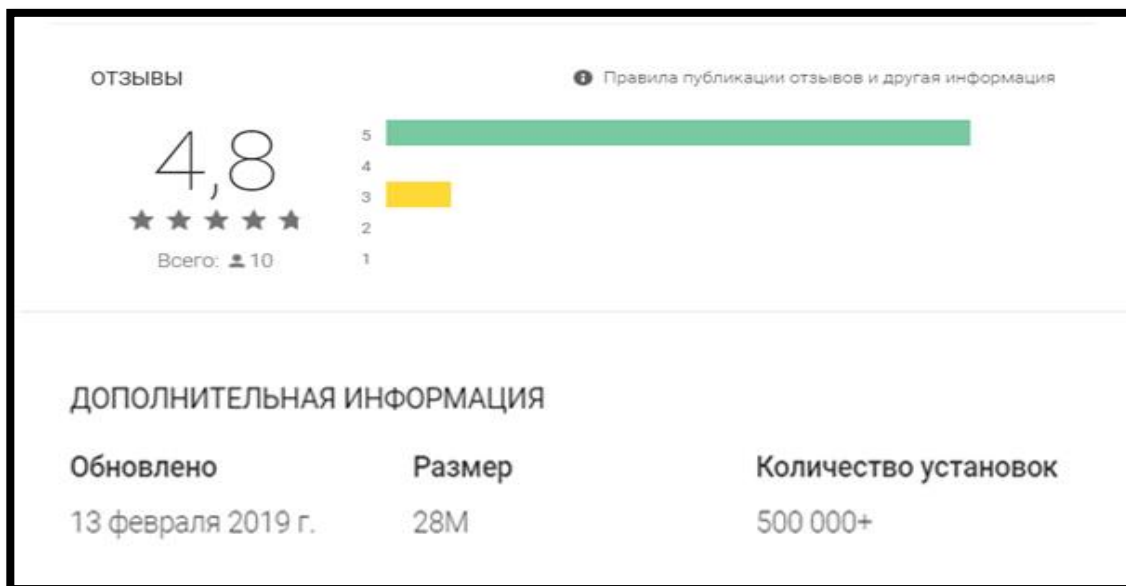


Рис. 1.2 Популярність першого додатку)

Завантаживши додаток перше що кинулось в очі це дизайн додатку(рисунок 1.3 та 1.4). Його тло виконано в коричневому та кремовому

тоні, через невіддале поєднання кольорів цей додаток виглядає дуже дешево, ніби його зробив на швидку руку школяр.

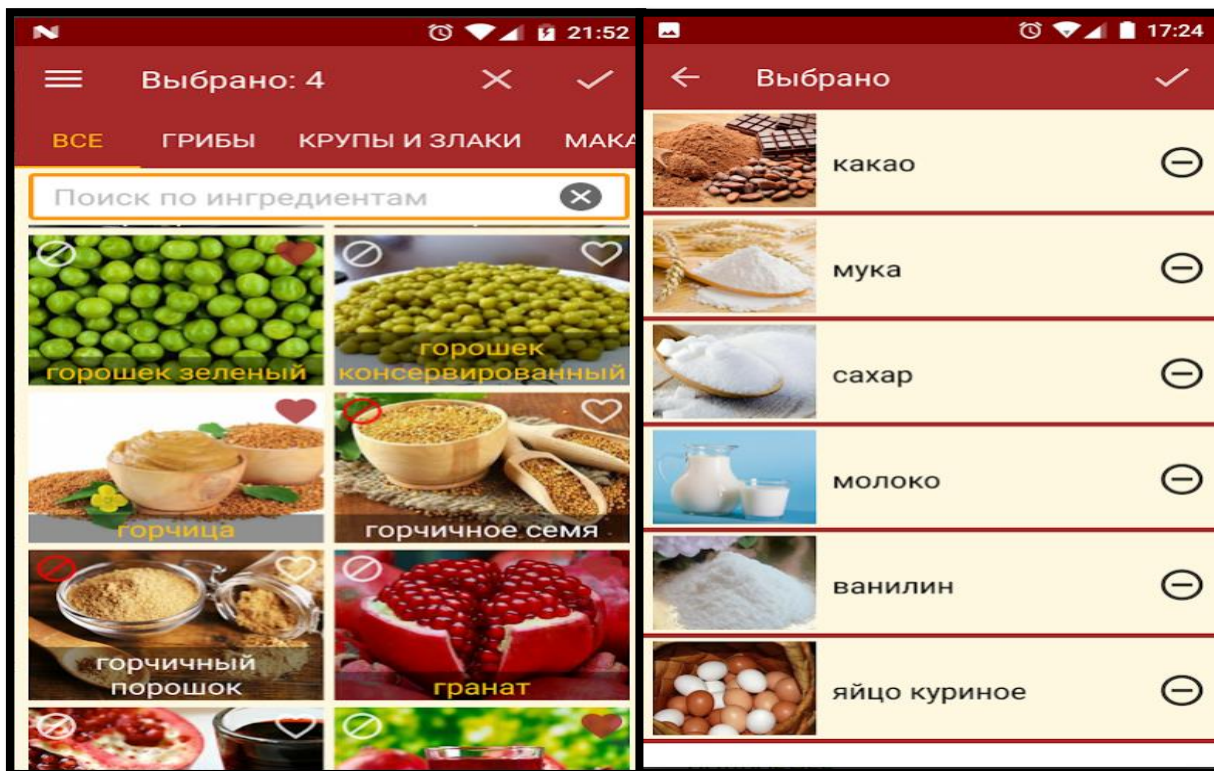


Рис. 1.3 а (ввід інгредієнтів)

Рис. 1.3 б (список вибраних)

Наступним мінусом є те що коли обираєш продукт який є в наявності не можна вибрати його кількість. Тобто якщо користувач ПЗ має 2 яйця, помідор, огірок, і він спробує знайти що він може собі приготувати однією з перших рекомендацій буде Голландський рулет для приготування якого потрібно як мінімум по 5 штук кожного з інгредієнтів відповідно користі з такого рецепту 0



Рис. 1.4 Рецепти

Хоч даний додаток мені не сподобався, в ньому все-таки є декілька плюсів. Першим плюсом функція додавання в корзину продуктів (Рис 1.5 а) яких не вистачає для приготування обраної страви. Це дуже зручно, коли ти нікуди не спішиш і хочеш приготувати для себе щось смачненьке. Особливо ця фішка припаде до душі тим людям, які, прийшовши в магазин, не можуть згадати що вони збирались купити, ця функція спростить їм життя та навчить деякій організованості.

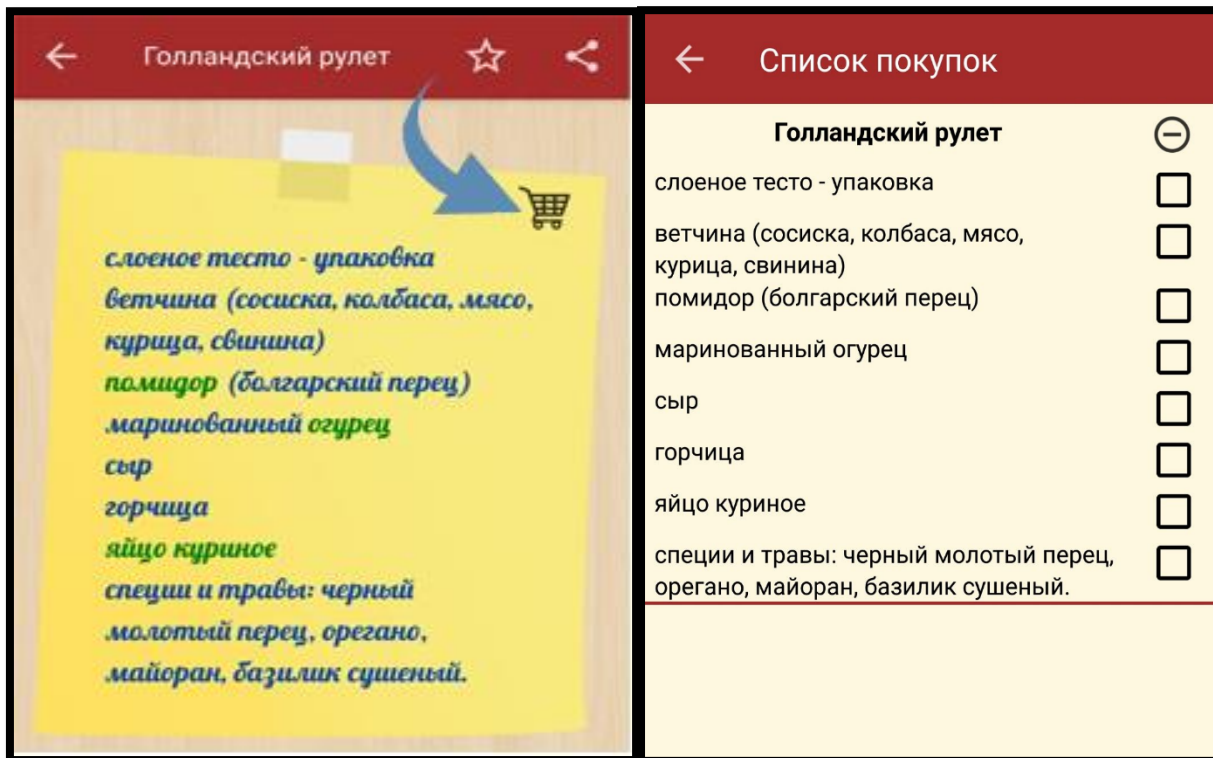


Рис. 1.5 а (іконка кошику)

Рис. 1.5 б (кошик)

Наступним було вирішено оглянути американський додаток який би мав схожий функціонал. Найпопулярнішим серед них виявився додаток під назвою “SuperCook”(Рис. 1.5 б).

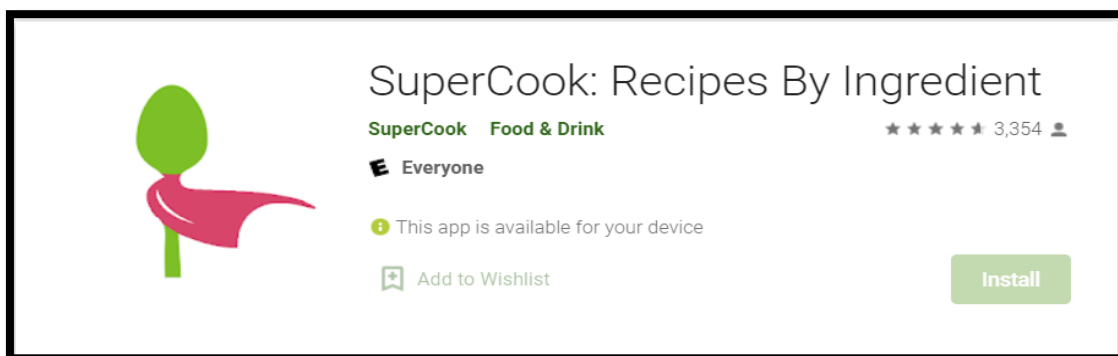


Рис. 1.6 Логотип другого додатку

Його дизайн значно відрізняється від додатків з схожим функціоналом. Кольорова гама підібрана дуже вдало, та виглядає по-сучасному, це додаток зовсім іншого рівня, ним хочеться покористуватись.

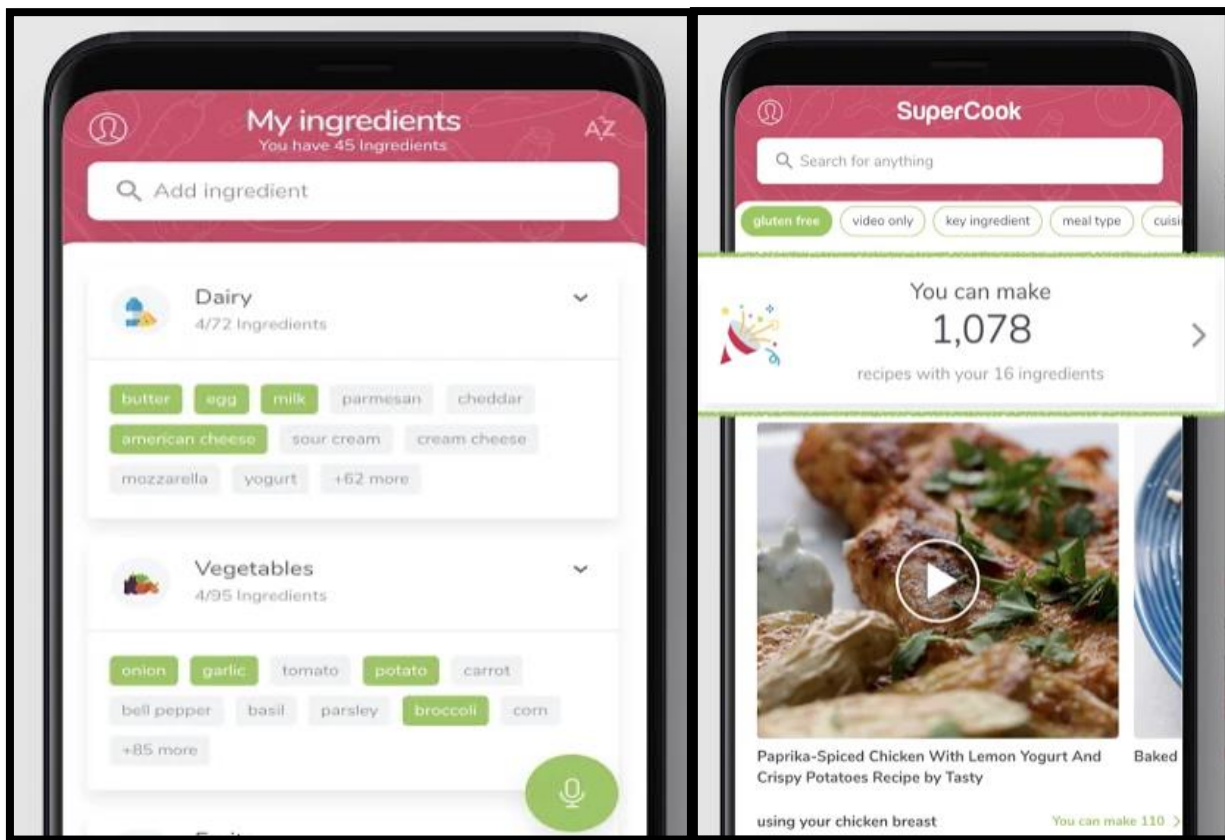


Рис. 1.7 а (інтерфейс)

Рис. 1.7 б (інтерфейс)

Трохи покористувавшись додатком та почитавши відгуки було виявлено такий же самий недолік як і в першому додатку – програма рекомендує страви які містять ті інгредієнти які користувач не має в наявності, підтвердженням цих слів є коментар зображений на рисунку 1.11

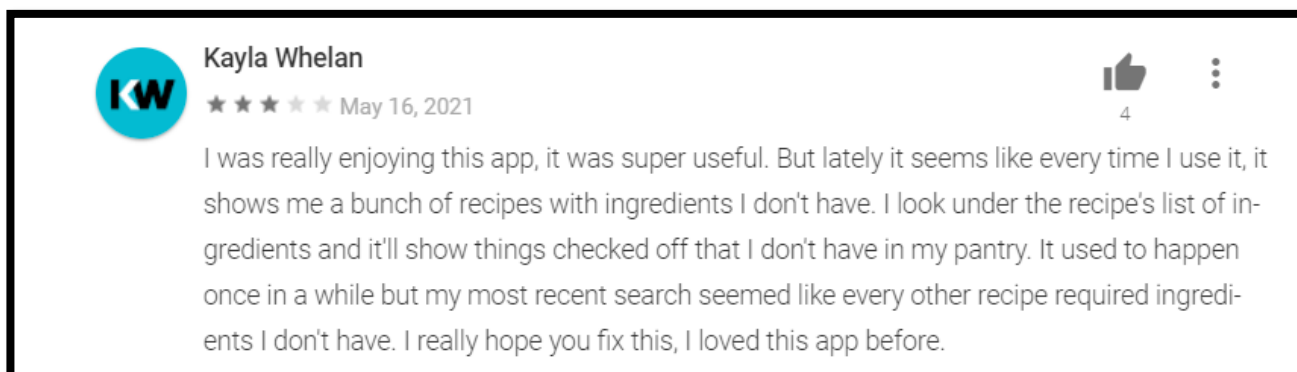


Рис. 1.8 відгук про другий додаток

Проаналізувавши схожі програмні рішення було створено скелет власного додатку, а саме:

- повинна бути сторінка, де користувач зможе задати бажану кількість калорій, щоб зменшити, збільшити або підтримати масу тіла;
- повинна бути сторінка, де користувач зможе швидко задати які продукти він має в наявності;
- повинна бути сторінка, де буде список рецептів відсортований за кількістю продуктів які є в користувача та вказаною ним калорійністю;
- сторінка з рецептом повинна містити фото готової страви, дані про його калорії та корисні речовини, список необхідних інгредієнтів та опис приготування;

## **1.2 Огляд використаних засобів розробки**

Даний проект був створений на платформі Android studio. Android Studio - це офіційне інтегроване середовище розробки (IDE) для розробки додатків для Android, засноване на IntelliJ IDEA.

Теоретично можна писати програми для Android у звичайному текстовому редакторі або командному рядку, але загальним підходом є використання інтегрованого середовища розробки. Цей інструмент інтегрує всі інструменти SDK та допомагає керувати ними простіше та зручніше. Android Studio є офіційною IDE для Android

Для створення додатку було обрано фреймворк Flutter та мову програмування Dart.

Flutter - це міжплатформенний набір інтерфейсів, призначений для повторного використання коду в таких операційних системах, як iOS та Android, а також дозволяє додаткам взаємодіяти безпосередньо з базовими службами платформи. Мета полягає в тому, щоб дозволити розробникам доставляти високопродуктивні програми, які почуваються природньо на різних платформах, враховуючи відмінності там, де вони існують, одночасно використовуючи якомога більше коду.

Під час розробки програми Flutter працюють у віртуальній машині, яка пропонує гаряче перезавантаження змін без необхідності повної перекомпіляції.

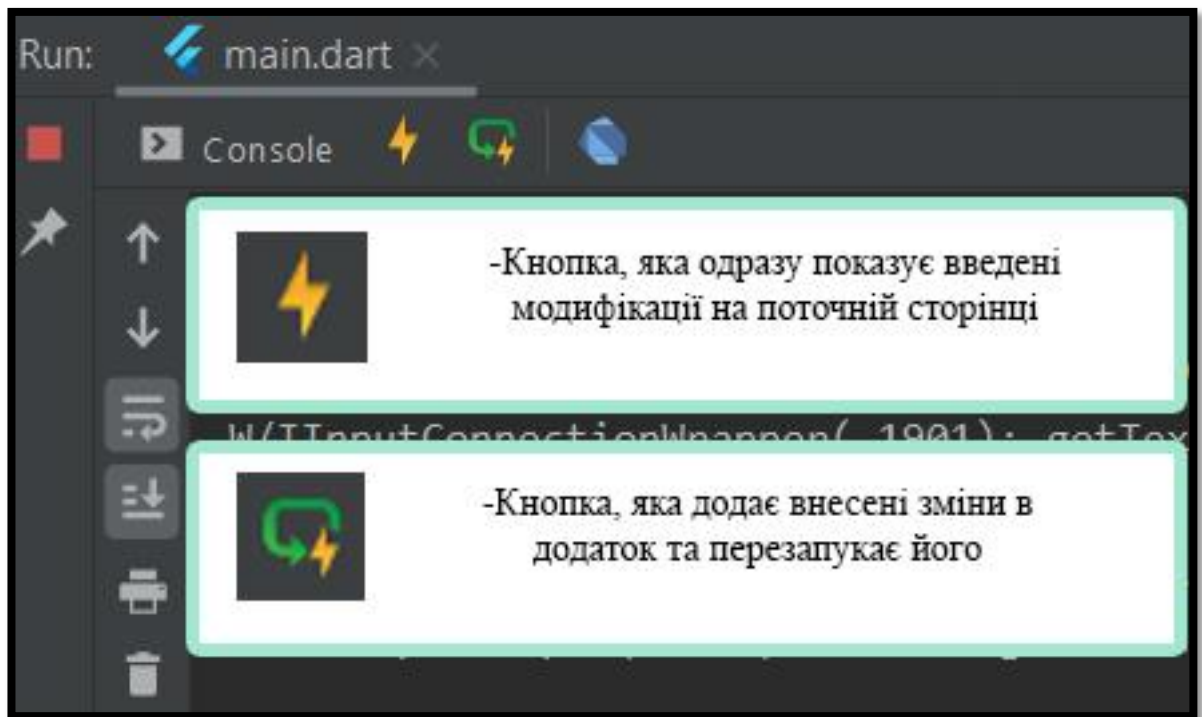


Рис. 1.12 Детальна інформація

Для випуску програми Flutter компілюються безпосередньо до машинного коду, будь то інструкції Intel x64 чи ARM, або до JavaScript, якщо націлено на Інтернет. Фреймворк має відкритий код, має дозвільну ліцензію BSD і має процвітаючу екосистему сторонніх пакетів, які доповнюють основні функціональні можливості бібліотеки.

Dart - це оптимізована клієнтом мова для розробки швидких додатків на будь-якій платформі. Його мета - запропонувати найпродуктивнішу мову програмування для розробки мультиплатформ у поєднанні з гнучкою платформою виконання для фреймворків програм.

## РОЗДІЛ 2 СТВОРЕННЯ ДИЗАЙНУ

### 2.1 Дизайн додатку

При розробці розробці даного програмного забезпечення було обрано кілька варіантів дизайну.



Рис. 2.1 Тестові палітри кольорів

Усі з вищенаведених варіантів є красивими та гармонійно виглядають, проте з урахуванням особливостей даного програмного продукту, та його орієнтованості на оздоровлення способу життя людини.

У 1949 році М. Люшер вперше опублікував свій тест, в якому була присутня таблиця «зеленого кольору». Згідно з уявленнями швейцарського психотерапевта, цей колір вирівнює пульс і дихання, нормалізує тиск, а також асоціюється з стабільністю, самоповагою.

Зелений часто присутній в рекламних роликах про органку і продуктами харчування, оскільки ми сприймаємо його, як символ свіжості, натуральності і близькості до природи.

Враховуючи вищенаведені результати проведених досліджень було вирішено обрати гаму зелених кольорів як основну палітру даного ПЗ оскільки даній колір асоціюється з природою, здоров'ям та корисними продуктами харчування.

## 2.2 Фото для додатку

Також важливою частиною дизайну є оформлення зображень продуктів харчування як інгредієнтів страв, так і готових страв. Варто зазначити що для розробки даного програмного забезпечення було сфотографовано перелік інгредієнтів що найбільш часто зустрічаються в холодильнику середньостатистичного студена. Також було сфотографовано страви що можна отримати завдяки комбінації вибраних інгредієнтів з наданого переліку. На нижченаведеному рисунку відображено приклад зображення продукту харчування та вже приготованої страви.

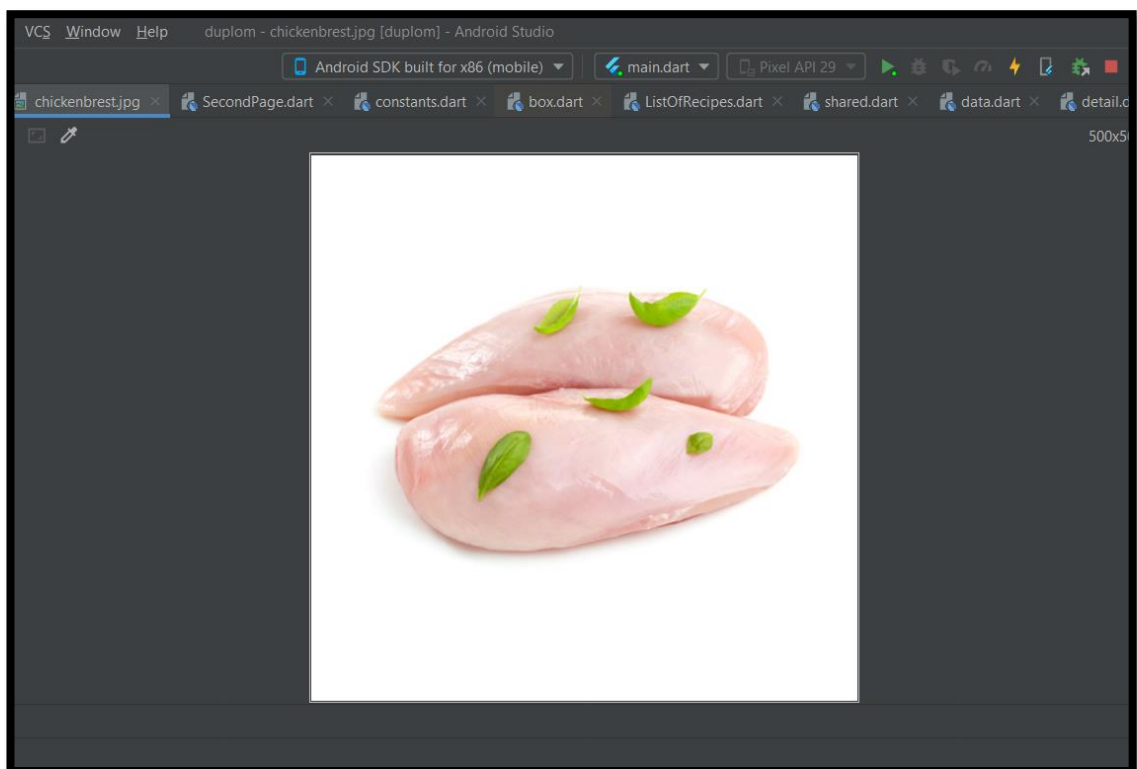


Рис. 2.2 фото інгредієнта

На рисунку 2.2 відображено інгредієнт - курячу грудку яка потім обрізається по формі кнопки за допомогою програмного коду. Аналогічні дії проведені з усіма інгредієнтами в програмі. Також варто зазначити що усі

продукти харчування сфотографовані на білому фоні за для спрощення сприйняття продукту користувачем.

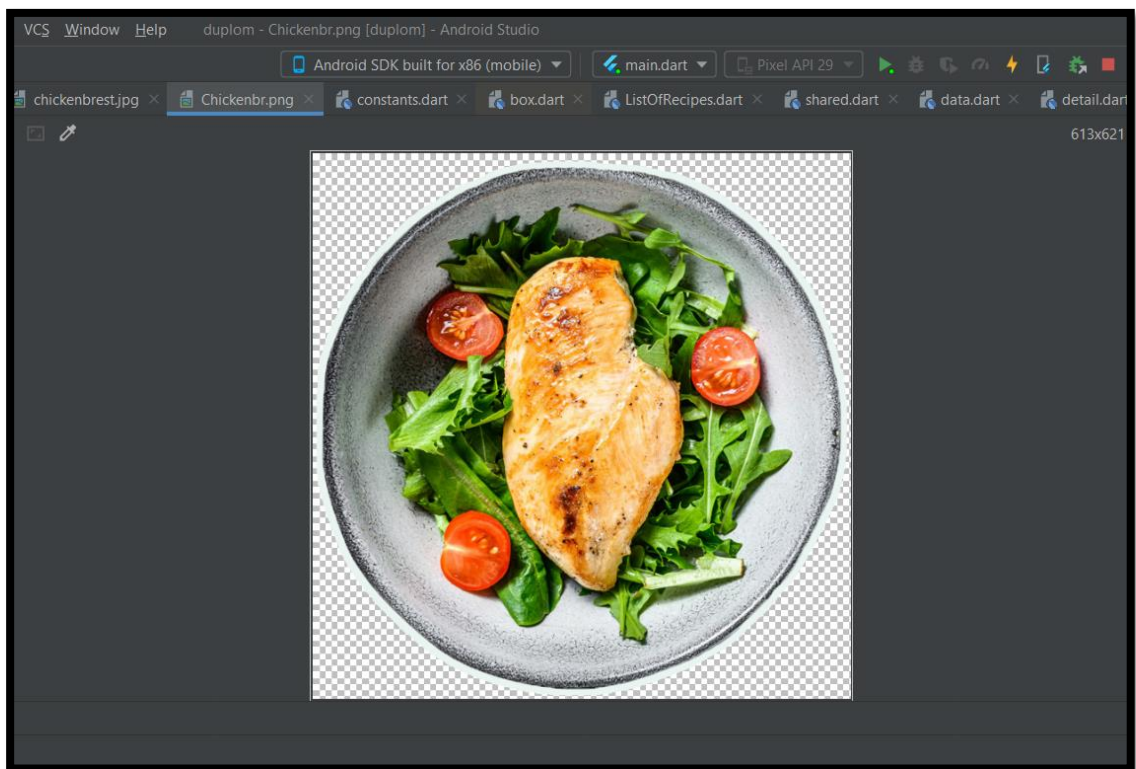


Рис. 2.3 фото страви

На вищенаведеному рисунку 2.3 відображено фотографію вже приготовленої страви як включає в себе інгредієнт з рисунку 2.2. Дане фото зроблено в форматі Png задля подальшої вдалої інтеграції зображення у сторінку що буде відображатись. Всі фотоматеріали були поміщені в папку “assets”. У додатку створеному на Flutter є спеціальний файл який містить в собі інформацію про фотографії, шрифти які можна використати в програмі, називається він “pubspec.yaml”

```

# To add assets to your application, add an assets section, like this:
assets:
  - assets/egg.jpg
  - assets/onion.jpg
  - assets/garlic.jpg
  - assets/cucumber.jpg
  - assets/chickenleg.jpg
  - assets/chickenbreast.jpg
  - assets/tomato.jpg
  - assets/macaroni.jpg
  - assets/greenbean.jpg
  - assets/spaghetti.jpg
  - assets/

```

Рис. 2.4 шляхи для фото

Обов'язковою процедурою є додавання папок які містять в собі необхідні файли (фото, шрифти) та шляхів до цих файлів. На вищенаведеному рисунку 2.4 зображено побудовані шляхи для фото в папці “assets”.

```

# To add custom fonts to your application, add a fonts section here,
# in this "flutter" section. Each entry in this list should have a
# "family" key with the font family name, and a "fonts" key with a
# list giving the asset and other descriptors for the font. For
fonts:
  - family: Chewy
    fonts:
      - asset: fonts/Chewy-Regular.ttf
  - family: ChelaOne
    fonts:
      - asset: fonts/ChelaOne-Regular.ttf

```

Рис 2.5 шляхи для шрифтів

Аналогічно, на вищенаведеному рисунку 2.5 відображено побудовані шляхи для фото в папці “assets”.

## РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ

### 3.1 Архітектура додатку

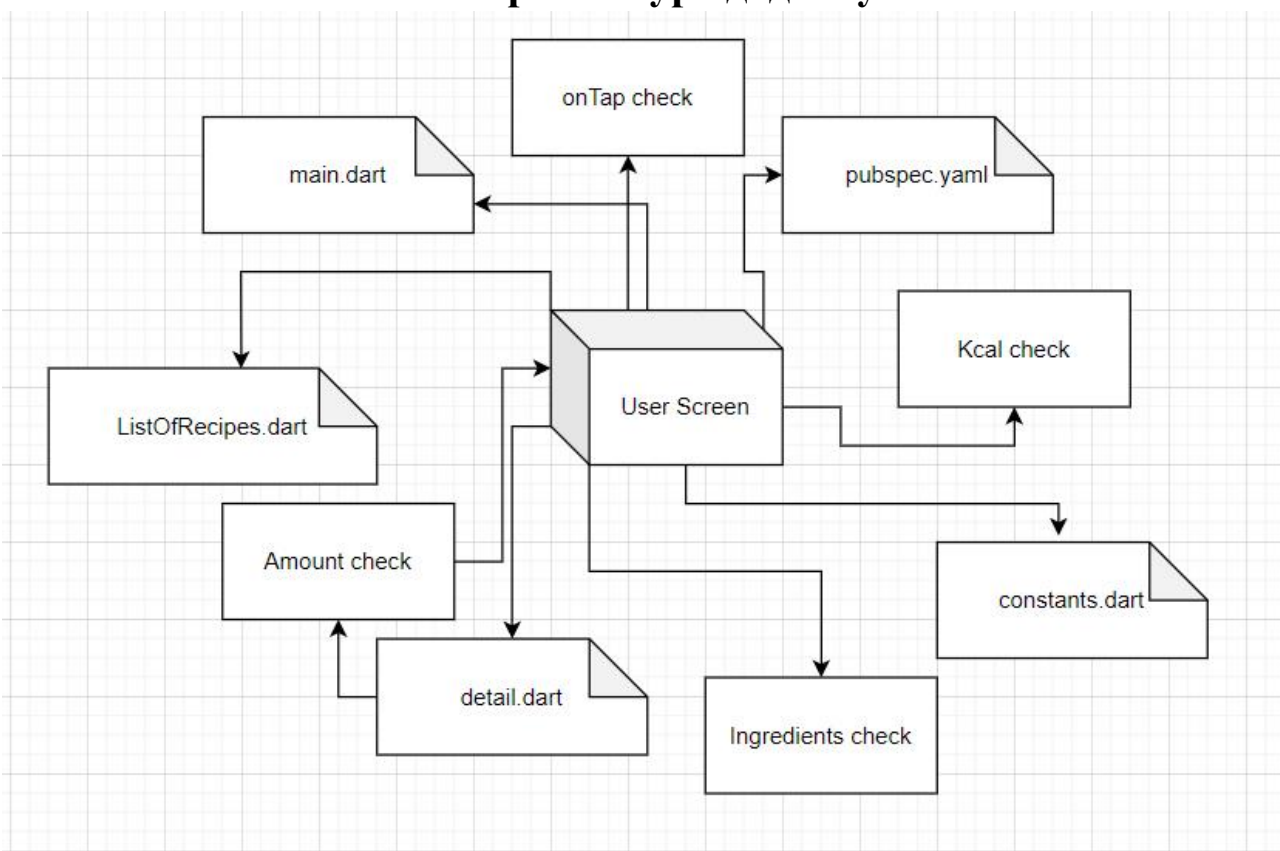


Рис. 3 Діаграма архітектури додатку

На рисунку 3 можна побачити UML діаграму створеного в ході цієї дипломної роботи додатку. На ній чітко відображено архітектуру розробленого ПЗ. Гарна архітектура це важлива частина в створенні мобільного додатку, вона означає вигоду – простота і ефективність. Програму з такою архітектурою легше змінювати, тестувати і налагоджувати. Додаток є ефективним оскільки виконує поставлені завдання і виконує функції в будь-яких умовах. Розроблений додаток є гнучким, можна змінити один елемент і це не стане фатальним для інших складових. В додаток можна додавати скільки завгодно функцій, якщо це буде потрібно. Додаток легко тестується, а значить, зменшується кількість помилок і збільшується його надійність. Елементи і структуру можна використовувати в інших проектах. Код в додатку написаний по всім стандартам і правилам, так що код зрозуміти зможе більша кількість людей.

### 3.2 Перша сторінка

На першій сторінці було вирішено створити поле де користувач може ввести скільки калорій він би хотів отримати від прийому їжі.

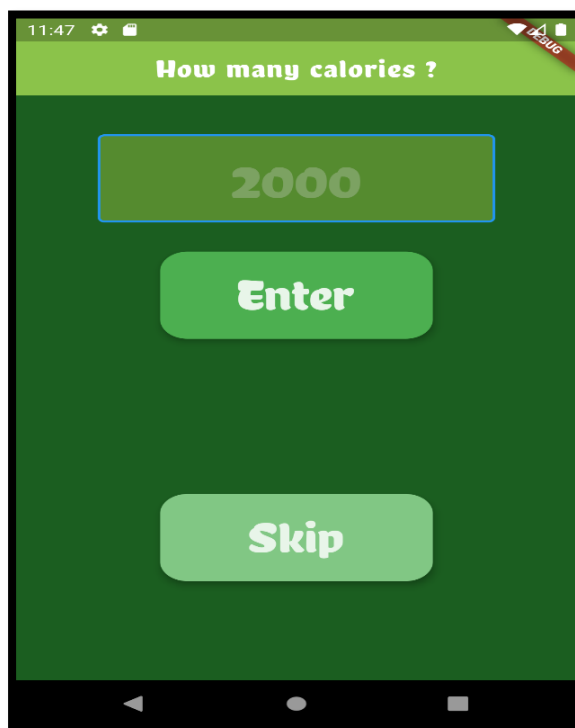


Рис 3.1 перша сторінка додатку

На рисунку 3.1 можна побачити як виглядає перша сторінка розробленого програмного забезпечення. З самого верху додатку розміщене питання – “How many calories”, що в перекладі означає – “Скільки калорій вам потрібно”. Далі під написом розміщене поле для вводу кількості калорій близько до якої повинна мстити страва, або сума декількох страв( салату з гарніром наприклад), коли студент ввів бажану кількість калорій, щоб перейти на наступну сторінку йому потрібно натиснути на кнопку “Enter”, і введені ним дані збержуться, якщо ж йому без різниці яку кількість калорій міститиме страва він може натиснути кнопку “Skip” і тоді параметр з калоріями не буде враховуватись при виводі рецептів на екран. Пройдемося поі всім елементам які є на даному рисунку.

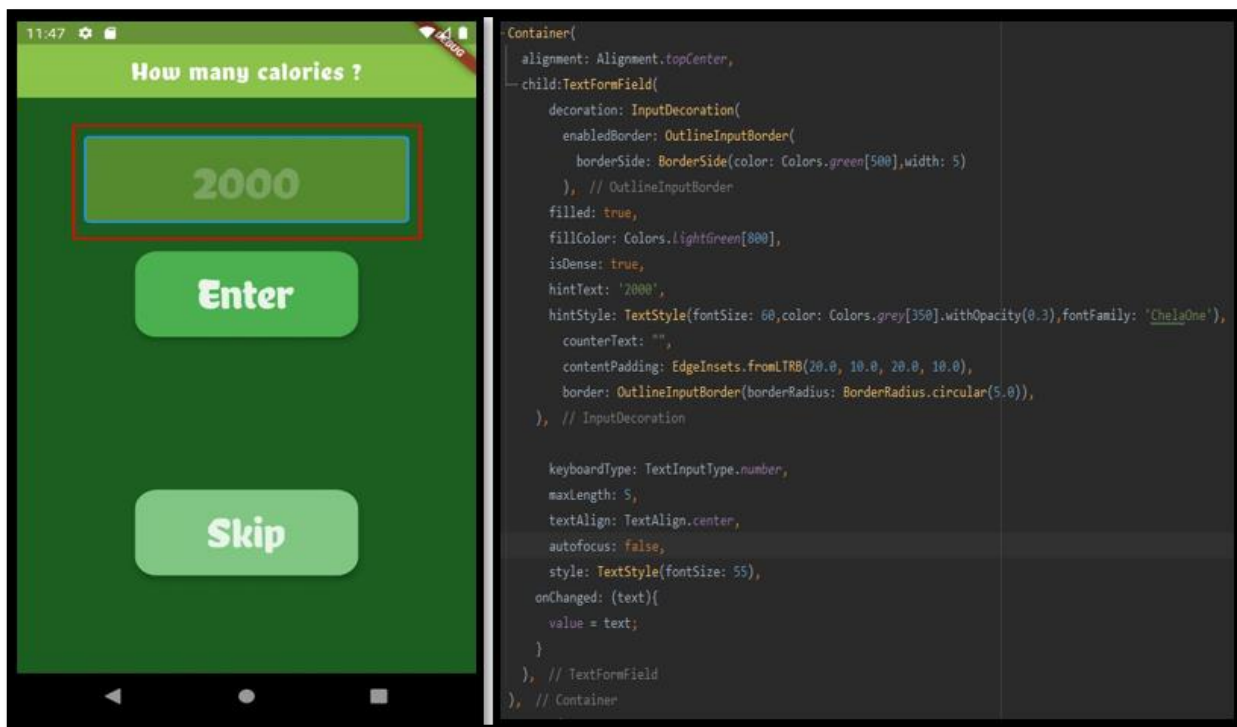


Рис. 3.2 а (поле вводу калорій)

Рис. 3.2 б (реалізація поля)

Першим елементом є поле для вводу кількості калорій яку повинна містити їжа яку користувач в подальшому приготує. З власного досвіду я знаю що харчування є важливою частиною здорового способу життя, особливо це важливо для студентів які займаються спортом і хочуть набрати м'язової маси, адже неможливо добитись прогресу в побудові тіла якщо в його основу ти закладаєш фаст-фуд, бутерброди та вермішель швидкого приготування. Дане поле є одним з головних в додатку, адже саме завдяки цьому полю користувач може слідкувати за кількістю калорій які він споживає, і виставляти її відповідно до своїх цілей. Якщо він хоче набрати маси необхідно споживати більше калорій ніж зазвичай, якщо студент бажає схуднути або “підсушитись” потрібно їсти менше, ну і також можна підтримувати масу тіла якщо повністю нею задоволений. В даному додатку усі рецепти направлені на покращення раціону студента та не включають ніяких шкідливих інгредієнтів.

Кількість калорій для збереження поточного ваги: 2341 калорій / день  
 Щоб набрати 5 кілограмів м'язової маси за 2 місяці вам потрібно збільшити споживання калорій до: 2976 калорій / день

**Калькулятор набору ваги, новий розрахунок.**

Введіть ваші параметри для розрахунку калорій для набору маси або схуднення

Підлога:  Чоловік  жіночий

вік:  років

Зріст:   см  футів / дюймів

вага:   кг  фунтів

Відсоток жиру:  %

Рівень фізичних навантажень:  ▼

Ви хочете:  ▼ на   кг  фунтів

За термін:   ▼

Рис. 3.3 (визначення необхідної кількості калорій)

Я скористався сайтом “Калькулятор набору ваги” для того щоб дізнатись скільки мені потрібно споживати калорій в день для збереження моєї ваги, а також скільки потрібно щоб за 2 місяці тренувань набрати 5 кілограм м’язової маси (Рис. 3.3). Тепер за допомогою створеного додатку я зможу отримати страви з потрібною мені кількістю калорій, і не витратити зайвий час та гроші на консультації тренерів чи пошуку інформації в інтернеті.

Для того щоб створити поле з вводом калорій я використав багато різних віджетів та команд. Найскладнішим в його створенні було створити текст з прикладом підказкою “2000”, але мені вдалось добитись бажаного ефекту за допомогою команди `color: Colors.grey[350].withOpacity(0.3)` (Рис 3.3)

Наступний елемент – кнопка Enter (виділений на рисунку 3.4 а) створенна для того, щоб підтвердити введену користувачем інформацію та рухатись далі. В випадку, якщо користувач натисне цю кнопку, але поле з калоріями пuste або менше чи рівне нулю, нічого не відбудеться. Якщо данні введені вірно то користувач зможе перейти на наступну сторінку.

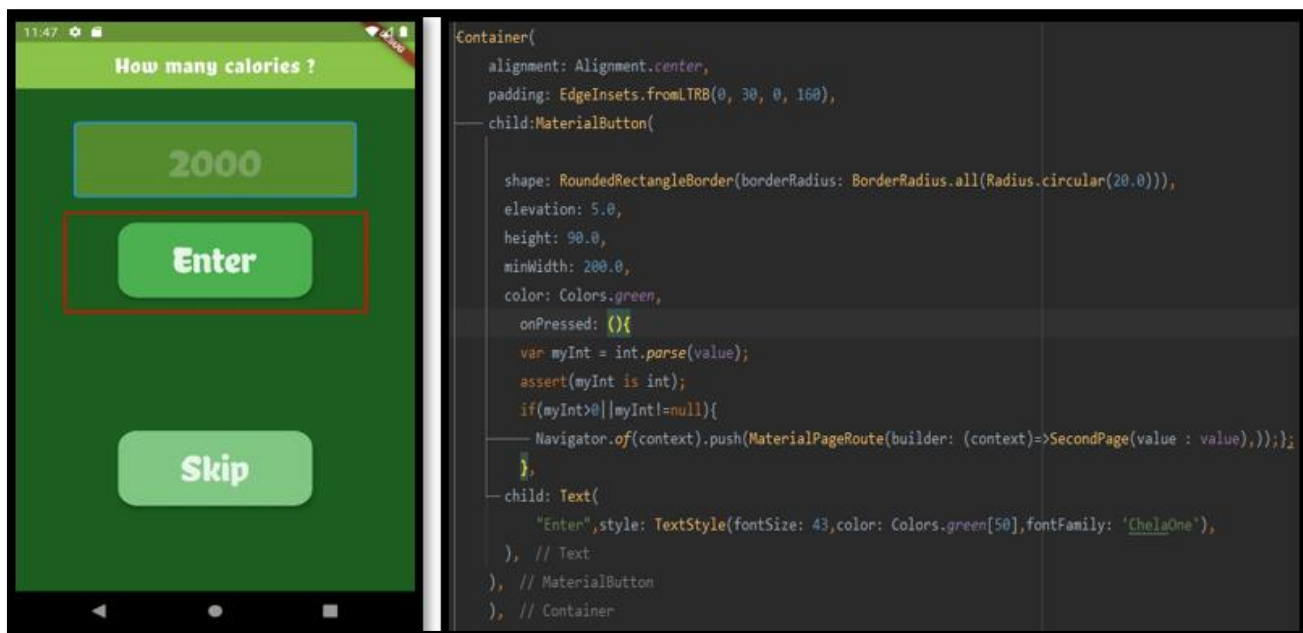


Рис. 3.4 а (кнопка підтвердження) Рис. 3.4 б (реалізація кнопки)

Щоб кнопка працювала в неї є спеціальний метод `onPressed()`, саме в цьому методі я реалізував перевірку даних та перехід на наступну сторінку. Дана команда виглядає так:

```
if(myInt>0||myInt!=null){
```

```
  Navigator.of(context).push(MaterialPageRoute(builder:
  (context)=>SecondPage(value:value),));};},
```

де змінні `myInt` та `value` містять в собі інформацію про калорії введені і користувачем.

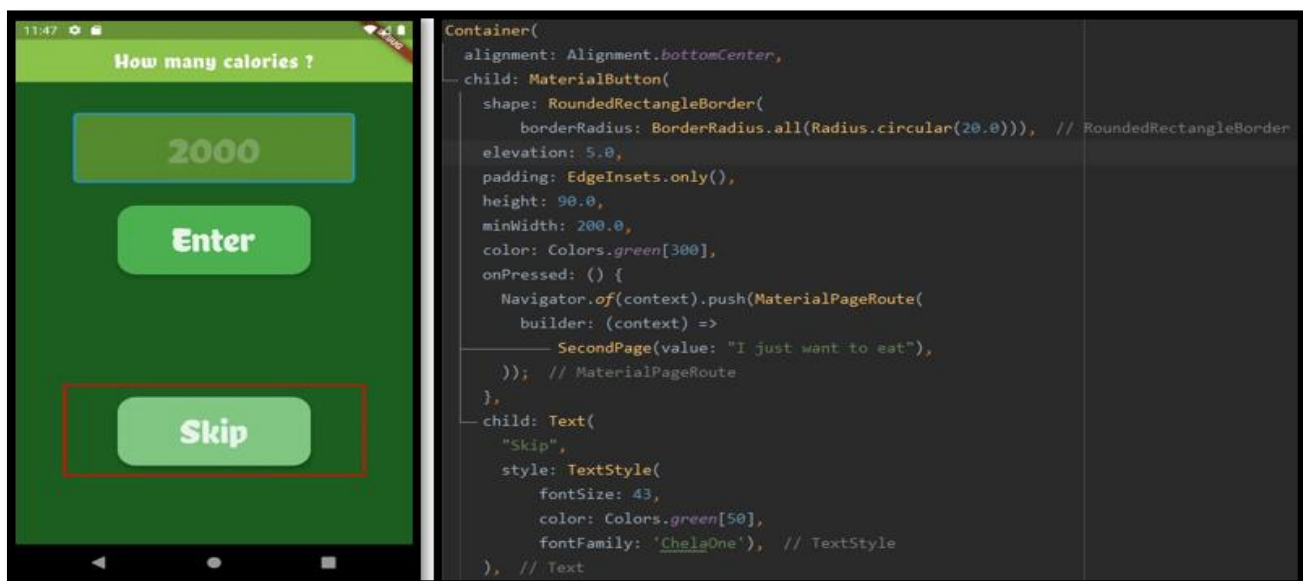


Рис 3.5 а (кнопка пропуску)

Рис 3.5 б (реалізація кнопки)

Останнім елементом на першому екрані є кнопка Skip (Рис. 3.5 а) яка дозволяє пропустити частину вводу калорій та перейти одразу до наступної. Ця кнопка створена для студентів яким байдуже скільки калорій їм потрібно в даний момент, і вони хочуть просто приготувати щось поїсти, на наступній сторінці на місці даних про калорії та і буде писати – “ I just want to eat”.

### 3.3 Друга сторінка

Під час планування додатку було вирішено що на другій сторінці будуть розміщена інформація про обрану користувачем кількість калорій, та кнопки з зображеннями продуктів які найчастіше зустрічаються в холодильнику середньостатистичного студента, тобто ніяких креветок, раків та ікри в даному додатку немає.

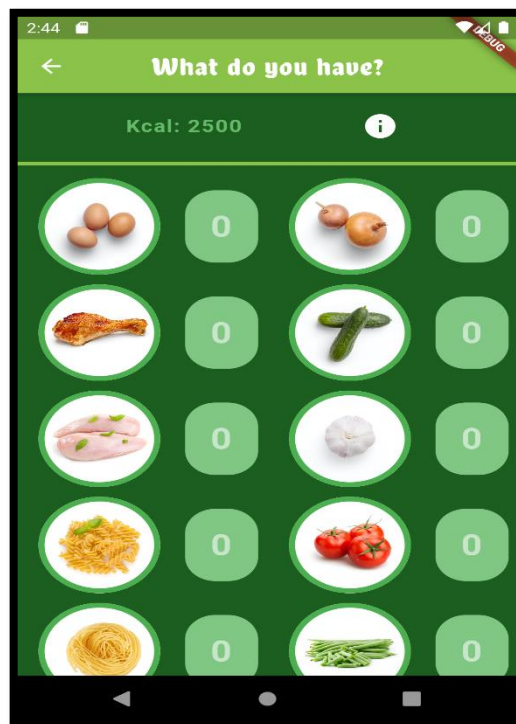


Рис. 3.6 сторінка вводу інгредієнтів

На рисунку 3.6 скріншот другої сторінки розробленого додатку. М'ясо та крупи та вироби розміщені з лівої сторони екрану, а овочі з правої. Також деякі інгредієнти розміщені таким чином щоб в поєднанні з сусідом вони утворювали страву, наприклад куряча грудка та часник, з них може вийти смачнющий та

корисний стейк, а з макаронів та помідорів чудова паста. При розробці UX дизайну найбільше уваги були сконцентровано саме на другому екрані. На власному досвіді я переконався що в додатках аналогах найбільше часу займає занесення інгредієнтів які є наявності в додаток. Як можна бачити з рисунку 3.9 дизайн вийшов досить ергономічним, в ньому немає ніяких зайвих функцій а лише необхідне. Також в ході розробки було вирішено не підписувати кнопки. Студенти постійно поглинають масу інформації, і сидіти вчитувати інгредієнти після виконання лабораторних чи домашніх завдань для них може бути напружно, а також по картинці мозок скоріше розуміє що він бачить за їжу ніж по прочитаному слову. Для того щоб користувачу було чітко ясно що за продукт на кнопці було вирішено всі продукти сфотографувати на білому фоні. Пройдемося по всім елементам які є на даному екрані.

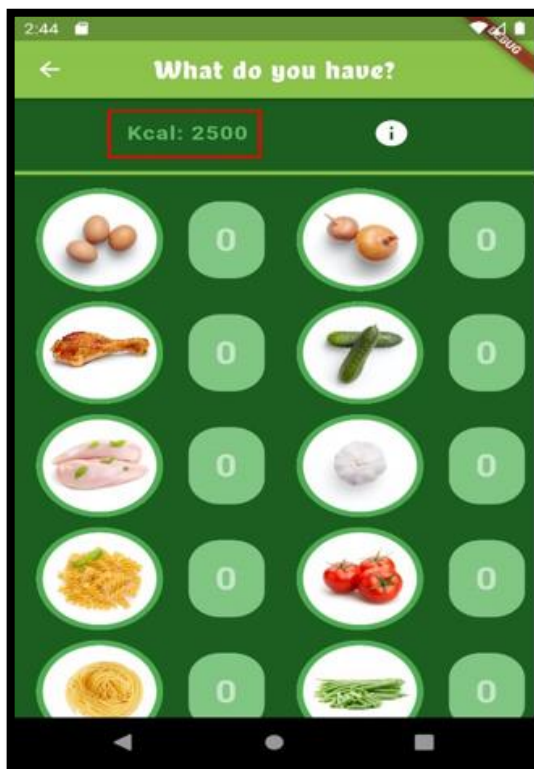


Рис. 3.7 а (поле калорій)



Рис. 3.7 б (реалізація поля калорій)

Першим видимим елементом є індикатор калорій які користувач ввів на першій сторінці. Він створений для того щоб студент завжди бачив ту кількість калорій яку він обрав. Якщо користувач захоче повернутись назад та змінити

вказану ним кількість калорій йому потрібно натиснути стрілочку направлену в ліво яка знаходиться вгорі екрану, або натиснути кнопку дії “назад”, яка є на всіх телефонах на платформі Android.

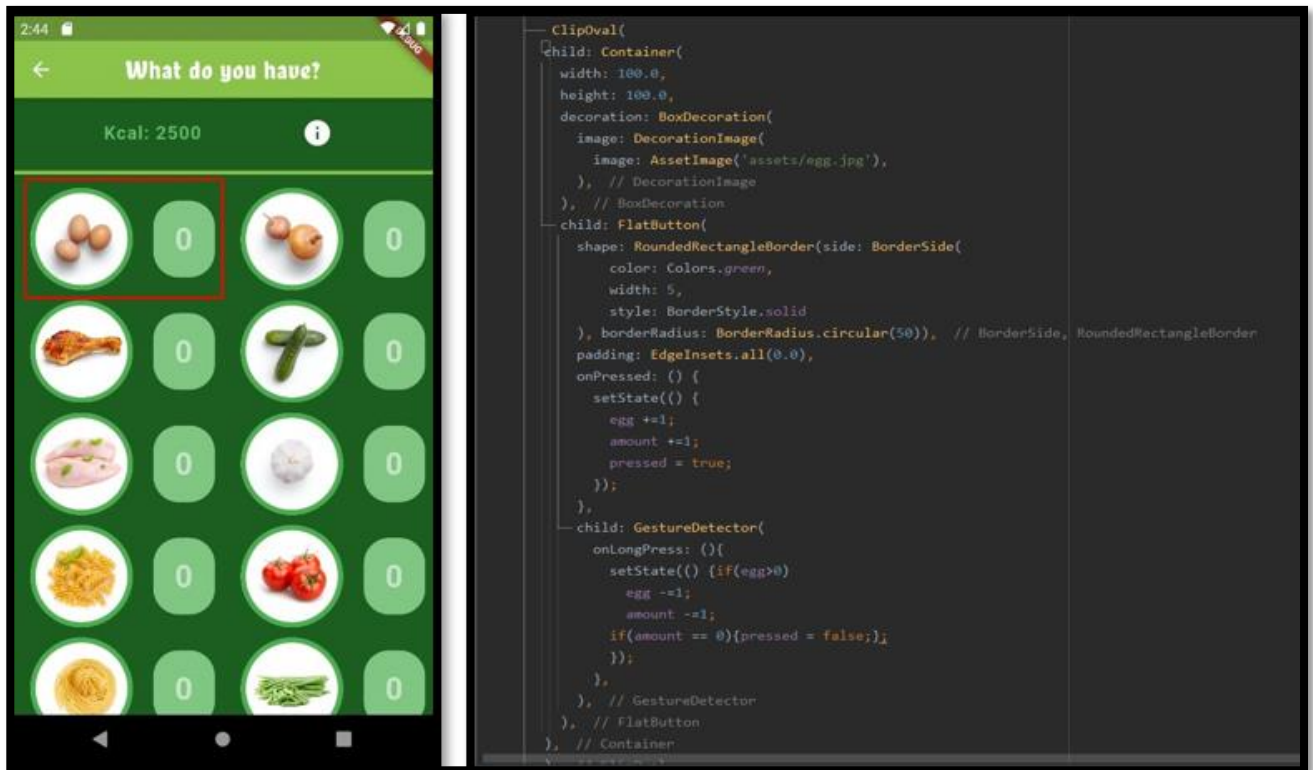


Рис. 3.8 а (кнопка з інгредієнтом) Рис. 3.8 б (реалізація кнопки та логіки)

Наступним елементом є кнопки, дизайн яких є авторською ідеєю. Щоб надати їм круглої форми було використано віджет `ClipOval()` та `RoundedRectangleBorder()`. Для того щоб керувати розміром кнопки, а саме їх висотою та шириною було використано команди `width` та `height`. Помістив фото в кнопку за допомогою команди `image: AssetImage('assets/egg.jpg')`, де 'assets/egg.jpg' шлях до бажаної картинки. Для кожного продукту була створена своя змінна яка зберігає в собі його кількість.

Щоб додати одиницю продукту потрібно натиснути на кнопку, цей спосіб добре підходить в даному випадку тому що можна швидко внести інгредієнти в додаток.

Щоб відняти одиницю продукту потрібно затримати кнопку на пару мілі секунд. В більшості схожих додатків зробили б ще дві кнопки з + та -, але було

вирішено відмовитись від даної реалізації ідеї. Даний спосіб було обрано щоб не накопичувати велику кількість кнопок на екрані, та щоб це було легко, швидко, та інтуїтивно.

При плануванні було вирішено щоб кнопка, яка буде давати можливість перейти на наступну сторінку та буде переносити інформацію з внесеними користувачем продуктами на наступний скрін, була невидима доти, доки не вибраний хоча б один інгредієнт зі списку.

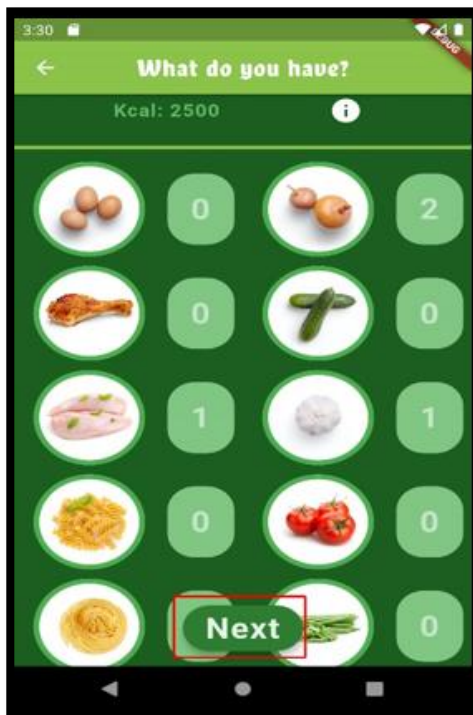


Рис. 3.9 а (кнопка далі)

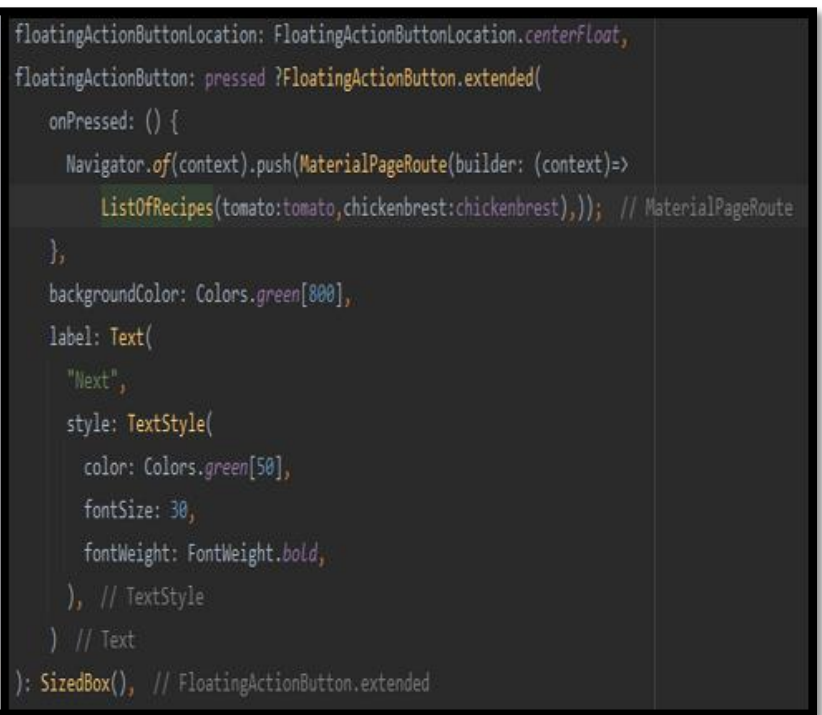


Рис. 3.9 б (реалізація кнопки далі)

Як можна бачити на рисунку 3.9 а кнопка для переходу на наступну сторінку є, а на рисунку 3.6 немає. Різниця в тому що на рисунку 3.9 а вже обрано декілька продуктів тому і кнопка з'явилась.

Щоб реалізувати дану можливість було добавлено дві змінних – ціле число amount (кількість) як з початку рівна нулю яка є лічильником всіх інгредієнтів та булеву pressed (натиснута) як з самого початку є false (не вірна).

В метод onPressed() кожної кнопки окрім змінної яка дорівнює кількості інгредієнту було добавлено змінну amount яка при натисканні зростає на одиницю та pressed яка стає "True".

Далі в віджет `GestureDetector()`, який використовується для зчитування рухів по екрану та надання їм функції. В даній роботі використовується метод `onLongPress()`, саме він забезпечує можливість зчитування довгого натиску на об'єкт. В цьому методі було створено декілька перевірок, перша – це перевірка на кількість інгредієнтів, щоб кількість інгредієнту не могла бути менше нуля. Виглядає вона так: `if(egg>0)`. Якщо ця умова правдива, тобто кількість продукту мінімум одиниця, то при затриманні кнопки з внесеного списку віднімається ця одиниця, а також вона віднімається від загальної кількості, тобто `amount`.

Друга перевірка потрібна для того щоб, якщо після додавань та віднімань інгредієнтів їх кількість стала рівна нулю, кнопка з переходом на наступну сторінку зникала. Дана перевірка була реалізована таким чином:

```
if(amount == 0){pressed = false;};
```

а якщо `pressed` не є правдивим кнопка не відображається на екрані.

Також, як можна бачити на рисунку 3.12 було створено поле для відображення кількості добавлених в додаток інгредієнтів.

```

|— SizedBox(width: 5),
Material(color: Colors.green[900],
|— child: ClipRRect(
|   borderRadius: BorderRadius.only(
|     topLeft: Radius.circular(25.0),
|     topRight: Radius.circular(25.0),
|     bottomLeft: Radius.circular(25.0),
|     bottomRight: Radius.circular(25.0),
|   ), // BorderRadius.only
|   child: Align(
|     alignment: Alignment.bottomRight,
|     heightFactor: 1,
|     widthFactor: 1,
|     child: Container(
|       color: Colors.green[300],
|       width: 60,
|       height: 75,
|       child: Center(
|         child: Text(
|           '$egg',
|           style: TextStyle(
|             color: Colors.green[100],
|             letterSpacing: 1.0,
|             fontSize: 30,
|             fontWeight: FontWeight.bold, // TextStyle
|           ), // Text
|         ), // Center
|       ), // Container
|     ), // Align
|   ), // ClipRRect
), // Material

```

Рис. 3.10 (реалізація поля виводу інформації)

Досягти такого вигляду вдалося завдяки ряду команд вказаних на рисунку 3.10 команда `color: Colors.green[300]` необхідна для встановлення кольору об'єкту,

`borderRadius: BorderRadius` для формування бажаної форми. Щоб передавати інформацію, яку містить змінна з кількістю добавленого продукту в віджеті, який потрібен для відображення тексту – `Text()`, використовується посилання на змінну з даними, виглядає воно таким чином: `'$egg'`.

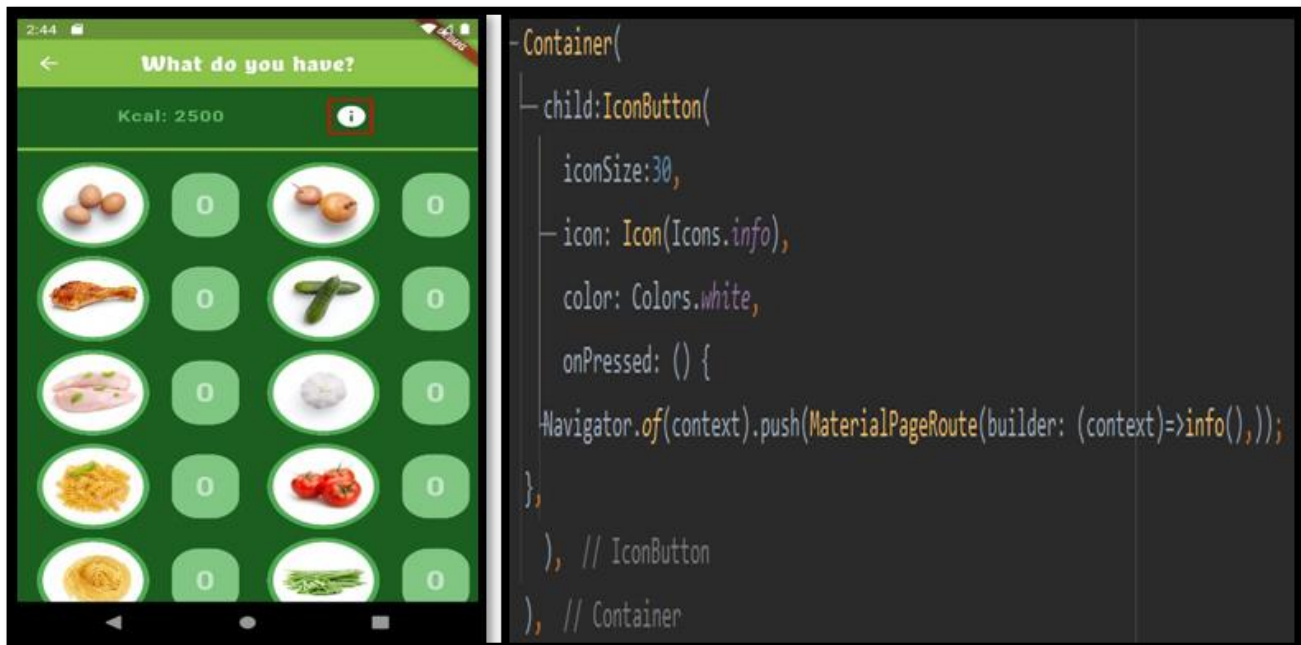


Рис. 3.11 а (Інформація)      Рис. 3.11 б (реалізація іконки з переходом)

Останнім елементом про який ще не було сказано є іконка, зображена на рисунку 3.11 а, з латинською буквою “i” тобто “information”. При натисканні для неї відкривається вказаний нижче скрін (Рис 3.12) з підказками про те, як додати продукт до списку в додатку, а також як його відняти якщо була додана лишня одиниця якої немає в наявності чи просто була введена лишня одиниця продукту.

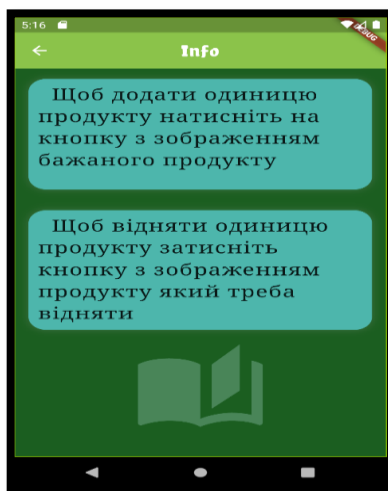


Рис. 3.12(сторінка з інформацією)

### 3.4 Третя сторінка

Після того як усі продукти було введено в додаток і натиснувши кнопку “Next”, з’являється третя сторінка додатку.

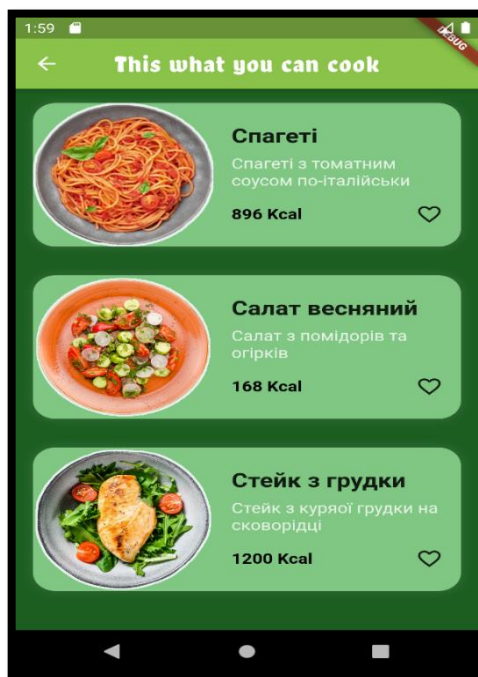


Рис 3.13 (сторінка з рецептами)

На цій сторінці, зображеній на рисунку 3.13, розміщені ті рецепти які підходять користувачу по калоріям та складаються з тих інгредієнтів які він вказав що має в наявності.

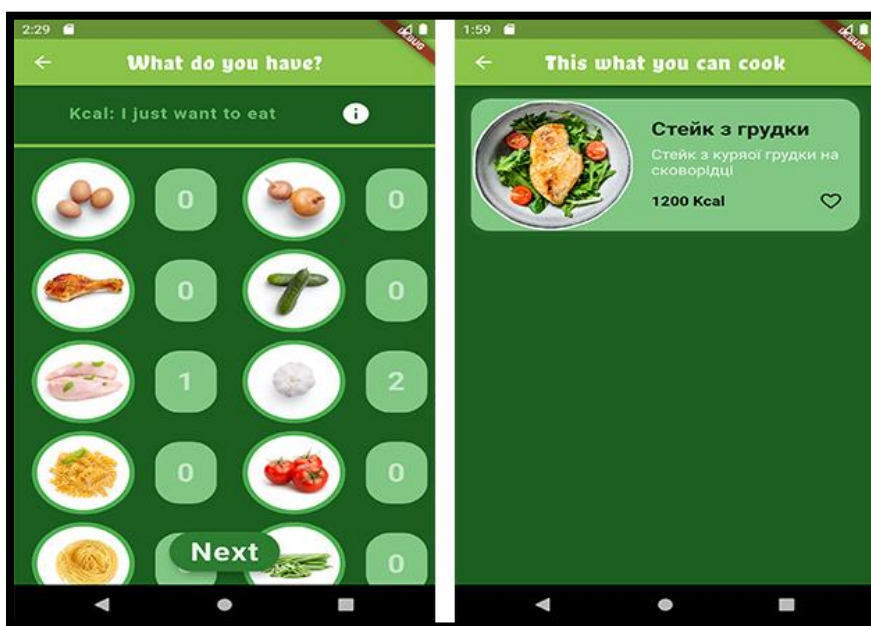


Рис. 3.14 а (обрані інгредієнти) Рис. 3.14 б (результат)

На рисунку 3.14 б Можна побачити результат який отримає користувач якщо введе в додаток що має в наявності лише курячу грудку та часник, програма запропонує йому приготувати стейк з грудки тому що з існуючого переліку страв лише на цю страву йому вистачає інгредієнтів.

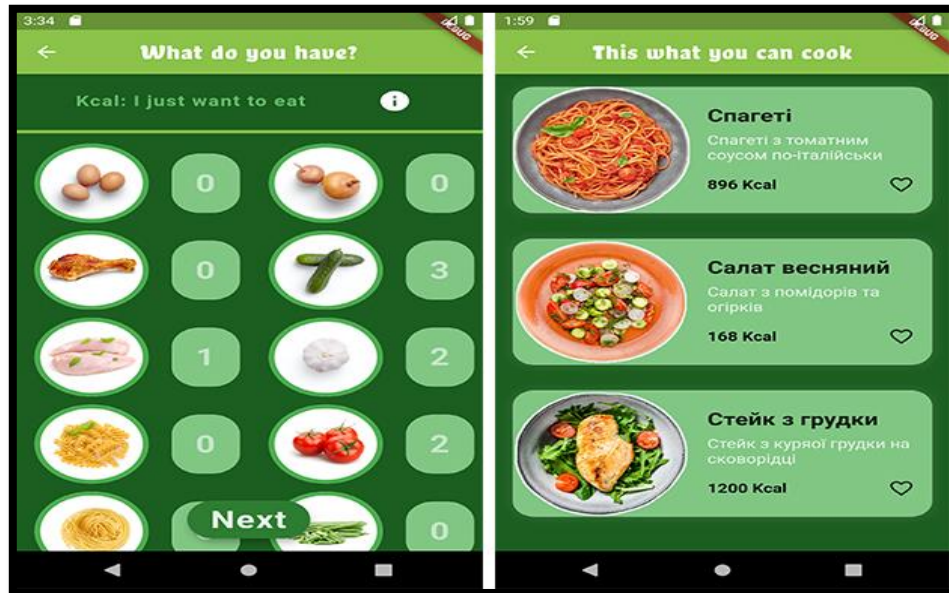


Рис. 3.15 а (обрані інгредієнти) Рис. 3.15 б (результат)

Проте, варто користувачу додати до списку помідорів, огірків та макаронів, як список страв значно зростає. Відображаються лише ті рецепти які підходять за калорійністю, але якщо на першому кроці їй не було вказано то відображаються усі.

Тепер пройдемося по елементах з яких складається сам рецепт. На рисунку 3. Частині В показано список з рецептів які користувач в змозі приготувати з наявних продуктів. Кожен рецепт складається з картинки, на якій зображена страва щоб користувач одразу мав чітке уявлення страви яку він матиме в кінці і одразу вирішив для себе хотів би він її з'їсти чи ні, з короткої назви рецепту, з розширеної назви рецепту, яка відображає окрім назви ще й основні інгредієнти страви, а також з кількістю калорій яку містить страва, та іконки з сердечком натиснувши на яку він може відмітити її як страву яка сподобалась і наступного разу коли він скористається додатком він побачить темне сердечко біля цієї страви і знатиме що її він вже готував і вона йому сподобалась.

### 3.5 Четверта сторінка

Після того як користувач обрав страву яку б він хотів приготувати і натиснувши на її іконку, він потрапляє на фінальну сторінку – з рецептом обраної страви.

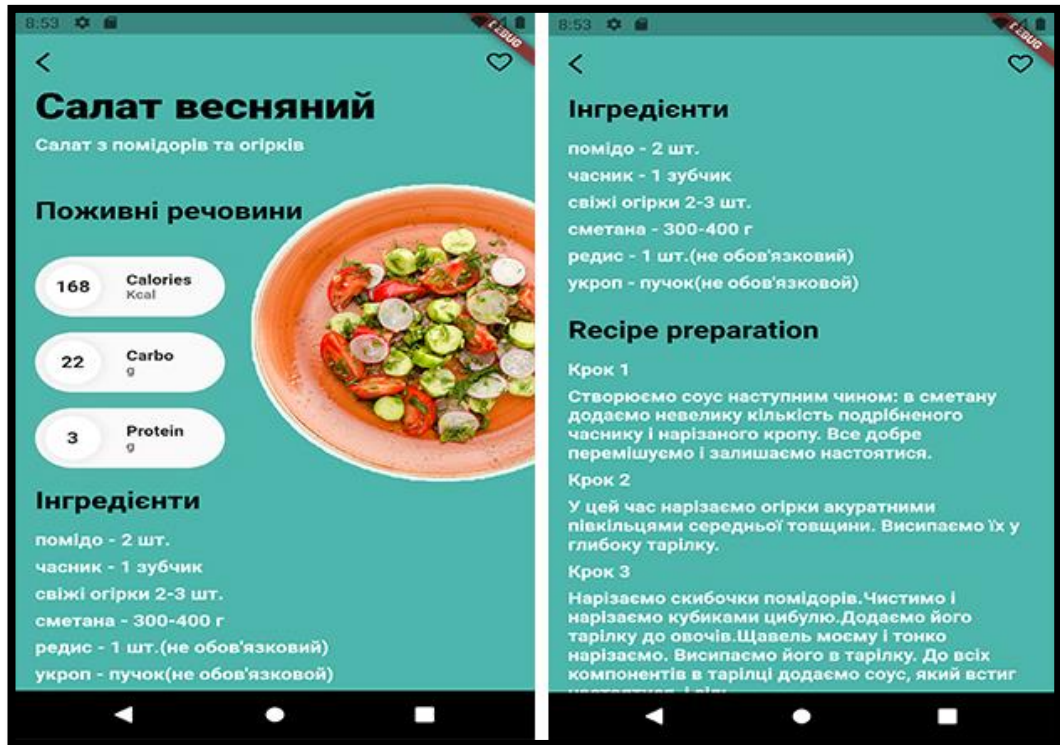


Рис. 3.16 а (детальна інформація про страву) Рис. 3.16 б( детальний рецепт)

На рисунках 3.16а та 3.16 б Показано як виглядає фінальна частина додатку, а саме рецепт за яким студент ввійшов в створений в ході даної дипломної роботи додаток. Сторінка з рецептом містить в собі всю потрібну інформацію і нічого зайвого, а саме: на ній написано про те скільки калорій містить страву та карбонів та протеїнів, з правої сторони екрану розміщене детальніше фото страви яку збирається приготувати користувач. В розділі “Інгредієнти” розміщений список необхідних для страви інгредієнтів, а під ним ідуть детальні кроки як приготувати страву.

## ВИСНОВКИ

В даній роботі було проведено дослідження програм для фітнесу та програм для приготування їжі, було проаналізовано їхні плюси та мінуси. Було виділено переваги та недоліки існуючих програмних продуктів та спроектовано і розроблено власне рішення з урахуванням потреб цільового сегменту.

Виконана робота демонструє перспективи використання Flutter у створенні додатків на такі мобільні платформи як Android та iOS.

В ході даної роботи було використано сучасне програмне забезпечення яке створює симуляцію телефону. Розроблене програмне забезпечення гнучке, можна внести зміни і це не порушить працездатність інших складових. В додаток можна вносити безліч різних функцій, якщо це буде потрібно. Додаток легко тестується, отже це означає, кількість помилок зменшується і збільшується його захист.

Елементи розробленого програмного забезпечення і структуру можна використовувати в інших проектах. Код чітко структурований тому придатний для розуміння іншими програмістами.

Завдяки Android studio та Flutter було спроектовано додаток який може позмагатися з багатьма додатками аналогами на сучасному ринку. Дана робота полегшить життя середньостатистичного студента, та при правильному використанні допоможе йому підкачатись чи схуднути залежно від його цілей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вовчанський М. О. Зелений колір [Електронний ресурс] / М. О. Вовчанський // Гармоничное развитие личности. – 2020. – Режим доступу до ресурсу: <https://mystroimmir.ru/psihologiya/cvet/zelenyj.html>.
2. Android studio [Електронний ресурс] // Вікіпедія. – 2021. – Режим доступу до ресурсу: [https://ru.wikipedia.org/wiki/Android\\_Studio](https://ru.wikipedia.org/wiki/Android_Studio).
3. Flutter documentation [Електронний ресурс] // flutter dev. – 2021. – Режим доступу до ресурсу: <https://flutter.dev/docs>.
4. Dart documentation [Електронний ресурс] // dart. – 2021. – Режим доступу до ресурсу: <https://dart.dev/overview>.
5. Phillips B. Android Programming: The Big Nerd Ranch Guide [Електронний ресурс] / Bill Phillips // Bill Phillips. – 2021. – Режим доступу до ресурсу: <https://www.amazon.com/Android-Programming-Ranch-GuideGuides/dp/0321804333>.
6. OOP [Електронний ресурс] // Ерап. – 2021. – Режим доступу до ресурсу: <https://training.epam.ua/#!/News/275?lang=ru>.
7. Алгоритмы [Електронний ресурс] // otus. – 2020. – Режим доступу до ресурсу: <https://otus.ru/nest/post/829/>.
8. Типы Баз Данных [Електронний ресурс] // proglib. – 2020. – Режим доступу до ресурсу: <https://proglib.io/p/11-tipov-sovremennyh-baz-dannyh-kratkie-opisaniya-shemy-i-primery-bd-2020-01-07>.
9. SQL documentation [Електронний ресурс] // microsoft. – 2021. – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>.
10. Java documentation [Електронний ресурс] // Oracle. – 2020. – Режим доступу до ресурсу: <https://docs.oracle.com/en/java/>.
11. kotlin doc [Електронний ресурс] // Oracle. – 2020. – Режим доступу до ресурсу: <https://kotlinlang.org/docs/home.html>.
12. patterns arc [Електронний ресурс] // itproger. – 2020. – Режим доступу до ресурсу: <https://tproger.ru/tag/design-patterns/>.

13. photography [Электронный ресурс] // dabl.v. – 2021. – Режим доступа до ресурсу: <https://unsplash.com/s/photos/buckwheat>.
14. UML diagrams [Электронный ресурс] // intellect. – 2020. – Режим доступа до ресурсу: <https://intellect.icu/uml-unified-modeling-language-diagrammy-klassov-i-sostoyanij-4300>.
15. Android Cookbook: [Электронный ресурс] // Ian F. Darwin. – 2019. – Режим доступа до ресурсу: <https://kupichitay.com.ua/product/android-cookbook-problems-and-solutions-for-android-developers-2nd-edition-ian-f-darwin/>.

## ДОДАТКИ

### Main

```
import 'package:duplom/FirstPage.dart';
import 'package:flutter/material.dart';

void main() => runApp(FitnesApp());

class FitnesApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: FirstPage(),
    );
  }
}
```

### FirstPage

```
import 'package:duplom/SecondPage.dart';
import 'package:flutter/material.dart';

class FirstPage extends StatefulWidget {
  @override
  _FirstPageState createState() => _FirstPageState();
}

class _FirstPageState extends State<FirstPage> {
  String value = "";

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```

backgroundColor: Colors.green[900],
resizeToAvoidBottomPadding: false,
appBar: AppBar(
  title: Text('How many calories ?',
    style: TextStyle(
      letterSpacing: 1, fontFamily: 'ChelaOne', fontSize: 25)),
  centerTitle: true,
  backgroundColor: Colors.lightGreen,
  elevation: 0.0,
),
body: Container(
  padding: EdgeInsets.fromLTRB(60, 40, 60, 0),
  child: Column(
    children: <Widget>[
      Container(
        alignment: Alignment.topCenter,
        child: TextFormField(
          decoration: InputDecoration(
            enabledBorder: OutlineInputBorder(
              borderSide:
                BorderSide(color: Colors.green[500], width: 5)),
            filled: true,
            fillColor: Colors.lightGreen[800],
            isDense: true,
            hintText: '2000',
            hintStyle: TextStyle(
              fontSize: 60,
              color: Colors.grey[350].withOpacity(0.3),
              fontFamily: 'ChelaOne'),
            counterText: "",

```

```

        contentPadding: EdgeInsets.fromLTRB(20.0, 10.0, 20.0, 10.0),
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(5.0)),
      ),
      keyboardType: TextInputType.number,
      maxLength: 5,
      textAlign: TextAlign.center,
      autofocus: false,
      style: TextStyle(fontSize: 55),
      onChanged: (text) {
        value = text;
      }
    ),
  Container(
    alignment: Alignment.center,
    padding: EdgeInsets.fromLTRB(0, 30, 0, 160),
    child: MaterialButton(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.all(Radius.circular(20.0))),
      elevation: 5.0,
      height: 90.0,
      minWidth: 200.0,
      color: Colors.green,
      onPressed: () {
        var myInt = int.parse(value);
        assert(myInt is int);
        if (myInt > 0 || myInt != null) {
          Navigator.of(context).push(MaterialPageRoute(
            builder: (context) => SecondPage(value: value),
          ));
        }
      }
    ),
  ),
)

```

```

    }
    ;
  },
  child: Text(
    "Enter",
    style: TextStyle(
      fontSize: 43,
      color: Colors.green[50],
      fontFamily: 'ChelaOne'),
    ),
  ),
),
Container(
  alignment: Alignment.bottomCenter,
  child: MaterialButton(
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.all(Radius.circular(20.0))),
    elevation: 5.0,
    padding: EdgeInsets.only(),
    height: 90.0,
    minWidth: 200.0,
    color: Colors.green[300],
    onPressed: () {
      Navigator.of(context).push(MaterialPageRoute(
        builder: (context) =>
          SecondPage(value: "I just want to eat"),
      ));
    },
  child: Text(
    "Skip",

```

```

        style: TextStyle(
          fontSize: 43,
          color: Colors.green[50],
          fontFamily: 'ChelaOne'),
      ),
    ),
  ],
),
),
);
}
}

```

## **SecondPage**

```

import 'package:duplom/ListOfRecipes.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:duplom/info.dart';

class SecondPage extends StatefulWidget {
  String value;

  SecondPage({ Key key, @required this.value }) : super(key: key);

  @override
  _SecondPageState createState() => _SecondPageState(value);
}

class _SecondPageState extends State<SecondPage> {
  String value;

```

```
bool pressed = false;
```

```
int amount = 0;
```

```
int egg = 0;
```

```
int onion = 0;
```

```
int chickenleg = 0;
```

```
int cucumber = 0;
```

```
int chickenbrest = 0;
```

```
int garlic = 0;
```

```
int macaroni = 0;
```

```
int tomato = 0;
```

```
int greenbean = 0;
```

```
int spaghetti = 0;
```

```
int carrot = 0;
```

```
int buckwheat = 0;
```

```
_SecondPageState(this.value);
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
  return Scaffold(
```

```
    backgroundColor: Colors.green[900],
```

```
    appBar: AppBar(
```

```
      centerTitle: true,
```

```
      title: Text("What do you have?",
```

```
        style: TextStyle(
```

```
          letterSpacing: 1, fontFamily: 'ChelaOne', fontSize: 25)),
```

```
      backgroundColor: Colors.lightGreen,
```

```
    ),
```

```

body: SingleChildScrollView(
  child: Column(
    children: <Widget>[
      Container(
        padding: EdgeInsets.only(top: 10),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: <Widget>[
            Align(
              alignment: Alignment.topLeft,
              child: Text(
                'Kcal: $value',
                style: TextStyle(
                  color: Colors.green[400],
                  letterSpacing: 1.0,
                  fontSize: 18,
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
          ],
        ),
      Container(
        child: IconButton(
          iconSize: 30,
          icon: Icon(Icons.info),
          color: Colors.white,
          onPressed: () {
            Navigator.of(context).push(MaterialPageRoute(
              builder: (context) => info(),
            ));
          },
        ),
      ),
    ],
  ),
)

```

```

    },
  ),
),
],
),
),

```

```

Container(
  padding: EdgeInsets.fromLTRB(0, 10, 0, 10),
  child: Divider(
    height: 10,
    thickness: 4,
    color: Colors.lightGreen,
  ),
),

```

```

///egg and onion

```

```

Container(
  padding: EdgeInsets.only(left: 10),
  alignment: Alignment.topLeft,
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: <Widget>[
      ClipOval(
        child: Container(
          width: 100.0,
          height: 100.0,
          decoration: BoxDecoration(
            image: DecorationImage(
              image: AssetImage('assets/egg.jpg'),

```

```

    ),
  ),
  child: FlatButton(
    shape: RoundedRectangleBorder(
      side: BorderSide(
        color: Colors.green,
        width: 5,
        style: BorderStyle.solid),
      borderRadius: BorderRadius.circular(50)),
    padding: EdgeInsets.all(0.0),
    onPressed: () {
      setState(() {
        egg += 1;
        amount += 1;
        pressed = true;
      });
    },
    child: GestureDetector(
      onLongPress: () {
        setState(() {
          if (egg > 0) egg -= 1;
          amount -= 1;
          if (amount == 0) {
            pressed = false;
          }
        });
      },
    ),
  ),
),

```

```

    ),
  ),
  SizedBox(width: 5),
  Material(
    color: Colors.green[900],
    child: ClipRRect(
      borderRadius: BorderRadius.only(
        topLeft: Radius.circular(25.0),
        topRight: Radius.circular(25.0),
        bottomLeft: Radius.circular(25.0),
        bottomRight: Radius.circular(25.0),
      ),
      child: Align(
        alignment: Alignment.bottomRight,
        heightFactor: 1,
        widthFactor: 1,
        child: Container(
          color: Colors.green[300],
          width: 60,
          height: 75,
          child: Center(
            child: Text(
              '$egg',
              style: TextStyle(
                color: Colors.green[100],
                letterSpacing: 1.0,
                fontSize: 30,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
        ),
      ),
    ),
  ),

```







```

children: <Widget>[
  ClipOval(
    child: Container(
      width: 100.0,
      height: 100.0,
      decoration: BoxDecoration(
        image: DecorationImage(
          image: AssetImage('assets/chickenleg.jpg'),
        ),
      ),
    ),
    child: FlatButton(
      shape: RoundedRectangleBorder(
        side: BorderSide(
          color: Colors.green,
          width: 5,
          style: BorderStyle.solid),
        borderRadius: BorderRadius.circular(50)),
      padding: EdgeInsets.all(0.0),
      onPressed: () {
        setState(() {
          chickenleg += 1;
          amount += 1;
          pressed = true;
        });
      },
      child: GestureDetector(
        onLongPress: () {
          setState(() {
            if (chickenleg > 0) chickenleg -= 1;
            amount -= 1;
          });
        },
      ),
    ),
  ],
)

```















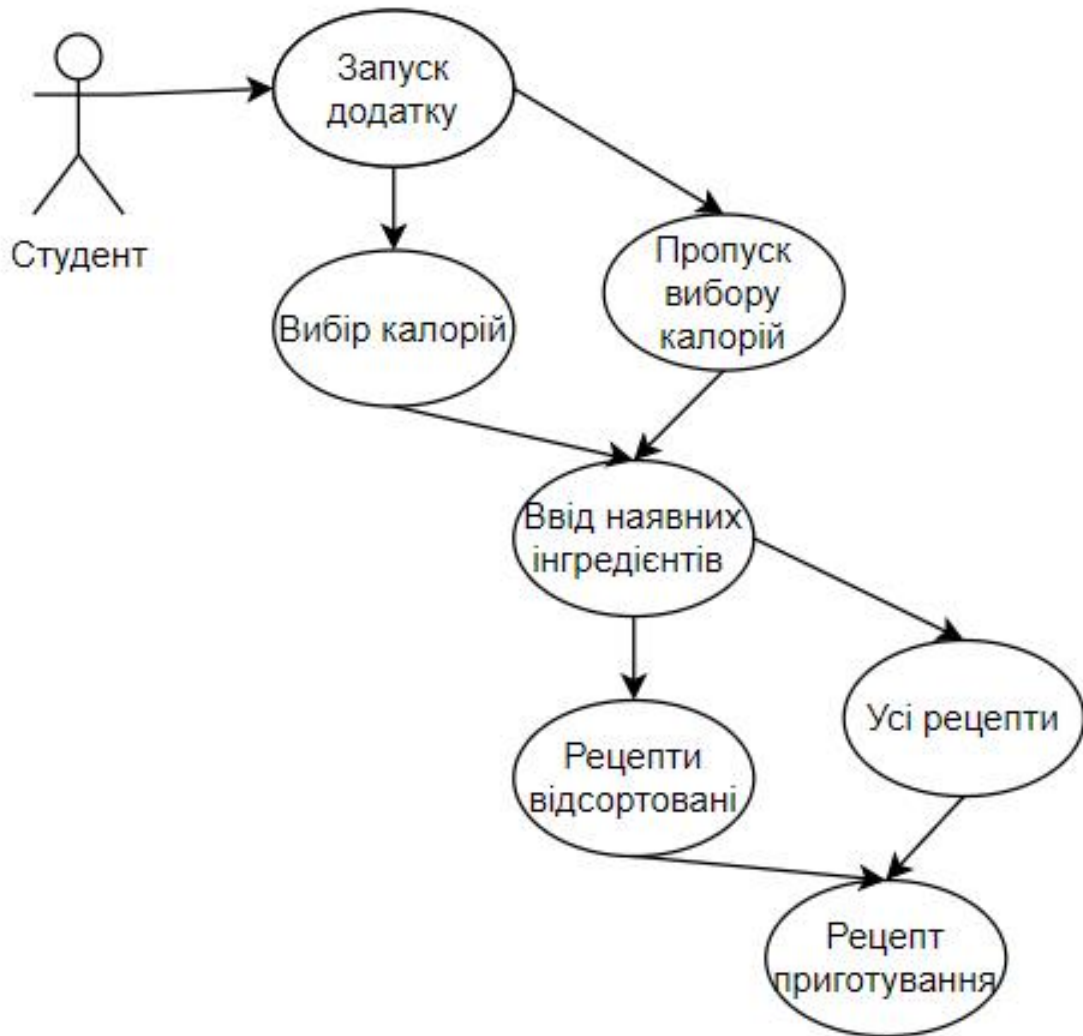
```

),
child: FlatButton(
  shape: RoundedRectangleBorder(
    side: BorderSide(
      color: Colors.green,
      width: 5,
      style: BorderStyle.solid),
    borderRadius: BorderRadius.circular(50)),
  padding: EdgeInsets.all(0.0),
  onPressed: () {
    setState(() {
      garlic += 1;
      amount += 1;
      pressed = true;
    });
  },
  child: GestureDetector(
    onLongPress: () {
      setState(() {
        if (garlic > 0) garlic -= 1;
        amount -= 1;
        if (amount == 0) {
          pressed = false;
        }
      });
    },
  ),
),
),
),
),

```

## Додаток 2

### Uml діаграма



Taras Shevchenko National University of Kyiv

Design and development of the server part of the web application for  
maintaining a database of publications

Software Architecture Document (SAD)

Content Owner: Oleksandr Mustafaiev

DOCUMENT NUMBER: 1.0

RELEASE: 1.0

RELEASE DATE: 01.06.2021

## TABLE OF CONTENTS

1. Introduction
  - 1.1. Purpose
  - 1.2. Scope
  - 1.3. Definitions
  - 1.4. References
2. Architecture Representation
3. Architectural Goals and Constraints
4. Sequence diagram
5. Architectural View
6. Size and Performance

## 1. Introduction

### 1.1. Purpose

This document provides a complete overview of the system architecture, using many different architectural views to depict different aspects of the system. It is designed to capture and transmit important architectural decisions made in the system.

### 1.2. Scope

This Software Architecture Document provides an architectural overview of software to aid in a student's diet. The software simplifies the student's cooking process. System has 3 modules: input module, video stream module and image analysis module.

### 1.3. Definitions

In flutter, Widget is a way to declare and construct UI. If you are familiar with the Android or iOS development then you might make the immediate connection with the views (on Android) or UIViews (on iOS).

### 1.4 References

Applicable references are:

- Flutter
- Dart
- App
- VirtualPhone
- Android

## 2. Architecture Representation

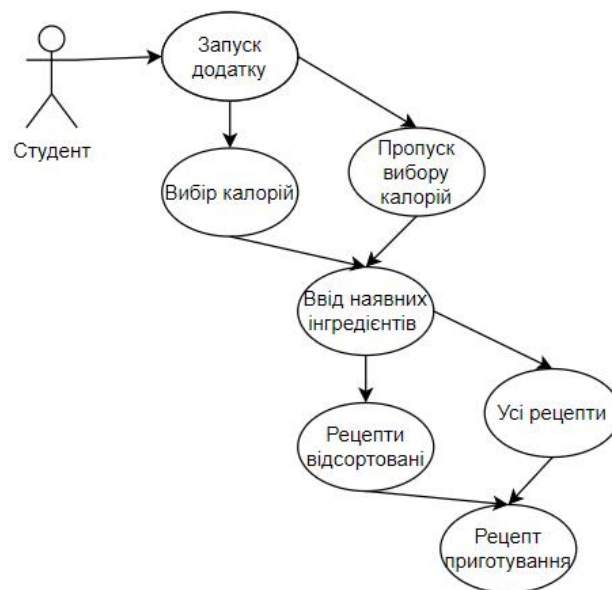
Event-driven architecture. In this case, the developer prescribes the behavior (reactions) for the program when any events occur. An event in the system is a significant change in its state. This architecture significantly reduces the response time of the program.

## 3. Architectural Goals and Constraints

Develop an application to simplify the process of cooking meals of students with a minimum number of operations to maximize the reduction of time for the entire cooking process.

## 4. Use-case View

A description of the use-case view of the software architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.



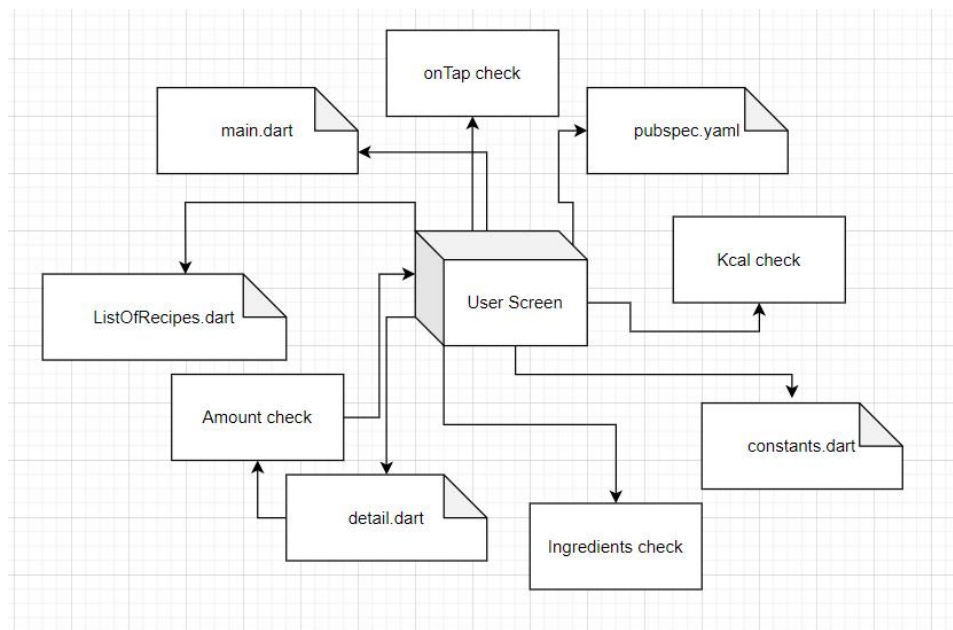
The programming system has 4 types of use cases:

- Data Entry - The user enters the desired calorie value. after pressing "Enter" the next page.
  - Entering ingredients - after entering the calories, you need to enter the ingredients that are available
  - Choosing a recipe - after entering the ingredients, you need to choose the recipe that is most suitable in terms of the number of calories.
  - Recipe - go to the page with the recipe and all the information you need about the preparation and usefulness of the dish
- Programming system has 3 types of use-cases:

## 5. Architectural View

Set of the most important decisions about the organization of the software system. The architecture includes:

selection of structural elements and their interfaces with which the system is composed, as well as their behavior within the framework of cooperation of structural elements, the connection of selected structural elements and behavior into ever larger systems, an architectural style that directs the entire organization - all elements, their interfaces, their cooperation and their connection



## 6. Size and Performance

The chosen software architecture supports key size and time requirements:

- The application must respond quickly to user actions
- Responses should arrive quickly with a delay of no more than 1 second.