

Міністерство освіти і науки України  
Державний вищий навчальний заклад  
«Прикарпатський національний університет імені Василя Стефаника»  
Факультет математики та інформатики  
Кафедра комп'ютерних наук та інформаційних систем

## МАГІСТЕРСЬКА РОБОТА

на тему “Система аналізу та оптимізації закупівель продуктів  
кінцевим споживачем”

Виконала: студентка 2 курсу, гр. КНМ-21  
спеціальності

122 комп'ютерні науки

(шифр і назва спеціальності)

Головко О.Р

(прізвище та ініціали студента)

Керівник д.т.н, проф. Петришин Л.Б

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Івано-Франківськ – 2020 р.

## АНОТАЦІЯ

Дана магістерська кваліфікаційна робота складається із переліку умовних скорочень, вступу, семи розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає – сто дві сторінки, робота ілюстрована двадцять шістьма рисунками, а список проаналізованих літературних джерел містить двадцять шість пунктів.

У магістерській кваліфікаційній роботі здійснено огляд та аналіз найбільш вживаних додатків-органайзерів для закупівлі продуктів. Проведено системний аналіз, що включає побудову дерева проблем та дерева цілей. Спроектовано та реалізовано інформаційно-довідкову систему для закупівлі продуктів.

В пояснювальній записці наведено функціональні та нефункціональні вимоги, опис функціонування системи за допомогою діаграм, опис програмних рішень, тестування та інструкції користувачу. Доцільність розробки системи підтверджено ґрунтовним економічним та маркетинговим аналізом.

## **ABSTRACT**

This master qualification work consists of a list of conditional abbreviations, an introduction, seven sections, conclusions, list of sources used and applications. The total volume of work is one hundred and two pages, the work is illustrated by twenty six drawings, and the list of analyzed literary sources contains twenty six points.

In the master qualification work the review and analysis of the most used application-organizers for the purchase of products was carried out. A systematic analysis has been carried out, including the construction of problem trees and objectives trees. The information and reference system for the purchase of products has been designed and implemented.

The explanatory note provides functional and non-functional requirements, description of system operation using diagrams, description of software solutions, testing and user instructions. The expediency of the system development is confirmed by sound economic and marketing analysis.

**ЗМІСТ**

АНОТАЦІЯ .....	3
ABSTRACT .....	4
ВСТУП.....	8
ХАРАКТЕРИСТИКА ОБ’ЄКТУ ДОСЛІДЖЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ .....	11
1.1. Основні поняття.....	11
1.2. Постановка задачі.....	12
Висновок до розділу.....	13
ОГЛЯД ЛІТЕРАТУРИ .....	14
1.3. Огляд платформ для мобільних додатків. ....	14
1.3.1. OS Android. ....	14
1.3.2. iOS.....	15
1.3.3. Telegram Bot.....	16
1.3.4. Вибір платформи. ....	16
1.4. Огляд аналогів. ....	16
1.4.1. Shopping List від maloii .....	17
1.4.2. Список покупок від Balu. ....	18
1.4.3. Смарт список покупок Listonic.....	19
1.4.4. Shopping List Buy me a pie!(Купи батон!).....	20
1.4.5. Список покупок August Software.....	21
1.4.6. Out of Milk - Grocery Shopping List.....	22
1.4.7. Каури Товари від Каури. ....	23
1.4.8. Порівняльна характеристика аналогічних програмних рішень. ....	25
Висновок до розділу.....	26
СИСТЕМНИЙ АНАЛІЗ.....	27

	6
1.5. Побудова дерева проблем та дерева цілей .....	27
1.5.1. Дерево проблем .....	27
1.5.2. Дерево цілей.....	29
1.6. Аналіз та вибір методів, алгоритмів та засобів розв’язання задачі .....	31
1.6.1. Огляд та аналіз мов програмування для Android-додатків.....	31
1.6.2. Програмування інтерфейсів за допомогою XML. ....	34
1.6.3. СКБД SQLite.....	35
1.6.4. Середовище розробки Android Studio. ....	36
1.7. Структурна схема системи з урахуванням інформаційних потоків. ....	37
Висновок до розділу.....	38
<b>РОЗРОБКА ТА ВІДОБРАЖЕННЯ АЛГОРИТМІВ РОБОТИ КОМПОНЕНТІВ ТА СИСТЕМИ .....</b>	<b>39</b>
1.8. Засоби проектування інформаційних систем. ....	39
1.8.1. Enterprise Architect.....	39
1.8.2. AllFusion Process Modeler .....	39
1.8.3. AllFusion ERwin Data Modeler. ....	40
1.9. Діаграми, що описують функціонування системи та її окремих блоків. .	40
1.9.1. Побудова діаграм IDEF0. ....	41
1.9.2. Відображення потоків даних за допомогою DFD-діаграми. ....	45
1.9.3. ER діаграма бази даних. ....	46
1.10. Use-case діаграма.....	48
Висновок до розділу.....	49
<b>РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ .....</b>	<b>51</b>
1.11. Загальна структура програмного проекту. ....	51
1.12. Розробка та опис програмних модулів.....	52

	7
1.13. Розробка та опис інтерфейсу користувача. ....	54
1.14. Опис використаних сторонніх бібліотек та модулів. ....	56
1.15. Опис проблем і нестандартних ситуацій, які виникали під час розробки та заходів для їх вирішення. ....	58
Висновок до розділу.....	59
<b>ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА</b> .....	60
1.16. Вимоги до апаратного забезпечення. ....	60
1.17. Інструкції щодо встановлення системи. ....	60
1.18. Тестування. ....	62
1.19. Інструкція користувачу.....	63
Висновок до розділу.....	69
<b>ВИСНОВКИ</b> .....	70
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	72

## ВСТУП

Застосунки для смартфонів заповнили різні сфери життя сучасної людини. Швидкість життя наростає, потрібно кудись бігти, щось шукати, з кимось бачитися, не забувати при цьому поїсти і поспати. Для організації власного часу сформувалась ціла гілка додатків: для організації зустрічей, робочих завдань, списків справ чи тренувань. Тут варто згадати, що необхідною складовою життя є їжа, і, враховуючи те, що більшість людей її готує вдома, існує потреба ходити за продуктами до супермаркету. Ось тут і з'являється проблема: як організувати великий список покупок, щоб не загубитися в магазині та не метушитися довкола однієї полицки. Для цього слугуватиме інформаційно-довідкова система для закупівлі продуктів.

**Актуальність дослідження** полягає в тому, що в сучасної людини виникає потреба організації власного часу, зокрема часу на закупівлю продуктів та приготування страв. Інформаційно – довідкова система для закупівлі продуктів складатиметься з розділу списки для закупівлі та розділу рецептів, з яких можна буде додавати інгредієнти до списку одним натисканням. Більше ніякої неорганізованості – навіть при вході до магазину можна зайти в свій список рецептів, обрати що хочеться поїсти на вечерю, до прикладу грецький салат, додати продукти з рецепту до списку, а він вже сам відсортується за категоріями товарів.

**Метою магістерської роботи** є аналіз та розробка інформаційно-довідкової системи для закупівлі продуктів, в якій буде зібрано персоналізований словник продуктів, а також списки покупок та рецепти користувача.

**Об'єкт дослідження** – методи та засоби організації особистих даних та часу користувача.

**Предмет дослідження** – розробка інформаційно-довідкової системи для організації закупівлі продуктів.

**Задача магістерської кваліфікаційної роботи** – проектування та розробка функціоналу інформаційно-довідкової системи для закупівлі продуктів, що

складається з списку продуктів до закупівлі та власних рецептів користувача.

Щоб виконати поставлену задачу, ставимо такі завдання:

- охарактеризувати об'єкт дослідження та зробити постановку задачі;
- оглянути літературні джерела та наявні на ринку аналогічні програмні рішення, розглянути їх переваги та недоліки;
- провести системний аналіз системи, розробити дерево проблем та дерево цілей, на основі проведеного аналізу розробити функціональні та нефункціональні вимоги;
- на основі сформульованих вимог спроектувати та розробити алгоритм роботи компонентів та системи в цілому;
- розробити програмне рішення, використовуючи діаграми, спроектовані на попередньому етапі;
- провести тестування компонентів системи та всієї системи в цілому, за необхідності внести правки в програмне рішення;
- розробити інструкції користувачу з установки та користування розробленою системою;
- розробити рекомендації щодо подальших оновлень системи;
- провести економічне та маркетингове обґрунтування доцільності проведення розробки.

**Практична цінність** магістерської кваліфікаційної роботи полягає у тому, що створену систему можна застосовувати як готовий окремий застосунок для організації власного часу в галузі закупівлі продуктів та приготування страв. Окремі частини розробленого програмного продукту можна використати при розробці веб-ресурсів з кулінарними рецептами, а також систем доставки продуктів.

В перспективі можливе розширення розробленої системи: надати можливість створення облікових профілів користувачів для синхронізації та групового управління списками, сайт для доступу до власних списків з браузера будь-якого пристрою, а також підключення API однієї з великих мереж



супермаркетів для формування з складеного списку покупок замовлення доставки продуктів додому.

Результатом виконання бакалавської кваліфікаційної роботи є формування пояснювальної записки, що містить детальний опис проведених робіт відносно кожного з зазначених вище завдань.

## ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1. Основні поняття.

Органайзер(англ.organizer) – спочатку цей термін визначав невелику книгу, що служить для організації персональної інформації, яка містить в собі календар, адресну та телефонну книгу, список справ, покупок, подій тощо. Наразі органайзери існують в паперовому(блокноти, щоденники, списки справ та покупок) і в комп'ютеризованому вигляді(окремі спеціалізовані додатки для смартфонів, комп'ютерів, веб-сайти тощо) [1].

Комп'ютерна програма-органайзер – це прикладна програма, що призначена для накопичення персональної інформації користувача, організації справ та контролю їх виконання. Перевагами комп'ютерних органайзерів є:

- можливість збереження великих обсягів інформації;
- швидкий і простий пошук;
- компактність та мобільність;
- встановлення автоматичних нагадувань;
- можливість синхронізації інформації.

Форми органайзера можуть бути різними: загальними, що мають в собі кілька функціональних компонентів та вузькоспеціалізованими(органайзери покупок, рецептів, фінансів, списки фільмів для перегляду, списки книг, тощо). Недолік багатофункціональних органайзерів саме в великій кількості функцій, з яких активно використовується лише декілька, також велика кількість функцій робить інтерфейс громіздким та заплутаним, користувачу простіше завантажити кілька різних додатків, які йому дійсно потрібні.

Основні вимоги до програм-органайзерів:

- інтуїтивно зрозумілий інтерфейс;
- простота використання;
- можливість налаштування «під себе».

Об'єктом дослідження магістерської кваліфікаційної роботи є організація персональної інформації в сфері закупівлі продуктів. Початково для закупівлі

продуктів робили списки на папері і в магазині подумки або ручкою викреслювали позиції, що вже в кошику. Це зручніше, ніж тримати весь список в пам'яті і менший ризик забути щось купити чи навпаки взяти зайве, але все ж можна забути готовий список вдома чи загубити дорогою до магазину. З активним проникненням в всі сфери життя інформаційних технологій, з'явилися додатки, що замінюють паперовий список покупок. Зараз же для зручності користування можна створити список і надіслати його комусь чи створювати списки для кількох людей, що автоматично синхронізуються. Досить корисним є сортування загального списку за категоріями: ринок, магазин, супермаркет, молочні продукти, побутова хімія, м'ясо, риба тощо.

## **1.2. Постановка задачі.**

Розроблюваний програмний продукт матиме назву «Buy`n`Cook»(купи та приготуй) та виконуватиме функції розумного списку покупок та організатора власних рецептів користувача.

Завданням програмного продукту є забезпечення зручної організації закупівлі продуктів користувача. Для виконання завдання повинно бути забезпечено можливість:

- створювати та редагувати список покупок;
- створити та наповнювати словник продуктів;
- сортувати продукти за категоріями;
- створювати власні категорії продуктів;
- надсилати список за допомогою електронної пошти, смс чи месенджерів;
- додавати нові рецепти до книги рецептів та редагувати вже існуючі;
- додавати до списку покупок всі інгредієнти з рецепту.

Для зручності використання додатку передбачається створення словника продуктів та сортування за стандартними категоріями – відділами звичайного супермаркету(молочний відділ, м'ясо, фрукти, овочі, заморожені продукти тощо).

### **Висновок до розділу**

В даному розділі магістерської кваліфікаційної роботи наведено характеристику об'єкту дослідження та поставлено завдання, які попередньо має виконувати додаток, що розробляється. Проведено дослідження історії органайзерів та наведено основні переваги комп'ютерних програм перед паперовими аналогами, а також переваг вузькоспеціалізованих органайзерів порівняно з багатофункціональними.

## ОГЛЯД ЛІТЕРАТУРИ

Як вже було сказано вище, початково організатори, зокрема організатори покупок, були аркушами паперу. Тому зараз люди організовують свої покупки як і за допомогою того ж паперу, так і за допомогою стандартних програм телефона чи комп'ютера, до прикладу «Нотаток». Також список покупок можна створити надсилаючи текстове повідомлення собі, та поширити його надсилаючи це ж повідомлення другу чи родичу. В цих випадках можна просто записати список, щоб його було важче загубити ніж його паперового аналога. Проте, варто зазначити, що списки з програми «Нотатки» та повідомлення мають такий же недолік – товари з нього видаляються стиранням всього слова, тому більшість користувачів просто не видаляють вже куплені позиції прямо в магазині коло полицки, а при нагоді вже вдома. Якщо список великий, в ньому важко орієнтуватися, а неможливість сортування змушує бігати кілька разів до одного і того ж стелажа.

Складення списку в стандартних програмах комп'ютера має крім вище зазначених ще один великий недолік, його важко взяти з собою. Звісно можна роздрукувати і зробити аналога паперовому списку, або ж надіслати з комп'ютера повідомлення і прочитати його на телефоні, як згадувалось раніше.

Зважаючи на все вищесказане, було зроблено висновок, що розроблювана система має бути представлена на мобільних платформах. Нижче розглянемо порівняння різних мобільних платформ та вибір найбільш вигідної для нас.

### **1.3. Огляд платформ для мобільних додатків.**

#### **1.3.1. OS Android.**

Android OS – платформа та операційна система для мобільних телефонів та планшетних комп'ютерів, створена на базі ядра Linux компанією Google. За даними Statista, станом на другий квартал 2018 року, на 88% проданих смартфонів в світі була встановлена операційна система Android.[2]

Більшість додатків для Android пишуться на мові Java, інструменти Android SDK компілюють написаний код та всі файли ресурсів в файл APK. В файлі APK є все що необхідно для роботи додатку, він також дозволяє

встановити додаток на будь-якому іншому пристрої з операційною системою Android.

Кожна програма Android, встановлена на пристрої, працює у власній "пісочниці" (ізолюваному програмному середовищі):

- операційна система Android представляє собою багатокористувацьку систему Linux, в якій кожен додаток є користувачем;
- за замовчуванням система призначає кожному додатку унікальний ідентифікатор користувача Linux (цей ідентифікатор використовується тільки системою); система встановлює права для всіх файлів в додатку так, щоб доступ до них був дозволений тільки користувачеві з ідентифікатором, що призначений цьому додатку;
- у кожного процесу є власна віртуальна машина (VM), тож код програми виконується ізолювано від інших додатків;
- за замовчуванням кожен додаток виконується у власному процесі Linux. Android запускає процес, коли потрібно виконати будь-який компонент додатка, а потім завершує процес, коли він більше не потрібен або коли системі потрібно звільнити пам'ять для інших додатків.[3]

### 1.3.2. iOS.

iOS – власна операційна система компанії Apple, розроблена спочатку для iPhone, а потім також вдосконалена для iPad. Є похідною OS X, тож за своєю природою є Unix-подібною системою. За даними Statista, станом на другий квартал 2018 року, на 11.9% проданих смартфонів в світі була встановлена операційна система iOS[2].

Додатки для цієї платформи розробляються мовою Swift за допомогою Xcode – інтегрованого середовища розробки для macOS, iOS, watchOS та tvOS. Додатки встановлюються лише з офіційного магазину додатків AppStore, тому для тестування додатків перед релізом використовується TestFlight[4].

### 1.3.3. Telegram Bot.

Telegram – це програмне забезпечення для смартфонів, планшетів і персональних комп'ютерів, що дозволяє користувачам обмінюватися текстовими повідомленнями та файлами. Обліковий запис користувача приєднується до номеру мобільного телефону, на який надсилається код для входу.

За допомогою спеціально розробленого API розробники можуть створювати ботів – спеціальні акаунти, що керовані програмами. Типові боти відповідають в персональних чи групових чатах, можуть здійснювати пошук в Інтернет та інші завдання. [5]

Перевагою створення Telegram бота є його кросплатформеність, адже бот працює на базі додатку Telegram, що реалізований на більшості сучасних операційних систем; а також відпадає необхідність встановлювати і виділяти пам'ять та ресурси для окремого додатку.

Недоліком виступає те, що в бота обмежений користувацький інтерфейс, тому він в більшості випадків є менш зручним за окремий додаток на певній ОС.

### 1.3.4. Вибір платформи.

Враховуючи все вище написане, обрано операційну платформу Android, як оптимальну для вирішення поставленої задачі. Вона гарантує користувачу зручний інтерфейс а для для розробника зручну побудову додатку.

Між ОС Android та iOS вибір на користь Android, що дає можливість охопити майже в десять разів більшу частину ринку. Telegram бот не вибрано через не зручний для користувача інтерфейс, що в більшості випадків спонукає кінцевого споживача користуватися розробленим додатком.

## 1.4. Огляд аналогів.

Сьогодні на ринку програмного забезпечення представлено велику кількість додатків, що допомагають користувачу з складенням списку покупок: від найпростіших, що нагадують нотатки на папері, до складних рішень, що включають сканування штрих-коду товару, пошук заданого товару в магазинах

поблизу та підрахування загального чеку по списку. Проте в кожному з цих додатків необхідні продукти додаються по черзі один за одним, не представлено додатка, в якому можна було б зберегти, до прикладу, рецепт страви і, коли виникне потреба, додати в список зразу всі інгредієнти, а не кожного разу вписувати однакові списки. Нижче розглянемо основні аналогічні рішення, що мають найбільш схвальні відгуки на платформі GooglePlay.

#### 1.4.1. Shopping List від maloiі

Дуже простий додаток, за дизайном схожий на блокнот для нотаток. Елементи списку додаються за допомогою введення з клавіатури. Натиснувши на елементи списку можна їх відмітити як вже куплені, позначити як важливі, редагувати або сканувати штрих-код товару. Список можна посортувати в алфавітному порядку, сховати куплені елементи або надіслати комусь. Налаштувати можна лише шрифт, купівлю елементу одним доторканням та мову інтерфейсу. Знімки екрану при використанні даного додатку наведено на рис 2.1.

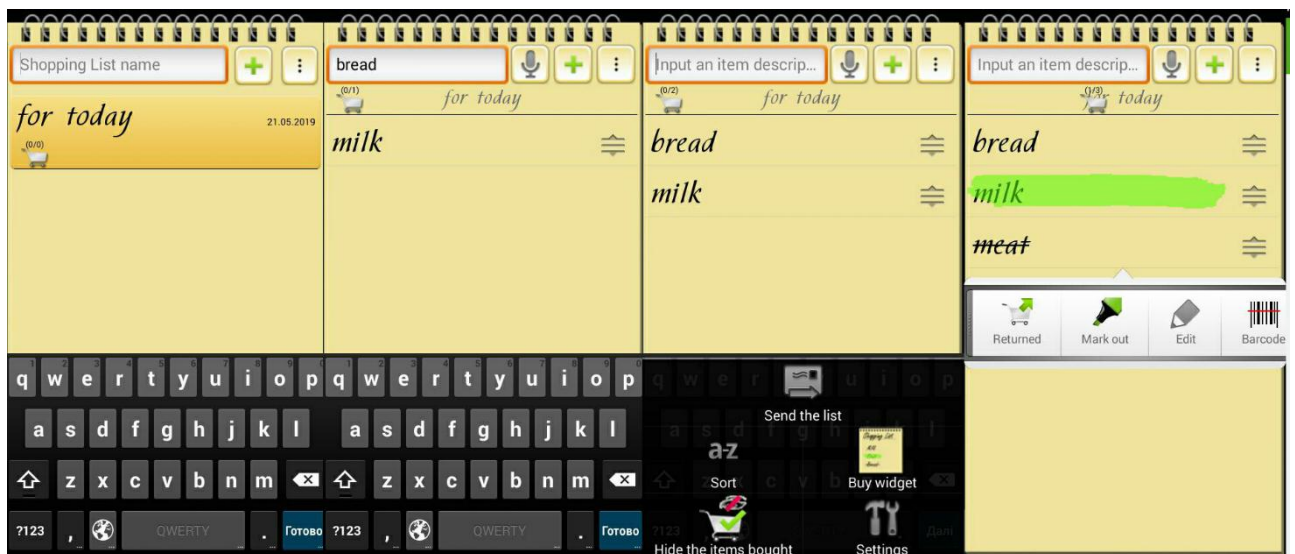


Рис.1.1 Знімки екрану при використанні додатку Shopping List від maloiі

Переваги даного додатку:

- простий, зрозумілий інтерфейс;
- голосове введення;
- виділення важливих елементів списку.

Недоліками можна вважати:



- застарілий дизайн;
- не зручне сортування.

#### 1.4.2. Список покупок від Valu.

Простий у використанні додаток, що дозволяє легко та швидко створити список покупок. Товари в список можна вносити з клавіатури або за допомогою голосового введення, є можливість введення кількості необхідних товарів(можна вибрати одиниці кількості, наприклад кілограми, пачки, десятки), їх ціни з підтримкою різних валют та нотаток. Є зручна можливість наповнювати «Мій список» з найчастіше затребуваними товарами, в поточний список можна додати зразу кілька товарів або всі одразу. В налаштуваннях є можливість зміни шрифтів, валюти, категорій товарів, сортування, оформлення екрану. Знімки екрану при використанні додатку «Список покупок від Valu» наведено на рис.2.2.

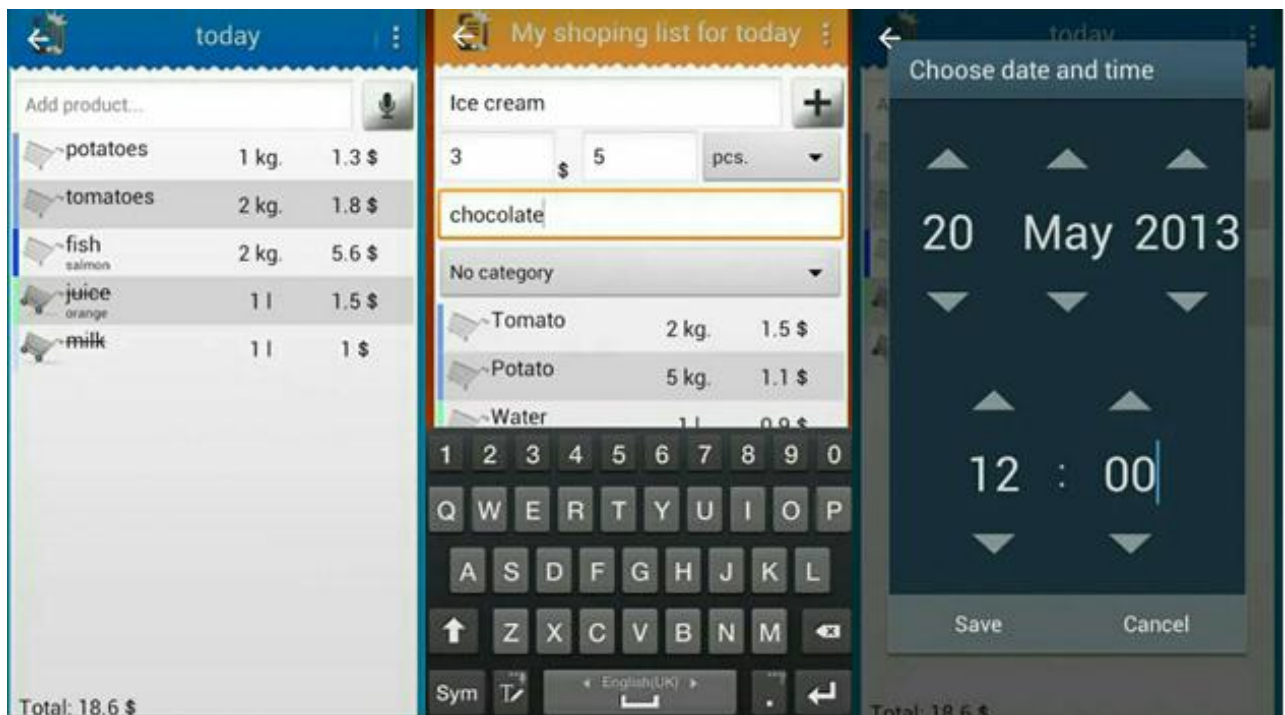


Рис.1.2 Знімки екрану при використанні застосунку Список покупок від Valu.

Переваги даного програмного продукту:

- голосовий набір;
- підтримка різних валют;
- підтримка категорій товару та створення власних категорій;

- нагадування;
- «Мій список» – можна зберегти найчастіше використовувані товари і в список покупок додавати товари звідти.

Недоліки:

- застарілий дизайн;
- відсутня синхронізація і можливість надіслати комусь готовий список;
- погана локалізація.

### 1.4.3. Смарт список покупок Listonic.

Простий у використанні, додавання та викреслювання продуктів зі списку одним дотиком. Є можливість голосового введення, редагувати список можна в додатку на телефоні, або на сайті виробника. Є можливість долучення інших користувачів до списку та контроль витрат. Також є сортування списку за категоріями товарів. В додатку є власна база товарів, розумний помічник пропонує найбільш релевантні позиції. Списки, що переміщено до кошика можна відновити та повторно використати. На рисунку 2.3 наведено знімки екрану при використанні додатку Listonic.

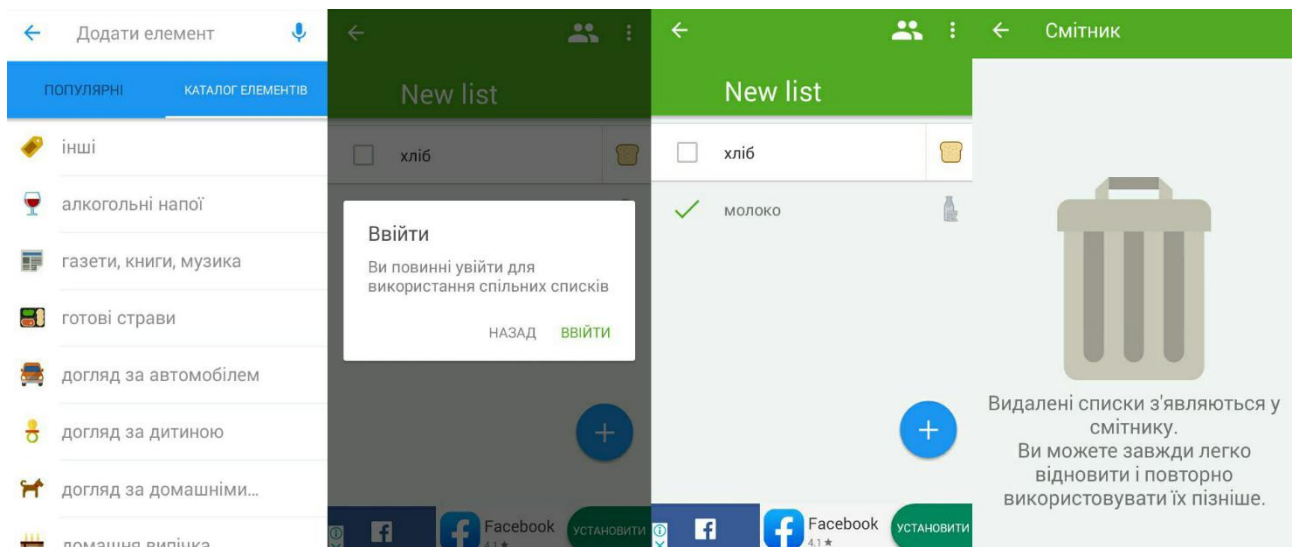


Рис.1.3 Знімки екрану при використанні додатку Listonic.

Переваги цього додатку:

- до товару можна додати ціну, кількість, та фото;

- синхронізація, спільні списки, доступ з сайту виробника, якщо створено обліковий запис;
- велика кількість категорій товарів, можливість редагування і додавання нових;
- підрахунок приблизної вартості списку та автоматичне додавання цін з останніх списків покупок.

Недоліки цього додатку:

- одиниці виміру товару(кілограми, пачки, десятки) необхідно вводити кожного разу;
- ділитися списком можуть лише зареєстровані користувачі, і з зареєстрованими користувачами, не передбачена можливість поширення через електронну пошту.

#### 1.4.4. Shopping List Buy me a pie!(Купи батон!).

Зручний в використанні додаток, що дозволяє створювати списки покупок з уже наявних в базі даних товарів. Після налаштування мови словника, при введенні додаток пропонує вибрати товар з бази або ввести свій, є можливість введення кількості товару. Для зареєстрованих користувачів є можливість синхронізації даних дані між пристроями на яких виконано вхід з одного облікового запису, додавання інших зареєстрованих користувачів до редагування та перегляду списку, а також є доступ через сайт виробника. Можна створити декілька списків покупок, сортування за категоріями товарів та створення власних категорій. Оплативши підписку користувач отримує більше кольорів для кращого поділу товарів за категоріями, не обмежену кількість списків та відсутність реклами.

На рисунку 2.4 наведено знімки екрану при використанні додатку Buy me a Pie!

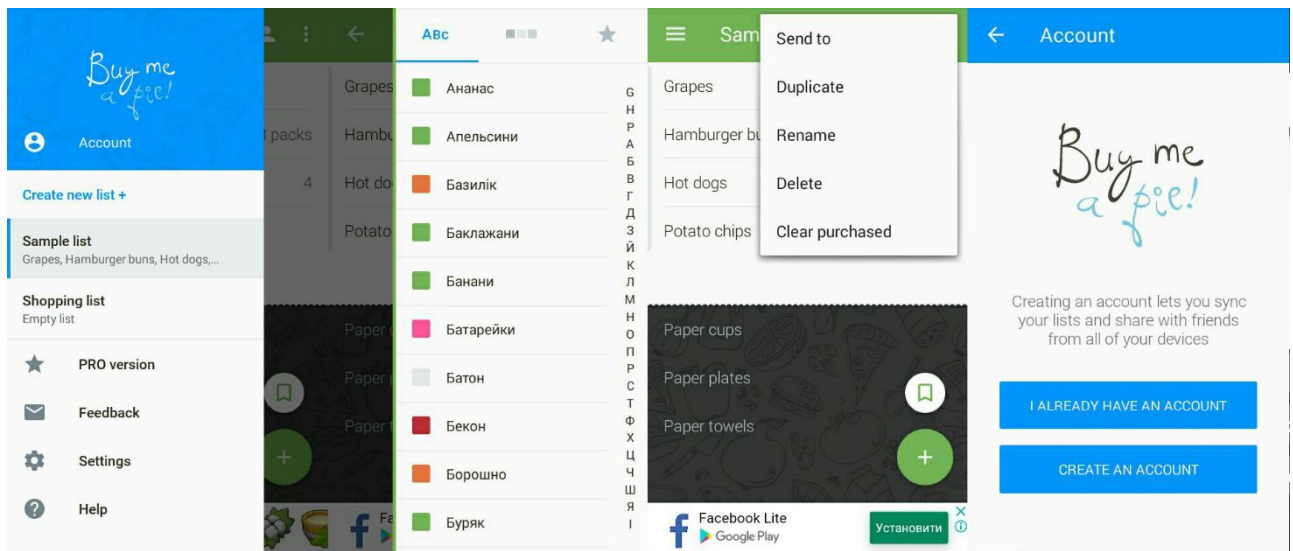


Рис.1.4 Знімки екрану при використанні додатку Buy me a pie!

#### Переваги:

- власний словник товарів, що уже розділені за категоріями;
- синхронізація та можливість доступу з будь-якої точки світу за допомогою сайту виробника;
- локалізація словника товарів більш ніж десятьма найпоширенішими мовами світу.

#### Недоліки:

- обмежена кількість списків, що можна зберігати одночасно;
- обмежена кількість користувачів, що можна додати до списку.

#### 1.4.5. Список покупок August Software.

Особливості цього програмного продукту: зрозумілий і інтуїтивно простий інтерфейс, швидке формування списку покупок, групування товарів в списку, редагування елементів списку, можливість вибору мови, створення декількох списків та пошук списку за датою. В налаштуваннях є можливість введення вартості товарів з словника, за допомогою цих цін можна підрахувати приблизну суму закупівлі, коли товар позначено як куплений – програма рахує кількість витрачених коштів. Створений список можна надіслати через пошту чи месенджери. Переглянути знімки екрану при використанні даного додатку можна на рисунку 2.5.

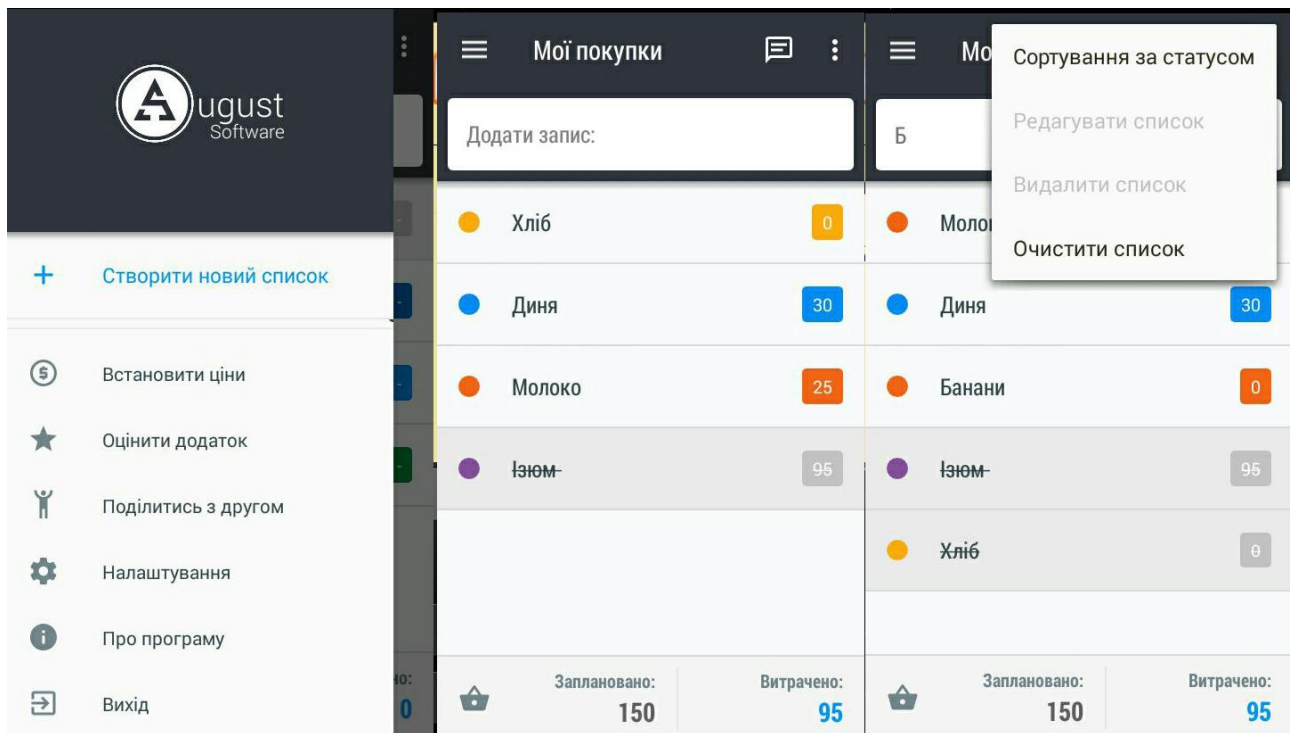


Рис.1.5 Знімки екрану при використанні додатка від August Software

Перевагами цього продукту є його простота, можливість встановлення цін на товари, велика кількість списків, що можуть бути збережені одночасно, пошук списків за датою створення. Недолік – відсутність різних грошових валют, введення цін не за пачку або кілограм, а просто абстрактних, в результаті обчислення дуже приблизні і далекі від реальності.

#### 1.4.6. Out of Milk - Grocery Shopping List.

Простий та невеликий за обсягом додаток, що дає можливість скласти список покупок та список інвентаризації комори. Є можливість синхронізації, доступу з сайту виробника, надсилання списку покупок електронною поштою. Товари сортуються за категоріями, які налаштовуються під потреби користувача. Є можливість копіювання або переміщення товарів з комори в список покупок або навпаки. Якщо введено вартість товарів то підраховує вартість на даний момент та вартість всього списку товарів. Знімки екрану при використанні даного додатку можна переглянути на рисунку 2.6.

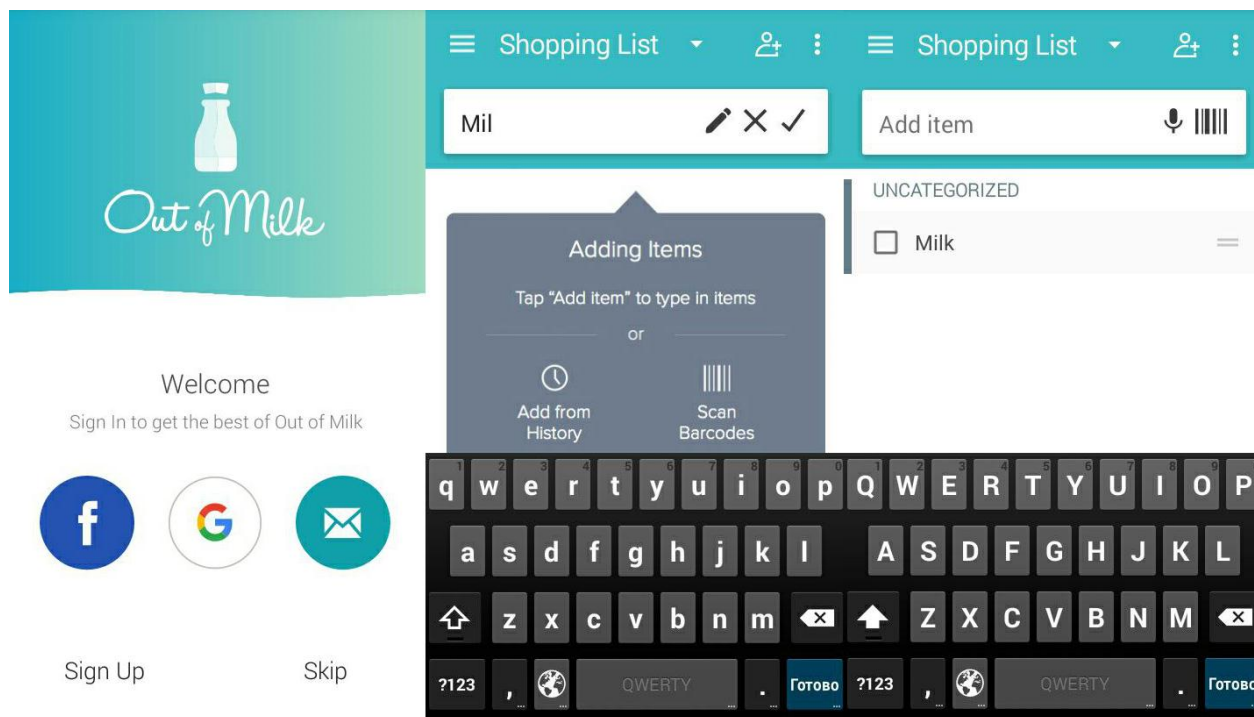


Рис.1.6 Знімки екрану при використанні додатку Out of Milk

Переваги даного програмного продукту:

- є можливість ввести недавні товари з історії;
- сканер штрих-кодів для введення товарів;
- комора, в якій позначено, що ще є в домі і швидко коли продукти закінчуються перенести їх в список покупок;
- синхронізація і доступ з сайту виробника, спільний доступ з іншими користувачами;
- сумарна сума покупок, сума вже в корзині, кількість елементів в списку та в корзині.

Недоліки:

- не зовсім зрозумілий інтерфейс, велике згромадження кнопок в впливаючому меню.

#### 1.4.7. Каури Товари від Каури.

Додаток відрізняється від інших з списку великим розміром та збільшеним функціоналом. Тут є пошук товару за штрих-кодом, тегом, виробником, порівняння цін на заданий товар в різних магазинах міста, можливість прокладення маршруту до магазину на карті. До товару можна

поставити оцінку, написати відгук, почитати відгуки інших користувачів, поділитися посиланням на товари в соціальних мережах та скласти список з родиною чи друзями. В додатку є величезна база товарів, яка кожного дня поповнюється користувачами(більш ніж 26000 товарів в тому числі ліків, продуктів, побутової техніки, книг тощо). Є рейтинг користувачів, побачивши відгук можна оцінити наскільки можна довіряти йому. Є список «Хочу в подарунок!», в якому записуються бажані подарунки і на певні свята список можна розіслати друзям, щоб отримати лише необхідне. Знімки екрану при використанні даного додатку наведено на рисунку 2.7.

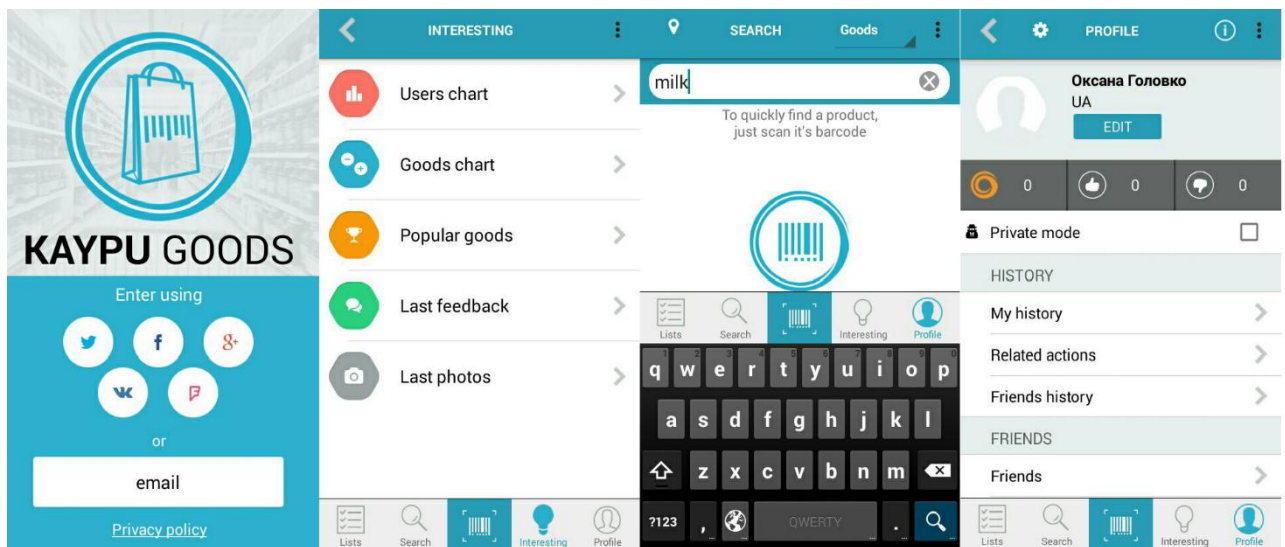


Рис.1.7 Знімки екрана при використанні додатку Каури Товари  
До переваг даного програмного продукту можна віднести:

- повну інформацію про товар: склад, калорійність, харчова цінність, об'єм для харчових продуктів; інформація про видавництво та тираж книг; характеристики комп'ютерної та побутової техніки тощо;
- база товарів, що поповнюється користувачами;
- рейтинг користувачів та відгуки про товари;
- можливість поширення списку покупок;
- додаткові можливості(карти навігації до найближчого магазину з потрібними товарами, список «Хочу в подарунок!»).

Недоліками можна вважати:

- громіздкий інтерфейс;
- наявність великої кількості функцій, що не використовуються більшістю користувачів;
- велика кількість ресурсів, що необхідні додатку та велика кількість дозволів, що він вимагає для встановлення.

#### 1.4.8. Порівняльна характеристика аналогічних програмних рішень.

На основі наведеного вище огляду аналогічних програмних рішень створимо таблицю з узагальненими характеристиками для наочнішого порівняння. Для порівняння обрано наступні характеристики: синхронізація між кількома пристроями та групове створення списку, можливість редагування з браузера, можливість надсилання списку, наявність бази даних товарів, групування за категоріями, додаткові способи введення, а також групове додавання товарів до списку і зручність інтерфейсу. Порівняння розглянутих аналогічних програмних рішень наведено в таблиці 2.1.

Таблиця 1.1

Порівняльна таблиця аналогічних програмних рішень

Назва додатку \ Характеристики	Maloii	Balu	Listonic	Buy me a pie!	August Software	Out of Milk	Капу
1	2	3	4	5	6	7	8
Синхронізація між кількома пристроями та групове створення списку	-	-	+	+	-	+	+
Можливість редагування з браузера	-	-	+	+	-	+	+



Продовження таблиці 2.1

1	2	3	4	5	6	7	8
Можливість надсилання списку	+	+	-	+	+	-	+
База даних товарів	-	-	+	+	+	+	+
Додаткові способи введення	+	+	+	+	-	+	+
Групове додавання товарів до списку	-	-	-	-	-	-	-
Зручний інтерфейс	-	-	+	+	-	-	-

Розглянувши згадані вище аналогічні програмні рішення, можна зробити висновок, що більшість додатків мають не дуже зручний інтерфейс, а також абсолютно в всіх додатках відсутнє групове додавання товарів до списку. Тому однією з цілей даної розробки має бути надання можливостей групового додавання товарів до списку на прикладі інгредієнтів з книги рецептів.

### **Висновок до розділу**

В даному розділі магістерської кваліфікаційної роботи проведено огляд та аналіз платформи для створення майбутньої інформаційної системи, в результаті якого вибрано OS Android. Також проведено огляд та порівняння аналогічних програмних рішень з платформи додатків Google Play. Оскільки в розглянутих додатках відсутнє групове додання товарів до списку покупок, поставлено ціль – розробку групового додавання продуктів.

## СИСТЕМНИЙ АНАЛІЗ

### 1.5. Побудова дерева проблем та дерева цілей

Щоб побудувати систему, яка буде потрібна користувачу необхідно провести системний аналіз предметної області, визначити основні проблеми. На основі визначених проблем будуємо дерево проблем, ставимо цілі, що вирішуватимуть задані проблеми та з них будуємо дерево цілей.

#### 1.5.1. Дерево проблем

Дерево проблем – це ієрархічна структура, що складається з основної проблеми (кореня) та її складових (гілки дерева). Кожну підпроблему можна поділити на декілька менших складових, залежно від рівня деталізації дерева[6].

Дерево проблем інформаційно-довідкової системи (див. рис. 3.1) для закупівлі продуктів містить в собі три основні гілки, кожна з яких має декілька підпроблем.

1. Непродуманий інтерфейс. Ця гілка об'єднує проблеми пов'язані з налаштуванням інтерфейсу та стандартними доступними функціями:
  - 1.1 Непродуманий словник товарів. В більшості розглянутих вище систем є власний словник товарів, проте основна проблема його в тому, що користувач не може додати нові «свої» продукти до власного словника. До прикладу, якщо в словнику є хліб, користувач не може налаштувати словник, щоб мати «хліб з висівками» та «хліб кукурудзяний».
  - 1.2 Категоризація товарів. Для зручного користування списком варто розробити категоризацію товарів, адже сортування за алфавітом зовсім не доречне коли йдеш до супермаркету. Тут значно корисніше буде поділити товари за відділами магазину, наприклад, молочні товари, м'ясо, алкоголь тощо.
  - 1.3 Персоналізація словника та категорій товарів. Як згадано вище, користувачу має бути зручно налаштовувати свій додаток під себе і свої потреби, в тому числі і категорії товарів. Приміром, якщо він

вибирає купувати деякі товари в супермаркеті, а інші на базарі, можна створити поділ за цим критерієм.

2. Відсутність зручного поширення списку.

2.1 Відсутність можливості копіювання списку товарів як тексту. В жодному з розглянутих аналогічних програмних рішень не знайдено дану функцію, що може бути достатньо зручно, адже не завжди користувачу необхідно надіслати список через один з доступних додатків на телефоні.

2.2 Відсутність можливості надсилання списку через найпопулярніші месенджери. Корисна функція, що спрощує поширення списку для друзів чи родини.

3. Відсутність групового додавання товарів до списку. Проблема є явною, оскільки в більшості розглянутих аналогів список створюється за допомогою додавання товарів по чергово по одному за раз.

3.1 Відсутність смітника. Досить зручно мати смітцеву корзину, звідки з часом можна витягнути список, що був створений, наприклад, для минулого пікніка і використати його повторно для планованого.

3.2 Відсутність книги рецептів. Корисна функція для тих, хто готує: рецепт створюється одного разу, потім з списку рецептів до списку покупок додаються всі інгредієнти.

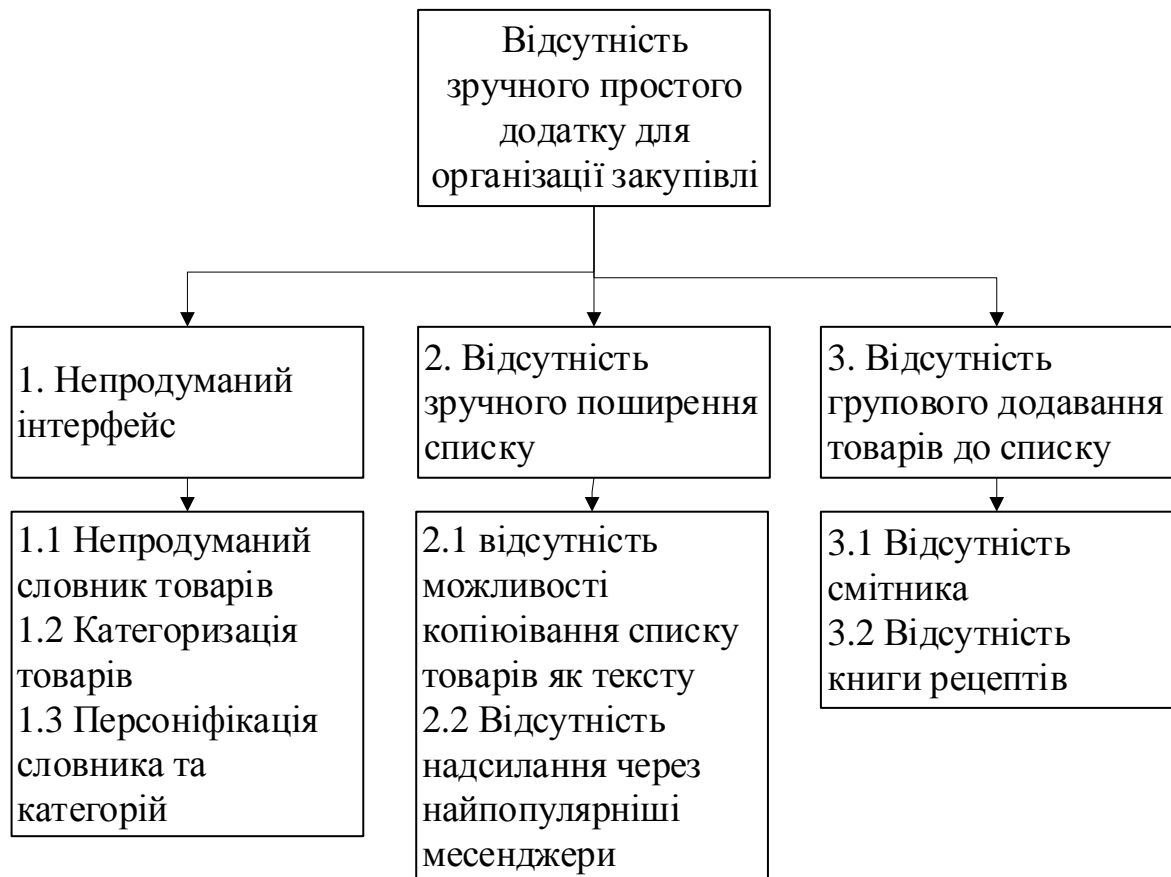


Рис.1.8 Дерево проблем системи.

Побудоване дерево проблем показує наявні проблеми, відсутні функції та не вирішені задачі, що мають бути розглянуті при розробці системи. Для побудови дерева проблем використано вище наведені недоліки аналогічних програмних рішень.

#### 1.5.2. Дерево цілей

Дерево цілей – це ієрархічна деревоподібна структура, що отримується поділом загальної цілі на підцілі, які в свою чергу теж можна розділити на менші підцілі, функції тощо. Графічно дерево зображають «кровою донизу», а головну ціль поміщають на найвищий рівень. Перевагою побудови дерева цілей при системному аналізі вважають можливість поділу великої неосяжної цілі на простіші завдання, які можна розв’язати відомими методами[6].

Дерево цілей зазвичай відображає дерево проблем, адже кожен знайдений проблему варто розглянути при розробці інформаційної системи.

Дерево цілей для інформаційно-довідкової системи для закупівлі продуктів складається з трьох основних цілей, що мають ряд підцілей(див. рис. 3.2).

1. Створення зручного інтерфейсу.
  - 1.1 Створення словника товарів. В словник товарів вноситимуться основні товари.
  - 1.2 Створення стандартних категорій товарів. Стандартна категоризація означає поділ за відділами магазину.
  - 1.3 Персоналізація категорій та словника товарів. Можливість додавання власних продуктів та спеціалізованих категорій товарів.
2. Створення зручного поширення списку.
  - 2.1 Створення можливості копіювання списку як тексту.
  - 2.2 Можливість поширення списку через найпопулярніші месенджери.
3. Розробка групового додавання товарів до списку.
  - 3.1 Розробка смітника. Аналог смітцевого відра в ОС Windows, перед видаленням певного списку його можна перемістити до смітника і витягнути звідти, якщо виникне потреба.
  - 3.2 Розробка книги рецептів. Список інгредієнтів з рецепту можна одним натисканням додати до списку покупок, виключає можливість забути внести в список найважливіший інгредієнт.

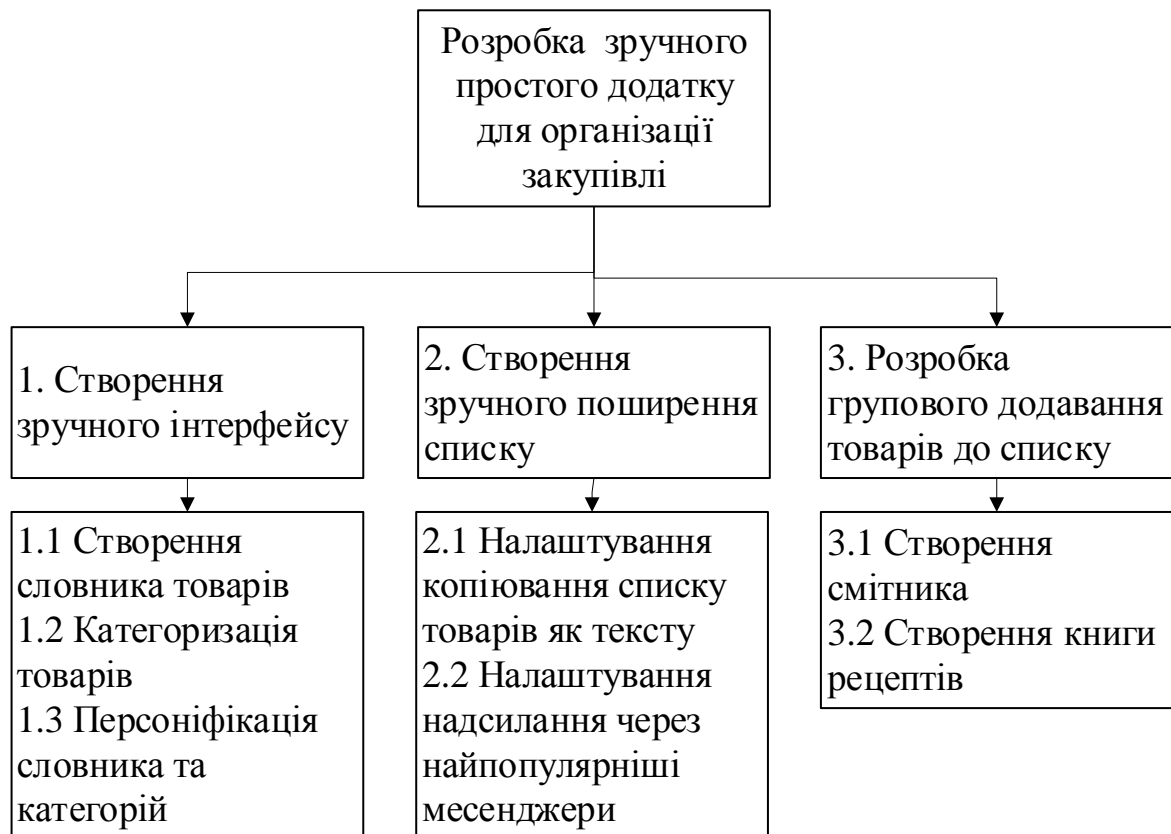


Рис.1.9 Дерево цілей.

Розроблене дерево цілей відображає поставлені цілі при створенні інформаційно-довідкової системи для закупівлі продуктів. Дерево формують три гілки: створення зручного інтерфейсу, створення зручного поширення списку та розробка групового додавання товарів до списку. Після досягнення наведених цілей система вважається успішно розробленою.

## 1.6. Аналіз та вибір методів, алгоритмів та засобів розв'язання задачі

### 1.6.1. Огляд та аналіз мов програмування для Android-додатків.

Варто зазначити, що програмувати додатки для обраної платформи Android можна різними мовами. В цьому пункті розглянемо Java, C/C++, C# та Basic, їх переваги та недоліки.

#### 1.6.1.1. Java.

Більшість додатків, що працюють на платформі Android розроблені мовою Java, в більшості тому що на офіційному сайті Android пропоновано Java як рекомендовану мову програмування[7].

Java – об'єктно-орієнтована мова програмування, що розробляється компанією Oracle(раніше Sun Microsystems). В офіційній реалізації Java-

програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною Java(JVM) для конкретної платформи, зокрема Android.

Основні переваги Java відносно інших мов програмування:

- проста, легка в вивченні мова;
- об'єктно-орієнтована мова, що дозволяє створювати модульні програми і декілька разів використовувати уже написані компоненти;
- інтерпретується на Android-пристрої за допомогою JVM[8];
- офіційно підтримується Google для створення Android-додатків;
- є велика кількість документації, літератури та велика спільнота, що допоможе у вирішенні проблем.

Недоліки використання Java:

- необхідно мати базові поняття про Gradle, мову розмітки XML та Android Manifest.

#### 1.6.1.2. C/C++.

C/C++ -- це дві мови програмування, що пов'язані між собою. Початково після розробки мови C++ її називали мовою «C з класами». Ця мова була досить популярна в дев'яностих роках минулого століття, зараз вона використовується для складних додатків, що вимагають більшого контролю над пам'яттю(здійснюється за допомогою вказівників) а також для embedded-програмування[9]. Для програмування на Android в Android Studio існує NDK.

Щодо програмування Android-додатків цими мовами можна відмітити такі переваги:

- використання бібліотек C/C++;
- контроль над пам'яттю, що дозволяє «видушити» більше ресурсів для свого додатку(наприклад при створенні ігор).

Проте недоліків значно більше:

- складно налаштувати та встановити;
- більше помилок при програмуванні;
- менш гнучка;

- контроль над пам'яттю зобов'язує розробника слідкувати за витокami та звільненням пам'яті.

#### 1.6.1.3. C#.

C# -- це об'єктно-орієнтована мова програмування, що розроблена компанією Microsoft на заміну C/C++, спрямована на те, щоб поєднати в собі силу C++, простоту Visual Basic та читатися трохи легше ніж Java. Подібно до Java, в C# розробнику не потрібно турбуватися про такі речі, як витік пам'яті та звільнення пам'яті.

Для розробки додатків використовується в поєднанні з ігровим двигуном Unity – для розробки ігор, або використовуючи Xamarin в Visual Studio. З використанням Xamarin можна створити кросплатформений додаток, що скомпілюється як на Android так і на iOS[10].

Переваги програмування на C#:

- кросплатформеність( можливість компіляції для Android та iOS).

Недоліки:

- складність розробки;
- складніший процес налаштування та відлагодження роботи програми;
- більший розмір програм відносно аналогів розроблених за допомогою Java та Swift.

#### 1.6.1.4. Visual Basic.

Visual Basic – це мова програмування, що створена Microsoft, що достатньо проста для вивчення початківцю. Створення додатків для Android на Visual Basic, відбувається за допомогою Basic4android.

Basic4android - це комерційна підгрупа Visual Basic на основі Windows, яка має IDE, графічний візуальний редактор, вбудовану документацію, покроковий відладчик і підтримку об'єктів. Basic4android компілює базовий вихідний код для Dalvik-сумісного, віртуального байт-коду. Basic4android також допомагає створювати файли APK для простої інсталяції на пристрої Android користувача. Basic4android підтримує Android версії 1.6 і вище[11].



Переваги:

- простота вивчення мови та створення простих додатків;
- компіляція в APK-файл;

Недоліки:

- не підтримується Android Studio, Unity, Visual Studio;
- середовище Basic4android платне;
- складно створити схожі до Java інтерфейси;
- складність підключення більшості бібліотек.

#### 1.6.1.5. Вибір мови програмування.

Зважаючи на розглянуті вище переваги та недоліки мов програмування Java, C/C++, C#, Visual Basic, можна зробити висновок, що найоптимальнішим вибором буде мова програмування Java.

C# Xamarin розробка підходить для невеликих додатків, та при необхідності розробки для декількох платформ. Вимагає додаткового вивчення документації та базового досвіду програмування для мобільних пристроїв. Visual Basic може бути досить простим засобом розробки Android-додатків, проте він має ряд суттєвих недоліків, оскільки не є офіційно підтримуваною мовою. C/C++ має перевагу в контролі пам'яті, що одночасно стає її недоліком, оскільки розробнику необхідно пам'ятати та слідкувати за перевантаженням чи звільненням пам'яті.

Java є офіційною мовою програмування на Android, вона підтримується Android Studio та більшість бібліотек та додаткових сервісів Google пропонує підключати за допомогою Android Studio та мови програмування Java. Проста та зрозуміла мова програмування, що дає можливість створити додаток для платформи Android легко та просто. В розробці при вирішенні проблемних питань в нагоді стане одна з найбільших спільнот – спільнота Java Android розробників.

#### 1.6.2. Програмування інтерфейсів за допомогою XML.

Для розробки інтерфейсів буде використано мову XML(Extensible Markup Language) – стандарт побудови розмітки за допомогою простих тегів,

структурованих ієрархічно даних, що був запропонований Wide Web Consortium для обміну даними між різними застосунками.

Android надає розробникам зручний довідник XML-елементів для класів View і їх підкласів. Розробники мають можливість гнучкого інтерфейсу програми, до прикладу, можна оголосити макети за замовчуванням, що включають елементи екрану, а потім додати код, що буде змінювати стан об'єктів під час виконання. В Android Studio є можливість попереднього перегляду створеного файлу XML – для цього достатньо відкрити файл XML та перейти на вкладку Layout (Макет). При відлагодженні макетів можна використати Hierarchy Viewer, за допомогою якого можна переглянути значення властивостей, а також повністю промальовані макети під час налагодження програми на пристрої або в емуляторі.

Перевага оголошення призначеного для користувача інтерфейсу в файлі XML полягає в тому, що таким чином можна ефективно відокремити інтерфейс додатку від коду, що управляє його поведінкою. Враховуючи, що всі описи користувацького інтерфейсу знаходяться поза межами коду, є можливість змінити інтерфейс без зміни і компіляції вихідного коду, до прикладу створити різні файли XML-макетів для різних мов. Також необхідно зазначити, що оголошення макетів в XML спрощує візуалізацію структури користувацького інтерфейсу[12].

### 1.6.3. СКБД SQLite.

SQLite – полегшена система керування реляційними базами даних, що поширюється безкоштовно, а з 2018 року є рекомендованим форматом зберігання структурованих даних від Бібліотеки Конгресу США[13].

API, які потрібно використовувати для бази даних на Android, доступні в пакеті android.database.sqlite. Одним з основних принципів баз даних SQL є схема, що відображена в операторах SQL, які використовуються для створення бази даних. Для створення схеми бази даних можна використати контрактний клас. Android зберігає базу даних у приватній папці програми – дані захищені,

оскільки за умовчанням ця область не доступна для інших програм або для користувача.

Клас `SQLiteOpenHelper` містить корисний набір API для керування вашою базою даних. Коли ви використовуєте цей клас для отримання посилань на вашу базу даних, система виконує потенційно тривалі операції створення та оновлення бази даних тільки в разі потреби, а не під час запуску програми. Все, що потрібно зробити – це викликати `getWritableDatabase ()` або `getReadableDatabase ()`.

Щоб скористатися `SQLiteOpenHelper`, треба створити підклас, який перебиває методи зворотного виклику `onCreate ()` і `onUpgrade ()`. Для запису даних слугує метод `insert()`, для читання – `query()`, для видалення – `delete()`, та для оновлення даних використовується `update()`[14].

#### 1.6.4. Середовище розробки Android Studio.

Android Studio – це офіційне інтегроване середовище розробки (IDE) для платформи Android. Середовище побудовано на основі IntelliJ IDEA від JetBrains. Крім потужного редактора коду IntelliJ і інструментів розробників, Android Studio пропонує ще більше функцій, які підвищують продуктивність під час створення програм для Android, таких як:

- гнучка система побудови на основі Gradle;
- швидкий і багатофункціональний емулятор;
- єдине середовище, де можна розробляти для всіх пристроїв Android;
- миттєве виконання – можливо змінити додаток без створення нового APK;
- шаблони коду та інтеграція GitHub допоможуть створити власний особливий дизайн програми та імпортувати зразок коду;
- широкі інструменти тестування;
- інструменти Lint для охоплення продуктивності, зручності, сумісності версій та інших проблем;
- підтримка C ++ і NDK;

- вбудована підтримка Google Cloud Platform, що дозволяє легко інтегрувати Google Cloud Messaging і App Engine.[15]

Тож Android Studio є оптимальним середовищем розробки для додатків платформи Android.

### 1.7. Структурна схема системи з урахуванням інформаційних потоків.

Структурна схема системи показує модулі системи та потоки даних, що циркулюють між ними. Структурна схема розроблюваної системи представлена на рисунку 3.3, та складається з трьох модулів: Model – модуль управління даними, View – відображення даних, графічний інтерфейс користувача та Control, що пов'язує ці модулі між собою, отримує вхідні дані та перетворює їх на команди для Model чи View.

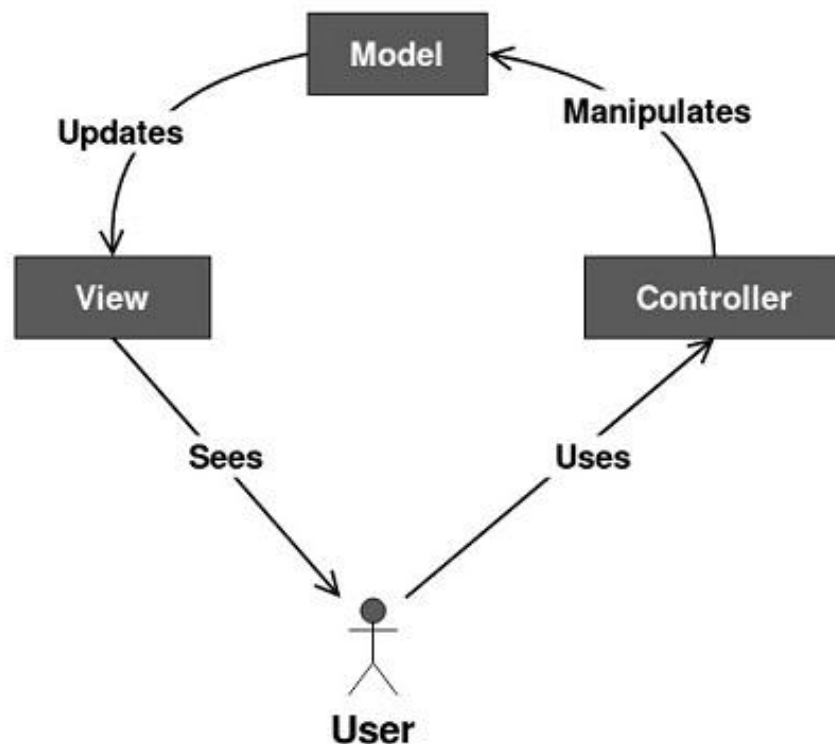


Рис.1.10 Структурна схема системи.

Така структура системи дозволяє розмежувати дані, що зберігаються базі даних та графічний інтерфейс, це дозволяє змінювати інтерфейс без втручання в роботу ядра системи, а також змінювати модель зберігання даних без зміни інтерфейсу. Загалом, така структура надає можливість використання окремих

компонентів програми в інших програмах в майбутньому а також побудувати систему так, щоб полегшити подальші зміни та розширення.

### **Висновок до розділу**

В даному розділі магістерської кваліфікаційної роботи проведено системний аналіз інформаційно-довідкової системи для закупівлі продуктів а саме побудовано дерево проблем та дерево цілей. Також проведено огляд, аналіз та вибір мови програмування: серед Java, C/C++, C#, Visual Basic обрано мову програмування Java як офіційно підтримувану Google для розробки Android-додатків. Описано вибрані засоби розробки інтерфейсів та бази даних, а також середовище програмування Android Studio. Також в цьому розділі розроблено та представлено структурну схему системи, що складається з трьох основних модулів.

## **РОЗРОБКА ТА ВІДОБРАЖЕННЯ АЛГОРИТМІВ РОБОТИ КОМПОНЕНТІВ ТА СИСТЕМИ**

### **1.8. Засоби проектування інформаційних систем.**

#### **1.8.1. Enterprise Architect**

Sparks System Enterprise Architect – набір UML інструментів для системного та бізнес аналізу, що може використовуватися на всіх стадіях розробки програмного забезпечення: формулювання вимог, аналіз, розробку, тестування та підтримку.

Інтегрована платформа для моделювання. Повний цикл моделювання для бізнес- та IT-систем, програмного забезпечення та системної інженерії, розробка в реальному часі та вбудована розробка. Вбудовані можливості управління вимогами дають змогу відстежувати нові специфікації більш ніж двадцяти відомих відкритих стандартів. Надає можливість генерації документації, генерації коду та реверс-інжинірингу[16].

#### **1.8.2. AllFusion Process Modeler**

AllFusion Process Modeler – потужний інструмент моделювання, що дозволяє проводити опис, аналіз, моделювання документування та реорганізацію складних бізнес-процесів. Займає одне з лідируючих місць в своєму сегменті ринку, в даний час випускається компанією Computer Associates.

Включає три стандартні методології: IDEF0 (функціональне моделювання), DFD (моделювання потоків даних) і IDEF3 (моделювання потоків робіт). Кожна з них може бути виконана окремо за допомогою BPwin, але їх сукупність дає аналітику повну картину предметної області клієнта.

Дозволяє удосконалювати бізнес-процеси, формулюючи і визначаючи альтернативні реакції на дії ринку, швидко виключати непродуктивні операції. Неefективні, неекономічні або надлишкові операції можуть бути легко виявлені і поліпшені, змінені або зовсім виключені – відповідно до цілей компанії[17].

### 1.8.3. AllFusion ERwin Data Modeler.

AllFusion Erwin Data Modeler – засіб проектування та документації баз та сховищ даних. Дозволяє проектувати багаторівневі архітектури, функціонал системи містить фізичну та логічну моделі, можна також створити фізично-логічну модель. Середовище підтримує генерацію скриптів створення бази даних для найпопулярніших реляційних СКБД, а також спеціальні методики формування баз даних, наприклад, за схемою зірки(star schema)[18].

### **1.9. Діаграми, що описують функціонування системи та її окремих блоків.**

Для проектування інформаційної системи буде використано методологію SADT – методологію структурного аналізу та проектування. Розроблена в 60-х роках минулого століття Дугласом Россом, на основі SADT розроблено стандарти IDEF. Сам процес моделювання може бути поділений на декілька етапів: опитування експертів, створення діаграм та моделей, поширення документації, оцінка адекватності моделей та їх придатності до подальшого використання. Сутність структурного підходу полягає в розбитті системи на функціональні підсистеми, що діляться на функції, ті на підфункції, і, зрештою, отримуємо список формальних процедур, що легко виконати. При цьому система зберігає цілісне представлення, в якому всі складові компоненти взаємопов'язані.

Базові принципи:

1. Розділай та володарюй – вирішення складних проблем шляхом розбиття їх на велику кількість менших задач, легких для розуміння та вирішення.
2. Принцип ієрархічного упорядкування – принцип організації складових частин системи у ієрархічні структури, з додаванням деталей на кожному рівні ієрархії.
3. Принцип абстрагування – виділення істотних компонентів системи та відволікання від несуттєвих.

4. Принцип формалізації – полягає в необхідності суворого методичного підходу до вирішення проблем.
5. Принцип несуперечливості – обґрунтованість і узгодженість елементів.
6. Принцип структурування даних – дані повинні бути структуровані та організовані.

Найчастіше використовуються такі діаграми:

- IDEF0 – методологія функціонального моделювання та графічного опису процесів, що призначена для розбиття та опису бізнес-процесів.
- DFD – діаграма потоків даних, методологія структурного аналізу, що має на меті представлення потоків даних в інформаційній системі.
- ERD – діаграма сутність-зв'язок, дані представлені в вигляді сутностей та зв'язків між ними, сутності можуть мати різні властивості(атрибути), що їх характеризують. Зазвичай використовується для проектування баз даних.

Для побудови згаданих вище моделей використано AllFusion Process Modeller 7, який призначений для розробки діаграм функціонального моделювання(IDEF0) та діаграм потоків даних(DFD)та ERwin Data Modeller, в якому зручно розробляти ERD діаграми.

#### 1.9.1. Побудова діаграм IDEF0.

Кожна діаграма IDEF0 складається з:

1. Функцій, що представлені графічно в вигляді прямокутних блоків;
2. Управління(контроль) – стрілки, що входять в верхню частину блоку, визначають за допомогою яких правил відбувається робота.
3. Механізм – стрілки, що входять в нижню частину блоку, визначають хто і(або) що виконує роботу функціонального блоку.
4. Входи – стрілки що входять в блок зліва та визначають вхідні дані роботи.



## 5. Виходи – результат виконаної роботи, стрілки, що виходять справа функціонального блоку[19].

На початку розробки діаграми IDEF0 будують контекстну діаграму. Вона складається лише з одного функціонального блоку, та стрілок контролю, механізмів, входів та виходів. На рисунку 4.1 представлено контекстну діаграму IDEF0 для інформаційно-довідкової системи для закупівлі продуктів.

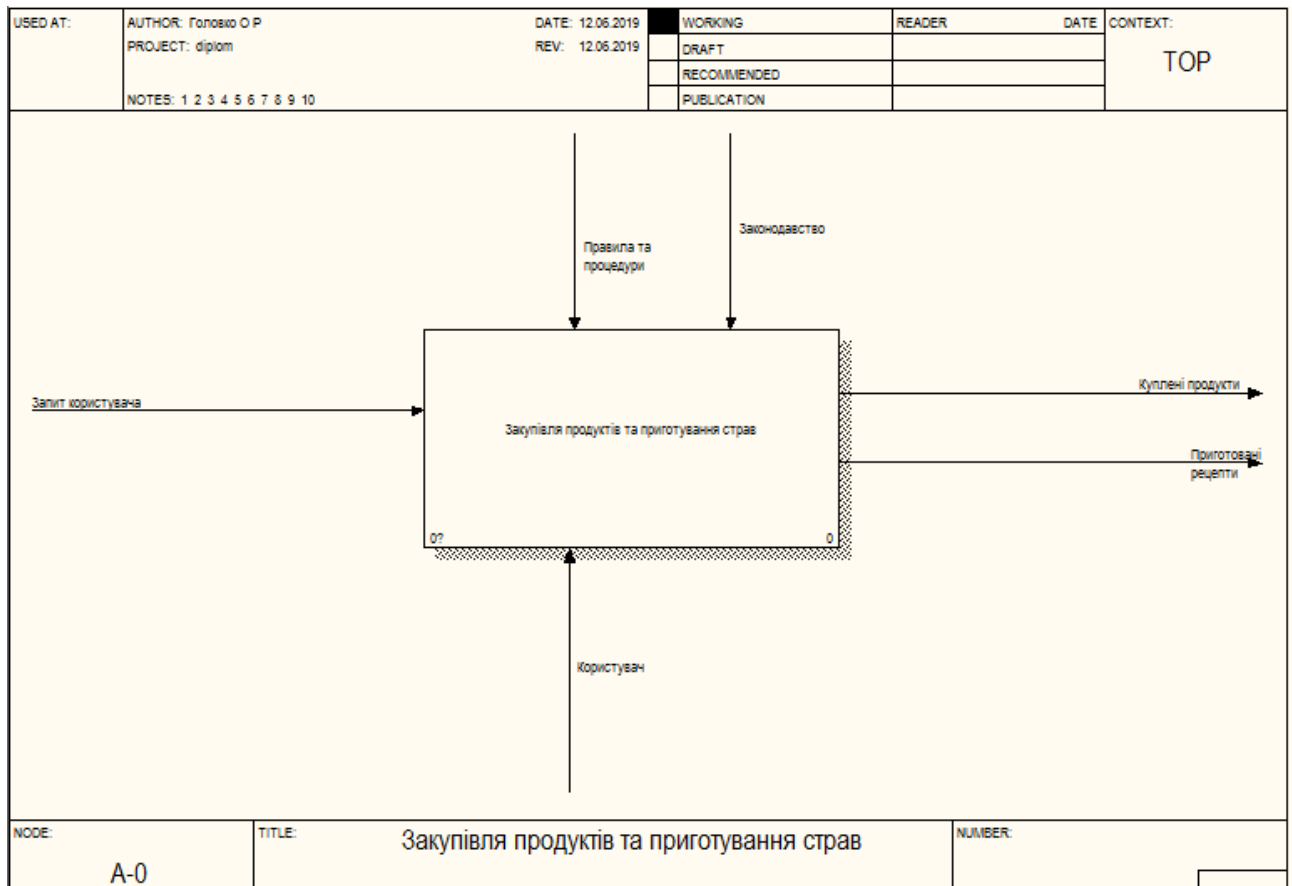


Рис.1.11 Контекстна діаграма IDEF0.

Оскільки дана діаграма нульового рівня на ній присутня робота «Закупівля продуктів та приготування страв», на вході якої запит користувача, на виході – куплені продукти та приготовані рецепти, робота відбувається під контролем правил та процедур інформаційної системи та українського законодавства, виконує дану роботу користувач.

Розглянемо декомпозицію діаграми нульового рівня до діаграми першого рівня декомпозиції. На цьому етапі варто зазначити, що жодна з стрілок не має «зникнути» або «з'явитися нізвідки». Ми розбиваємо роботу A0 на декілька простіших робіт, проте на вході та виході мають бути ті самі дані, це

означатиме, що система структурована та цілісна. Діаграма першого рівня декомпозиції представлена на рис. 4.2.

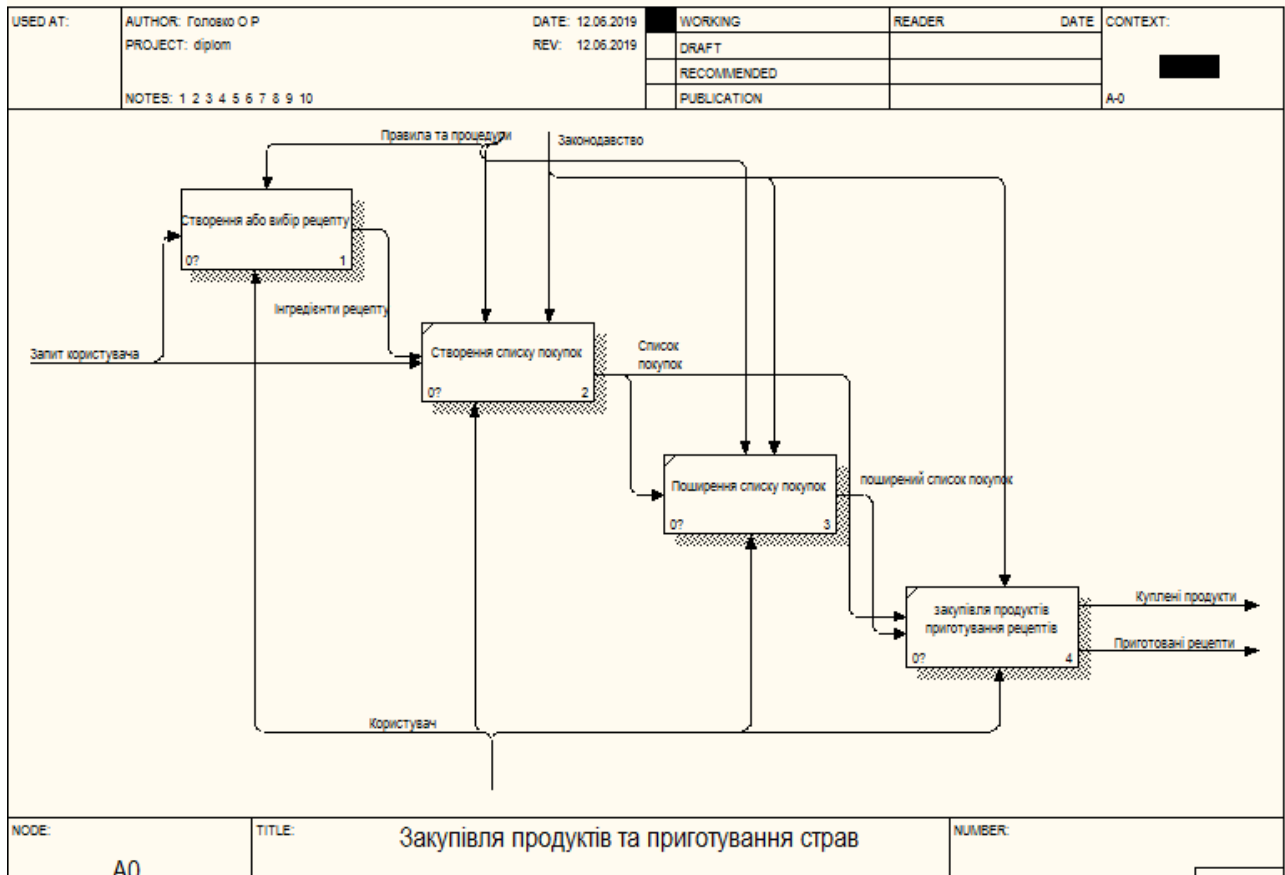


Рис.1.12 Діаграма декомпозиції першого рівня.

З рисунку видно, що роботу «Закупівля продуктів та приготування страв» нульового рівня розбито на чотири роботи першого рівня декомпозиції:

1. Пошук або створення рецепту. Робота що відповідає за створення рецептів або пошук уже створених в базі даних. Контроль – Процедури та правила, механізм – користувач, на вході – запит користувача, на виході – інгредієнти рецепту.
2. Створення списку покупок. Робота відповідає за створення списку покупок, він може бути створений вручну, або автоматичним додаванням потрібних інгредієнтів з обраних рецептів. На вході – запит користувача та інгредієнти з обраних рецептів, на виході – готовий список покупок, контроль за допомогою законодавства та правил і процедур, механізм – користувач.

3. Поширення списку покупок. Виконує поширення списку покупок доступним чином. На вході список покупок – на виході поширений список покупок, контроль – законодавство та правила і процедури, роль механізму виконує користувач.
4. Закупівля продуктів та приготування страв. Робота з закупівлі продуктів та приготування страв, виконує користувач на основі списку покупок та обраних рецептів.

Оскільки методологія IDEF0 передбачає якомога глибшу діаграму, розіб'ємо роботу «Створення або вибір рецепту» на п'ять підфункцій(див. рис. 4.3).

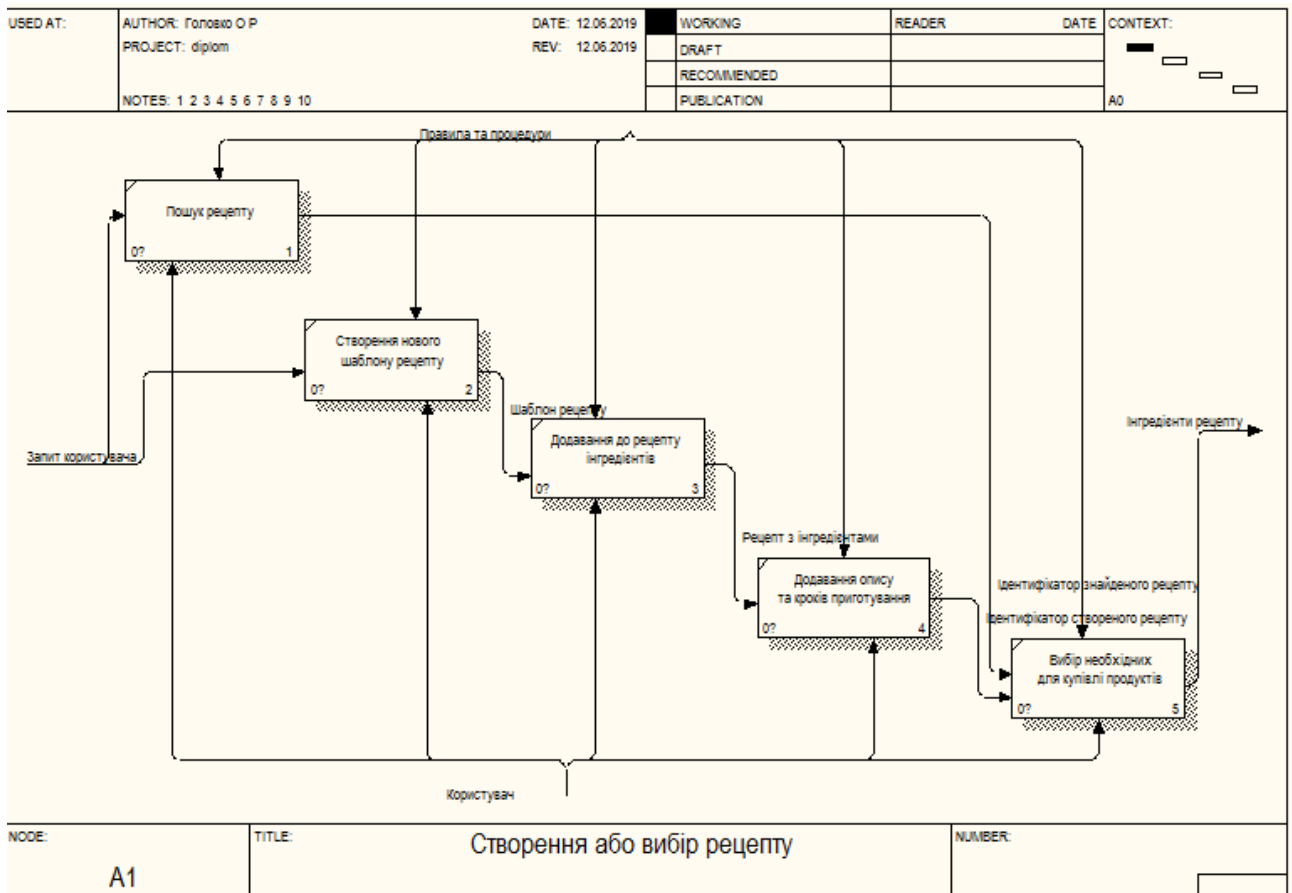


Рис.1.13 Діаграма декомпозиції другого рівня роботи «Створення або вибір рецепту».

Дана діаграма складається з п'яти робіт:

1. Пошук рецепту. Здійснюється користувачем, на вході запит, на виході рецепт що відповідає вимогам користувача.

2. Створення нового шаблону рецепту. Користувач ініціює створення нового рецепту.
3. Додавання інгредієнтів до рецепту. Отримавши на вході шаблон, до нього додаються інгредієнти.
4. Додавання опису та кроків приготування. До отриманого на вході рецепту, користувач додає кроки приготування та опис. На виході – готовий новостворений рецепт.
5. Вибір необхідних для купівлі продуктів. На вхід отримує рецепт(створений або знайдений), користувач обирає продукти(інгредієнти), які на даний момент необхідно придбати. Тож, на виході отримуємо список інгредієнтів до придбання.

#### 1.9.2. Відображення потоків даних за допомогою DFD-діаграми.

Діаграми потоків даних використовуються для опису зовнішніх відносно системи джерел та адресатів даних, сховища та бази даних, логічні функції та потоки даних між вищезазначеними елементами в системі. Як і в діаграмах IDEF0, кожна робота діаграми потоків даних може бути розділена на кілька менших в межах однієї нотації[20].

Зовнішня сутність може бути матеріальним предметом або людиною, що надає системі ззовні дані, або отримує їх від системи. Робота може бути визначена як на рівні контекстної діаграми, так і в вигляді підфункцій деякого рівня декомпозиції.

На рисунку 4.4 зображено діаграму потоків даних системи, що розробляється, першого рівня декомпозиції.

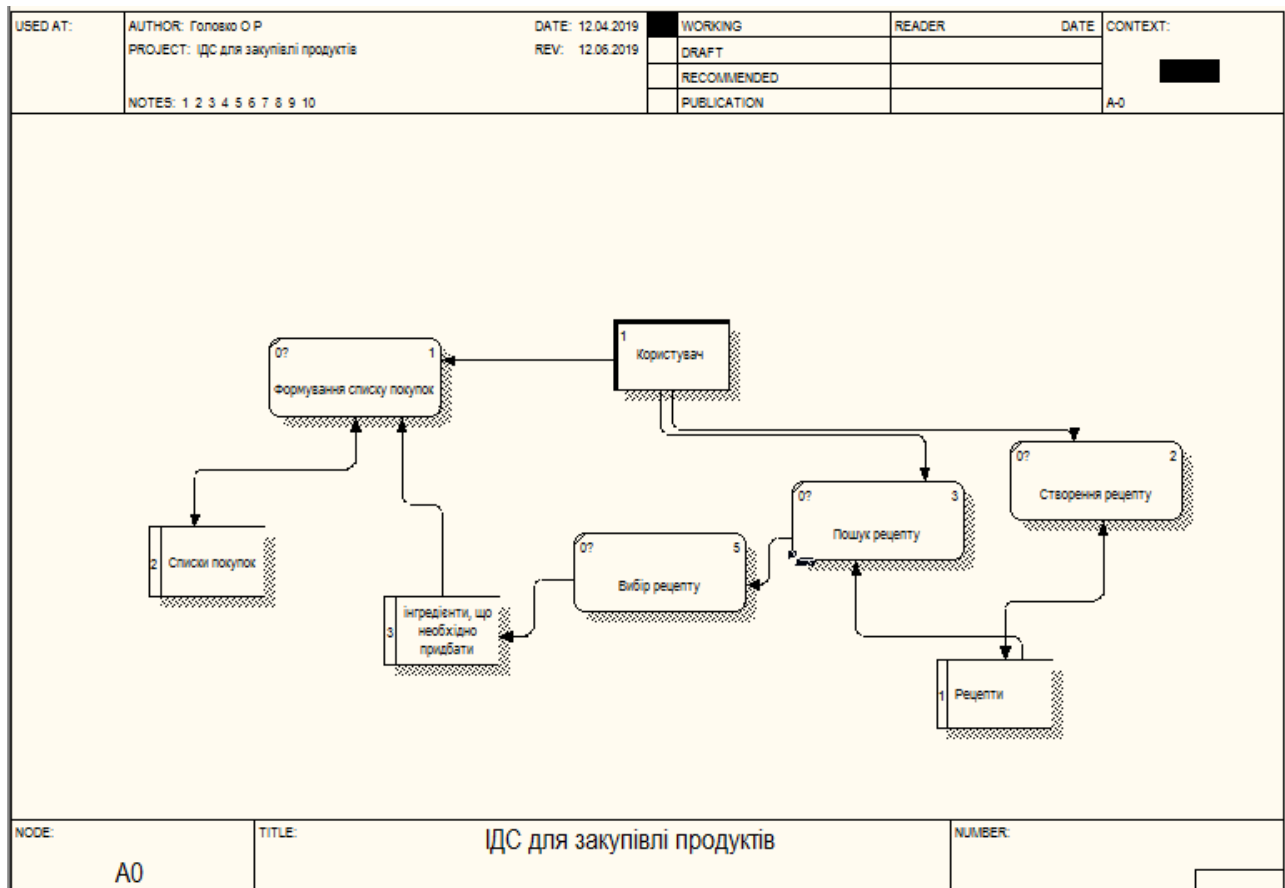


Рис.1.14 Діаграма потоків даних системи.

Суть створення діаграми потоків даних – відобразити бази даних (інгредієнти, що необхідно придбати, рецепти, списки покупок) та зовнішні сутності (користувач) для розуміння звідки інформація береться і як вона рухається в системі.

### 1.9.3. ER діаграма бази даних.

Для розробки системи необхідно створити місце де зберігатимуться її дані, тому в даному підпункті розглянемо побудову схеми бази даних інформаційної системи, що розробляється. На діаграмі зображають сутності (майбутні таблиці БД) та зв'язки між ними. На цьому етапі зв'язки можуть бути один-до-багатьох, один-до-одного та багато-до-багатьох, який при побудові бази даних замінюється проміжними таблицями [21]. Схема бази даних представлена на рис. 4.5.

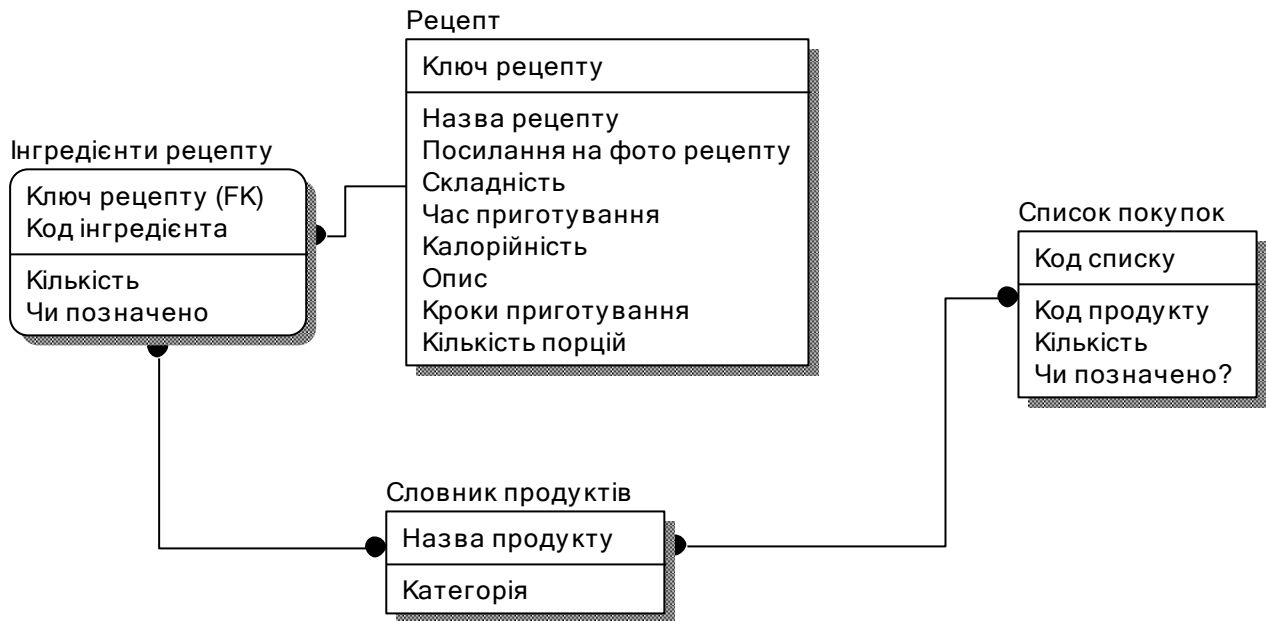


Рис.1.15 Схеми бази даних системи.

Схеми складається з чотирьох сутностей: Рецепт, Інградієнти рецепту, Словник продуктів, Список покупок. Головна сутність – власне Список покупок, що містить в собі поля: Код продукту(ПК), Назва продукту(FK) з словника продуктів, Кількість та Чи позначено? – відмітку про те, чи продукт позначено як куплений. В Словнику продуктів є два атрибути: Назва продукту(ПК), та Категорія продукту. Між собою ці сутності пов’язані зв’язком багато-до-багатьох, адже в список продуктів можуть входити багато продуктів з словника продуктів, і продукти з словника продуктів можуть використовуватися в багатьох списках продуктів. З Словником продуктів також поєднана ще одна сутність – Інградієнти рецепту. До сутності Інградієнти рецепту належать такі атрибути: Ключ рецепту – вторинний ключ, під’єднує інградієнти до рецепту; Ключ інградієнта – первинний ключ, Кількість інградієнта, що необхідна для рецепту та Чи позначено – атрибут, що відповідає за відмітку необхідних для покупки інградієнтів. Зв’язок між цими сутностями також багато-до-багатьох.

Сутність Рецепт має атрибути: Ключ рецепту – унікальний ідентифікатор та первинний ключ, Назва рецепту, Кількість порцій, Складність приготування, Калорійність, Час приготування, Опис рецепту та Кроки приготування. Ця сутність має зв’язок один-до-багатьох з Інградієнтами рецептів, оскільки до

одного рецепту може бути додано багато інгредієнтів, а в різних рецептах інгредієнти різні.

### **1.10. Use-case діаграма.**

Use-case діаграма – діаграма прецедентів( також перекладають діаграма випадків використання) – Uml-діаграма, на якій зображено відношення між акторами та прецедентами в системі. Набір усіх прецедентів системи визначає функціональні вимоги, за допомогою яких можна сформулювати технічне завдання[22].

Суб'єкт – система, що розглядається. Актори – користувачі або інші системи, що взаємодіють з системою, що розробляється.

Між прецедентами можуть бути два типи відношень:

- Відношення включення(include) – в деякій точці базового прецеденту як його складова використовується частина іншого прецеденту. Прецедент, що включається не використовується автономно.
- Відношення розширення(extend) – визначає необов'язкову поведінку системи та може використовуватися для моделювання окремих субпотоків, що виникають лише за певних обставин.

Діаграма випадків використання для інформаційно-довідкової системи для закупівлі продуктів представлена на рис. 4.6.

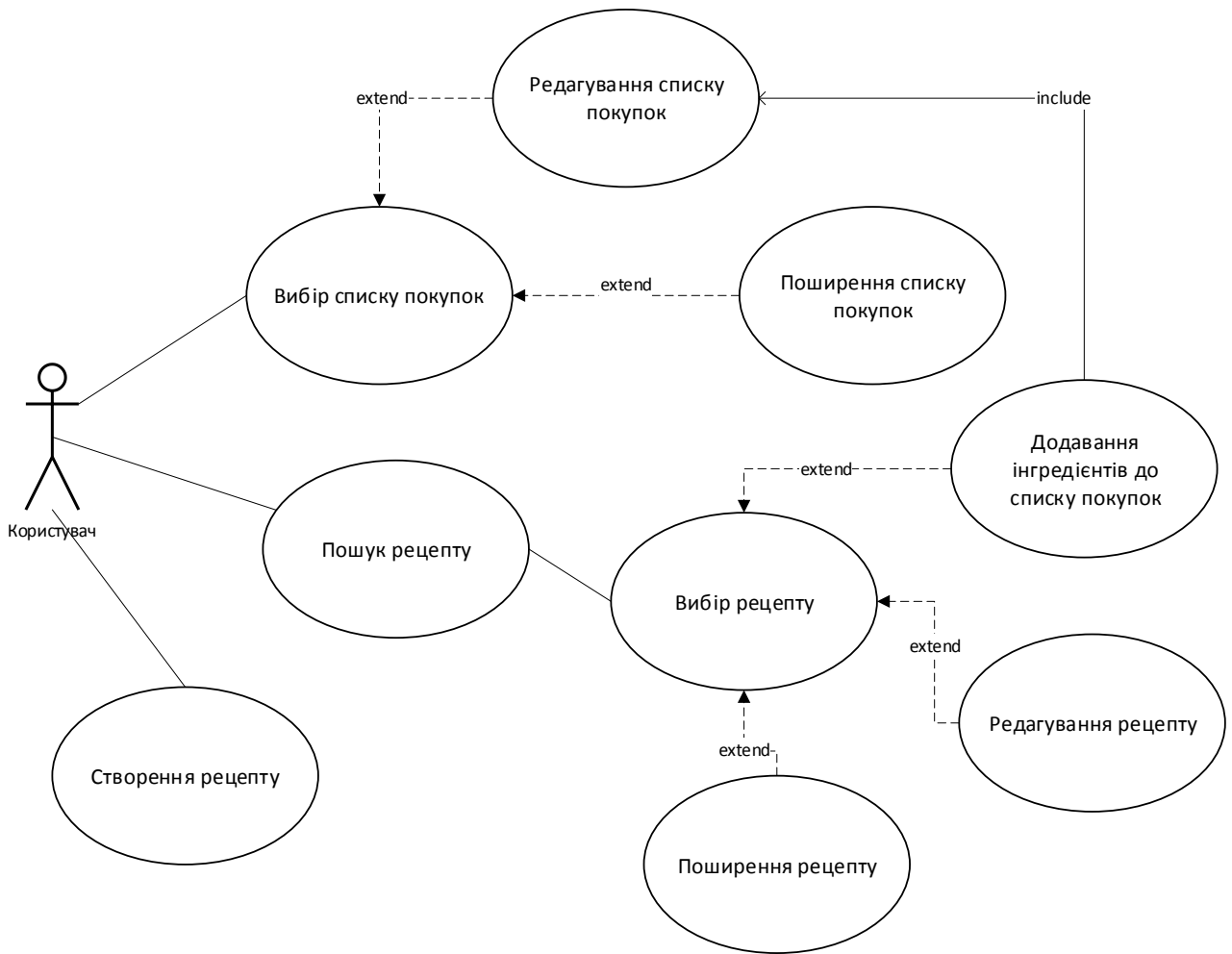


Рис.1.16 Діаграма варіантів використання.

Враховуючи, що дана система призначена для збереження особистих рецептів та списків покупок, в системі є лише один актор – користувач. Він може створити рецепт та список покупок, знайти рецепт в власній базі даних, перейти на сторінку рецепту та поширити його чи відредагувати. В користувача є можливість обрати з інгредієнтів рецепту ті, які на даний момент необхідно купити та відправити їх в поточний список покупок. Список покупок можна редагувати: додавати, видаляти, редагувати елементи та їх кількість, сортувати тощо. Список покупок також можна поширити зручним для користувача способом.

### Висновок до розділу

Даний розділ магістерської кваліфікаційної роботи присвячений розробці та відображенню роботи компонентів системи. Описано вибрану методологію проектування, розроблено функціональні діаграми IDEF0 нульового, першого



та другого рівнів декомпозиції, що дозволяє розкласти дану інформаційну систему на прості зрозумілі функціональні одиниці, які згодом будуть реалізовані. Розроблена діаграма потоків даних пояснює рух даних в системі, схема бази даних в майбутньому буде використана для розробки бази даних. Діаграма прецедентів показує функціонал системи, враховуючи обов'язкову та необов'язкову поведінку.

## РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ

### 1.11. Загальна структура програмного проекту.

За структурним підходом, необхідно спочатку розробити загальну структуру проекту, а вже потім розглядати та розробляти окремі його частини. Розглянемо діаграму компонентів системи що створюється.

Діаграма компонентів – це UML діаграма, на якій зображаються компоненти, залежності та зв'язки між ними. Компоненти можуть об'єднуватись за допомогою структурних зв'язків, які передбачають, що один з компонентів надає послуги, необхідні іншому.

Діаграма компонентів розробляється для візуалізації структури коду системи, забезпечення багатократного використання окремих фрагментів програми, а також для представлення руху даних в системі[23].

На рисунку 5.1 зображена діаграма компонентів інформаційно-довідкової системи для закупівлі продуктів.

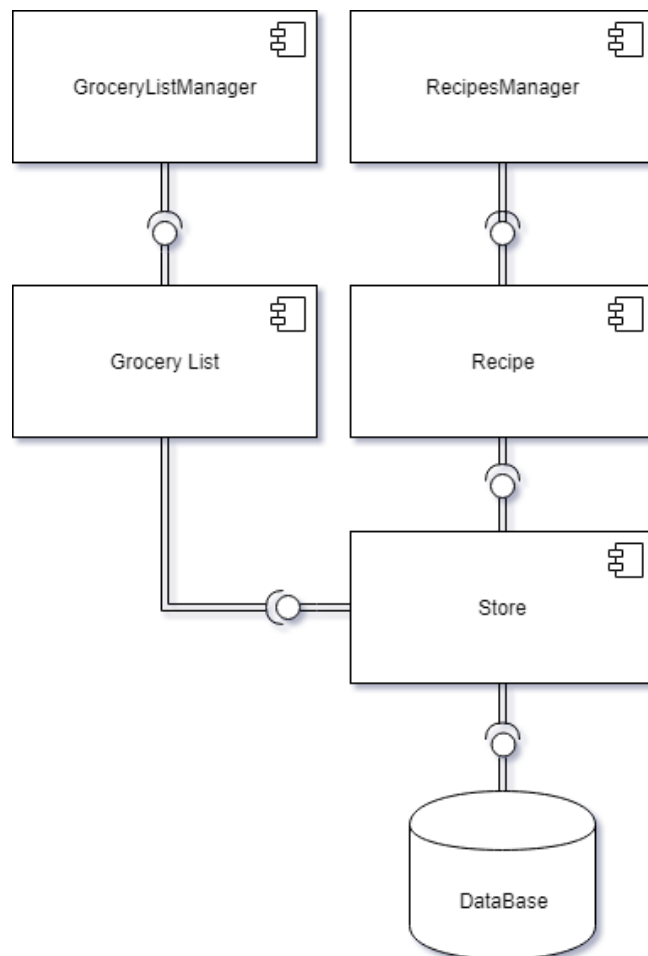


Рис.1.17 Діаграма компонентів.

Як видно з діаграми, систему можна поділити на п'ять компонентів:

- Компонент GroceryListManager відповідає за інтерфейс управління списками покупок(додавання, редагування, видалення списків покупок тощо).
- Компонент GroceryList – інтерфейс управління списком покупок: додавання нових елементів до списку покупок, видалення, редагування, поширення списку.
- Компонент RecipesManager – управляє списками рецептів: дає можливість створити новий рецепт, видалити рецепт.
- Компонент Resipe – відповідає за інтерфейс управління рецептом. До його функцій належить редагування рецепту( редагування списку інгредієнтів, редагування опису та кроків приготування)та поширення рецепту.
- Компонент Store відповідає за взаємодію вище згаданих компонентів з базою даних.

### **1.12. Розробка та опис програмних модулів.**

Оскільки основна задача при програмуванні даної інформаційної системи – розробка класів, що забезпечать взаємодію бази даних з графічним інтерфейсом, найголовнішим з згаданих вище компонент, я вважаю компонент Store. Класи цього компоненту дають можливість додавання, редагування, видалення записів з бази даних. Тож, коли розроблено цей компонент – залишається програмування інтерфейсів та налаштування виклику методів компоненту Store з певного екрану програми.

Розглянемо класи компоненту Store, їх поля та методи, що представлені в таблиці 5.1.

## Класи компоненту Store

Назва	Поля	Методи
1	2	3
GroceryStore	Logger, Context, DataBase	Open(), close(), addItem(), getItem(), changeStrokeVisibilityGroceryItem(), removeGroceryItem(), getStrokedGroceryItem(), getNonStrokedGroceryItem(), deleteAllGroceryItems(), getAllGroceryItemsCursor(), addIngredientsToGroceryList(), update()
RecipeIngredientNameStore	Logger, Context, DataBase	Open(), close(), addIngredient(), ingredientExist(), autocompleteIngredients()
RecipeIngredientStore	Logger, Context, DataBase	Open(), close(), getIngredients(), addIngredients(), removeIngredients()
ResipeStepsStore	Logger, Step_Columns, Step_Ingredients_Columns, Context, DataBase	Open(), close(), getSteps(), removeSteps(), addSteps(),

1	2	3
ResipeStore	Logger, Context, DataBase	Open(), close(), empty(), getRecipesForQuery(), exist(), resipeResponse(), replaceall(), replace()
SQLiteDB	DBName, DBVersion, IngredientTable, IngredientNameTable, GroceryTable, RecipeIngredientTable, RecipeStepsTable, RecipeTable, RecipeStepsIngredientTable	deleteDataBase(), onCreate(), onUpgrade(), createIngredientNameTable(), createGroceryTable(), createRecipeIngredientTable(), createRecipeStepsTable(), createRecipeTable(), createRecipeStepsIngredientTable()

Як видно з таблиці, цей компонент відповідає за комунікацію між базою даних та системою. Тобто запити на зміну бази даних ідуть не напряму з форми чи класів, а саме через компонент Store. Це дозволяє розробити і легко вносити зміну в структуру бази даних, додавши нові поля до таблиць та методи для їх використання. З інших програмних модулів для зміни таблиць баз даних викликаються методи цього компоненту.

### **1.13. Розробка та опис інтерфейсу користувача.**

Для користувача є важливою стабільна робота системи, але не менш важливим є зручний інтерфейс. Згідно огляду аналогічних програмних рішень, проведеного в розділі 2, можна зробити висновок, що популярність та рейтинг програми на платформі Google Play прямопропорційно залежить від зручності додатку.

На рисунку 5.2 зображено головне меню програми, яке викликається гортанням вліво або натисканням на іконку в верхній лівій частині програми. В

меню є чотири пункти: Списки покупок, Рецепти, Смітник та Налаштування. Вони відображають головний функціонал системи.

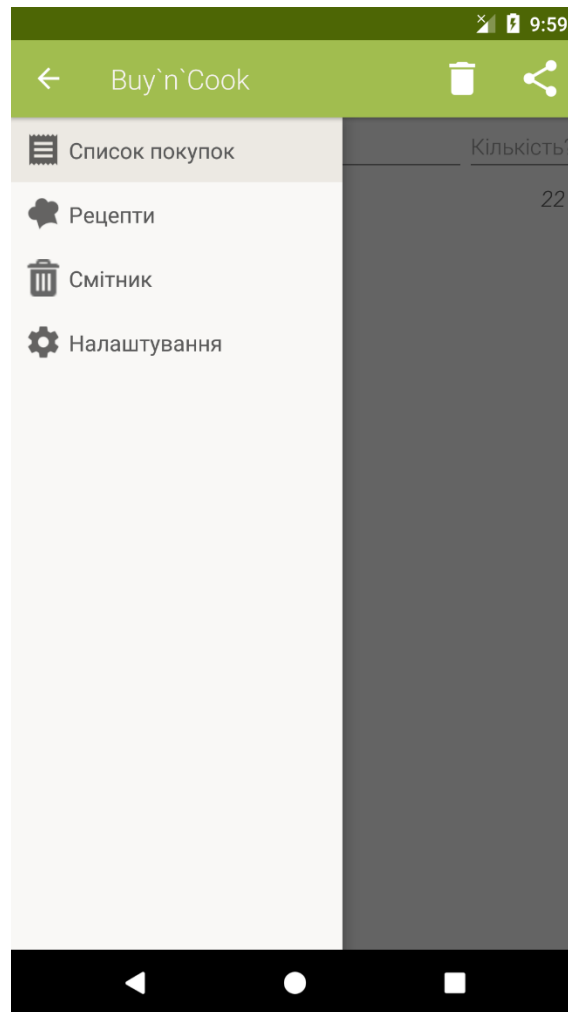


Рис.1.18 Головне меню додатку.

Крім головного меню є:

- Відображення списку покупок. Має в собі кнопки для видалення елементів списку та поширення списку.
- Відображення списків покупок з кнопками для їх менеджменту.
- Відображення списку рецептів. В верхній частині вікна присутня кнопка пошуку, за допомогою якої здійснюється пошук за назвою в базі рецептів.
- Відображення рецепту. На верхній панелі кнопки розташовані кнопки редагування та поширення рецепту, в центрі розміщено фото рецепту та три вкладки:

- Інгредієнти рецепту, з кнопкою для додавання вибраних інгредієнтів до поточного списку покупок. Самі інгредієнти відображаються в вигляді маркованого списку. Над списком інгредієнтів є поле для відображення необхідної кількості порцій для приготування.
- Опис рецепту, включає в себе короткий опис рецепту, складність рецепту та калорійність.
- Остання вкладка – кроки приготування рецепту.
- Відображення налаштувань. В налаштуваннях можна вибрати чи відображати категорії товарів, продукти, що в рецептах вважаються не маркованими, очистити базу даних, відправити електронний лист з відгуком розробнику, а також відображаються відомості про версію системи.

#### **1.14. Опис використаних сторонніх бібліотек та модулів.**

При розробці даної інформаційно-довідкової системи використано бібліотеки Google Analytics.

Аналітика Google – це зручний та багатофункціональний сервіс для аналізу додатків та веб-сайтів. Хоча Google Analytics зазвичай використовується для веб-сервісів, існує також окреме відгалуження, що зветься Google Analytics for Mobile Apps, що відповідає за аналітику мобільних додатків. За допомогою простих у розумінні звітів дає можливість:

1. Зрозуміти кількість користувачів мобільного додатку, їхні характеристики та місцезнаходження;
2. Визначити які дії виконують користувачі в додатку;
3. Виміряти платежі в програмі, кількість здійснених покупок, дохід від програми;
4. Налаштувати звіти, специфічні для окремих видів бізнесу;
5. Візуалізувати шляхи навігації користувача;
6. Розділити дані, щоб зрозуміти поведінку різних груп користувачів;

7. Використати отримані дані Google Analytics для інших Google продуктів[23].

Google Analytics можна налаштувати «під себе», як сказано вище, хоча вона надає досить суттєві дані розробнику і при використанні налаштувань за замовчанням:

- кількість користувачів та сеансів;
- тривалість сеансу;
- операційні системи;
- моделі пристроїв;
- географічні дані.

Для підключення Google Analytics необхідно виконати наступне:

1. Додайте в файл AndroidManifest.xml дозволу INTERNET і ACCESS\_NETWORK\_STATE.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

2. Додайте в файл build.gradle залежність на рівні проекту:

```
dependencies { // ..
    classpath 'com.google.gms:google-services:3.0.0'
}
```

3. Додайте в файл app / build.gradle наступну залежність для сервісів Google Play:

```
dependencies {
    // ...
    compile 'com.google.android.gms:play-services-analytics:10.2.4'
}
```

4. Створіть файл app / src / res / xml / global\_tracker.xml наступного змісту:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
```



```
<string name="ga_trackingId" translatable="false">
${YOUR_TRACKING_ID} </string>
</resources>
```

5. Додайте відстеження дій. На цьому етапі ви відправляєте іменованій перегляд екрану в Google Analytics, коли користувач відкриває або перемикає екран в додатку. Код повинен виконувати наступні дії:забезпечувати загальнодоступний лічильник в підкласі Application; перевизначати метод зворотного виклику для роботи в пріоритетному режимі; вказувати назву екрану і виконувати відстеження.
6. Надішліть подію. Задайте значення поля екрану в лічильнику, а потім надішліть звернення. У прикладі нижче для відправки події використовується HitBuilders.EventBuilderEvent.

```
mTracker.send(new HitBuilders.EventBuilder()
    .setCategory("Action")
    .setAction("Share")
    .build());
```

В розроблюваній системі Аналітика Google використовується для збору даних про збої системи, та відвідування різних екранів додатку. За допомогою отриманих даних можна відстежувати проблемний код, що призводить до помилок, а також визначити найпотрібніші користувачу функції, а згодом використати це при якісному оновленні системи.

### **1.15. Опис проблем і нестандартних ситуацій, які виникали під час розробки та заходів для їх вирішення.**

При розробці для тестування програмного застосунку використовувався емулятор Android-пристрою, а саме Pixel під керуванням ОС Android 7.1.1 Nougat. Дана віртуальна машина займає на диску близько 15 гігабайт. За замовчанням, на віртуальному пристрої встановлено англійську локалізацію та відсутня українська клавіатура. Було прийнято рішення не встановлювати пакет для української локалізації, оскільки бракує пам'яті на диску мого ПК. Тому, всі інгредієнти та інші дані, що вводяться користувачем виконуються

англійською мовою за допомогою англійської клавіатури. Інтерфейс додатку розроблено українською мовою.

### **Висновок до розділу**

Даний розділ магістерської кваліфікаційної роботи присвячений розробці програмного коду застосунку. На цьому етапі розроблено та пояснено діаграму компонентів, описано розробку окремих модулів програми та інтерфейсів а також використання сторонніх бібліотек при розробці. Наприкінці розділу подано опис проблем, що виникли при розробці даного програмного продукту та шляхи їх вирішення.

## ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

### 1.16. Вимоги до апаратного забезпечення.

Для встановлення та функціонування системи необхідно, щоб апаратне забезпечення(мобільний телефон) відповідало таким вимогам:

- операційна система Android версії не нижче ніж Nougat 7.1.
- Не менше 100 мб пам'яті на жорсткому диску для зберігання інформації.

При виконанні описаних вище апаратних вимог система працюватиме вірно.

### 1.17. Інструкції щодо встановлення системи.

Android-додатки можна встановлювати двома способами:

1. З офіційного магазину додатків Google Play;
2. За наявності APK-файлу програми.

Google Play – це крамниця застосунків, розроблена компанією Google, яка дає можливість власникам пристроїв з операційною системою Android завантажувати та купувати застосунки, книги, музику та фільми. Рахунок розробника, що надає можливість публікувати програми коштує 25 доларів[24].

Щоб встановити додаток за допомогою Google Play необхідно перейти за прямим посиланням на цей додаток, або знайти його в пошуку магазину. Перейшовши на сторінку додатку, треба натиснути кнопку встановити та надати необхідні дозволи програмі.

Для того, щоб встановити програму за допомогою APK-файлу необхідно спочатку в налаштуваннях операційної системи дозволити встановлення додатків з невідомих джерел. За замовчуванням ця функція вимкнена з міркувань безпеки. На рисунку 6.1 зображено послідовність дій.

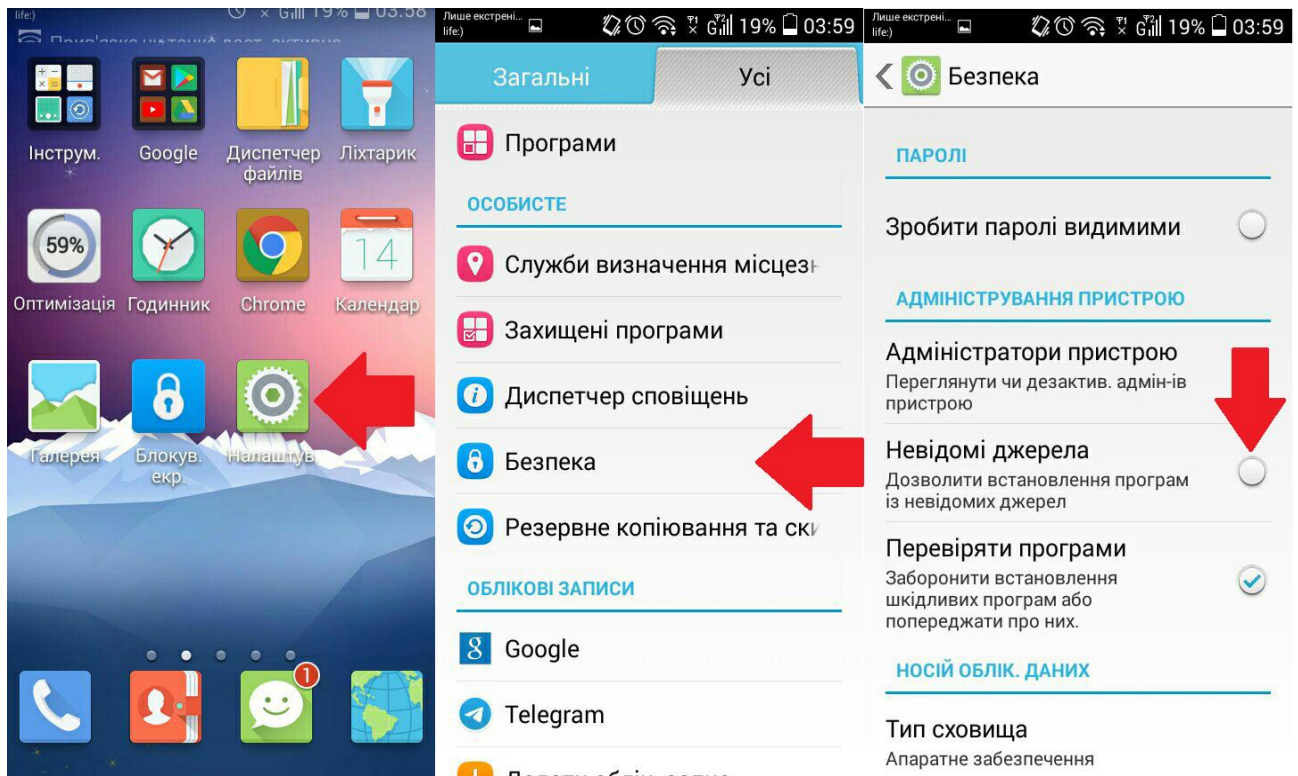


Рис.1.19 Послідовність дій для дозволу встановлення додатків з невідомих джерел.

Після встановлення дозволів, система дозволяє встановлювати додатки з невідомих джерел.

Тепер необхідно передати файл з застосунком до файлової системи апаратного забезпечення, це можна зробити в будь-який доступний спосіб: через usb-провід, за допомогою Bluetooth тощо.

Наступний крок – знайти файл в файловій системі, можна зробити за допомогою стандартного диспетчера файлів, або сторонніх додатків. Наш файл називається BUYnCOOK.apk, що знаходиться в папці Download.(див. рис. 6.2).

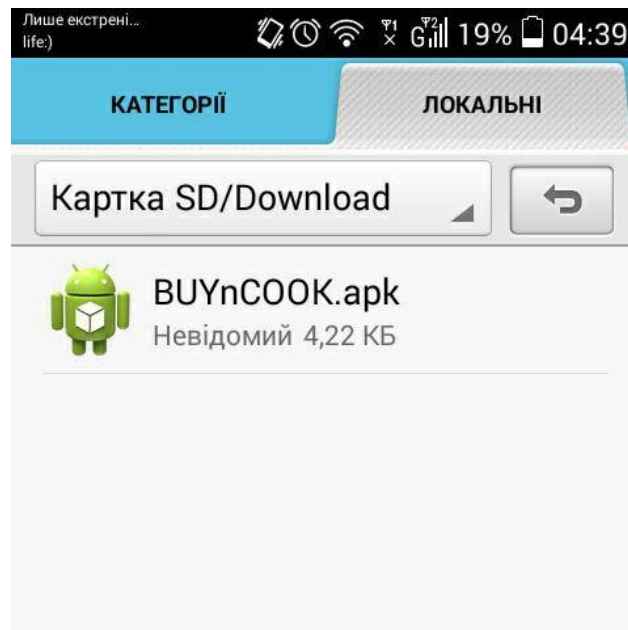


Рис.1.20 Файлова система пристрою.

Відкривши даний файл, система запитає дозволи на використання галереї фото, щоб додавати їх до рецептів. Після підтвердження дозволів користувачем, менеджер пакетів почне налаштування та встановлення додатку.

Програма поширюється на безкоштовній основі.

### 1.18. Тестування.

Тестування системи – необхідний етап при розробці, що допомагає виявити та виправити помилки виконання.

Для тестування обрано частину програми, що відповідає за групування елементів списку: коли в списку вже є яблуко 1 кілограм, і додаємо ще один, має бути один рядок списку в якому записано яблуко 2 кілограми, тощо. Також частину коду, що відповідає за приведення даних до єдиного формату.

А також частину програми, що містить в собі множення інгредієнтів рецепту на кількість порцій для отримання максимально точної необхідної кількості продуктів.

Виконано автоматизоване тестування за допомогою бібліотеки `junit.Assert.assertEquals`. Бібліотека Junit – кросплатформенна бібліотека для тестування програмного забезпечення на мові Java[25].

Код тестів можна переглянути в додатку А.

Результати тестування можна переглянути в таблиці 6.1.

Результати автоматизованого тестування модулю групування елементів списку

Виклик методу	Очікуваний результат	Реальний результат
mergeAmounts("2 bla", "2 blub")	"2 bla + 2 blub"	"2 bla + 2 blub"
mergeAmounts("2", "3")	"5"	"5"
mergeAmounts("120 kg", "130 kg")	"250 kg"	"250 kg"
mergeAmounts("1/4", "2/4")	"3/4"	"3/4"
mergeAmounts("3/2", "2/4")	"2"	"2"
mergeAmounts("1/5 blub", "6/10 blub")	"4/5 blub"	"4/5 blub"
mergeAmounts("1/4", "0,5")	"0,75"	"0,75"
mergeAmounts("1/5 blub", "0,6 blub")	"0,8 blub"	"0,8 blub"
multiplyAmount("2/4", 1, 1)	"1/2"	"1/2"
multiplyAmount("1/4", 4, 5)	"5/16"	"5/16"
multiplyAmount("1/4", 1, 4)	"1"	"1"
multiplyAmount("1/3", 7, 5)	"5/21"	"5/21"
formatAmount("2,3")	"2,3"	"2,3"
formatAmount("2.3")	"2,3"	"2,3"
formatAmount("2.3kg")	"2,3 kg"	"2,3 kg"
formatAmount(" 322.332kg ")	"322,332 kg"	"322,332 kg"

Провівши тестування модулю групування елементів списку можна зробити висновок, що він працює вірно В роботі його можна буде подивитися в наступному підпункті що відповідає за мануальне тестування системи.

### **1.19. Інструкція користувачу.**

При відкритті додатку, відкривається останній список покупок. Прогорнувши вліво можна побачити головне меню програми.

1. Головне меню програми(рис.6.3). Звідси можна перейти до будь-якої частини програми.
  - a. Списки покупок. Кнопка, що відображає списки покупок, при натисканні відображається екран Списків покупок.
  - b. Рецепти. Кнопка, що відображає рецепти, при натисканні відображає екран списку рецептів.
  - c. Смітник. Кнопка, що відображає смітник. При натисканні відображає екран смітника.
  - d. Налаштування. Кнопка, що відкриває налаштування.

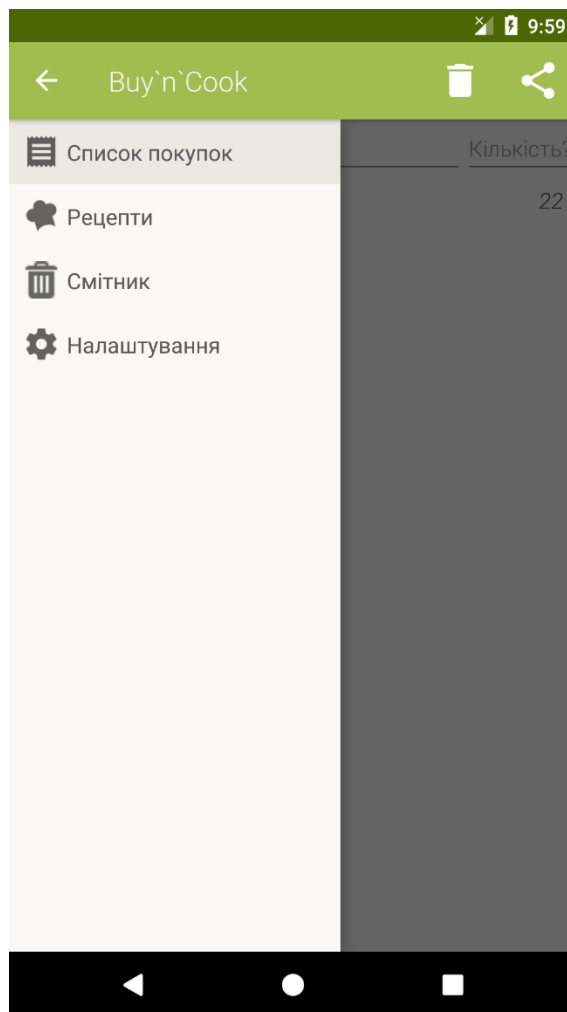


Рис.1.21 Головне меню програми.

2. Списки покупок. Екран на якому відображаються списки покупок. Можна вибрати необхідний та провести з ним такі маніпуляції:
  - a. Поширити список. Функція що викликається натисканням на піктограмку поширення в вікні відображення списку. Викликає

діалог поширення(рис.6.4). Діалог включає в себе такі елементи: копіювати в буфер обміну, електронна пошта, повідомлення або зберегти на диску. Внизу екрану можна вибрати чи це одноразово ви вибираєте таке поширення, чи можна встановити його за замовчуванням.

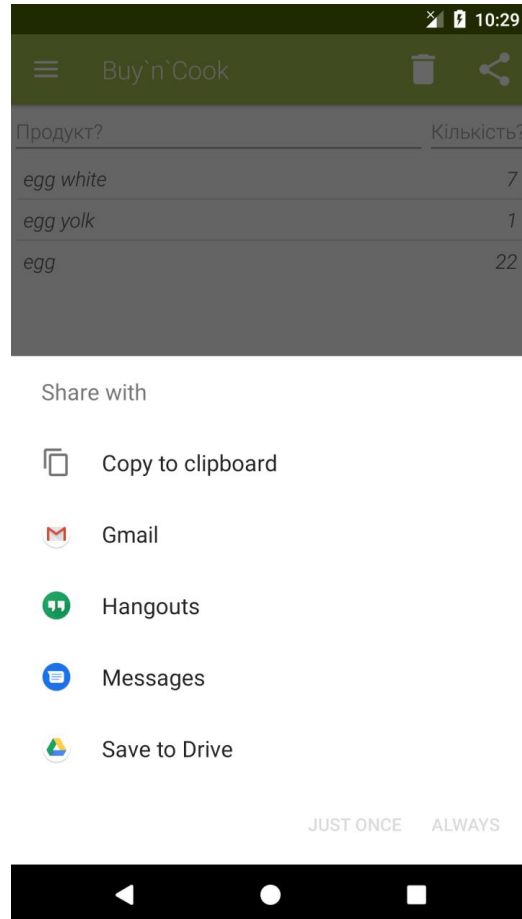


Рис.1.22 Діалог поширення списку покупок.

- в. Додати елемент до списку. Відбувається при заповненні полів та натисканні кнопки «ОК!» на клавіатурі. Елементи, що додаються можуть бути додані з бази даних назв продуктів, якщо потрібний інгредієнт в ній відсутній – нововведений елемент зберігається в базу. Якщо елемент з таким іменем в списку уже є, то нова кількість продукту додається до уже записаної раніше(рис.6.5.).



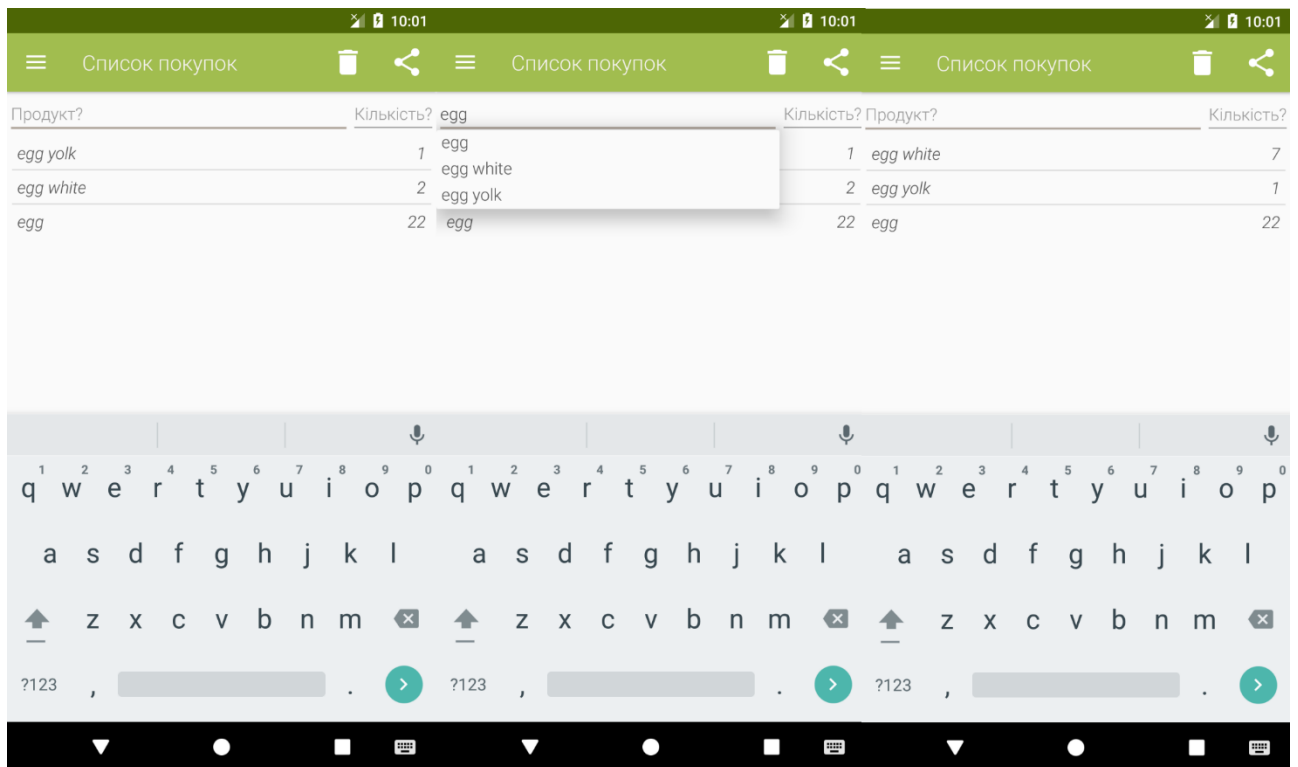


Рис.1.23 Додавання елементу до списку, порядок виконання зліва – направо.

- с. Редагувати елемент списку. Відбувається при довгому затисканні одного з введених елементів списку. Користувачу виводиться діалог редагування елементу списку. Після натиснення клавіші «ОК!», база даних списку оновлюється.
  - д. Видалити елемент списку. Видалити елемент списку легко натиснувши на елемент списку.
  - е. Видалити всі елементи списку. При натисканні на піктограму смітника в верхній частині екрану, користувач бачить запит на підтвердження очищення всього списку покупок.
3. Рецепти. Екран відображення рецептів. Включає в себе такі можливості:
- а. Пошук рецепту за назвою. Натиснувши на піктограму пошуку відкривається рядок для введення назви і відбувається пошук по базі даних.
  - б. Відображення рецепту. Після натискання на рядок, що має назву якогось рецепту програма відправляє користувача на екран цього рецепту.

- c. Поширення рецепту. На екрані відображення рецепту в верхній частині розміщена піктограма поширення. Після натискання на неї викликається діалог поширення з вибором за допомогою якого інструменту поширити рецепт.
  - d. Редагування рецепту. На екрані відображення рецепту в верхній частині відображена піктограма редагування. Натиснувши на неї користувач отримує можливість редагувати обраний рецепт.
  - e. Позначити інгредієнти необхідні для закупівлі. В нижній частині вікна знаходиться вкладка з інгредієнтами. Там можна змінити інгредієнти, що необхідні для закупівлі натисканням на радіо-кнопку.
  - f. Змінити кількість порцій. На вкладці інгредієнти є поле, що відповідає за кількість порцій страви, для яких ми готуємо та будемо купувати продукти. В цьому полі вказуємо кількість людей, що будуть їсти.
  - g. Надіслати обрані інгредієнти на задану кількість порцій до поточного списку покупок. Викликається за допомогою кнопки плюсік посередині екрану. Відправляє до списку покупок назви продуктів та їх кількість, враховуючи кількість порцій.
  - h. Переглянути опис страви. Перемістившись на другу вкладку в нижній частині екрану відкривається опис страви. До нього належить також оцінка калорійності, оцінка складності приготування та короткий опис страви.
  - i. Переглянути кроки приготування страви. Для цього необхідно переміститися на третю вкладку на екрані рецепту. Тут відображається процес приготування крок-за-кроком.
4. Смітник. Тут відображаються нещодавно видалені списки покупок.
- a. Очистити смітник. В верхній частині екрану смітника розміщено іконку смітника. Натиснувши на неї, можна очистити смітник.

Перед видаленням система ще раз запитає в вас чи ви точно готові цього позбутися.

- б. Повернути список покупок. На кожному списку покупок відображено кнопку повернути. Перед поверненням треба буде підтвердити свої дії.

5. Налаштування. Вікно налаштувань, показує поточні налаштування системи та дозволяє їх змінити(рис.6.).

- а. Інгредієнти, що не позначаються. Тут можна записати список інгредієнтів, що завжди є вдома і їх не потрібно кожного разу додавати до списку покупок. За замовчуванням там є сіль, перець, вода та олія.

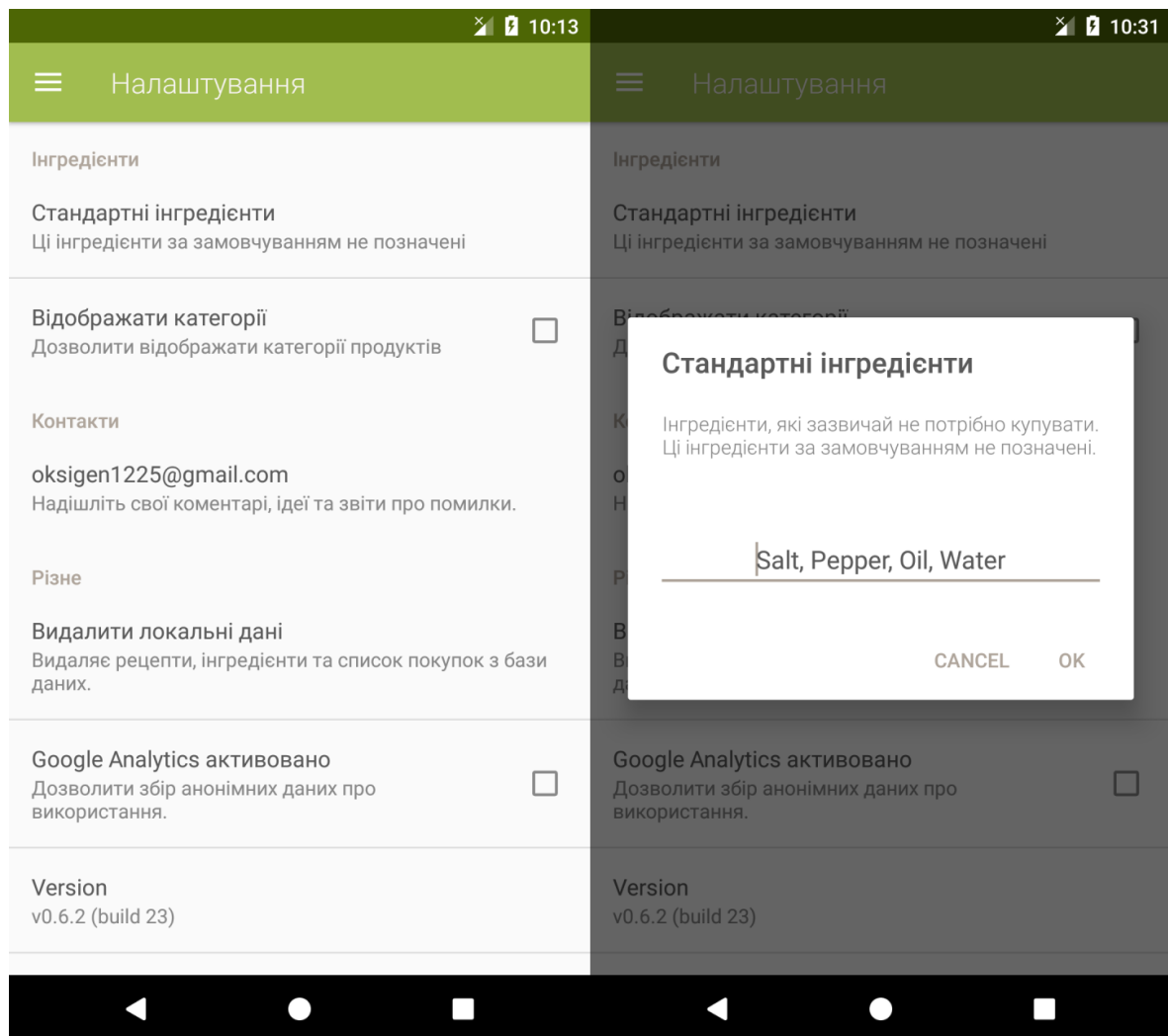


Рис.1.24 Налаштування. Зліва початковий екран налаштувань, справа – викликано діалог змінити інгредієнти, що не позначаються.

- b. Відображення категорій товарів. Для зручності розроблена категоризація товарів, якщо комусь зручніше без неї – необхідно просто зняти відмітку з цього поля.
- c. Контакти. Тут можна надіслати листа розробникам відносно збоїв системи чи ідей покращення.
- d. Очищення бази даних. Тут можна одразу всю базу даних очистити. Потребує підтвердження з боку користувача, щоб запобігти випадковому натисканню.
- e. Збір даних для Google Analytics. Цей радіо-баттон відповідає за погодження користувача на поширення анонімної інформації до розробника з метою покращення роботи додатку.
- f. Версія програми. Відображено версію програми.

### **Висновок до розділу**

У цьому розділі магістерської кваліфікаційної роботи проведено тестування модулів системи, описано вимоги до апаратної складової, а також інструкція з встановлення додатку для користувача за допомогою двох способів.

Після проведеного тестування, можна зробити висновок, що протестовані модулі працюють вірно.

Також в даному розділі наведено знімки екрану при виконанні певних функцій системою, ручне тестування застосунку.

## ВИСНОВКИ

При виконанні даної магістерської кваліфікаційної роботи було розроблено інформаційно-довідкову систему для закупівлі продуктів, пройшовши основні етапи створення програмних продуктів: формулювання вимог, системний аналіз, проектування, розробка, тестування, установка та підтримка.

На етапі формулювання вимог проведено аналіз предмету дослідження та розроблено постановку задачі. Огляд літератури включив в себе огляд та вибір платформи для реалізації програмного продукту а також огляд, опис та аналіз аналогічних програмних рішень.

Проведено системний аналіз предметної області, знайдено імовірні проблеми при розробці інформаційної системи, сформовано дерево проблем та дерево цілей, що визначає цілі, які мають бути досягнуті в результаті розробки. Виходячи з системного аналізу спроектовано бізнес-процеси в інформаційній системі за допомогою методології IDEF0; розглянуто потоки даних в системі та описано діаграмою DFD; розроблено схему бази даних, що відображається на ER-діаграмі; діаграма випадків використання (Use-case) документує функціональні вимоги.

На основі спроектованих моделей розроблено окремі модулі системи, які врешті об'єднались в одне ціле. Проведено автоматизоване тестування системи за допомогою JUnit. Для користувача описано вимоги до апаратної частини та інструкцію з встановлення двома способами. Для користувача наведена детальна зрозуміла інструкція з ілюстраціями, що спрощує роботу з програмою.

В економічній частині даної магістерської кваліфікаційної роботи проведено економічний та маркетинговий аналіз, зібрано інформацію про ринки збуту. Вибрано первинну стратегію розвитку існуючого продукту, оскільки ринок не новий і потреба в організації закупівлі не виникла нещодавно, ми використовуємо кращі частини вже розроблених програмних застосунків і додаємо свій авторський функціонал. Оскільки в планах нема виходу на нові ринки чи впровадження кардинально інших бізнесів, вторинна стратегія –

розвитку продукту. Для підтвердження правильності вибору стратегії проведено бюджетування проекту з урахуванням всіх можливих витрат на розробку. В результаті обрахунків виявилось, що система є досить прибутковою, тож є всі підстави для її створення.

Загалом, результатом виконання магістерської кваліфікаційної роботи є розроблена інформаційно-довідкова система для закупівлі продуктів з додатковим модулем групового додавання товарів з рецептів. Очікується, що дана система буде розвиватись надалі. Можливі майбутні реалізовані функції – створення груп користувачів, що можуть створювати групові списки покупок, а також підключення сканування штрих-коду та API для замовлення доставки з місцевих або глобальних мереж супермаркетів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Органайзер.—[Електронний ресурс] –Режим доступу:  
<https://slovnyk.me/dict/vts/%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D0%B0%D0%B9%D0%B7%D0%B5%D1%80>
2. Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018.—[Електронний ресурс] –Режим доступу:  
<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>
3. About Android —[Електронний ресурс] –Режим доступу:  
<https://developer.android.com/about>
4. Android Documentation —[Електронний ресурс] –Режим доступу:  
<https://developer.android.com/guide/components/fundamentals>
5. iOS Documentaion —[Електронний ресурс] –Режим доступу:  
<https://developer.apple.com/develop/>
6. Telegram Bot Documentation —[Електронний ресурс] –Режим доступу:  
<https://core.telegram.org/bots>
7. Системний аналіз[Навчальний посібник] /Катренко А.В. – Львів, Новий світ, 2000 — 396с.
8. Java Documentation —[Електронний ресурс] –Режим доступу:  
<https://www.oracle.com/technetwork/java/intro-141325.html>
9. C/C++ Documentation —[Електронний ресурс] –Режим доступу:  
<http://www.cplusplus.com/>
10. Documentation Visual Studio Xamarin —[Електронний ресурс] –Режим доступу: <https://docs.microsoft.com/xamarin/>
11. Documentation Basic4android—[Електронний ресурс] –Режим доступу:  
<https://www.b4x.com/b4a.html#documentation>
12. Declaring Layout —[Електронний ресурс] –Режим доступу:  
<https://developer.android.com/guide/topics/ui/declaring-layout>
13. Recommended Formats Statement—[Електронний ресурс] –Режим доступу: <https://www.loc.gov/preservation/resources/rfs/data.html>

14. Data storage —[Электронный ресурс] –Режим доступа: <https://developer.android.com/training/data-storage/sqlite#java>
15. Android Studio Documentation —[Электронный ресурс] –Режим доступа: <https://developer.android.com/studio/intro>
16. Enterprise Architect—[Электронный ресурс] –Режим доступа: <https://sparxsystems.com/products/ea/>
17. AllFusion Process Modeler —[Электронный ресурс] –Режим доступа: <https://erwin.com/products/erwin-business-process/>
18. AllFusion Erwin Data Modeler—[Электронный ресурс] –Режим доступа: <https://erwin.com/products/erwin-data-modeler/>
19. Методология IDEF0 —[Электронный ресурс] –Режим доступа: [https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6\\_2](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2)
20. Основи теорії систем та системного аналізу/Сорока К.О. ХНАМГ,2004 — 291с.
21. What is Entity Relationship Diagram—[Электронный ресурс] –Режим доступа: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>
22. Use-case diagram—[Электронный ресурс] –Режим доступа: <https://searchsoftwarequality.techtarget.com/definition/use-case>
23. What is the Component diagram—[Электронный ресурс] –Режим доступа: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/>
24. Google Analytics. —[Электронный ресурс] –Режим доступа: <https://developers.google.com/analytics/devguides/collection/android/v4/>
25. Google Play —[Электронный ресурс] –Режим доступа: <https://play.google.com>
26. JUnit—[Электронный ресурс] –Режим доступа: <https://junit.org/junit5/>