

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

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Безпека комп'ютерних систем і мереж»

Група: 4КБ-01

Дипломний проєкт

здобувача освіти денної форми навчання

КБ.01.12.000.ДП

**ПОПОВА ОЛЕКСАНДРА
РУСЛАНОВИЧА**

м. Одеса
2024 р.

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

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Безпека комп'ютерних систем і мереж»

Група: 4КБ-01

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

**Розробка мобільного додатку
для обліку особистих фінансів**

Проектний матеріал складається з пояснювальної записки на 78 сторінках та графічного (презентаційного) матеріалу на 11 аркушах (слайдах).

Дипломник _____ (Попов О. Р.)
Керівник _____ (Скорнякова О. В.)

Консультанти:

з економічного розділу _____ (Іванченков В. С.)
з розділу охорони праці та техніки безпеки _____ (Чорновол Н. І.)
з нормоконтролю _____ (Петрашова В. І.)
старший консультант _____ (Кривченко Ю. В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю. В.)
Завідувач відділення _____ (Скорнякова О. В.)

Захист «20» 06 2024 р.

Протокол ЕК № 4

Оцінка ЕК 4 (добре) / 458.

Секретар ЕК _____

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

Відділення	<u>комп'ютерних систем</u>	Комісія	<u>КТ та ПІ</u>
Спеціальність	<u>123 «Комп'ютерна інженерія»</u>		
Освітня програма	<u>«Безпека комп'ютерних систем і мереж»</u>		

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І. В.
" 16 " 11 2024 року

ЗАВДАННЯ
на дипломний проєкт

Здобувачеві освіти Попову Олександр Рустановичу
1. Тема проєкту Розробка мобільного додатку для обліку особистих фінансів

Затверджена наказом по коледжу від " 02 " листопада 2023 р., наказ № 244-А2-ОД

2 Термін здачі закінченого проєкту 10.06.2024 р.

3. Вихідні дані до проєкту

1. Застосувати графічний дизайн користувача (GUI) у стилі MaterialYou
2. Реалізувати дружній користувацький досвід (UX)
3. Розробити алгоритм збереження фінансів засобами мови програмування Dart
4. Реалізувати мобільний застосунок засобами фреймворку Flutter

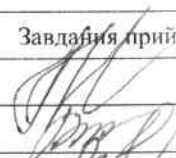
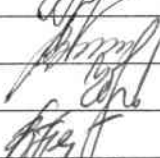
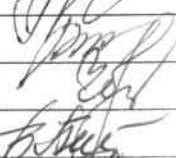
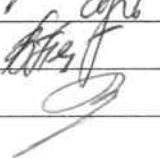
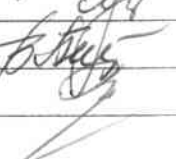
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз предметної області. 2. Технології та засоби розробки (проєктування).
3. Проєктування дизайну застосунку. 4. Проєктування застосунку.
5. Розробка мобільного застосунку. 6. Тестування створеного мобільного застосунку.
7. Економічний розрахунок. 8. Аспекти охорони праці та техніки безпеки

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

Презентація Power Point – 11 слайдів
(Загальні відомості; Стек технологій; Стилізація графічного інтерфейсу;
Навігація у застосунку; Архітектура застосунку; Процес розробки;
Програмний код основної логіки; Демонстрація основної логіки;
Зв'язок з ОПП)

6. Консультанти по проєкту, із зазначенням розділів проєкту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Скорнякова О. В.		
Економічний розділ	Іванченков В. С.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

7. Дата видачі завдання

15.01.2024

Керівник

Скорнякова О. В.


(підпис)

Завдання прийняв до виконання

Попов О. Р.


(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проєкту (роботи)	Термін виконання етапів дипломного проєкту (роботи)	Відмітка про виконання
1	Формування вступу	29.04.24	виконано
2	Аналіз предметної області	10.05.24	виконано
3	Підбір технічної літератури	19.05.24	виконано
4	Вибір технологій та засобів розробки (проєктування)	20.05.24	виконано
5	Проектування дизайну мобільного застосунку	22.05.24	виконано
6	Проектування архітектури мобільного застосунку	24.05.24	виконано
7	Реалізація мобільного застосунку	27.05.24	виконано
8	Тестування створеного мобільного застосунку	29.05.24	виконано
9	Оформлення пояснювальної записки	31.05.24	виконано
10	Оформлення графічної (презентаційної) частини	01.06.24	виконано
11	Економічний розрахунок	02.06.24	виконано
12	Опис охорони праці та техніки безпеки	09.06.24	виконано
13	Аналіз результатів проєктування	13.06.24	виконано
14	Підготовка доповіді для захисту	16.06.24	виконано

Дипломник


(підпис)

Керівник


(підпис)

[illegible]

ЗМІСТ

ВСТУП	7
1 ОСНОВНИЙ РОЗДІЛ	9
1.1 Аналіз предметної області	9
1.1.1 Наявні рішення-аналоги на ринку	9
1.1.1 Обрані технології мобільної розробки	14
1.2 Проектування мобільного застосунку	18
1.2.1 Технічне завдання на розробку	18
1.2.2 Проектування UI частини мобільного застосунку	19
1.2.3 Проектування UX частини мобільного застосунку	20
1.2.4 Проектування архітектури мобільного застосунку	22
1.3 Реалізація мобільного застосунку	23
1.3.1 Середовище розробки	23
1.3.2 Розробка бази для застосунку	25
1.3.3 Розробка екрану “Планування”	27
1.3.4 Розробка екрану “Поради”	32
1.3.5 Розробка екрану “Довідка”	34
1.3.6 Розробка екрану “Автор”	35
1.3.7 Розробка екрану “Вихід”	36
1.3.8 Будівання графіку	37
1.4 Мануальне тестування (Manual QA)	38
1.4.1 Тестування екрану “Планування”	39
1.4.2 Тестування екрану “Поради”	45
1.4.3 Тестування екрану “Довідка”	47
1.4.4 Тестування екрану “Автор”	48
1.4.5 Тестування екрану “Вихід”	49
2 ЕКОНОМІЧНИЙ РОЗДІЛ	50
2.1 Резюме	50

2.2	Визначення трудомісткості розробки програмного забезпечення	50
2.3	Розрахунок ціни програмного продукту	53
3	РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	56
3.1	Вступ	56
3.2	Аналіз умов праці й забезпечення безпеки при виконання основних видів робіт на об'єкті дипломного проектування	56
3.3	Гігієнічні вимоги до виробничого середовища	56
3.3.1	Вимоги до приміщення експлуатації ПК	57
3.3.2	Гігієнічні вимоги до параметрів повітря приміщень із ПК	57
3.3.3	Виробниче освітлення	58
3.3.4	Електробезпека	58
3.3.5	Організація робочих місць із ПК	59
3.4	Пожежна безпека	60
	ВИСНОВКИ	62
	ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	63
	ДОДАТОК А. Програмний код основної логіки мобільного застосунку	64
	ДОДАТОК Б. Слайди мультимедійної презентації	73

ВСТУП

Мобільні застосунки для обліку особистих фінансів завоюють все більшу популярність. Ці інструменти дають можливість користувачам чітко контролювати свої доходи та витрати, а також аналізувати власну фінансову діяльність. Автоматизовані системи стають ключовим фактором, який спрощує ці процеси та робить їх більш зручними.

Широкий спектр функцій мобільних застосунків для фінансового обліку включає в себе автоматизовані інструменти для звітності, управління рахунками та категоризації витрат. Ці функціональні можливості роблять процес управління фінансами більш ефективним та комфортним для користувачів.

Важливо підкреслити, що мобільні застосунки для обліку особистих фінансів не лише допомагають користувачам краще контролювати свої витрати, але й можуть слугувати потужним інструментом для досягнення фінансових цілей. Завдяки можливостям обліку, аналітиці та порадам, які пропонують ці застосунки, користувачі можуть краще планувати свої витрати, економити кошти та інвестувати в майбутнє.

Крім того, мобільні застосунки для обліку особистих фінансів можуть стати цінним інструментом для фінансової освіти. Завдяки доступним та зрозумілим порадам та рекомендаціям, які пропонують ці застосунки, користувачі можуть покращити свої фінансові знання та навички, що допоможе їм приймати більш обґрунтовані та вигідні фінансові рішення.

Метою цього дипломного проєкту є розробка мобільного застосунку для обліку особистих фінансів. Застосунок має дозволяти користувачам чітко контролювати свої доходи та витрати, а також аналізувати власну фінансову діяльність.

Технологічний стек для реалізації проєкту включає мову програмування Dart та фреймворк Flutter. Dart відомий своєю швидкістю та гнучкістю, що робить його ідеальним вибором для розробки мобільних додатків. Flutter, у свою

					КБ 01. 12 000. 00 ДП ПЗ	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

чергу, забезпечує зручну та ефективну розробку інтерфейсів користувача з інтуїтивно зрозумілим дизайном.

Практична цінність отриманих результатів полягає в тому, що розроблена система буде корисною для користувачів, сприяючи їхній фінансовій грамотності та забезпечуючи зручні інструменти для аналізу та контролю їхніх доходів і витрат. Завдяки цьому, користувачі зможуть приймати більш обґрунтовані фінансові рішення, ефективно планувати свої витрати та досягати своїх фінансових цілей.

					КБ 01. 12 000. 00 ДП ПЗ	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз предметної області

1.1.1 Наявні рішення-аналоги на ринку

Для пошуку рішення-аналогів використовуємо магазин застосунків Google Play Market.

Розглянемо перше рішення-аналог – “Money Lover”.

Money Lover – це мобільний додаток для ведення особистого бюджету і відстеження витрат. Він дозволяє користувачам вносити доходи та витрати, створювати бюджети на місяць, категоризувати витрати, а також аналізувати свою фінансову діяльність за допомогою звітів і графіків [1].

На рис 1.1. зображено сторінку застосунку “Money Lover” в Google Play Market.

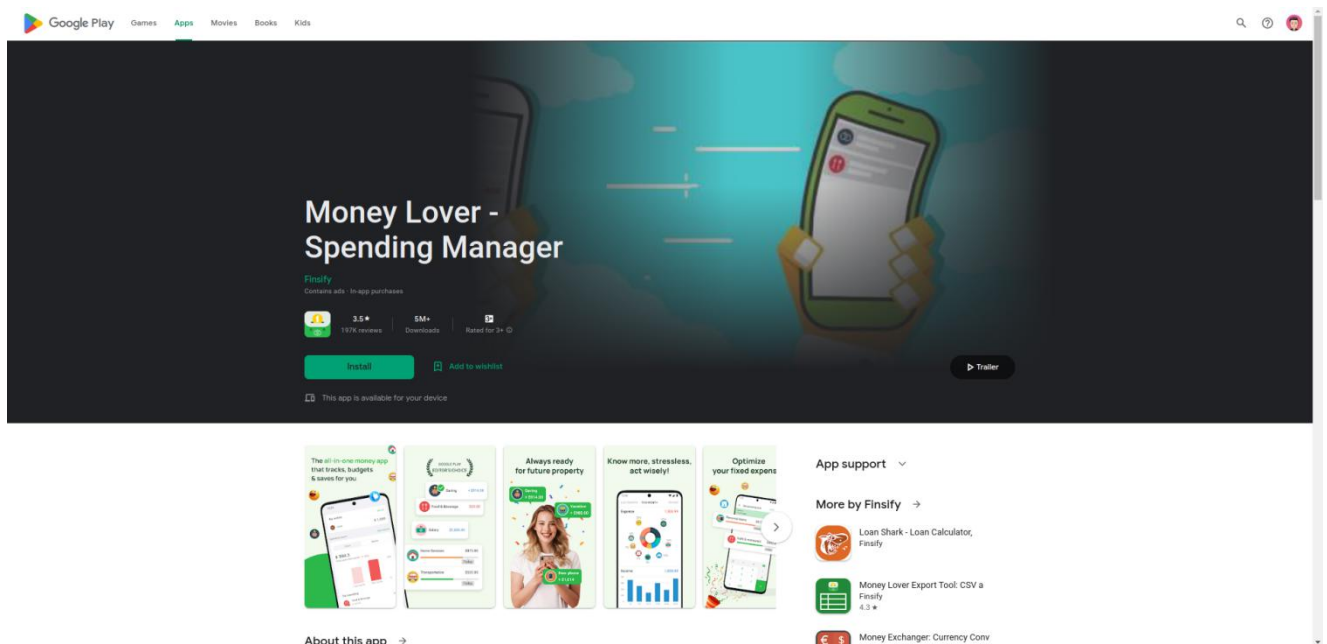


Рисунок 1.1. Сторінка застосунку “Money Lover” в Google Play Market

Переваги:

Простий у використанні: інтерфейс досить інтуїтивний, що дозволяє новачкам швидко орієнтуватися.

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						9
Ізм.	Лист	№ докум.	Підпис	Дата		

Функція аналізу витрат і доходів: користувачі можуть отримувати звіти та графіки, що допомагають зрозуміти, куди йде гроші та як можна ефективніше управляти фінансами.

Можливість налаштування бюджетів для кожної категорії: це дозволяє краще контролювати витрати та планувати фінансові цілі.

Недоліки:

Можливо, не досить інтуїтивний інтерфейс: хоча для багатьох користувачів інтерфейс зрозумілий, деяким користувачам може знадобитися додатковий час для звикання.

Деякі функції доступні лише у преміум-версії: це може бути обмеженням для тих, хто шукає повний функціонал без додаткових витрат.

Розглянемо друге рішення-аналог – “Expensify”.

Expensify – це додаток, спрямований на відстеження витрат і облік витрат для бізнесу та особистих потреб. Він дозволяє користувачам сканувати квитанції, створювати звіти про витрати, автоматизувати процеси відшкодування витрат для бізнесу та багато іншого [2].

На рис 1.2. зображено сторінку застосунку “Expensify” в Google Play Market.

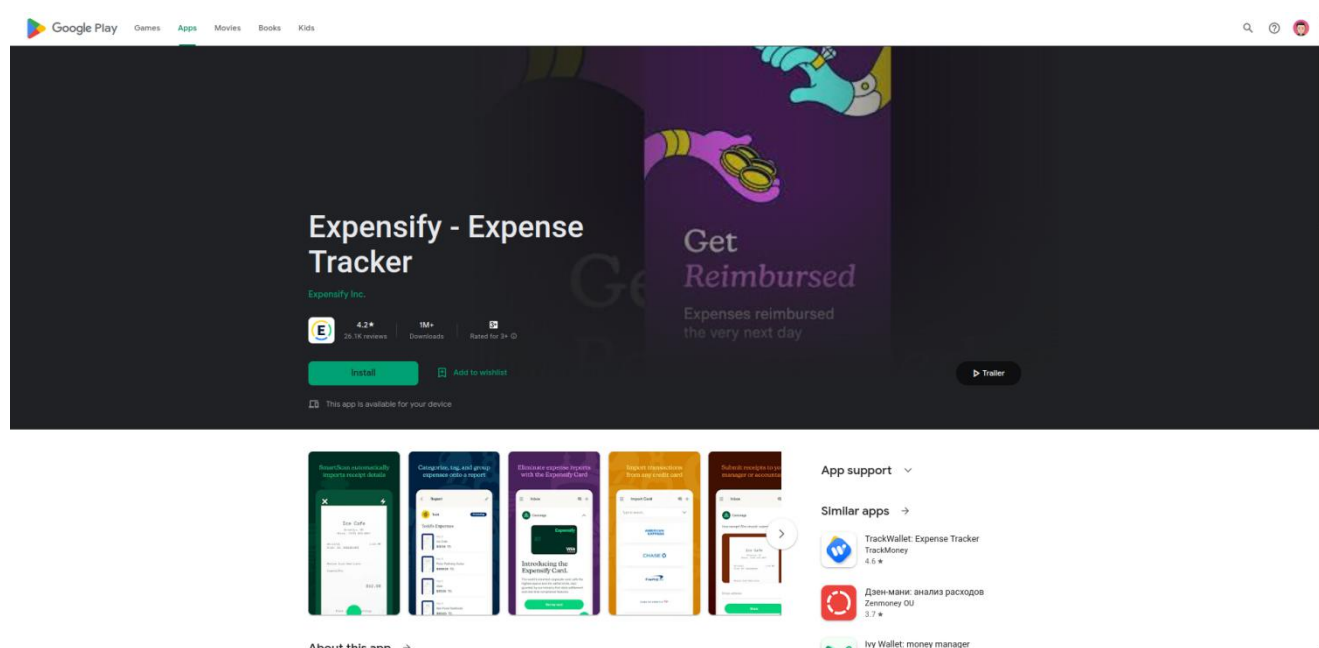


Рисунок 1.2. Сторінка застосунку “Expensify” в Google Play Market

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						10
Ізм.	Лист	№ докум.	Підпис	Дата		

Переваги:

Можливість сканування квитанцій для автоматичного внесення витрат: це значно полегшує процес ведення обліку витрат та звітності.

Автоматизований процес звітності про витрати для бізнесу: додаток спрощує процеси звітності для корпоративних користувачів.

Інтеграція з іншими програмами для фінансового обліку: це дозволяє користувачам зручно інтегрувати Expensify з іншими інструментами, які вони вже використовують.

Недоліки:

Висока вартість платних версій: це може відлякувати користувачів, які шукають більш доступні або безкоштовні альтернативи.

Для особистих користувачів може бути зайвими функціонал, спрямований на бізнес: деякі функції можуть бути надто складними або непотрібними для індивідуального використання.

Розглянемо третє рішення-аналог – “Wally”.

Wally – це додаток для ведення особистого бюджету, схожий на Money Lover. Він дозволяє користувачам створювати бюджети, вносити витрати і доходи, категоризувати їх та аналізувати фінансову діяльність за допомогою звітів [3].

На рис 1.3. зображено сторінку застосунку “Wally” в Google Play Market.

					<i>КБ 01. 12 001. 00 ДП ПЗ</i>	Арк.
						11
Ізм.	Лист	№ докум.	Підпис	Дата		

Wally: AI Personal Finance

Wally Global Inc.

10K+

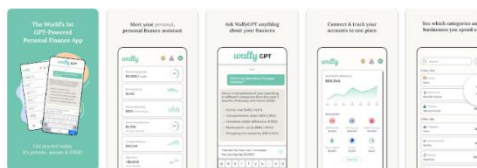
Downloads

Rated for 3+ ⓘ

Install

Add to wishlist

This app is available for your device



About this app →

The world's 1st GPT powered personal finance app is here!
Get started today. It's private, secure and FREE (no premium subscriptions, hidden charges, or ads)
The #1 personal finance management app now enhanced with WallyGPT and unparalleled privacy and security.
TOP RATED GLOBALLY SINCE 2014...

Updated on
Aug 23, 2023

Finance

App support

Similar apps →

- Дневник: анализ расходов
ZenMoney OU
3.7 ★
- My Wallet: money manager
My Apps
4.7 ★
- Mavio: Expense Manager
Mavio
4.5 ★
- TrackWallet: Expense Tracker
TrackMoney
4.5 ★
- YNAB
ynab.com
4.2 ★
- Finance Pro: Expense control
Jorge de la Hoz...

Рисунок 1.3. Сторінка застосунку “Wally” в Google Play Market

Переваги:

Простий у використанні та естетичний інтерфейс: це робить процес ведення обліку витрат більш приємним і зручним.

Можливість встановлення цілей та нагадувань про бюджет: додаток допомагає користувачам триматися у межах фінансових цілей та планів.

Інтеграція з іншими фінансовими додатками: це дозволяє користувачам зручно переміщувати дані між різними інструментами.

Недоліки:

Відсутність деяких функцій, які доступні в конкуруючих додатках: це може зробити Wally менш привабливим для тих, хто шукає специфічні функції або можливості.

Можливість аналізу деяких типів витрат може бути обмеженою: деякі користувачі можуть відчувати, що деякі аспекти їх фінансової діяльності не можуть бути адекватно відслідковані або проаналізовані за допомогою додатку.

Розглянемо порівняльну таблицю аналогів.

У табл. 1.1. наведено порівняльний аналіз аналогів-рішень.

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						12
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 1.1. Порівняльний аналіз аналогів рішень

Характеристика	Money Lover	Expensify	Wally
Види користувачів	Особисті фінанси	Бізнес, корпоративні витрати	Особисті фінанси
Спрощений ввід витрат	Так	Так	Так
Автоматизація звітності	Ні	Так	Ні
Можливість сканування квитанцій	Ні	Так	Ні
Функція аналізу витрат	Так	Так	Так
Інтеграція з іншими програмами	Ні	Так	Так
Вартість	Безкоштовно, преміум-версія	Платна	Безкоштовно, преміум-версія
Платформи	iOS, Android	iOS, Android	iOS, Android

За результатами аналізу можна визначити, що Money Lover підходить для особистих фінансів, забезпечуючи простий у використанні інструмент для ведення обліку витрат та аналізу фінансової діяльності. Expensify найбільше підходить для бізнесу, забезпечуючи автоматизовані процеси звітності про витрати та інтеграцію з іншими програмами. Wally відзначається зручним інтерфейсом для особистих фінансів та можливістю встановлення цілей, проте може відчувати обмеження в аналізі певних типів витрат.

1.1.1 Обрані технології мобільної розробки

Наразі в контексті мобільної розробки існує нативна та кросплатформена розробка.

Нативна розробка являє собою створення мобільних додатків безпосередньо для конкретної платформи, використовуючи її рідні мови програмування та інструменти, такі як Swift або Objective-C для iOS і Kotlin або Java для Android.

Кросплатформена розробка являє собою створення мобільних додатків, які можуть працювати на кількох платформах (iOS, Android тощо) з єдиним кодовим базисом, за допомогою фреймворків, таких як Flutter або React Native.

Розглянемо порівняльну таблицю аналогів [4, 5].

У табл. 1.2. наведено порівняльний аналіз підходів до мобільної розробки.

Таблиця 1.2. Порівняльний аналіз підходів до мобільної розробки

Характеристика	Нативна розробка	Кросплатформена розробка
Доступність на різних платформах	Окремі додатки для кожної платформи	Один додаток для різних платформ
Ефективність ресурсів пристрою	Максимальне використання ресурсів	Обмежений доступ до ресурсів
Відчуття користувача	Краще відчуття користувача	Менш краще відчуття користувача
Швидкість розробки	Повільніше	Швидше
Ресурси розробника	Більше	Менше
Підтримка нових функцій	Швидше впровадження	Менш швидке впровадження

Нативна розробка надає краще відчуття користувача та максимальне використання ресурсів пристрою, проте вона зазвичай потребує більше часу та ресурсів розробника. З іншого боку, кросплатформена розробка дозволяє зменшити витрати та прискорити процес розробки, але може призвести до менш задовільного відчуття користувача та обмеженого доступу до ресурсів пристрою.

Враховуючи вищезазначений аналіз, обираємо кросплатформену розробку. Розглянемо популярність наявність кроплатформних технологій на ринку за даними сервісу “statista”.

Statista являє собою один з найбільших онлайн-порталів для статистичних даних та досліджень [6].

На рис 1.4. зображено топ кросплатформних технологій на ринку за даними сервісу “statista”.

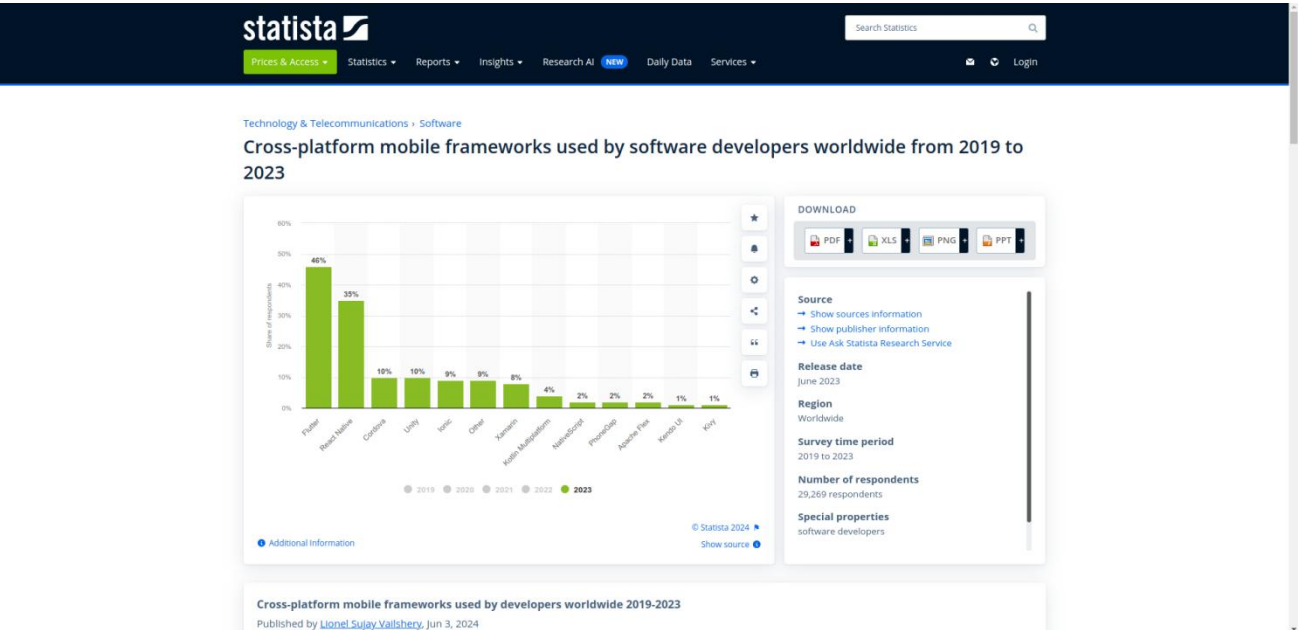


Рисунок 1.4. Топ кросплатформних технологій на ринку за даними сервісу “statista”

Розглянемо порівняльну таблицю найпопулярніших кросплатформних технологій.

Flutter являє собою відкритий фреймворк від Google для створення кросплатформених додатків за допомогою мови Dart.

React Native являє собою фреймворк від Facebook, який дозволяє створювати кросплатформенні мобільні додатки за допомогою JavaScript та React [7].

У табл. 1.3. наведено порівняння найпопулярніших кросплатформних технологій.

Таблиця 1.3. Порівняльний аналіз кросплатформних технологій

Характеристика	Flutter	React Native
Мова програмування	Dart	JavaScript (зазвичай з використанням React)
Власні UI компоненти	Так	Ні
Продуктивність	Висока	Порівняно висока
Швидкість розробки	Швидка	Порівняно швидка
Спільнота та підтримка	Зростаюча	Велика
Навчальний крива	Помірна	Помірна
Вартість розробки	Може бути меншою, оскільки має власні компоненти	Зазвичай стандартні витрати

Розглянемо порівняльну таблицю мов програмування Dart та JavaScript.

У табл. 1.2. наведено порівняльний аналіз підходів до мобільної розробки.

Таблиця 1.4. Порівняльний аналіз кросплатформних технологій

Засоби	Dart	JavaScript
Тип	Статично типізована мова	Динамічно типізована мова
Синтаксис	Схожий на Java, C#, Kotlin	Схожий на C, Java
Використання	Основна мова в Flutter	Основна мова в React Native, створення веб-додатків, серверні застосунки
Компіляція	Компілюється в байткод (VM) або JavaScript (Web)	Компіляція в JavaScript
Швидкодія	Зазвичай швидший за JavaScript	Швидкість залежить від движка JavaScript та реалізації

За результатами порівняння, Flutter відрізняється власними UI компонентами та високою продуктивністю, що робить його привабливим вибором для швидкого розробки мобільних додатків. У той же час, React Native користується великою спільнотою та має помірну навчальну криву, що робить його привабливим для команд, які більше звикли до JavaScript та React.

Таким чином, маємо наступний стек обраних технологій.

На рис 1.5. зображено стек обраних технологій.

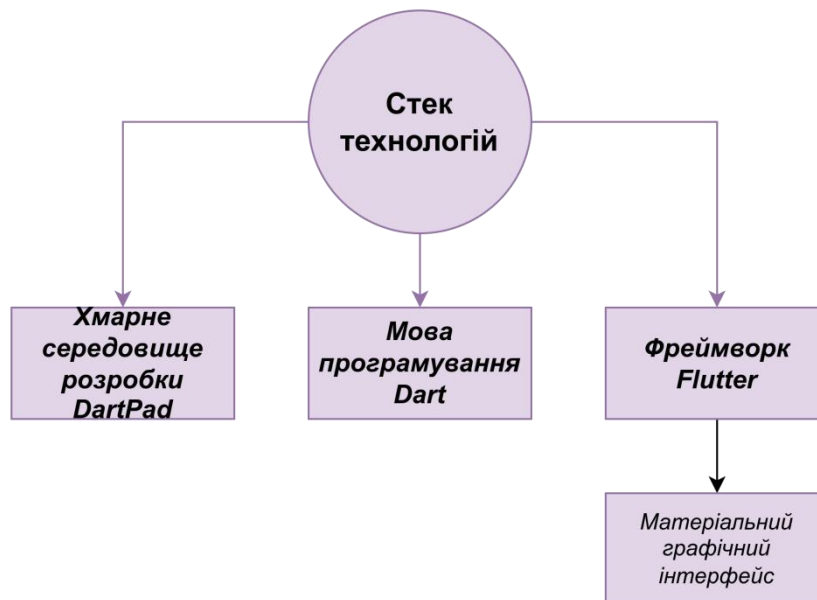


Рисунок 1.5. Стек обраних технологій

1.2 Проектування мобільного застосунку

1.2.1 Технічне завдання на розробку

Розглянемо аспекти технічного завдання на розробку мобільного застосунку:

Мінімалістичний графічний інтерфейс: Для досягнення цього аспекту потрібно забезпечити чистий та простий дизайн інтерфейсу, уникати зайвих деталей та зберігати простоту в кольорах, шрифтах та ефектах.

Зрозумілий досвід користувача: Для створення зрозумілого досвіду користувача важливо розробити послідовний та інтуїтивно зрозумілий інтерфейс, який дозволить користувачам легко взаємодіяти з застосунком та виконувати необхідні дії.

Облік власних фінансів: Для цього застосунку необхідно реалізувати можливість внесення даних про доходи та витрати користувачів, розробити систему категорій для класифікації фінансових операцій та забезпечити можливість аналізу цих даних.

Будування графіків для аналізу витрат: Для цього аспекту необхідно реалізувати функціонал побудови графіків та діаграм для візуалізації фінансової активності користувача, що дозволить легко розуміти їхні витрати та доходи.

Збереження даних в кешу застосунку: Для оптимізації продуктивності та швидкодії застосунку важливо реалізувати механізми збереження даних в кешу пристрою, що дозволить користувачам швидше отримувати доступ до їхніх фінансових даних, навіть у відсутності Інтернет-з'єднання.

1.2.2 Проєктування UI частини мобільного застосунку

У якості стилю графічного інтерфейсу мобільного застосунку буде використовуватися стиль "MaterialYou" (Google Material Design 3).

"MaterialYou" (Material Design 3) являє собою оновлену версію дизайну Google Material Design, яка надає користувачам більше можливостей для персоналізації інтерфейсу [8].

Основні властивості "MaterialYou" включають можливість користувача вибирати основний та акцентний кольори, що застосовуються до елементів інтерфейсу.

Цей підхід дозволяє створювати індивідуальний та унікальний вигляд додатків, які відповідають стилю та вподобанням кожного користувача.

Крім того, "MaterialYou" включає нові стилі та компоненти, які розширюють можливості дизайну та покращують користувацький досвід.

На рис 1.5. зображено компоненти MaterialYou.

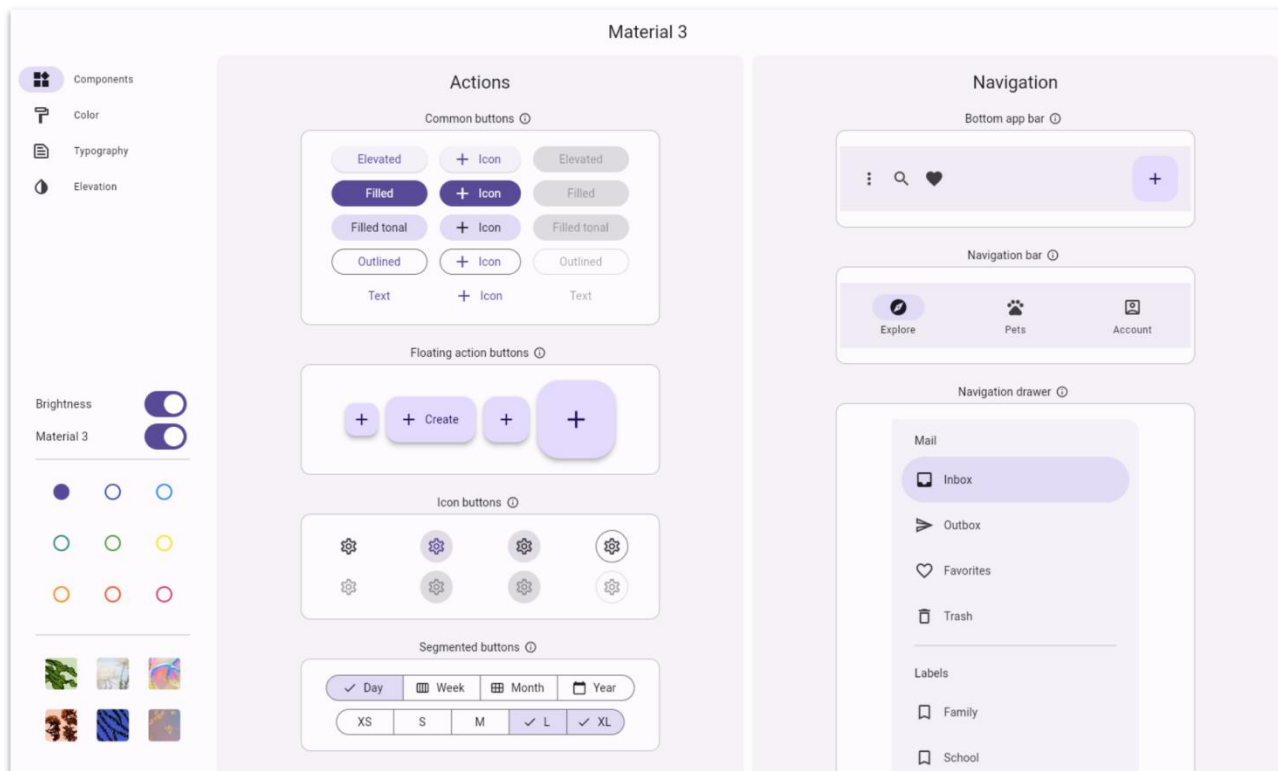


Рисунок 1.5. Компоненти MaterialYou

1.2.3 Проектування UX частини мобільного застосунку

У якості засобу навігації буде використовуватися макет застосунку з BottomNavigationBar.

BottomNavigationBar в Flutter являє собою виджет, який надає навігаційну панель знизу екрана з фіксованою кількістю елементів, кожен з яких представляє окремий екран або сторінку додатку. В основному, BottomNavigationBar використовується для швидкого доступу до основних функцій або розділів додатку [7].

BottomNavigationBar складається з фіксованої кількості BottomNavigationBarItem, кожен з яких представляє одну пункт навігації. Кожен BottomNavigationBarItem має значок і, за бажанням, текстовий опис.

BottomNavigationBar дозволяє користувачу обирати активний елемент шляхом натискання на відповідний пункт навігації. При виборі пункта, його колір або значок може змінюватися, щоб вказати на активний стан.

BottomNavigationBar зазвичай використовується разом з пакетом навігації, таким як Navigator або MaterialApp, для зручної навігації між екранами додатку. Кожен BottomNavigationBarItem може бути пов'язаний з певним екраном за допомогою ідентифікаторів маршрутів або інших методів навігації.

BottomNavigationBar дозволяє налаштовувати розмір, колір, значок і текст кожного пункту навігації, а також зовнішній вигляд загального фону. Він є потужним і простим у використанні інструментом для створення ефективної навігації в мобільних додатках Flutter, дозволяючи користувачам швидко переходити між основними частинами додатку та забезпечуючи зручну інтеракцію.

На рис 1.6. Зображено навігацію між екранами у мобільному застосунку.

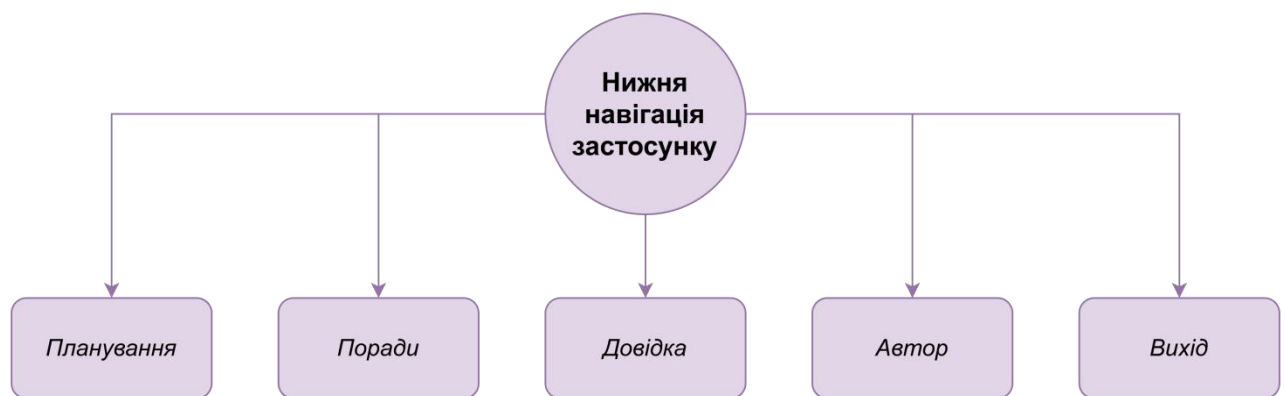


Рисунок 1.6. Навігація між екранами у мобільному застосунку

На рис 1.7. Зображено макет навігації у застосунку.

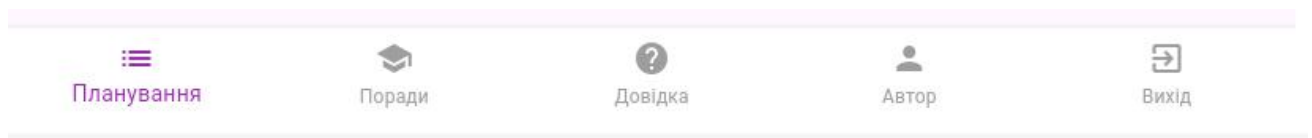


Рисунок 1.7. Навігація між екранами у застосунку

1.2.4 Проєктування архітектури мобільного застосунку

У якості архітектури мобільного застосунку використовується концепція екранів (сторінок).

Архітектура застосунку Flutter, що базується на екранах, використовує підхід, де кожен екран або сторінка додатку представлений окремим віджетом.

Основні властивості цієї архітектури включають розділення логіки, представлення та управління станом для кожного екрана.

Кожен екран може мати свій власний віджет, який відповідає за візуальне представлення і взаємодію з користувачем.

Цей підхід дозволяє легко масштабувати та управляти окремими частинами додатку, що полегшує розробку, тестування та підтримку. Це сприяє повторному використанню коду, оскільки окремі віджети можна використовувати на різних екранах або навіть у різних додатках. Крім того, вона дозволяє забезпечити однорідний дизайн і взаємодію для користувача, оскільки кожен екран може мати свою унікальну візуальну структуру, але при цьому залишатися частиною загального дизайну додатку.

Крім того, архітектура застосунку Flutter, що базується на екранах, сприяє відокремленню логіки додатку та полегшує переходи між різними частинами програми.

Ще однією перевагою цієї архітектури є простота в тестуванні. Оскільки логіка і представлення кожного екрана розділені, їх можна легко тестувати окремо, що полегшує виявлення та виправлення помилок. Крім того, завдяки використанню віджетів, можна використовувати автоматизовані тести для перевірки відображення та взаємодії з користувачем на кожному екрані.

					<i>КБ 01. 12 001. 00 ДП ПЗ</i>	Арк.
						22
Ізм.	Лист	№ докум.	Підпис	Дата		

На рис 1.8. зображено архітектури Flutter застосунку.

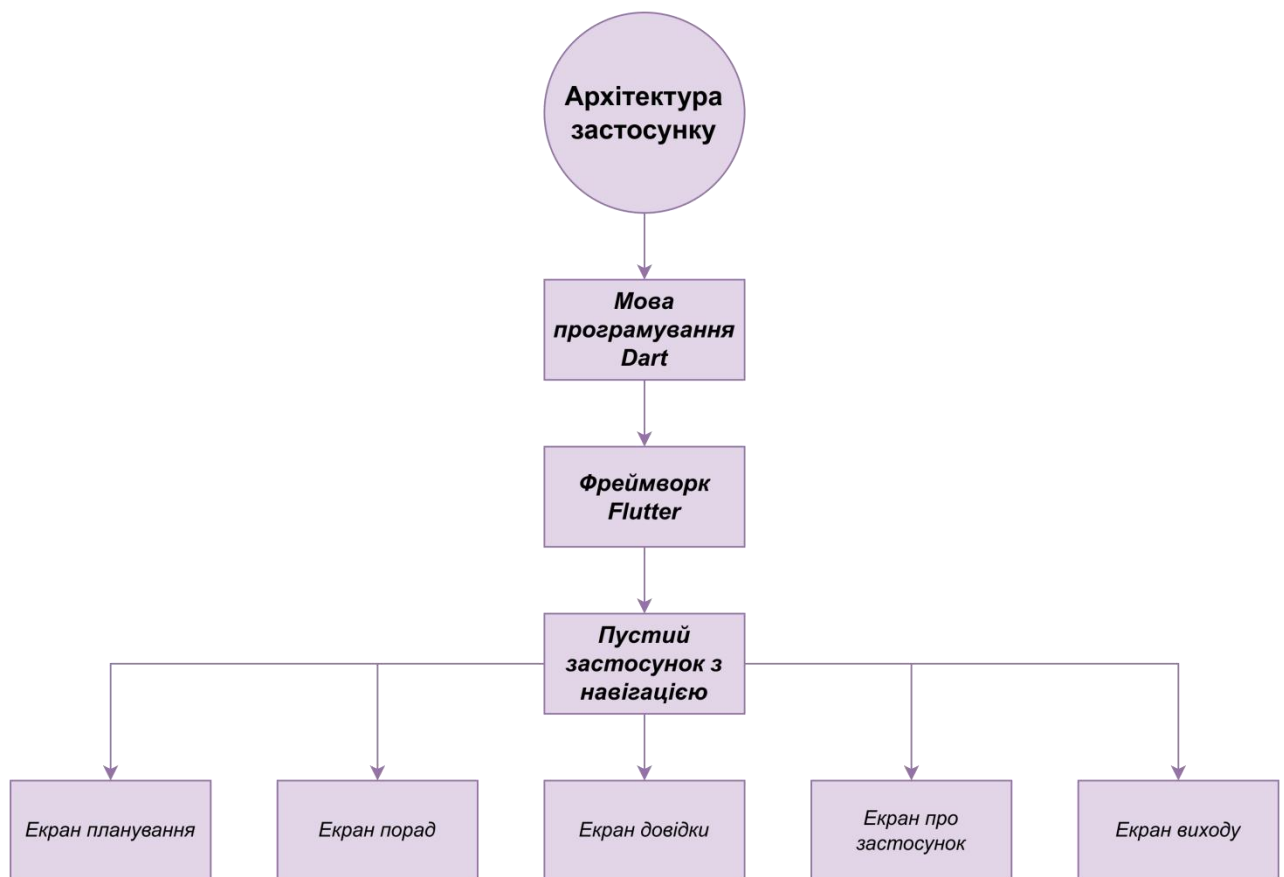


Рисунок 1.8. Архітектура Flutter застосунку

1.3 Реалізація мобільного застосунку

1.3.1 Середовище розробки

У якості середовища розробки було обрано DartPad.

DartPad - це інтерактивне середовище програмування в браузері для мови програмування Dart, яке дозволяє швидко пробувати, тестувати та демонструвати код без необхідності встановлення будь-якого додаткового програмного забезпечення.

Основні властивості DartPad включають можливість виконувати Dart код безпосередньо в браузері, автоматичне форматування коду та надання підказок під час введення.

Це середовище ідеально підходить для швидкого прототипування, вивчення мови Dart та демонстрації коду під час співпраці над проєктами.

Завдяки DartPad можна легко ділитися кодом з колегами, студентами або співробітниками для спільної роботи та отримання зворотного зв'язку.

На рис 1.9 зображено хмарне середовище розробки DartPad.

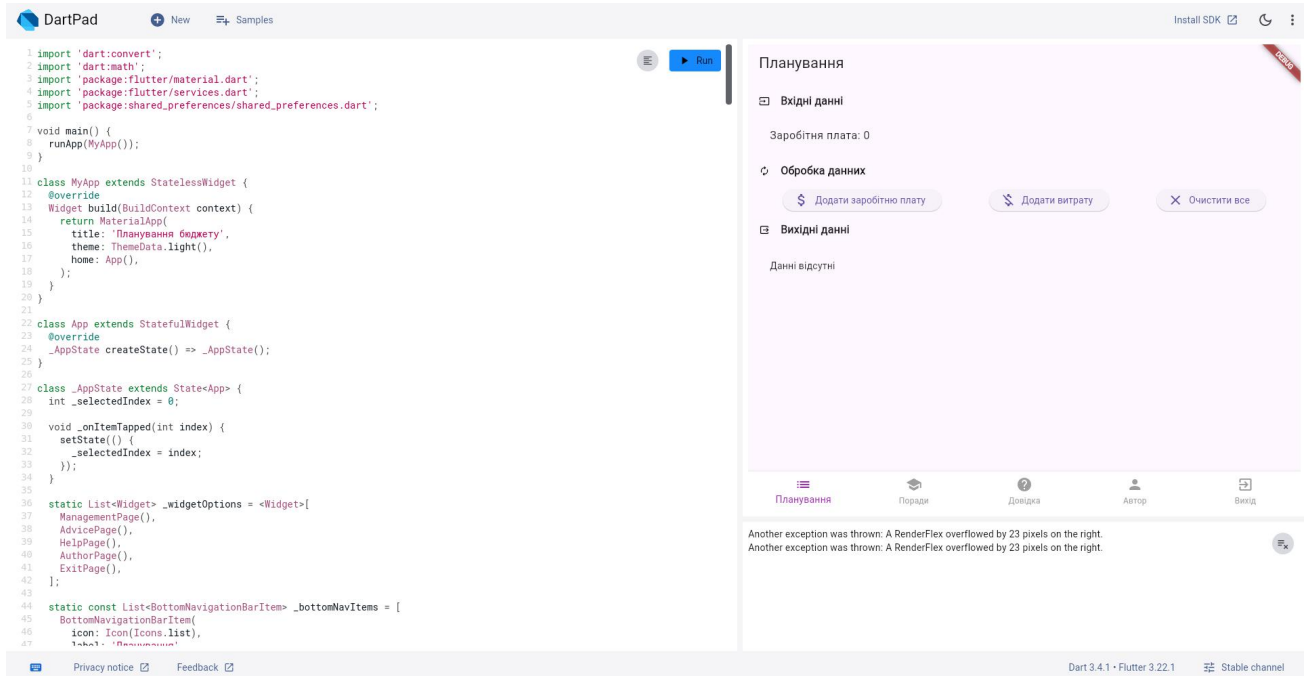


Рисунок 1.9. Середовище розробки DartPad

Розглянемо початковий код мобільного застосунку.

Нижче наведено код з підключенням бібліотек та з точкою входу в застосунок.

```
import 'dart:convert';
import 'dart:math';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:shared_preferences/shared_preferences.dart';

void main() {
  runApp(MyApp());
}
```

Даний код є початковим кодом мобільного застосунку, написаного за допомогою фреймворку Flutter для розробки інтерфейсу користувача. Він використовує пакети Dart для обробки даних, рандомізації та доступу до спільних налаштувань, а також пакети Flutter для побудови інтерфейсу та роботи з ресурсами пристрою.

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						24
Ізм.	Лист	№ докум.	Підпис	Дата		

Головна функція main() запускає основний віджет застосунку MyApp(), який буде містити усі інші віджети та логіку застосунку.

1.3.2 Розробка бази для застосунку

Розглянемо код бази для застосунку.

Нижче наведено код зі створенням породжувального класу застосунку “App”.

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Планування бюджету',
      theme: ThemeData.light(),
      home: App(),
    );
  }
}

class App extends StatefulWidget {
  @override
  _AppState createState() => _AppState();
}

class _AppState extends State<App> {
  int _selectedIndex = 0;

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  static List<Widget> _widgetOptions = <Widget>[
    ManagementPage(),
    AdvicePage(),
    HelpPage(),
    AuthorPage(),
    ExitPage(),
  ];

  static const List<BottomNavigationBarItem> _bottomNavItems = [
    BottomNavigationBarItem(
      icon: Icon(Icons.list),
      label: 'Планування',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.school),
      label: 'Поради',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.help),
      label: 'Довідка',
    ),
    BottomNavigationBarItem(

```

```

        icon: Icon(Icons.person),
        label: 'Автор',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.exit_to_app),
        label: 'Вихід',
      ),
    ],
  );

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          _bottomNavItems[_selectedIndex].label!,
        ),
      ),
      body: _widgetOptions.elementAt(_selectedIndex),
      bottomNavigationBar: BottomNavigationBar(
        type: BottomNavigationBarType.fixed,
        backgroundColor: Colors.white,
        selectedItemColor: Colors.purple,
        unselectedItemColor: Colors.grey,
        items: _bottomNavItems,
        currentIndex: _selectedIndex,
        onTap: _onItemTapped,
      ),
    );
  }
}

```

Цей код визначає структуру мобільного застосунку "Планування бюджету" на базі Flutter.

Клас MyApp є статичним віджетом, що налаштовує тему застосунку та визначає його головну сторінку App.

Клас App є StatefulWidget, який створює об'єкт стану _AppState, де відбувається основна логіка роботи застосунку.

У стані _AppState визначається поточний вибраний індекс _selectedIndex, що відповідає активному елементу нижньої навігаційної панелі.

Метод _onItemTapped змінює стан застосунку при натисканні на елементи навігаційної панелі. Для кожного з п'яти елементів панелі визначені відповідні сторінки (ManagementPage, AdvicePage, HelpPage, AuthorPage, ExitPage).

У методі build для Scaffold визначається верхній AppBar з динамічним заголовком та тіло, яке змінюється в залежності від вибраного індексу, а також налаштовується нижня навігаційна панель.

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						26
Ізм.	Лист	№ докум.	Підпис	Дата		

1.3.3 Розробка екрану “Планування”

Розглянемо код класу, що відповідає за екран “Планування”.

Нижче наведено код класу, що відповідає за екран “Планування”.

```
class ManagementPage extends StatefulWidget {
  @override
  _ManagementPageState createState() => _ManagementPageState();
}

class _ManagementPageState extends State<ManagementPage> {
  double salary = 0;
  Map<String, double> expenses = {};

  @override
  void initState() {
    super.initState();
    _loadData();
  }

  _loadData() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    String expensesString = prefs.getString('expenses') ?? '{}';
    setState(() {
      salary = prefs.getDouble('salary') ?? 0;
      expenses = Map<String, double>.from(json
        .decode(expensesString)
        .map((key, value) => MapEntry(key, value.toDouble())));
    });
  }

  _saveData() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.setDouble('salary', salary);
    await prefs.setString('expenses', json.encode(expenses));
  }

  void _clearData() async {
    setState(() {
      salary = 0;
      expenses = {};
    });
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.setDouble('salary', salary);
    await prefs.setString('expenses', json.encode(expenses));
  }

  void _showAddExpenseDialog() {
    String category = '';
    String amount = '';
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('Додати витрату'),
          content: SingleChildScrollView(
            child: Column(
              children: [
                TextField(

```

```

        decoration: InputDecoration(labelText: 'Категорія'),
        onChanged: (value) {
          category = value;
        },
      ),
      TextField(
        decoration: InputDecoration(labelText: 'Сума'),
        onChanged: (value) {
          amount = value;
        },
      ),
    ],
  ),
),
actions: [
  TextButton(
    onPressed: () {
      setState(() {
        if (amount.isNotEmpty && category.isNotEmpty) {
          double expenseAmount = double.parse(amount);
          if (expenses.containsKey(category)) {
            expenses[category] =
              (expenses[category] ?? 0) + expenseAmount;
          } else {
            expenses[category] = expenseAmount;
          }
          salary -= expenseAmount;
          _saveData();
        }
      });
      Navigator.of(context).pop();
    },
    child: Text('Додати'),
  ),
  TextButton(
    onPressed: () {
      Navigator.of(context).pop();
    },
    child: Text('Відмінити'),
  ),
],
);
},
);
}

void _showAddSalaryDialog() {
  String newSalary = '';
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Додати заробітню плату'),
        content: SingleChildScrollView(
          child: Column(
            children: [
              TextField(
                decoration: InputDecoration(labelText: 'Заробітня плата'),
                onChanged: (value) {
                  newSalary = value;
                },
              ),
            ],
          ),
        ),
      );
    },
  );
}

```

```

        ),
      ],
    ),
  ),
  actions: [
    TextButton(
      onPressed: () {
        if (newSalary.isNotEmpty) {
          setState(() {
            salary += double.parse(newSalary);
            _saveData();
          });
        }
        Navigator.of(context).pop();
      },
      child: Text('Додати'),
    ),
    TextButton(
      onPressed: () {
        Navigator.of(context).pop();
      },
      child: Text('Відмінити'),
    ),
  ],
);
},
);
}

```

```

@override
Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: [
        Row(
          children: [
            Icon(
              Icons.input,
              color: Colors.black,
              size: 16.0,
            ),
            SizedBox(width: 16),
            Text(
              'Вхідні данні',
              style: TextStyle(
                fontSize: 16.0,
                fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
        SizedBox(height: 16),
        ListTile(
          title: Text('Заробітня плата: $salary'),
        ),
        SizedBox(height: 16),
        Row(
          children: [
            Icon(

```

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						29
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        Icons.autorenew,
        color: Colors.black,
        size: 16.0,
      ),
      SizedBox(width: 16),
      Text(
        'Обробка даних',
        style: TextStyle(
          fontSize: 16.0,
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
  ),
  SizedBox(height: 16),
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceAround,
    children: [
      ElevatedButton.icon(
        onPressed: _showAddSalaryDialog,
        icon: Icon(Icons.attach_money),
        label: Text('Додати заробітню плату'),
      ),
      ElevatedButton.icon(
        onPressed: _showAddExpenseDialog,
        icon: Icon(Icons.money_off),
        label: Text('Додати витрату'),
      ),
      ElevatedButton.icon(
        onPressed: _clearData,
        icon: Icon(Icons.clear),
        label: Text('Очистити все'),
      ),
    ],
  ),
  SizedBox(height: 16.0),
  Row(
    children: [
      Icon(
        Icons.output,
        color: Colors.black,
        size: 16.0,
      ),
      SizedBox(width: 16),
      Text(
        'Вихідні данні',
        style: TextStyle(
          fontSize: 16.0,
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
  ),
  SizedBox(height: 16.0),
  SimpleLinearChart(expenses: expenses),
  Expanded(
    child: ListView.builder(
      itemCount: expenses.length,
      itemBuilder: (BuildContext context, int index) {
        String category = expenses.keys.elementAt(index);
        double amount = expenses.values.elementAt(index);

```

```

        return ListTile(
            title: Text('$category: $amount'),
        );
    },
),
),
1,
),
);
}
}

```

Цей код визначає сторінку `ManagementPage`, яка є `StatefulWidget`, для управління заробітною платою та витратами користувача. Клас `_ManagementPageState` зберігає поточну заробітну плату та витрати у вигляді мапи категорій та сум. У методі `initState` викликається `_loadData`, який завантажує збережені дані з `SharedPreferences` і оновлює стан виджету. Метод `_saveData` зберігає поточні значення заробітної плати та витрат до `SharedPreferences`.

Метод `_clearData` скидає всі дані до нульових значень та очищає збережені дані в `SharedPreferences`. Метод `_showAddExpenseDialog` відкриває діалогове вікно, де користувач може додати нову витрату, вказавши категорію та суму. При натисканні кнопки "Додати" у діалоговому вікні, витрата додається до мапи витрат, оновлюється заробітна плата та викликається `_saveData`.

Метод `_showAddSalaryDialog` відкриває діалогове вікно для додавання нової заробітної плати. Після введення суми і натискання кнопки "Додати" заробітна плата оновлюється та зберігається через `_saveData`. Метод `build` створює інтерфейс сторінки з різними елементами, включаючи заголовки, кнопки та списки.

На сторінці є секція "Вхідні данні", яка показує поточну заробітну плату користувача. Користувач може додати нову заробітну плату, нову витрату або очистити всі дані за допомогою відповідних кнопок у секції "Обробка даних". Секція "Вихідні данні" містить графік, що відображає витрати користувача за категоріями.

Графік створюється за допомогою виджету `SimpleLinearChart`, який отримує мапу витрат для побудови графіка. Під графіком розташований список,

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						31
Ізм.	Лист	№ докум.	Підпис	Дата		

що відображає кожну категорію витрат та суму витрат у цій категорії. Для кожної категорії створюється елемент ListTile з назвою категорії та сумою витрат.

Таким чином, цей код забезпечує функціонал для відстеження заробітної плати та витрат користувача, а також дозволяє додавати нові витрати та заробітну плату, зберігати ці дані локально і відображати їх у вигляді графіка та списку.

На рис 1.10 зображено алгоритм додавання категорії.

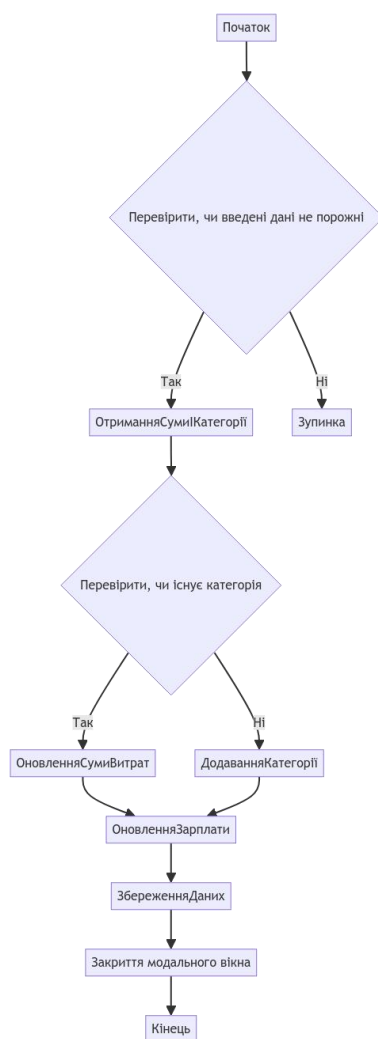


Рисунок 1.10. Алгоритм додавання категорії.

1.3.4 Розробка екрану “Поради”

Розглянемо код класу, що відповідає за екран “Поради”.

Нижче наведено код класу, що відповідає за екран “Поради”.

```

class AdvicePage extends StatelessWidget {
  final List<String> adviceList = [
    "Розгляньте можливість розподілу бюджету на категорії, такі як їжа, транспорт,
    розваги тощо.",
    "Спробуйте встановити місячні цілі з економії та тримайтеся їх, покладаючи
    гроші відразу по отриманню зарплати.",
    "Використовуйте бюджетні додатки або аркуші електронних таблиць для ведення
    обліку витрат та доходів.",
    "Аналізуйте свої витрати щомісяця, щоб визначити, на яких областях можна
    зекономити.",
    "Не забувайте про екстрені фонди та заощадження на непередбачені витрати.",
    "Розгляньте можливість інвестування частини доходу для майбутнього зростання
    капіталу."
  ];

  String getRandomAdvice() {
    final random = Random();
    return adviceList[random.nextInt(adviceList.length)];
  }

  @override
  Widget build(BuildContext context) {
    String advice = getRandomAdvice();
    return Center(
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              advice,
              style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
              textAlign: TextAlign.justify,
            ),
          ],
        ),
      ),
    );
  }
}

```

Цей код визначає сторінку AdvicePage, яка є StatelessWidget, для відображення випадкової поради з управління фінансами. Сторінка містить список порад у вигляді рядків, що зберігаються в adviceList. Метод getRandomAdvice використовується для вибору випадкової поради з цього списку за допомогою генератора випадкових чисел. У методі build створюється інтерфейс сторінки, який містить одну випадкову пораду.

При побудові сторінки викликається getRandomAdvice, щоб отримати випадкову пораду, яка зберігається в змінній advice. Інтерфейс сторінки складається з віджета Center, який центрує вміст на екрані, та віджета Padding

для додавання відступів. Внутрішньо використовується Column, щоб розмістити текст з порадою по центру сторінки.

Текст поради відображається у віджеті Text, з заданим стилем, що включає розмір шрифту та товщину. Текст також вирівнюється за допомогою textAlign: TextAlign.justify, щоб забезпечити гарний вигляд на сторінці. Отже, цей код відображає одну випадкову пораду з списку, що допомагає користувачеві з управлінням фінансами.

1.3.5 Розробка екрану “Довідка”

Розглянемо код класу, що відповідає за екран “Довідка”.

Нижче наведено код класу, що відповідає за екран “Довідка”.

```
class HelpPage extends StatelessWidget {
  @override
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: [Center(
          child: Text(
            'Облік власних фінансів є важливим з кількох причин:\n\n'
            '1. Контроль витрат: Ведення обліку дозволяє бачити, куди йдуть ваші гроші, і допомагає контролювати витрати.\n\n'
            '2. Планування бюджету: Знання своїх доходів і витрат дозволяє ефективно планувати бюджет і уникати фінансових труднощів.\n\n'
            '3. Досягнення фінансових цілей: Правильне управління фінансами допомагає досягати короткострокових і довгострокових фінансових цілей.\n\n'
            '4. Зменшення стресу: Коли ви знаєте свій фінансовий стан і маєте план, це зменшує стрес і робить вас впевненішими у майбутньому.\n\n'
            '5. Прийняття обґрунтованих рішень: Ведення фінансового обліку допомагає приймати обґрунтовані фінансові рішення, зокрема щодо інвестицій та великих покупок.\n\n'
            'Загалом, облік власних фінансів є основою фінансової стабільності та добробуту.',
            style: TextStyle(fontSize: 14),
            textAlign: TextAlign.justify,
          ),
        ),
      ],
    );
  }
}
```

Цей код визначає сторінку HelpPage, яка є StatelessWidget, і надає корисну інформацію щодо управління особистими фінансами. У методі build створюється інтерфейс сторінки з відступами за допомогою виджета Padding, що

додає рівномірні відступи по всьому контенту. Всередині Padding розташований виджет Column, який містить один елемент Center для центрування тексту.

Центрований текст містить інформацію про важливість ведення обліку власних фінансів, розбита на пункти. Текст відображає п'ять ключових причин для ведення фінансового обліку: контроль витрат, планування бюджету, досягнення фінансових цілей, зменшення стресу та прийняття обґрунтованих рішень. Використовується стиль TextStyle для встановлення розміру шрифту, що дорівнює 14.

Текст вирівнюється за допомогою textAlign: TextAlign.justify, що забезпечує рівномірне вирівнювання тексту по ширині. Таким чином, сторінка HelpPage інформує користувача про переваги ведення фінансового обліку, допомагаючи їм зрозуміти, як це може покращити їх фінансову стабільність та добробут.

1.3.6 Розробка екрану “Автор”

Розглянемо код класу, що відповідає за екран “Автор”.

Нижче наведено код класу, що відповідає за екран “Автор”.

```
class AuthorPage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Center(  
      child: Text('Попов О. Р., 2024',  
        style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold)),  
    );  
  }  
}
```

Цей код визначає сторінку AuthorPage, яка є StatelessWidget, для відображення інформації про автора. У методі build створюється інтерфейс сторінки, яка центрує текст на екрані за допомогою виджета Center. Віджет Text містить текст "Попов О. Р., 2024".

Стиль тексту задається за допомогою TextStyle, який встановлює розмір шрифту 16 і товщину шрифту FontWeight.bold. Таким чином, текст відображається крупним та жирним шрифтом. Цей код створює просту сторінку, яка відображає ім'я автора та рік.

Сторінка використовується для надання інформації про автора мобільного застосунку.

1.3.7 Розробка екрану “Вихід”

Розглянемо код класу, що відповідає за екран “Автор”.

Нижче наведено код класу, що відповідає за екран “Вихід”.

```
class ExitPage extends StatelessWidget {  
  void _exitApp(BuildContext context) {  
    // Close the app  
    SystemNavigator.pop();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Center(  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: [  
          ElevatedButton(  
            onPressed: () => _exitApp(context),  
            child: Text('Закрити застосунок'),  
          ),  
        ],  
      ),  
    );  
  }  
}
```

Цей код визначає метод `_exitApp` та створює інтерфейс сторінки для виходу з мобільного застосунку. Метод `_exitApp` використовує `SystemNavigator.pop()`, щоб закрити застосунок. У методі `build` створюється інтерфейс сторінки, який центрує контент за допомогою виджета `Center`.

Всередині `Center` розташований `Column`, який вирівнює свої дочірні елементи по центру за допомогою `mainAxisAlignment: MainAxisAlignment.center`. У `Column` міститься одна кнопка `ElevatedButton`. При натисканні на кнопку викликається метод `_exitApp`, який закриває застосунок.

Текст на кнопці "Закрити застосунок" вказує на її функціональність. Цей код забезпечує користувача можливістю закрити застосунок натисканням на кнопку. Весь інтерфейс сторінки складається з одного централізованого елемента `ElevatedButton`.

Цей підхід гарантує простоту та зрозумілість для користувача, який хоче вийти із застосунку.

1.3.8 Будування графіку

Розглянемо код класу, що відповідає за малювання графіку.

Нижче наведено код класу, що відповідає за малювання графіку.

```
class SimpleLinearChart extends StatelessWidget {
  final Map<String, double> expenses;

  SimpleLinearChart({required this.expenses});

  @override
  Widget build(BuildContext context) {
    final List<String> labels = [];
    final List<double> values = [];
    expenses.forEach((key, value) {
      if (value > 0) {
        labels.add(key);
        values.add(value);
      }
    });
    if (values.isEmpty) {
      return Container(
        height: 200,
        width: double.infinity,
        padding: EdgeInsets.all(16.0),
        child: Text('Данні відсутні'),
      );
    }

    final double maxValue = values.reduce((value, element) => value > element ?
value : element);

    return Container(
      height: 200,
      width: double.infinity,
      padding: EdgeInsets.all(16.0),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        crossAxisAlignment: CrossAxisAlignment.end,
        children: List.generate(labels.length, (index) {
          return Column(
            mainAxisAlignment: MainAxisAlignment.end,
            children: [
              Text(labels[index]),
              Container(
                width: 20,
                height: (values[index] / maxValue) * 150, // Adjusted to fit
within the chart area
                color: Colors.purple,
              ),
            ],
          );
        })),
    ),
  ),
}
```

```
);  
}
```

Цей код визначає віджет "SimpleLinearChart", який будує просту лінійну діаграму витрат на основі переданого словника витрат.

У методі "build" створюються списки міток (labels) та значень (values) на основі переданого словника витрат, де ключі представляють категорії витрат, а значення - витрати за кожну категорію.

Якщо значення витрат в словнику відсутні або дорівнюють 0, відображається текст "Данні відсутні" в контейнері з висотою 200 та шириною, що займає всю доступну ширину.

У протилежному випадку, розраховується максимальне значення витрат (maxValue), після чого для кожної категорії витрат будується колонка, де висота колонки відповідає відносній величині витрат за цю категорію.

Результуючий віджет "SimpleLinearChart" відображає горизонтальну лінійну діаграму, де кожна колонка представляє категорію витрат, а висота колонки відповідає величині витрат за цю категорію в порівнянні з іншими категоріями.

1.4 Мануальне тестування (Manual QA)

Мануальне тестування мобільного застосунку являє собою процес, під час якого тестувальники вручну перевіряють функціональність, інтерфейс та взаємодію додатку з користувачем без використання автоматизованих скриптів або інструментів.

Під час мануального тестування, тестувальник виконує різні дії в застосунку, спостерігає за реакцією програми та перевіряє відповідність очікуванням.

Цей процес дозволяє виявити проблеми, які можуть бути важко автоматизувати, такі як аспекти взаємодії з користувачем, а також оцінити загальний вигляд та відчуття застосунку.

Мануальне тестування є важливою складовою тестування мобільних додатків, оскільки воно дозволяє виявити проблеми, які можуть бути важко

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						38
Ізм.	Лист	№ докум.	Підпис	Дата		

виявити за допомогою автоматизованих засобів, і забезпечує високу якість та коректність додатку перед його випуском.

1.4.1 Тестування екрану “Планування”

Екран "Планування" відповідає за управління особистими фінансами у застосунку. На цьому екрані користувач може вводити дані про свою заробітну плату, витрати та інші доходи.

Завдяки можливості додавання заробітної плати і витрат, користувач може вести облік своїх фінансів та контролювати витрати.

Після введення даних, на екрані будується список з графіком, який візуалізує розподіл доходів та витрат за певний період часу, що допомагає користувачу краще розуміти свою фінансову ситуацію та планувати подальші дії.

На рис 1.11 зображено екран “Планування” без даних.

Планування

DEBUG

Вхідні данні

Заробітня плата: 0

Обробка даних

\$ Додати заробітню плату

\$ Додати витрату

X Очистити все

Вихідні данні

Данні відсутні

Планування

Поради

Довідка

Автор

Вихід

Рисунок 1.11. Екран “Планування” без даних

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						40
Ізм.	Лист	№ докум.	Підпис	Дата		

На рис 1.12 зображено модальне вікно з додаванням заробітної плати.

Планування

Вхідні данні

Заробітня плата: 0

Обробка даних

Додати заробітню плату

Додати витрату

Очистити все

Вихідні данні

Данні відсутні

Додати заробітню плату

Заробітня плата

Додати Відмінити

Планування Поради Довідка Автор Вихід

Рисунок 1.12. Модальне вікно додавання заробітної плати

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						41
Ізм.	Лист	№ докум.	Підпис	Дата		

На рис 1.13 зображено екран “Планування” з доданими даними.

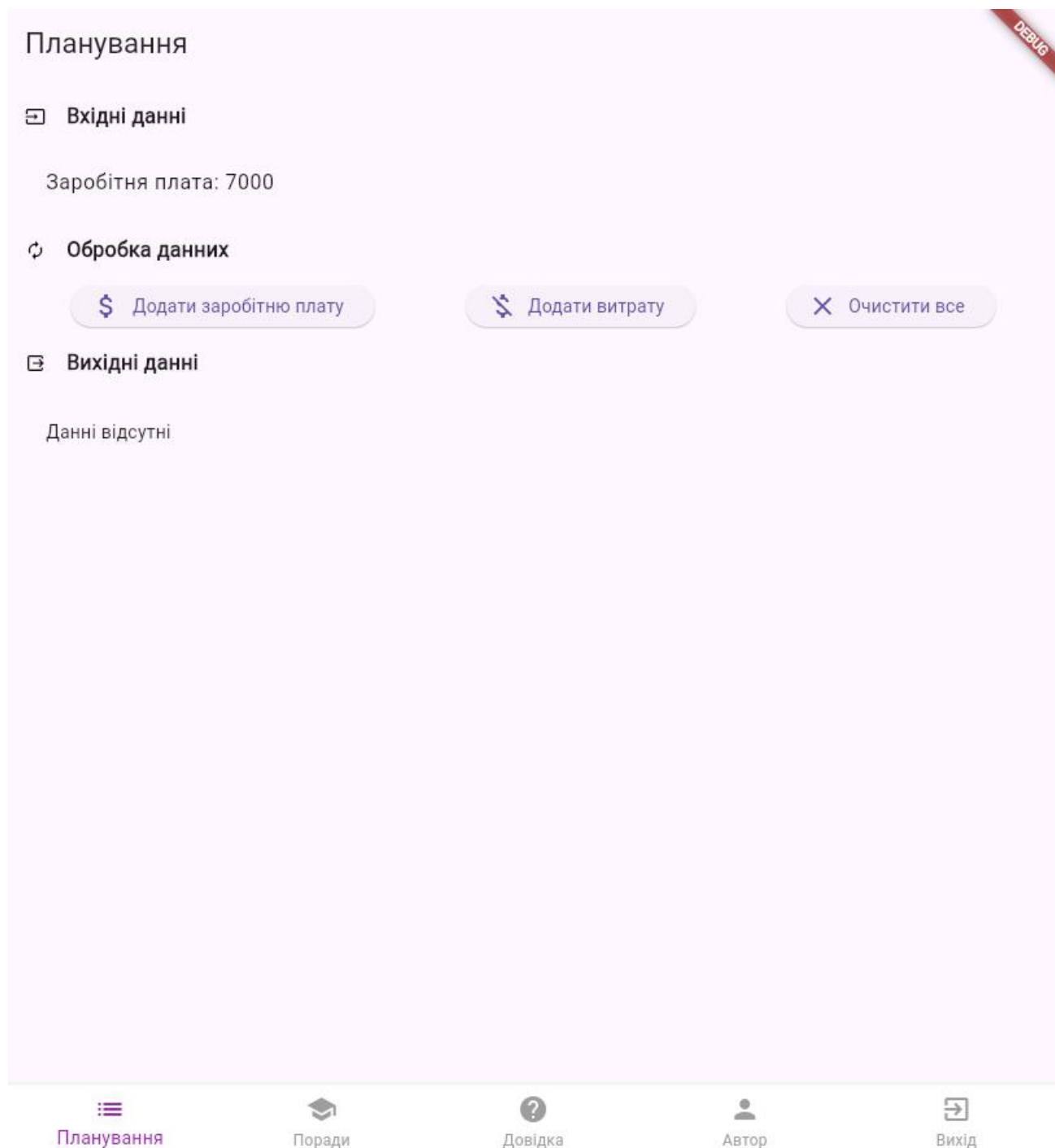


Рисунок 1.13. Екран “Планування” з доданими даними

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						42
Ізм.	Лист	№ докум.	Підпис	Дата		

На рис 1.14 зображено модальне вікно з додаванням витрати.

Планування

Вхідні данні

Заробітня плата: 7000

Обробка даних

Додати заробітню плату

Додати витрату

Очистити все

Вихідні данні

Данні відсутні

Додати витрату

Категорія

Сума

Додати Відмінити

Планування Поради Довідка Автор Вихід

Рисунок 1.14. Модальне вікно додавання витрати

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						43
Ізм.	Лист	№ докум.	Підпис	Дата		

На рис 1.15 зображено екран “Планування” з обробленими даними.

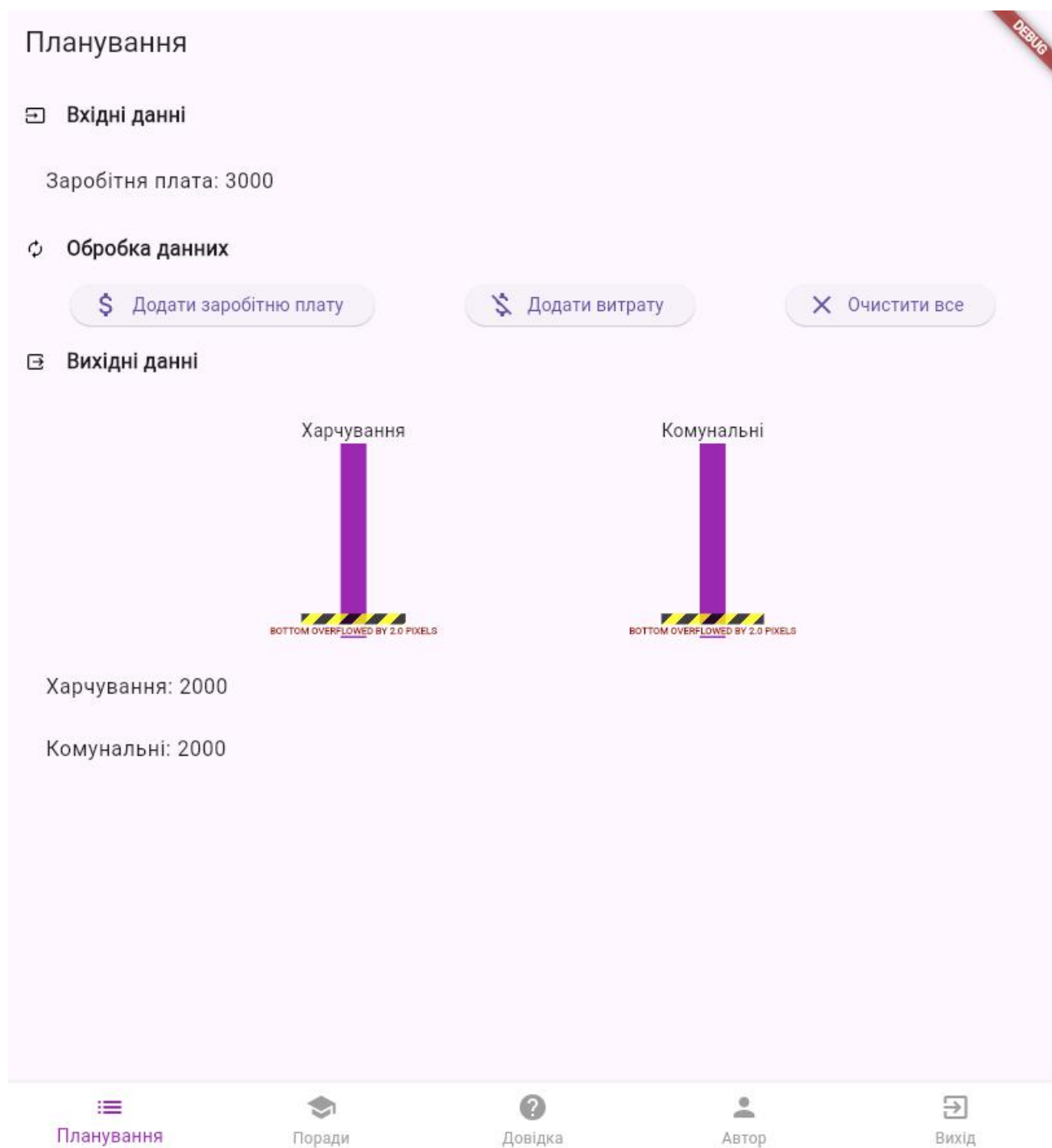


Рисунок 1.15. Екран “Планування” з обробленими даними

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						44
Ізм.	Лист	№ докум.	Підпис	Дата		

1.4.2 Тестування екрану “Поради”

Екран "Поради" надає користувачам корисні поради та рекомендації з управління особистими фінансами, які допомагають ефективніше розпоряджатися грошима та досягати фінансових цілей. Кожен перехід на цей екран супроводжується новою випадковою порадою.

На рис 1.16 зображено екран “Поради” з випадковою порадою.



Рисунок 1.16. Екран “Поради”

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						45
Ізм.	Лист	№ докум.	Підпис	Дата		

На рис 1.17 зображено екран “Поради” з наступною порадою.



Рисунок 1.17 Екран “Поради”

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						46
Ізм.	Лист	№ докум.	Підпис	Дата		

1.4.3 Тестування екрану “Довідка”

Екран "Довідка" надає користувачам інформацію та пояснення щодо різних аспектів функціонування застосунку, включаючи інструкції з використання, відповіді на часті запитання тощо.

На рис 1.18 зображено екран “Довідка”.



Рисунок 1.18. Екран “Довідка”

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						47
Ізм.	Лист	№ докум.	Підпис	Дата		

1.4.4 Тестування екрану “Автор”

Екран "Автор" містить інформацію про розробника мобільного застосунку застосунку.

На рис 1.19 зображено екран “Автор”.



Рисунок 1.19. Екран “Автор”

1.4.5 Тестування екрану “Вихід”

Екран "Вихід" дозволяє користувачам завершити роботу з застосунком шляхом закриття його і виходу до головного екрану пристрою або до іншого застосунку.

На рис 1.20 зображено екран “Вихід”.

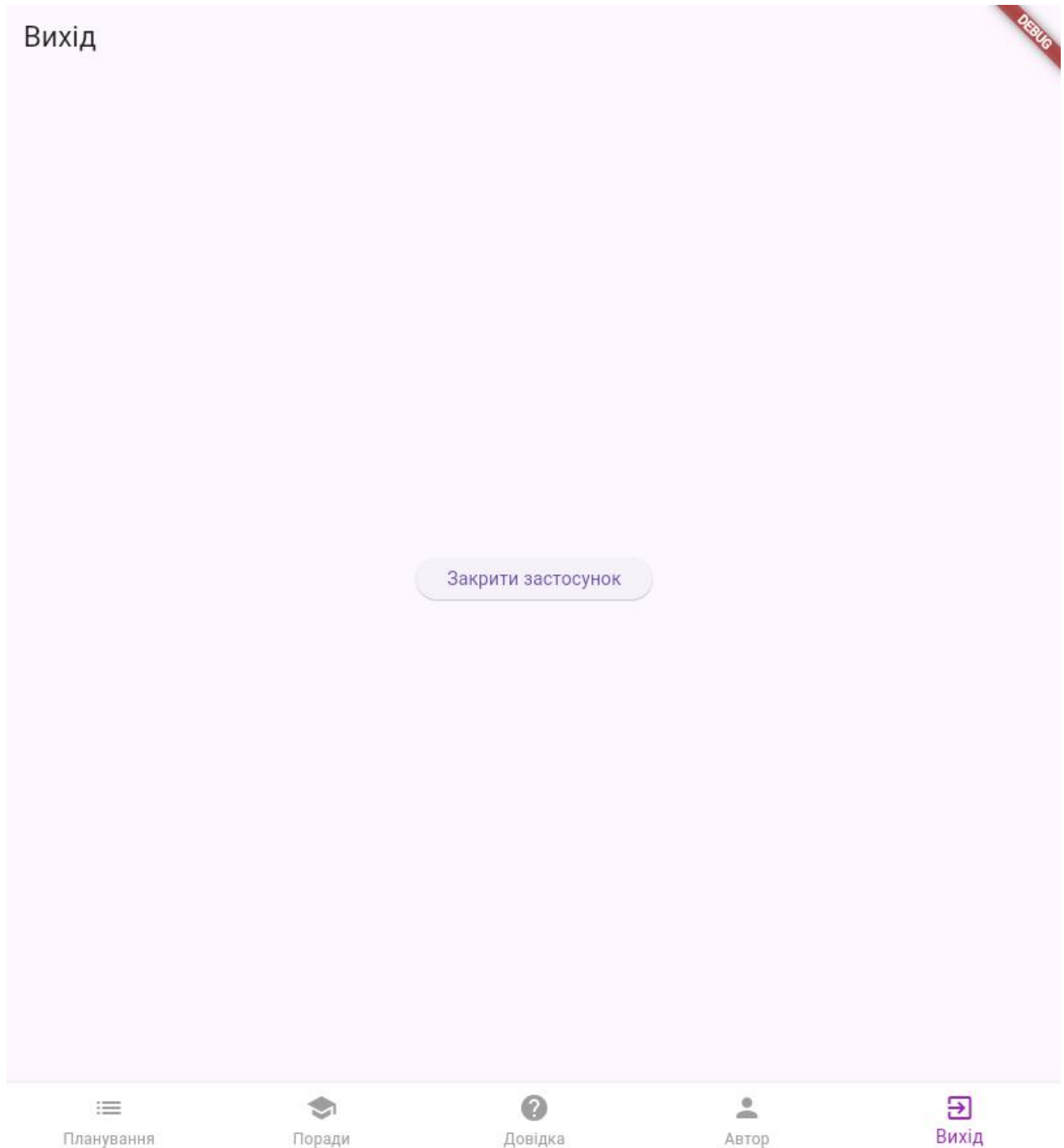


Рисунок 1.20. Екран “Вихід”

					КБ 01. 12 001. 00 ДП ПЗ	Арк.
						49
Ізм.	Лист	№ докум.	Підпис	Дата		

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

В даному дипломному проєкті було розроблено мобільний застосунок для обліку особистих фінансів, що ґрунтується на передових технологіях програмування та розробки. Дослідження показало, що впровадження такого застосунку має сенс у поліпшенні процесів управління фінансами, забезпечуючи зручність використання, ефективність та доступність.

Ефективність програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

Проведемо розрахунки визначення трудомісткості розробки та вартості даного програмного продукту.

2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_o , усл. машинних командах
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750

У табл. 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

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

Таблиця 2.2. Обсяг команд

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7 \div 0,8$): $T_{ар}=229 \times 0,8=183,2$ (люд/годин).

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

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

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

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

Для розрахунку необхідні наступні коефіцієнти:

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

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

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

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

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L ₁)	0,15	0,12	0,12
ТП (L ₂)	0,16	0,15	0,11
РП (L ₃)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення К _н
А	Принципово нові ПО	1,75 – 1,2
Б	ПО – розвиток визначеног параметричного ряду	1,0 – 0,8
В	ПО маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

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

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ}=T_a \cdot L_1 \cdot K_n = 183,2 \cdot 0,12 \cdot 0,7 = 15,39 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість розробки технічного проєкту

$$T_{ТП}=T_a \cdot L_2 \cdot K_n = 183,2 \cdot 0,11 \cdot 0,7 = 17,42 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки робочого проєкту

$$T_{rp} = T_a * L_3 * K_n * K_t = 183,2 * 0,61 * 0,7 * 0,7 = 54,76 \text{ (люд/годин)} \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{tz}=2$ (стр), розробка ТП $N_{tp}=49$ (стр), розробка робочого проєкту $N_{rp}=8$ (стр), пояснювальна записка відповідно $N_{pz}=20$ (стр).

Розрахунок зведений у табл. 2.6.

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
1.ТЗ	$T_{P_{TЗ}}=15,39$	$T_{KK}=0,7*N_{TЗ}=0,7*2=1,4$	$T_{HK}=0,15*N_{TЗ}=0,15*2=0,30$
2.Розробка ТП	$T_{P_{TP}}=14,12$	$T_{KK}=0,7*N_{TP}=0,7*49=34,3$	$T_{HK}=0,15*N_{TP}=0,15*49=7,35$
3.Розробка РП	$T_{P_{RP}}=54,76$	$T_{KK}=0,7*N_{RP}=0,7*8=5,6$	$T_{HK}=0,15*N_{RP}=0,15*8=1,2$
4.Розробка ПЗ	$T_{PЗ}=1,5**N_{PЗ}=1,5*20=30,0$	$T_{KK}=0,7*N_{PЗ}=0,7*20=14,00$	$T_{HK}=0,15*N_{PЗ}=0,15*20=3,00$
Усього, в т.ч.:	181,42		
- на розробку	$T_p=114,27$		
- контроль		$T_{KK}=55,3$	
- нормоконтроль			$T_{HK}=11,85$

2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, загальні витрати на розробку ПП. Розрахунок основної заробітної плати виконавців приведений у табл. 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2024» встановлено мінімальну заробітну плату у місячному розмірі з 1 квітня 2024 року - 8000 гривень; мінімальну погодинну тарифну ставку – 46,00 грн.

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудовісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	114,27	50,00	5713,50
2.Контроль керівника	55,30	80,00	4424,00
3.Нормоконтроль	11,85	80,00	948,00
Усього	-	-	30= 11085,50

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

Таблиця 2.8. Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	79	4.00	316,0
Транспортно – заготівельні Витрати (10%)				$B_{тр_з} = 0,1 \times B_{м1} = 0,1 \times 316 = 31,0$
Усього				$B_{м} = B_{м1} + B_{тр_з} = 347.0$

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

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	347,0	B_m (див. табл. 2.8)
2. Основна заробітна плата	11085,50	Z_o (див. табл. 2.7)
3. Додаткова заробітна плата	1108,55	$Z_d = 0,1 \times Z_o = 11085,50 \times 0,1$
4. Відрахування до єдиного фонду соціального внеску	2682,70	$B_{\epsilon.c.v.} = 0,22 \times (Z_o + Z_d) = 0,22 \times (11085,50 + 1108,55)$
5. Накладні витрати	4434,20	$B_{нак.} = 0,4 \times Z_o = 0,4 \times 11085,50$
6. Повна собівартість	19657,95	$C_{пов} = B_m + Z_o + Z_d + B_{\epsilon.c.v.} + B_{нак.} = 347,0 + 11085,50 + 1108,55 + 2682,70 + 4434,20$

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

$$П = (C_{пов} \cdot P) / 100 = (19657,95 \cdot 10) / 100 = 1965,80 \text{ грн. (2.4)}$$

Де p – плановий рівень рентабельності (10-20%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{п} + П = 19657,95 + 1965,80 = 21623,75 \text{ грн. (2.5)}$$

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

$$Ц_p = Ц_o + ПДВ = 21623,75 + 21623,75 \cdot 0,2 = 25948,49 \text{ грн. (2.6)}$$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Вступ

Сучасний розвиток технічного та технологічного стану виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. Сьогодні, напевно, важко уявити компанію, господарська діяльність в якій здійснювалась би без використання комп'ютерної техніки. Через масовий характер робіт, що виконуються працівниками за допомогою комп'ютера, законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці при роботі з комп'ютером.

В розділі охорони праці дипломного проекту розглядається питання розробки мобільного додатку для обліку особистих фінансів. Тому об'єктом дослідження беремо безпеку праці на робочому місці програміста.

3.2 Аналіз умов праці й забезпечення безпеки при виконанні основних видів робіт на об'єкті дипломного проектування

Аналіз умов праці показує, що на працівників можуть негативно впливати наступні фізичні та психофізіологічні фактори:

- підвищені або знижені температура, вологість повітря робочої зони;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- підвищені іонізація повітря та рівень електромагнітних випромінювань;
- нервово-психічні та фізичні перевантаження.

3.3 Гігієнічні вимоги до виробничого середовища

На робочому місці програміста повинні бути створені умови для безпечної та високопродуктивної праці.

					<i>КБ 01. 12 003. 00 ДП ПЗ</i>	Арк.
						56
Ізм.	Лист	№ докум.	Підпис	Дата		

3.3.1 Вимоги до приміщення експлуатації ПК

Приміщення, в яких планується установка та подальша робота з комп'ютером, повинні відповідати проектній документації будинку. Роботодавець повинен враховувати санітарні нормативи освітлення, вимоги до параметрів мікроклімату (температура, відносна вологість), ступеня і сили вібрації, звукового шуму і вогнестійкості приміщення, а також характеристики електромагнітного, ультрафіолетового та інфрачервоного полів

Роботодавцю заборонено установлювати комп'ютери в приміщеннях, розташованих у підвалах будинків. Для уникнення можливих аварій та замикань, поряд з приміщеннями, де вестиметься робота з комп'ютером (над чи під ними), також не дозволяється проведення робіт, що потребують здійснення надмірно вологих технологічних процесів.

3.3.2 Гігієнічні вимоги до параметрів повітря приміщень із ПК

Робота за енерговитратами відноситься до категорії легких робіт 1а, 1б, тому повинні дотримуватися наступні вимоги згідно ДСанПіН 3.3.2.-007-98.

У таблиці 3.1 наведено норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕМ.

Таблиця 3.1. Норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕМ

Пора року	Категорія робіт	Температура повітря, °С, не більше	Відносна вологість повітря %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
	Легка-1б	21-23	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1
	Легка-1б	22-24	40-60	0,1

Рівні позитивних і негативних іонів у повітрі приміщень з ВДТ мають відповідати санітарно-гігієнічним нормам № 2152-80.

У таблиці 3.2 наведено санітарно-гігієнічні нормам № 2152-80.

Таблиця 3.2. Санітарно-гігієнічні нормам № 2152-80

Рівні	Число іонів в 1 см ³ повітря	Число іонів в 1 см ³ повітря
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

Для підтримки в приміщенні нормального, що відповідає гігієнічним вимогам, складу повітря, видалення з нього шкідливих речовин використовують вентиляцію. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення і видаляється внаслідок різниці температур. Але вона має низку недоліків. Тому дипломним проектом передбачено встановлення припливно-витяжної системи вентиляції, а також можливе застосування кондиціонерів.

3.3.3 Виробниче освітлення

Освітлення приміщення має природне та штучне походження. Природне освітлення подається через віконні прорізи, бокове. Для штучного освітлення у приміщенні використовуються люмінесцентні лампи, які в порівнянні з лампами розжарювання мають ряд істотних переваг. Так за спектральним складом світла вони близькі до природного світла, мають підвищену світлову віддачу, триваліший термін служби. Норма освітленості на робочих місцях складає 300-500лк.

3.3.4 Електробезпека

Значення сили струму, що проходить через організм людини, залежить від напруги, під якою перебуває людина й від опору ділянки тіла, до якого

прикладена ця напруга. Джерелом живлячої напруги є мережа змінного струму з напругою 229В, на яку поширюється ГОСТ 25861-83.

Для попередження поразок електричним струмом необхідно чітко й у повному обсязі виконувати правила провадження робіт і правил технічної експлуатації. Необхідно виключити можливість доступу оператора до частин устаткування, що працює під небезпечною напругою, до неізольованим частинам, призначеним для роботи при малій напрузі й не підключеним до захисного заземлення, а також підводити електроживлення до ПЕОМ від розетки за допомогою спеціальної вилки із заземлюючим контактом.

3.3.5 Організація робочих місць із ПК

Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкцій всіх елементів робочого місця та їх взаємного розташування, ергономічним вимогам, з урахуванням характеру і особливостей трудової діяльності (ДСанПіН 3.3.2.-007-98).

Конструкція робочого місця й взаємне розташування всіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи. Конструкція робочих меблів дає можливість забезпечувати можливість індивідуального регулювання їх відповідно до потреб працівника для підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей, на відстані близько 70 см, що укладається в припустимі рамки від 60 до 90 см. Частота мерехтіння екрана дорівнює 100 Гц, що відповідає умові більше 70 Гц.

На рисунку 3.1 зображено конструкцію робочого місця користувача ПК.

					<i>КБ 01. 12 003. 00 ДП ПЗ</i>	Арк.
						59
Ізм.	Лист	№ докум.	Підпис	Дата		



Рисунок 3.1. Конструкція робочого місця користувача ПК

Для зниження нервово-емоційного напруження, стомлювання, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільно впроваджувати виконання комплексу вправ, які наведені у Державних санітарних правилах і нормах роботи з візуальними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98.

3.4 Пожежна безпека

Під пожежною безпекою розуміють систему державних і суспільних заходів, спрямованих на охорону від вогню людей і власності. Пожежна безпека приміщень, що мають електричні мережі, регламентується ГОСТ 12.1.033-81, ГОСТ 12.1.004-85. Робота програміста повинна вестися в приміщенні, що відповідає категорії Д пожежної безпеки (негорючі речовини й матеріали в холодному стані).

Всі приміщення повинні бути забезпечені первинними засобами пожежогасіння: пожежним водопостачанням (пожежні крани ПК), пожежні щити з набором пожежного інструменту, вуглекислотними або порошковими вогнегасниками. У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефону 101 пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації і приступити до ліквідації пожежі.

Будівлі укомплектовані пожежними щитами з набором інструментів, біля щитів – бочки з водою, ящики з піском.

Виробничі приміщення мають запасні виходи. Двері повинні мати освітлений надпис «Запасний вихід». План евакуації вивішується на видному місці у основного виходу із приміщення.

					КБ 01. 12 003. 00 ДП ПЗ	Арк.
						61
Ізм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

Для дипломного проєкту було розроблено мобільний застосунок для обліку особистих фінансів, що ґрунтується на передових технологіях програмування та розробки. Дослідження показало, що впровадження такого застосунку має сенс у поліпшенні процесів управління фінансами, забезпечуючи зручність використання, ефективність та доступність.

Розробка програмного забезпечення проводилася з використанням інноваційних технологій, таких як Dart та Flutter. Цей стек дозволив створити кроссплатформений застосунок, який можна використовувати на пристроях Android та iOS. Dart є мовою програмування, що використовується для створення мобільних застосунків з високою продуктивністю та гнучкістю, а Flutter – це фреймворк з відкритим кодом, який використовує Dart для створення інтерфейсів користувача з інтуїтивно зрозумілим дизайном.

Створений застосунок включає ряд функціональних можливостей, включаючи автоматизоване ведення обліку доходів та витрат, їх категоризацію, створення бюджетів, аналітику та звіти. Всі аспекти, визначені технічним завданням, були ретельно розглянуті, включаючи аналіз предметної області, опис використаних технологій та інструментів, розробку інтерфейсу користувача, тестування та налагодження, а також висновок.

У пояснювальній записці висвітлено всі аспекти, що визначені технічним завданням. Це включає дослідження області застосування, детальний опис використаних технологій та інструментів, а також оцінку економічної доцільності впровадження системи. Крім того, відзначено питання з охорони праці та наведено перелік джерел, що підтверджує обґрунтованість і актуальність проєкту.

					<i>КБ 01. 12 000. 00 ДП ПЗ</i>	Арк.
						62
Ізм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Money Lover Spending Manager Play Market. [Веб-сайт]. URL: <https://play.google.com/store/apps/details?id=com.bookmark.money&hl=en/>.
2. Expensify Expense Tracker Play Market. [Веб-сайт]. URL: <https://play.google.com/store/apps/details?id=org.me.mobiexpensifyg&hl=en/>.
3. Wally AI Personal Finance Play Market. [Веб-сайт]. URL: <https://play.google.com/store/apps/details?id=me.wally&hl=en/>.
4. О. Шевченко. Крос-платформне програмування: Навчальний посібник. Львів: Вид-во ЛТЕУ, 2019.
5. К. О. Радченко. Розробка мобільних застосунків для OS Android: Навчальний посібник. Київ: КПІ ім. Ігоря Сікорського, 2023.
6. statista. [Веб-сайт]. URL: <https://www.statista.com/>.
7. Flutter Docs. [Веб-сайт]. URL: <https://docs.flutter.dev/>.
8. MaterialYou. [Веб-сайт]. URL: <https://material.io/blog/announcing-material-you/>.

ДОДАТОК А. Програмний код основної логіки мобільного застосунку

```
// import 'dart:convert';
import 'dart:math';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:shared_preferences/shared_preferences.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Планування бюджету',
      theme: ThemeData.light(),
      home: App(),
    );
  }
}

class App extends StatefulWidget {
  @override
  _AppState createState() => _AppState();
}

class _AppState extends State<App> {
  int _selectedIndex = 0;

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  static List<Widget> _widgetOptions = <Widget>[
    ManagementPage(),
    AdvicePage(),
    HelpPage(),
    AuthorPage(),
    ExitPage(),
  ];

  static const List<BottomNavigationBarItem> _bottomNavItems = [
    BottomNavigationBarItem(
      icon: Icon(Icons.list),
      label: 'Планування',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.school),
      label: 'Поради',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.help),
      label: 'Довідка',
    ),
  ],
}
```

```

    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.person),
      label: 'Автор',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.exit_to_app),
      label: 'Вихід',
    ),
  ],
);

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(
        _bottomNavItems[_selectedIndex].label!,
      ),
    ),
    body: _widgetOptions.elementAt(_selectedIndex),
    bottomNavigationBar: BottomNavigationBar(
      type: BottomNavigationBarType.fixed,
      backgroundColor: Colors.white,
      selectedItemColor: Colors.purple,
      unselectedItemColor: Colors.grey,
      items: _bottomNavItems,
      currentIndex: _selectedIndex,
      onTap: _onItemTapped,
    ),
  );
}

class ManagementPage extends StatefulWidget {
  @override
  _ManagementPageState createState() => _ManagementPageState();
}

class _ManagementPageState extends State<ManagementPage> {
  double salary = 0;
  Map<String, double> expenses = {};

  @override
  void initState() {
    super.initState();
    _loadData();
  }

  _loadData() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    String expensesString = prefs.getString('expenses') ?? '{}';
    setState(() {
      salary = prefs.getDouble('salary') ?? 0;
      expenses = Map<String, double>.from(json
        .decode(expensesString)
        .map((key, value) => MapEntry(key, value.toDouble())));
    });
  }

  _saveData() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.setDouble('salary', salary);
  }
}

```

```

    await prefs.setString('expenses', json.encode(expenses));
}

void _clearData() async {
  setState(() {
    salary = 0;
    expenses = {};
  });
  SharedPreferences prefs = await SharedPreferences.getInstance();
  await prefs.setDouble('salary', salary);
  await prefs.setString('expenses', json.encode(expenses));
}

void _showAddExpenseDialog() {
  String category = '';
  String amount = '';
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Додати витрату'),
        content: SingleChildScrollView(
          child: Column(
            children: [
              TextField(
                decoration: InputDecoration(labelText: 'Категорія'),
                onChanged: (value) {
                  category = value;
                },
              ),
              TextField(
                decoration: InputDecoration(labelText: 'Сума'),
                onChanged: (value) {
                  amount = value;
                },
              ),
            ],
          ),
        ),
        actions: [
          TextButton(
            onPressed: () {
              setState(() {
                if (amount.isNotEmpty && category.isNotEmpty) {
                  double expenseAmount = double.parse(amount);
                  if (expenses.containsKey(category)) {
                    expenses[category] =
                      (expenses[category] ?? 0) + expenseAmount;
                  } else {
                    expenses[category] = expenseAmount;
                  }
                  salary -= expenseAmount;
                  _saveData();
                }
              });
              Navigator.of(context).pop();
            },
            child: Text('Додати'),
          ),
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },

```

```

        },
        child: Text('Відмінити'),
      ),
    ],
  );
},
);
}

void _showAddSalaryDialog() {
  String newSalary = '';
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Додати заробітню плату'),
        content: SingleChildScrollView(
          child: Column(
            children: [
              TextField(
                decoration: InputDecoration(labelText: 'Заробітня плата'),
                onChanged: (value) {
                  newSalary = value;
                },
              ),
            ],
          ),
        ),
        actions: [
          TextButton(
            onPressed: () {
              if (newSalary.isNotEmpty) {
                setState(() {
                  salary += double.parse(newSalary);
                  _saveData();
                });
              }
              Navigator.of(context).pop();
            },
            child: Text('Додати'),
          ),
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: Text('Відмінити'),
          ),
        ],
      );
    },
  );
}

```

```

@override
Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: [
        Row(
          children: [

```

```

        Icon(
          Icons.input,
          color: Colors.black,
          size: 16.0,
        ),
        SizedBox(width: 16),
        Text(
          'Вхідні данні',
          style: TextStyle(
            fontSize: 16.0,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
    SizedBox(height: 16),
    ListTile(
      title: Text('Заробітня плата: $salary'),
    ),
    SizedBox(height: 16),
    Row(
      children: [
        Icon(
          Icons.autorenew,
          color: Colors.black,
          size: 16.0,
        ),
        SizedBox(width: 16),
        Text(
          'Обробка даних',
          style: TextStyle(
            fontSize: 16.0,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
    SizedBox(height: 16),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceAround,
      children: [
        ElevatedButton.icon(
          onPressed: _showAddSalaryDialog,
          icon: Icon(Icons.attach_money),
          label: Text('Додати заробітню плату'),
        ),
        ElevatedButton.icon(
          onPressed: _showAddExpenseDialog,
          icon: Icon(Icons.money_off),
          label: Text('Додати витрату'),
        ),
        ElevatedButton.icon(
          onPressed: _clearData,
          icon: Icon(Icons.clear),
          label: Text('Очистити все'),
        ),
      ],
    ),
    SizedBox(height: 16.0),
    Row(
      children: [
        Icon(

```

```

        Icons.output,
        color: Colors.black,
        size: 16.0,
    ),
    SizedBox(width: 16),
    Text(
        'Вихідні данні',
        style: TextStyle(
            fontSize: 16.0,
            fontWeight: FontWeight.bold,
        ),
    ),
],
),
SizedBox(height: 16.0),
SimpleLinearChart(expenses: expenses),
Expanded(
    child: ListView.builder(
        itemCount: expenses.length,
        itemBuilder: (BuildContext context, int index) {
            String category = expenses.keys.elementAt(index);
            double amount = expenses.values.elementAt(index);
            return ListTile(
                title: Text('$category: $amount'),
            );
        },
    ),
),
],
),
);
}
}

```

```

class AdvicePage extends StatelessWidget {
    final List<String> adviceList = [
        "Розгляньте можливість розподілу бюджету на категорії, такі як їжа, транспорт, розваги тощо.",
        "Спробуйте встановити місячні цілі з економії та тримайтеся їх, покладаючи гроші відразу по отриманню зарплати.",
        "Використовуйте бюджетні додатки або аркуші електронних таблиць для ведення обліку витрат та доходів.",
        "Аналізуйте свої витрати щомісяця, щоб визначити, на яких областях можна зекономити.",
        "Не забувайте про екстрені фонди та заощадження на непередбачені витрати.",
        "Розгляньте можливість інвестування частини доходу для майбутнього зростання капіталу."
    ];

    String getRandomAdvice() {
        final random = Random();
        return adviceList[random.nextInt(adviceList.length)];
    }

    @override
    Widget build(BuildContext context) {
        String advice = getRandomAdvice();
        return Center(
            child: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.center,

```

```

        children: [
          Text(
            advice,
            style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
            textAlign: TextAlign.justify,
          ),
        ],
      ),
    ),
  );
}
}

class HelpPage extends StatelessWidget {
  @override
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: [Center(
          child: Text(
            'Облік власних фінансів є важливим з кількох причин:\n\n'
            '1. Контроль витрат: Ведення обліку дозволяє бачити, куди йдуть ваші гроші, і допомагає контролювати витрати.\n\n'
            '2. Планування бюджету: Знання своїх доходів і витрат дозволяє ефективно планувати бюджет і уникати фінансових труднощів.\n\n'
            '3. Досягнення фінансових цілей: Правильне управління фінансами допомагає досягати короткострокових і довгострокових фінансових цілей.\n\n'
            '4. Зменшення стресу: Коли ви знаєте свій фінансовий стан і маєте план, це зменшує стрес і робить вас впевненішими у майбутньому.\n\n'
            '5. Прийняття обґрунтованих рішень: Ведення фінансового обліку допомагає приймати обґрунтовані фінансові рішення, зокрема щодо інвестицій та великих покупок.\n\n'
            'Загалом, облік власних фінансів є основою фінансової стабільності та добробуту.',
            style: TextStyle(fontSize: 14),
            textAlign: TextAlign.justify,
          ),
        ),
      ],
    ),
  );
}

class AuthorPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Text('Попов О. Р., 2024',
        style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold)),
    );
  }
}

class ExitPage extends StatelessWidget {
  void _exitApp(BuildContext context) {
    // Close the app
    SystemNavigator.pop();
  }
}

```

```

@override
Widget build(BuildContext context) {
  return Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        ElevatedButton(
          onPressed: () => _exitApp(context),
          child: Text('Закрити застосунок'),
        ),
      ],
    ),
  );
}

class SimpleLinearChart extends StatelessWidget {
  final Map<String, double> expenses;

  SimpleLinearChart({required this.expenses});

  @override
  Widget build(BuildContext context) {
    final List<String> labels = [];
    final List<double> values = [];
    expenses.forEach((key, value) {
      if (value > 0) {
        labels.add(key);
        values.add(value);
      }
    });
    if (values.isEmpty) {
      return Container(
        height: 200,
        width: double.infinity,
        padding: EdgeInsets.all(16.0),
        child: Text('Данні відсутні'),
      );
    }

    final double maxValue = values.reduce((value, element) => value > element ? value :
    element);

    return Container(
      height: 200,
      width: double.infinity,
      padding: EdgeInsets.all(16.0),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        crossAxisAlignment: CrossAxisAlignment.end,
        children: List.generate(labels.length, (index) {
          return Column(
            mainAxisAlignment: MainAxisAlignment.end,
            children: [
              Text(labels[index]),
              Container(
                width: 20,
                height: (values[index] / maxValue) * 150, // Adjusted to fit within the
chart area
                color: Colors.purple,
              ),
            ],
          );
        })
      ],
    );
  }
}

```

```

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

```

ДОДАТОК Б. Слайди мультимедійної презентації

Розробка мобільного додатку для обліку особистих фінансів

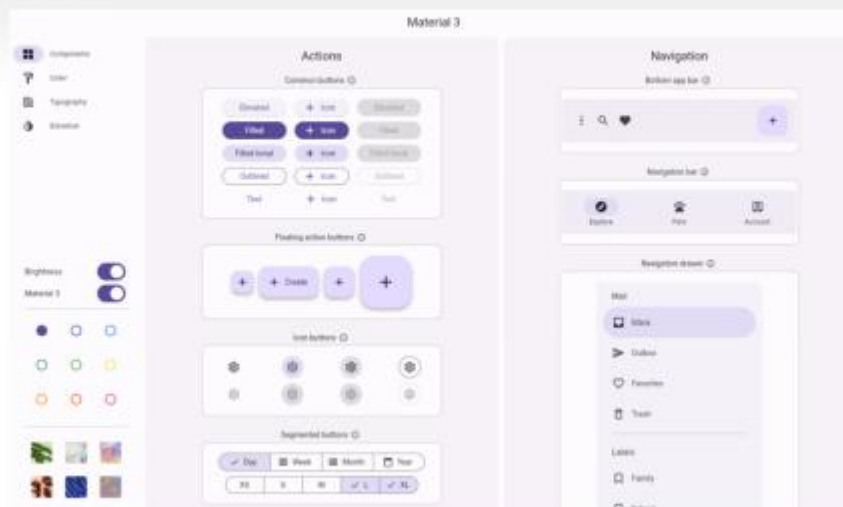
Попов Олександр Русланович



Загальні відомості



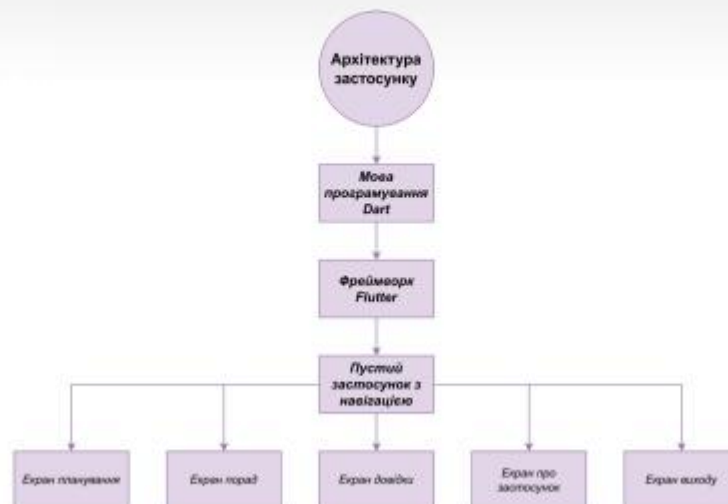
- Мобільні застосунки для обліку особистих фінансів стають популярними завдяки своїм функціям контролю доходів та витрат, а також аналізу фінансової діяльності.
- Вони надають зручні інструменти для категоризації витрат та автоматизованої звітності, сприяючи ефективному управлінню фінансами та досягненню фінансових цілей.
- Ці застосунки також можуть служити інструментом для фінансової освіти, надаючи користувачам доступні поради та рекомендації для покращення їхніх фінансових знань і навичок.



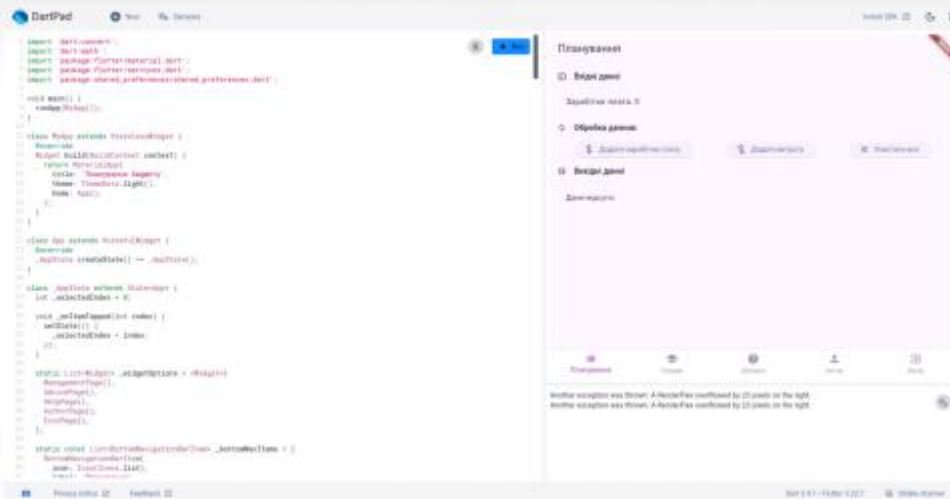
Навігація у застосунку



Архітектура застосунку



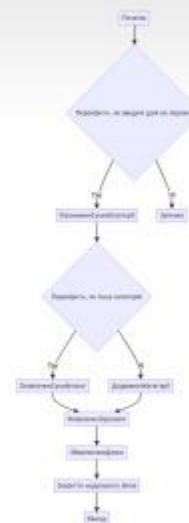
Процес розробки



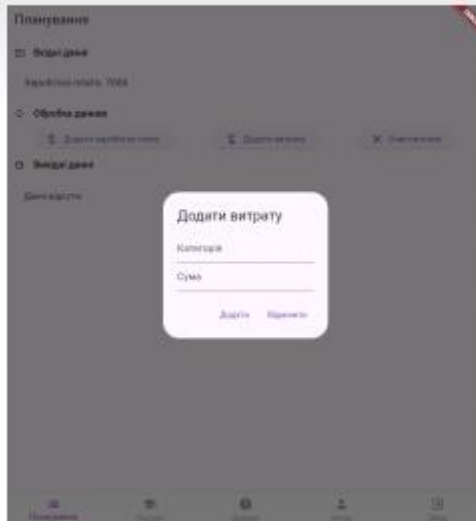
Програмний код основної логіки



```
TextButton(  
  onPressed: () {  
    setState(() {  
      if (amount.isNotEmpty && category.isNotEmpty) {  
        double expenseAmount = double.parse(amount);  
        if (expenses.containsKey(category)) {  
          expenses[category] =  
            (expenses[category] ?? 0) +  
            expenseAmount;  
        } else {  
          expenses[category] = expenseAmount;  
        }  
        salary -= expenseAmount;  
        _saveData();  
      }  
    });  
    Navigator.of(context).pop();  
  },  
),
```



Демонстрація основної логіки



Зв'язок з ОПП



- `SharedPreferences` - це пакет для Flutter, який дозволяє зберігати прості дані ключ-значення на постійному сховищі на мобільному пристрої.
- У методі `initState` викликається `_loadData`, який завантажує збережені дані з `SharedPreferences` і оновлює стан виджету. Метод `_saveData` зберігає поточні значення зарплатної плати та витрат до `SharedPreferences`.

```
_loadData() async {  
  SharedPreferences prefs = await  
    SharedPreferences.getInstance();  
  String expensesString =  
    prefs.getString('expenses') ?? '{}';  
  setState(() {  
    salary =  
      prefs.getDouble('salary') ?? 0;  
    expenses = Map<String,  
      double>.from(json  
        .decode(expensesString)  
        .map((key, value) =>  
          MapEntry(key, value.toDouble())));  
  });  
}
```

Дякую за увагу!



ВІДГУК

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

Попова Олександра Руслановича

(прізвище, ім'я та по батькові)

Спеціальність: *123 «Комп'ютерна інженерія»*

Освітня програма: *«Безпека комп'ютерних систем і мереж»*

Тема дипломного проекту:

Розробка мобільного додатку для обліку особистих фінансів

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) *Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка дипломного проекту виконана якісно, у повному обсязі. В дипломному проекті здобувачем проаналізовано існуючі аналоги, проведено аналіз їх недоліків, переваг. Конкретизовано на основі проведеного аналізу вимоги до пристрою, підібрано елементну базу для реалізації системи. В дипломному проекті в останніх розділах проаналізовано економічні показники та питання охорони праці. Графічна частина складається з слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.*

б) самостійність роботи над проектом:

Протягом виконання дипломного проекту здобувач освіти поступово та послідовно виконував всі етапи, проявив ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувач освіти виконував самостійно, з урахуванням рекомендацій керівника.

в) теоретична підготовка випускника (випускниці):

Здобувач освіти під час роботи над дипломним проектом вивчив багато літературних та інтернет-джерел за даною тематикою. Вважаю, що теоретична підготовка дипломника достатня та в цілому відповідає державним вимогам до фахівців відповідного рівня кваліфікації. Здобувач готовий до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувач освіти показав вміння організовано працювати над поставленим завданням, використовувати останні досягнення науки та техніки в предметній галузі, здійснювати пошук технічного рішення на підставі відповідної навчальної та науково-технічної літератури, використовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами.

Оцінка розрахункової частини Добре
Оцінка графічної частини Відмінно
Загальна оцінка Добре

Прізвище, ім'я, по батькові керівника дипломного проекту к.пед.н. Скорнякова Олена Володимирівна

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», зав.відділенням комп'ютерних систем, викладач спецдисциплін циклової комісії комп'ютерної техніки та програмної інженерії

Підпис 

«14» 06 2024 р.

РЕЦЕНЗІЯ

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

Попова Олександра Руслановича

(прізвище, ім'я та по батькові)

Спеціальність *123 «Комп'ютерна інженерія»*

Освітня програма *«Безпека комп'ютерних систем і мереж»*

Керівник дипломного проекту (роботи) *Скорянкова Олена Володимирівна*

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) *Розробка мобільного додатку для обліку особистих фінансів*

Обсяг розрахунково-пояснювальної записки *78* сторінок

Обсяг графічної (презентаційної) частини *11* аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі обліку особистих фінансів та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 11 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки добра, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту Реалізовано мобільний застосунок для обліку особистих фінансів, що дозволяє вводити заробітну плату та розбивати її за категоріями витрат.

Розроблене програмне базується на сучасній технології – Flutter.

GUI застосунку реалізовано в сучасному стилі – MaterialYou. Для категорій витрат будується графік. Дані захищено засобами XOR.

д) основні недоліки дипломного проекту _____

Програмне забезпечення зберігає статистику витрат лише на поточну заробітну плату.

В основному розділі пояснювальної записки основні алгоритми описані не дуже деталізовано.

Пояснювальна записка має деякі помилки оформлення.

Оцінка розрахункової частини _____ Добре

Оцінка графічної частини _____ Добре

Загальна оцінка _____ Добре

Прізвище, ім'я, по батькові рецензента _____ Стайкуца Сергій Володимирович

Місце роботи і посада рецензента Державний університет інтелектуальних технологій і зв'язку, к.ф.н., доцент кафедри КБ та ТЗІ



Сергій 2024 р.

Ім'я користувача:
Катерина Григоріївна Краснокутська

ID перевірки:
1016353669

Дата перевірки:
12.06.2024 19:02:37 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.06.2024 19:04:14 EEST

ID користувача:
100011688

Назва документа: 4КБ-01_Попов_О

Кількість сторінок: 45 Кількість слів: 5141 Кількість символів: 42139 Розмір файлу: 2.08 MB ID файлу: 1016157596

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

5.08%
Схожість

Найбільша схожість: 1.6% з Інтернет-джерелом (<https://morioh.com/p/794100ffd901>)

5.08% Джерела з Інтернету

381

Сторінка 47

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

3

Підозріле форматування

9
сторінок

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Попов Олександр Русланович,
здобувач освіти гр. 4КБ-01, та

Скорнякова Олена Володимирівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка мобільного додатку для обліку особистих фінансів» (автор роботи – Попов О.Р., керівник роботи – Скорнякова О.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

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

Виконавець



/ Попов О.Р. /

Керівник



/ Скорнякова О.В. /

«10» червня 2024 р.