

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

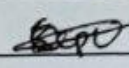
на тему:

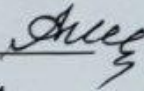
Інформаційна система моніторингу витрат особистого бюджету з
прогнозуванням майбутніх доходів і витрат

Виконав: студент 2-го курсу

групи ПІ-22мз спеціальності

121 – Інженерія програмного забезпечення

Богач Катерина Сергіївна 

Керівник: к.т.н., доц. каф. ПЗ Ткаченко О.М. 

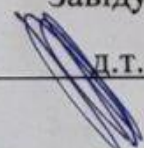
« 07 » червня 2024 р.

Опонент: к.т.н., проф. каф. ЗІ Кондратенко Н.Р. 

« 07 » червня 2024 р.

Допущено до захисту

Завідувач кафедри ПЗ

 д.т.н., проф. Романюк О. Н.

(прізвище та ініціали)

« 10 » червня 2024 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти II-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Романюк О. Н.

« 12 » березня 2024 р.

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Богач Катерина Сергіївна

1. Тема роботи – інформаційна система моніторингу витрат особистого бюджету з прогнозуванням майбутніх доходів і витрат.

Керівник роботи: Ткаченко Олександр Миколайович, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від « 11 » березня 2024 р. № 81.

2. Строк подання студентом роботи

3 червня 2024 р.

3. Вихідні дані до роботи: мова розробки Dart, Flutter, операційна система - Windows 10, метод експоненційного згладжування.

4. Зміст розрахунково-пояснювальної записки: вступ, обґрунтування вибору методу розробки постановки задачі дослідження, вибір методу прогнозування та розробка архітектури мобільного додатку, розробка модулів

програмного засобу, тестування розробленого програмного засобу, економічна частина, висновки.

5. Перелік графічного матеріалу: створення блок-схем алгоритму розроблення макету мобільного додатку, перетворення формул у код.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Ткаченко О.М., к.т.н., доцент кафедри ПЗ	11.03.24 <i>Міч</i>	06.06.24 <i>Міч</i>
5	Кавецький В.В., к.е.н., доц. кафедри ЕПВМ	26.04.24 <i>Вав</i>	10.05.24 <i>Вав</i>

7. Дата видачі завдання 11 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування вибору методу розробки та постановка задач дослідження	12.03.2024- 28.03.2024	Виконав
2	Вибір методу прогнозування та розробка архітектури мобільного додатку	29.03.2024- 11.03.2024	Виконав
3	Розробка модулів програмного засобу	12.03.2024- 12.04.2024	Виконав
4	Тестування розробленого програмного засобу	13.04.2024- 05.06.2024	Виконав
5	Економічна частина	26.04.24- 10.05.24	Виконав

Студентка

Богач
(підпис)

Богач К.С.

(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи

Ткаченко
(підпис)

Ткаченко О.М.

(прізвище та ініціали)

АНОТАЦІЯ

УДК 336.74

Богач К.С. Розробка інформаційної системи моніторингу витрат особистого бюджету з прогнозуванням майбутніх доходів та витрат магістерська кваліфікаційна робота зі спеціальності 121 Інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2024. 91с.

На укр. мові. Бібліогр. : 34 назв ; рис. : 16; табл. 8.

Під час виконання магістерської кваліфікаційної роботи було розроблено мобільний додаток для моніторингу витрат особистого бюджету з прогнозуванням майбутніх доходів та витрат.

У процесі створення програмного продукту було розроблено діаграми діяльності та станів на основі яких було запропоновано архітектуру інтегрованої інформаційної системи, яка об'єднує в собі функції моніторингу витрат сімейного бюджету і прогнозування майбутніх доходів та витрат.

Було проведено аналіз аналогів на ринку моніторингу особистих витрат. На основі результату порівняння було виокремлено як їх переваги, які потрібно було удосконалити, так і недоліки, яких потрібно було уникнути під час розробки програмного продукту. Проаналізовано доступні програмні забезпечення для реалізації мобільного застосунку та обрано найбільш підходящу. Розроблено інтерфейс мобільного додатку.

Тестування програмного продукту показало, що метод експоненційного згладжування забезпечує найменшу похибку прогнозування.

Виконано розрахунки та економічний аналіз в результаті якого було підтверджено доцільність, а також економічну вигідність для інвесторів та економічних установ.

Ключові слова: Dart, Flutter, експоненційне зглазування, тестування програмних засобів, прогнозування майбутніх надходжень та витрат.

ABSTRACT

Bohach K. S. Development of information system for personal budget expenses with forecasting future income and expenses Master's Qualification Work in the specialty 121 software engineering, educational program - software engineering. Vinnitsa: VNTU, 2024. 91 with.

In Ukrainian speech Bibliogr. : 34 titles; Fig. : 16; table 8.

During the master's qualification work, a mobile application was developed to monitor the costs of the personal budget with the forecasting of future income and expenses.

In the process of creating a software product, diagrams of activity and states were developed on the basis of which the architecture of an integrated information system was proposed, which combines the functions of monitoring the costs of the family budget and forecasting future income and expenses.

Analogues were analyzed in the personal cost monitoring market. On the basis of the result of the comparison, both their advantages, which needed to be improved and the disadvantages, which had to be avoided when developing a software product. The available mobile application software is analyzed and the most suitable is chosen. The mobile app interface has been developed.

Testing the software showed that the exponential smoothing method provides the lowest forecasting error.

Calculations and economic analysis were performed as well as the feasibility of investors and economic institutions.

Keywords: Dart, Flutter, exponential frosting, software testing, future revenues and expenses.

ЗМІСТ

ВСТУП.....	8
1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	12
1.1 Аналіз сучасного стану питання	12
1.2 Порівняльний аналіз аналогів	14
1.3 Аналіз методів вирішення поставленої задачі	23
1.4 Постановка задач на проектування	25
1.5 Висновки	26
2 ВИБІР МЕТОДУ ПРОГНОЗУВАННЯ ТА РОЗРОБКА АРХІТЕКТУРИ МОБІЛЬНОГО ДОДАТКУ	27
2.1 Загальна характеристика методів моніторингу та прогнозування витрат	27
2.2 Особливості застосування методу середнього для прогнозування витрат	28
2.3 Особливості застосування експоненційного згладжування для прогнозування витрат	30
2.4 Особливості застосування метода Хольта-Вінтерса для прогнозування витрат	32
2.5 Особливості застосування метода простого рухомого середнього для прогнозування витрат	35
2.6 Обґрунтування вибору методу прогнозування	37
2.7 Розробка архітектури мобільного додатку	38
2.8 Висновки	43
3. РОЗРОБКА МОДУЛІВ ПРОГРАМНОГО ЗАСОБУ	44
3.1 Варіантний аналіз та обґрунтування вибору засобів реалізації програмного продукту	44
3.2 Розробка структури інтерфейсу для мобільного додатку	48
3.3 Розробка програмних модулів мобільного додатку	53
3.4 Реалізація методу прогнозування експоненційним згладжуванням	60

3.5 Висновки	62
4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАСОБУ	63
4.1 Аналіз методів тестування	63
4.2 Тестування розробленого програмного продукту	66
4.3 Оцінювання достовірності майбутніх витрат	68
4.4 Інструкція користувача	73
4.5 Висновки	77
5 ЕКОНОМІЧНА ЧАСТИНА.....	78
5.1 Проведення комерційного аудиту науково-технічної розробки	78
5.2 Розрахунок витрат на здійснення науково-технічної розробки.	83
5.3 Розрахунок економічної ефективності та обґрунтування економічної доцільності комерціалізації науково-технічної розробки.	86
5.4 Висновки	90
ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	93
ДОДАТКИ.....	96
Додаток А. Технічне завдання	97
Додаток Б. Протокол перевірки роботи	101
Додаток В Лістинг розробленого додатку додатку	102
Додаток Г Таблиця із чек-лістом та тест-кейсами	131
Додаток І Ілюстративна частина	140

ВСТУП

Обґрунтування вибору теми дослідження

Вести записи про витрати - це невід'ємний елемент управління фінансами як на особистому, так і на комерційному рівні. Ця практика дозволяє докладно відстежувати всі витрати, аналізувати бюджет, визначати можливості для економії та встановлювати пріоритети. Використання засобів для запису витрат, таких як програми і додатки, забезпечує зручний та ефективний спосіб фіксації і аналізу витрат, допомагаючи автоматизувати процес обробки інформації та створювати звіти з фінансової діяльності. Такий підхід є важливим кроком для досягнення фінансової стабільності та ефективного управління фінансами.

Ведення записів про витрати дозволяє більш об'єктивно оцінювати ефективність фінансових стратегій та розробляти більш оптимальні плани бюджетування. Це також допомагає виявляти недоліки у фінансових звичках та змінювати їх на більш продуктивні. Крім того, засоби запису витрат можуть забезпечити індивідуальне налаштування і адаптацію до конкретних потреб користувача, що робить процес керування фінансами більш гнучким та персоналізованим. Такий підхід відображає сучасні тенденції у фінансовому плануванні та сприяє здоровій фінансовій поведінці як на особистому, так і на професійному рівні.

Вибір теми має свою обґрунтованість з кількох причин. По-перше, управління особистими фінансами є важливою складовою фінансової грамотності та успішного фінансового планування. Відстеження витрат допомагає розуміти, куди йдуть гроші, і знайти способи раціоналізувати витрати.

Друга причина полягає у тому, що при зростанні складності фінансових умов важливо мати ефективний механізм контролю за витратами. Моніторинг витрат допомагає уникнути перевитрат та забезпечує фінансову прогнозованість, а отже більшу стабільність. Варто зауважити що у сучасному світі з використанням доступних технічних можливостей, контроль за витратами, аналіз результатів і прогнозування майбутніх фінансових потреб, а також

постійний доступ до фінансових записів можна значно спростити завдяки смартфонам, якщо завантажити на нього відповідний додаток.

По-третє, розробка додатку для моніторингу особистого бюджету користувача є необхідною через численні недоліки наявних рішень на ринку. Багато існуючих додатків не задовольняють потреби користувачів повною мірою. Наприклад, деякі з них функціонують лише за умови використання банківської карти певного банку, що обмежує їхню універсальність. Інші програми є платними, що може бути недоступним або непривабливим для частини користувачів. Крім того, значна частина таких додатків не підтримує локалізацію українською мовою, що ускладнює їх використання для україномовних користувачів. Розробка нового додатку, який би враховував ці аспекти, могла б значно покращити користувацький досвід та забезпечити більшу доступність і зручність у моніторингу особистих фінансів.

Таким чином, обрана тема є актуальною, оскільки сприяє ефективному управлінню фінансами та забезпечує фінансову стабільність.

Зв'язок роботи з науковими програмами, планами, темами. Дослідження було проведено згідно з планом наукових досліджень кафедри програмного забезпечення.

Мета та задачі дослідження. *Метою магістерської кваліфікаційної роботи* є підвищення фінансової грамотності користувачів мобільного додатку, що допоможе їм уникати нераціональних витрат і сприятиме зростанню їх добробуту.

Основними задачами дослідження є:

- Провести аналітичний огляд сучасних програмних засобів, які застосовуються для планування особистого бюджету.
- Розробити математичну модель для прогнозування майбутніх витрат.
- Розробити алгоритми та програмні засоби для створення мобільного програмного продукту.
- Провести тестування розробленого програмного продукту.

- Обрахувати економічний ефект від впровадження розробленого програмного продукту.

Об'єктом дослідження є процеси моніторингу поточних та прогнозування фінансових витрат.

Предметом дослідження є методи прогнозування майбутніх витрат сімейного бюджету та засоби розробки програмних систем.

Методи досліджень: методи математичного моделювання для вибору методу прогнозування майбутніх витрат; методи проектування програмного забезпечення для розробки архітектури системи, алгоритмів та програмного забезпечення; методи математичної статистики для аналізу результатів тестування програмного продукту.

Наукова новизна отриманих результатів.

1. **Вперше** запропоновано архітектуру інтегрованої інформаційної системи, яка об'єднує в собі функції моніторингу витрат сімейного бюджету і прогнозування майбутніх доходів та витрат, забезпечує автоматизовану обробку фінансової інформації, що дозволяє користувачам точно передбачати свої майбутні фінансові потреби.

2. **Отримав** подальшого розвитку метод експоненційного згладжування, який відрізняється від існуючих значенням параметра згладжування α , що дозволило зменшити середнє значення абсолютної похибки прогнозу майбутніх витрат.

Практичне значення отриманих результатів. Розроблено алгоритми та програмне забезпечення, яке дозволить користувачам ефективніше відслідковувати свої витрати, а також надасть можливість систематичного планування і прогнозування майбутніх витрат. Використання такого програмного засобу сприятиме формуванню стійких фінансових навичок та покращенню фінансової грамотності серед користувачів.

Особистий внесок здобувача.

Всі наукові досягнення, які представлені у магістерській кваліфікаційній роботі, отримані автором особисто.

Апробація матеріалів магістерської кваліфікаційної роботи.

Основні положення магістерської кваліфікаційної роботи доповідалися та обговорювалися на Міжнародній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)»

Публікації. Основні результати досліджень опубліковано в 1 науковій праці, що входить у матеріали Міжнародної науково-практичної інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)» [1].

1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз сучасного стану питання

Питання особистого бюджету завжди було актуальним, але в сучасних реаліях набуває все більшого значення. У останні роки спостерігається значне підвищення інтересу до моніторингу витрат та планування витрат, а також різних способів для заощадження від скриньок для накопичення і до журналів із конвертами. На цьому тлі виникає інша проблема – доступність та зручність у використанні.

Ведення особистого бюджету є важливим елементом фінансової самодисципліни та контролю. Це допомагає людям краще розуміти свої фінансові потреби, здійснювати свідомі рішення щодо витрат та збереження коштів, а також планувати майбутні фінансові цілі. Ведення бюджету допомагає уникнути небажаних витрат та навіть зекономити фінанси.

Існує кілька методів моніторингу особистого бюджету, які можуть бути використані залежно від індивідуальних вподобань та стилів життя. Ось декілька з них[2]:

1 Запис у журналі або на папері: Цей традиційний метод включає ведення записів про всі доходи та витрати в ручному журналі або на аркушах паперу. Він дозволяє більш детально аналізувати фінансові показники та ручно контролювати кожну транзакцію.

2 Електронні таблиці: Використання програм, таких як Microsoft Excel або Google Sheets, для ведення обліку фінансів. Це надає більше можливостей для організації даних та створення різноманітних таблиць і графіків для аналізу.

3 Мобільні додатки для ведення бюджету: Існують різноманітні мобільні додатки, які допомагають вести облік фінансів, автоматично відстежуючи доходи та витрати, надаючи аналітичні звіти та навіть нагадуючи про платежі.

4 Онлайн сервіси для управління бюджетом: Ці сервіси дозволяють об'єднати всі фінансові дані з різних джерел, таких як банківські рахунки та кредитні картки, та надають інструменти для аналізу та планування бюджету.

5 Фінансові менеджери або консультанти: Професійні фінансові консультанти можуть надати індивідуальні поради щодо управління бюджетом та фінансового планування.

Використання мобільного додатку для моніторингу особистого бюджету має кілька переваг порівняно з іншими методами:

1 Зручність та доступність: Мобільні додатки завжди під рукою, оскільки більшість людей носять свої смартфони з собою. Це дозволяє вносити фінансові транзакції миттєво, навіть у русі, що значно полегшує ведення бюджету.

2 Аналітика та звітність: Мобільні додатки зазвичай мають вбудовані інструменти аналізу витрат, які надають користувачам зрозумілі звіти та графіки щодо їх фінансового стану. Це допомагає зробити більш обґрунтовані фінансові рішення.

3 Безпека і приватність: На відміну від звичайного записника, доступ до телефону людини має лише сама людина, тож дані про бюджет, тобто витрати та надходження знаходяться під захистом від сторонніх очей

Перевага використання мобільного додатку для моніторингу бюджету полягає в його зручності та доступності. Порівняно з традиційними методами, такими як ведення записів на папері або в електронних таблицях, мобільний додаток завжди під рукою, що дозволяє швидко вносити фінансові транзакції в будь-який час та в будь-якому місці. Такий додаток також може автоматично відстежувати витрати та надавати аналітичні звіти, що робить процес моніторингу бюджету більш ефективним та зручним для користувача.

1.2 Порівняльний аналіз аналогів

Для оцінки актуальності розробки було проведено детальний порівняльний аналіз найбільш популярних додатків, які використовуються для моніторингу особистого бюджету. Цей аналіз дозволив вивчити широкий спектр функціональності, що пропонується на ринку, включаючи інструменти відстеження витрат, аналітику бюджету, нагадування про платежі та інші корисні можливості. Було докладно оцінено їхні переваги та недоліки, а також звернута увага на те, які саме функції можуть бути найбільш корисними для користувачів.

Для порівняння було обрано наступні додатки:

- 1 Moneygraph+;
- 2 CoinKeeper: spending tracker;
- 3 Goodbudget: Budget & Finance.

Розпочнемо аналіз порівнюваних варіантів, щоб виявити їхні переваги та недоліки. Оцінимо кожен аналог, звертаючи увагу на їхні можливості, ефективність та придатність для наших потреб.

Goodbudget: Budget & Finance - це додаток для управління особистим бюджетом, який дозволяє користувачам зручно вести облік своїх фінансів. Його ключовою особливістю є використання системи "енвелопів", що дозволяє розділити бюджет на категорії та призначити певну суму грошей кожній категорії.

За допомогою Goodbudget, користувачі можуть вносити доходи, витрати та перекази між різними межами досяжності, щоб керувати своїми фінансами. Додаток надає зручний інтерфейс з можливістю створення різних бюджетів для різних потреб, а також аналітичні звіти для відстеження фінансового стану[3].

Додаток Goodbudget: Budget & Finance має кілька переваг:

- Система "енвелопів" для керування бюджетом: Goodbudget використовує інноваційну систему "енвелопів", яка дозволяє користувачам легко

розділити свій бюджет на різні категорії та відстежувати витрати в кожній з них. Це сприяє кращому контролю за фінансами та плануванню видатків.

- Зручний інтерфейс користувача: Goodbudget має дружній та інтуїтивно зрозумілий інтерфейс, що робить процес управління бюджетом максимально простим і зручним.

- Аналітичні звіти: Додаток надає користувачам можливість генерувати аналітичні звіти про їхні фінанси, що дозволяє зробити більш обґрунтовані фінансові рішення та вдосконалити свою стратегію управління бюджетом.

- Доступність на різних платформах: Goodbudget доступний як для iOS, так і для Android, що робить його досить універсальним і дозволяє користувачам використовувати його на різних мобільних пристроях.

- Синхронізація даних: Додаток пропонує можливість синхронізувати дані між різними пристроями, що дозволяє вести облік фінансів навіть у випадку зміни пристрою.

Незважаючи на свої переваги, додаток Goodbudget: Budget & Finance також має деякі недоліки:

- Обмежена безкоштовна версія: Безкоштовна версія додатку має обмеження по кількості "енвелопів" та функцій, що можуть обмежити користувачів у веденні детального обліку свого бюджету.

- Необхідність ручного внесення даних: Для точного відстеження фінансів користувачам часто потрібно вручну вносити дані про кожну транзакцію, що може бути часом і ресурсозатратним процесом.

- Відсутність автоматичного імпорту даних з банківських рахунків: Додаток не надає можливості автоматичного імпорту даних про транзакції з банківських рахунків, що може призвести до нестабільності і неточності в обліку фінансів.

- Синхронізація може бути недостатньо ефективною: Деякі користувачі повідомляли про проблеми з синхронізацією даних між пристроями, що може призвести до втрати даних або дублювання записів.
- Відсутність деяких додаткових функцій: У порівнянні з деякими конкурентами, Goodbudget може не мати деяких розширених функцій, таких як можливість інтеграції з іншими фінансовими сервісами або функції прогнозування витрат.

Інтерфейс Goodbudget: Budget & Finance зображено на рисунку 1.1

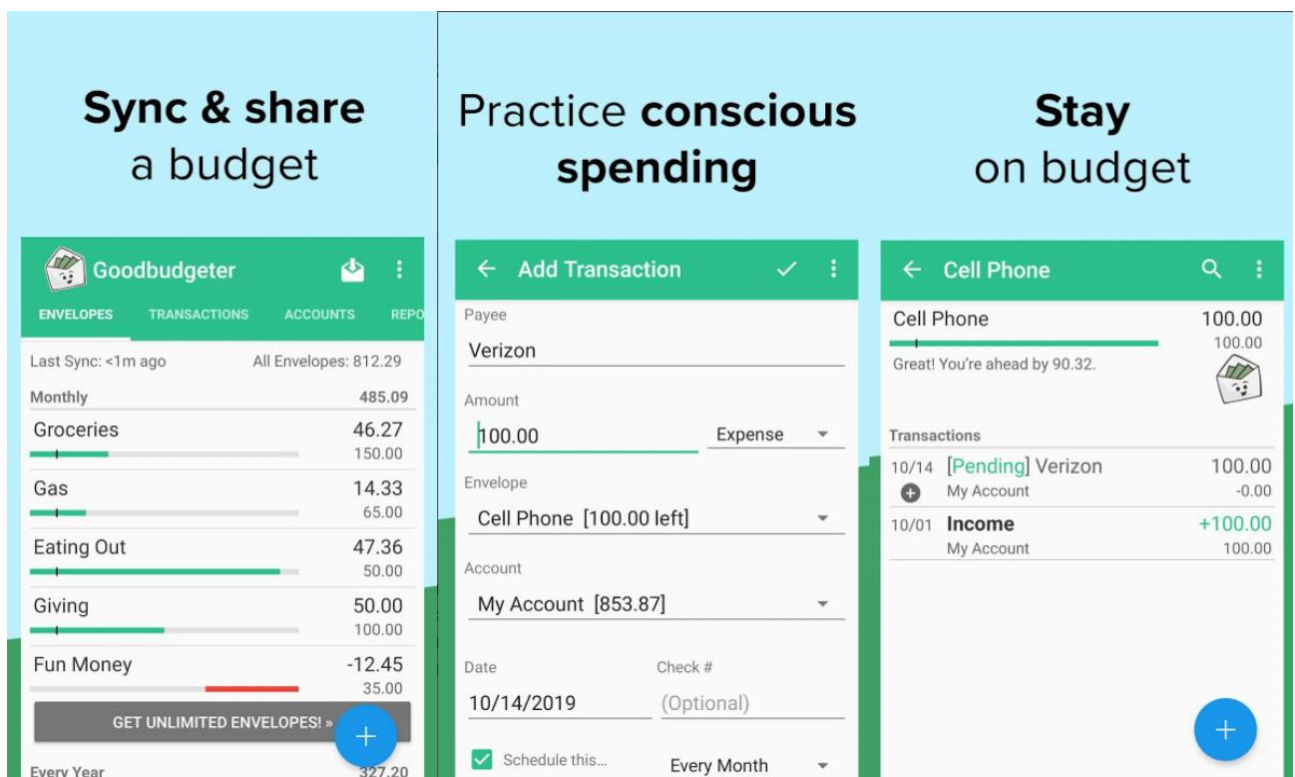


Рисунок 1.1 – Інтерфейс додатку Goodbudget: Budget & Finance

CoinKeeper: spending tracker – це додаток для ведення особистого бюджету та відстеження витрат, який допомагає користувачам керувати своїми фінансами зручно та ефективно. Основною його функціональністю є відстеження витрат, внесення доходів, створення бюджетів та аналіз фінансових потоків[4].

Переваги додатку CoinKeeper: spending tracker включають:

- Категоризація витрат: Користувачі можуть легко створювати різні категорії для своїх витрат, що допомагає структурувати бюджет та аналізувати витрати за окремими напрямками.
- Аналіз фінансів: Додаток надає зручні інструменти для аналізу фінансового стану, включаючи звіти та графіки, що допомагають користувачам зрозуміти, куди йдуть їхні гроші та як ефективно керувати бюджетом.
- Синхронізація даних: Додаток підтримує синхронізацію даних між різними пристроями, що дозволяє користувачам вести облік фінансів навіть на різних пристроях.
- Доступність на різних платформах: CoinKeeper доступний як для iOS, так і для Android, що робить його доступним для користувачів різних мобільних платформ.
- Мобільні додатки: Наявність мобільних додатків дозволяє користувачам вносити фінансові транзакції в будь-який час та в будь-якому місці зі свого смартфона, що забезпечує зручність та доступність.

Недоліки додатку CoinKeeper: spending tracker включають:

- Обмежена безкоштовна версія: Безкоштовна версія додатку має обмежені можливості або включає рекламу, що може обмежити функціональність та зручність користування.
- Необхідність ручного внесення даних: Для точного відстеження фінансів користувачам часто потрібно вручну вносити дані про кожну транзакцію, що може бути часом і ресурсозатратним процесом.
- Відсутність автоматичного імпорту даних: Додаток не надає можливості автоматичного імпорту даних про транзакції з банківських рахунків, що може призвести до нестабільності і неточності в обліку фінансів.
- Відсутність деяких додаткових функцій: У порівнянні з деякими конкурентами, CoinKeeper може не мати деяких розширених функцій, таких як можливість інтеграції з іншими фінансовими сервісами або функції прогнозування витрат.

- Платні функції: Деякі розширені функції можуть бути доступні лише за плату, що може збільшити витрати користувача на використання додатку.

Інтерфейс CoinKeeper: spending tracker зображено на рисунку 1.2

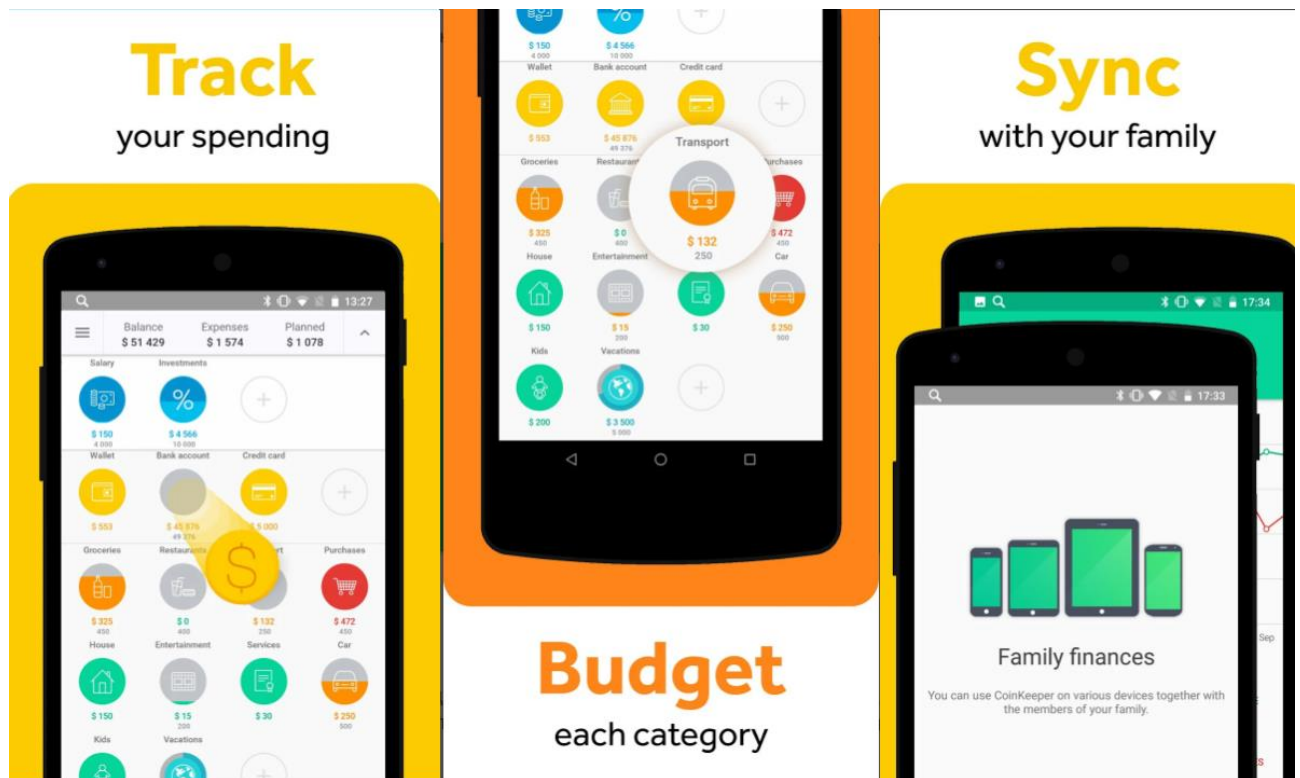


Рисунок 1.2 – Інтерфейс додатку CoinKeeper: spending tracker

Moneygraph+ – це додаток для ведення особистого бюджету та аналізу фінансів, який надає користувачам зручні інструменти для керування своїми фінансами. Основною метою додатку є допомога користувачам відстежувати свої доходи та витрати, а також аналізувати їх для прийняття обґрунтованих фінансових рішень[5].

Переваги додатку Moneygraph+ включають:

- Простий та зручний інтерфейс: Додаток має інтуїтивно зрозумілий і легкий у використанні інтерфейс, що дозволяє користувачам швидко організувати свої фінанси.

- Відстеження витрат та доходів: Moneygraph+ дозволяє користувачам легко вносити дані про свої доходи та витрати, що допомагає зрозуміти, куди йде гроші та як краще управляти бюджетом.
- Графіки та звіти: Додаток надає зручні графіки та звіти, що дозволяють користувачам аналізувати свої фінанси, виявляти тренди витрат та робити прогнози.
- Синхронізація даних: Moneygraph+ підтримує синхронізацію даних між різними пристроями, що дозволяє вести облік фінансів навіть на різних пристроях.
- Підтримка різних валют: Додаток підтримує різні валюти, що робить його зручним для користувачів з усього світу.

Недоліки додатку Moneygraph+ включають:

- Відсутність безкоштовної версії: Додаток може бути платним або містити обмежену безкоштовну версію з обмеженим функціоналом.
- Необхідність ручного внесення даних: Для ведення обліку фінансів користувачам може доводитися вносити дані про транзакції, що може бути часом і ресурсозатратним процесом.
- Відсутність автоматичного імпорту даних: Додаток може не мати можливості автоматичного імпорту даних про транзакції з банківських рахунків, що ускладнює процес ведення обліку фінансів.
- Можливі проблеми з синхронізацією даних: Деякі користувачі можуть зіткнутися з проблемами синхронізації даних між різними пристроями, що може призвести до втрати даних або дублювання записів.
- Обмежена підтримка функцій: У порівнянні з деякими конкурентами, Moneygraph+ може не мати деяких розширених функцій, таких як можливість інтеграції з іншими фінансовими сервісами або функції прогнозування витрат.

Інтерфейс Moneygraph+ зображено на рисунку 1.3

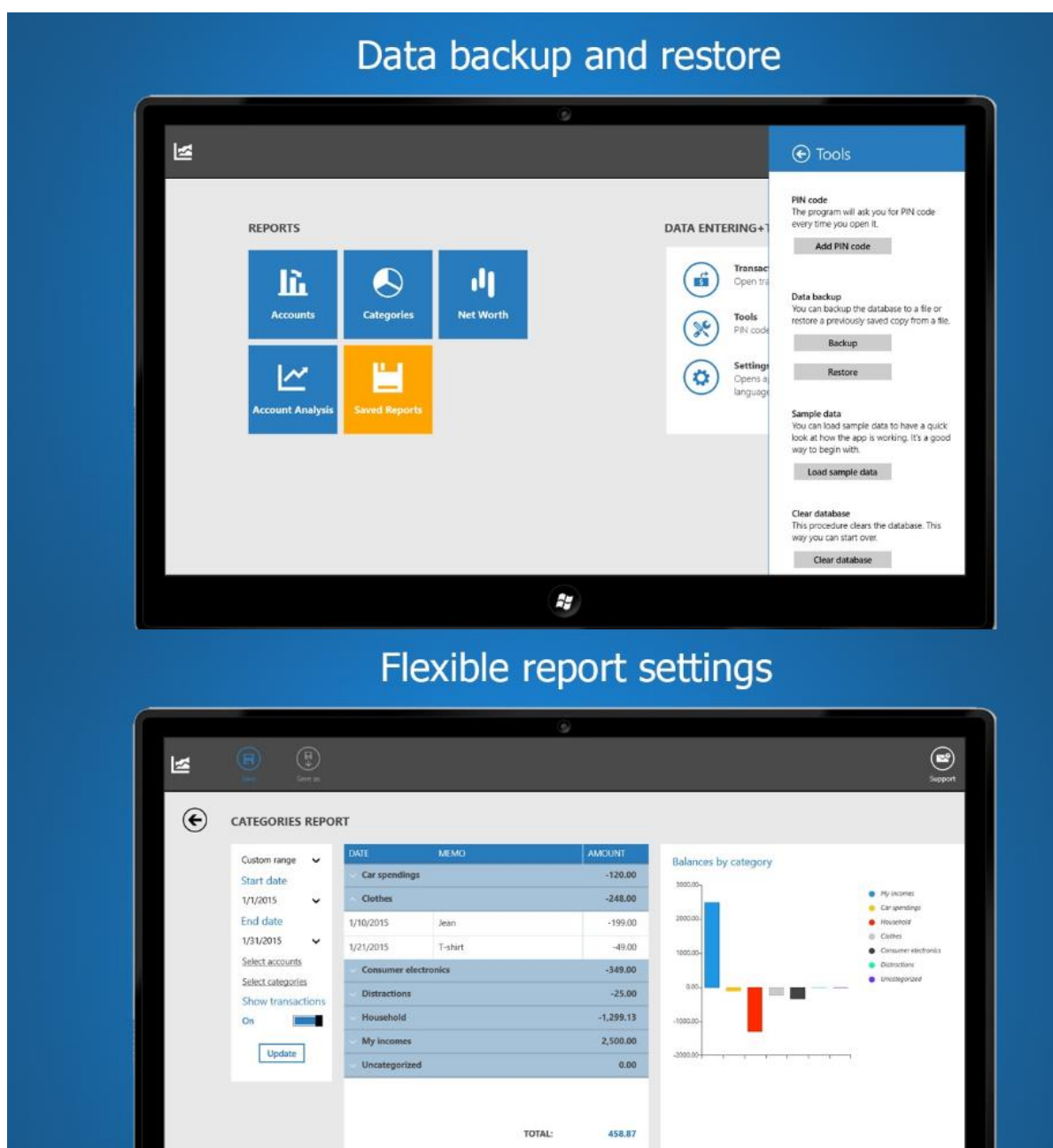


Рисунок 1.3 – Інтерфейс додатку Moneygraph+

Отже, в результаті аналізу вищезгаданих додатків була створена таблиця порівняння аналогів – таблиця 1.1. У цій таблиці наведено результати порівняння аналогів за основними критеріями, які є найважливішими для розробки додатку. Ця таблиця відображає переваги та недоліки кожного додатку, а також їхні можливості та обмеження з точки зору потреб користувача.

Таблиця 1.1 – Порівняння аналогів

Критерії	Moneygraph+	CoinKeeper	Goodbudget	CostTracker+
Можливість формування звіту	+	+	+	+
Безкоштовність	-	+/-	+/-	+
Зручний інтерфейс	+	+	-	+
Доступний для Android та iOS	-	+	+	+
Планування майбутніх витрат	-	-	-	+
Внесення даних про витрати та надходження	+	+	+	+
Всього:	3	4,5	3,5	6

Під час проведення порівняльного аналізу аналогів, було розглянуто основні параметри, які впливають на ефективність та зручність використання додатків. Враховані критерії спрямовані на забезпечення оптимальної продуктивності та зручності користування платформою.

Одним із обраних критеріїв було "Можливість формування звіту". По-перше, звіти дозволяють користувачам отримувати узагальнену інформацію про свої фінанси за певний період часу. Це допомагає зрозуміти, куди йдуть гроші, виявити зайві витрати та здійснити необхідні корективи в бюджеті. По-друге, звіти можуть слугувати основою для аналізу фінансового стану. Вони дозволяють виявити тенденції витрат, розподіл бюджету на різні категорії, а також визначити області, де можна зекономити. Крім того, звіти можуть бути корисними для планування майбутніх фінансових кроків. Вони дозволяють зробити прогнози щодо майбутніх доходів та витрат, що допомагає створити стратегію управління бюджетом.

Інший критерій "Безкоштовність" було обрано із кількох причин. Безкоштовний додаток дозволяє всім бажаючим отримати доступ до інструментів управління своїми фінансами без необхідності витратити гроші на придбання програми. Багато людей вже мають обмежений бюджет, і витрати на програмне забезпечення можуть бути надмірними. Безкоштовний додаток допомагає зменшити фінансове навантаження на користувачів. Для тих, хто тільки починає вести облік своїх фінансів, важливо мати доступ до безкоштовних інструментів, які дозволяють ознайомитися з процесом без ризику втрати коштів.

Не менш важливим у будь-якому додатку є зручність та інтуїтивно зрозумілість інтерфейсу. У першу чергу, зручний інтерфейс визначається його легкістю використання, що дозволяє користувачам швидко оволодіти програмою і ефективно користуватися нею без зайвих зусиль. Простота та інтуїтивно зрозумілий інтерфейс полегшують навігацію та сприяють швидкому освоєнню програми. Зручний інтерфейс також покращує користувацький досвід, створюючи комфортні умови для використання додатка. Позитивне сприйняття та задоволення від користування програмою забезпечуються завдяки приємному та зручному дизайну. Крім того, зручний інтерфейс забезпечує максимальну ефективність використання, дозволяючи швидко та легко виконувати необхідні завдання. Він спрощує взаємодію з програмою та дозволяє користувачам максимально сконцентруватися на вирішенні своїх фінансових задач.

Обрання критерію "Доступний для Android та iOS" обумовлене кількома ключовими моментами. По-перше, це забезпечує максимальну доступність програми для широкого кола користувачів. Наявність додатка на обох платформах Android та iOS означає, що він може задовольнити потреби більшої аудиторії користувачів, оскільки обидві платформи мають велику кількість прихильників. По-друге, це дозволяє користувачам мати гнучкість у виборі пристрою для використання програми. Підтримка обох платформ дозволяє користувачам використовувати додаток на будь-якому пристрої, що зручно для них. Крім того, підтримка обох платформ сприяє конкурентоспроможності

програми на ринку додатків. Це розширює потенційну аудиторію користувачів і дозволяє досягти більшого поширення програми серед мобільних користувачів.

Можливість планувати майбутні витрати є ключовим аспектом для додатку подібного типу. Планування майбутніх витрат дозволяє користувачам визначити свої фінансові цілі та спланувати витрати заздалегідь, уникнути непередбачених витрат та забезпечити фінансову стабільність в майбутньому. Заздалегідь заплановані витрати допомагають раціонально розподілити фінансові ресурси на різні категорії та покращити ефективність управління бюджетом, включаючи основні потреби, вкладання та розваги. Планування майбутніх витрат може слугувати мотивацією для досягнення фінансових цілей, стимулюючи більш осмислене витрачання коштів та прагнення до фінансової відповідальності.

Можливість внесення даних про витрати та надходження є головною функцією додатку для моніторингу особистого бюджету. Без можливості точного реєстрування цих фінансових транзакцій, додаток втрачає здатність надавати користувачам об'єктивну картину їхнього фінансового стану та здійснювати ефективний аналіз їхніх фінансів.

У відсутність можливості внесення даних про витрати та надходження, додаток стає непридатним для ведення особистого бюджету. Це перешкоджає користувачам в правильному визначенні їхніх фінансових цілей, а також у розробці стратегій з ефективного управління своїми фінансами. Таким чином, можливість внесення даних про витрати та надходження є ключовим елементом, який визначає функціональність та ефективність додатку для моніторингу особистого бюджету.

1.3 Аналіз методів вирішення поставленої задачі

Після детального аналізу аналогів додатків для моніторингу особистого бюджету, а також врахування основних потреб користувачів, команда розробників прийняла рішення щодо створення додатку, який відповідав би

вимогам зручного та інтуїтивно зрозумілого інтерфейсу, а також привабливої кольорової гами.

Однією з головних особливостей додатку став його інтуїтивно зрозумілий інтерфейс, який спрощує введення даних про витрати та надходження, а також забезпечує зручний доступ до основних функцій. Зокрема, були розроблені зручні інструменти для введення фінансових транзакцій та швидкого перегляду поточного стану рахунку.

Однією з ключових особливостей програми стала функція створення графіків, що забезпечує візуалізацію витрат та надходжень до особистого бюджету. Ця функція надає користувачам можливість швидко та зрозуміло оцінити розподіл своїх фінансових ресурсів, виявити основні тенденції та визначити пріоритетні напрямки для оптимізації бюджету. Графічне відображення витрат та надходжень дозволяє користувачам легко виявляти патерни та залежності у своєму фінансовому житті. Наприклад, вони можуть швидко побачити, в які періоди часу вони зазвичай витрачають більше коштів, і прийняти заходи для ефективнішого управління своїми витратами. Завдяки цій функції, користувачі можуть приймати обґрунтовані фінансові рішення, орієнтуючись на чітку та зрозумілу інформацію, надану у вигляді графіків. Вони можуть здійснювати свої фінансові плани та стратегії на основі об'єктивних даних, що сприяє підвищенню їхньої фінансової грамотності та ефективності управління бюджетом.

Поміж інших важливих функцій додатку є розроблена спеціальна функція прогнозування майбутніх доходів та витрат користувача. Ця функція використовує дані про витрати і надходження з попередніх тижнів для створення прогнозу на майбутні періоди. Це надає користувачам можливість розпланувати свої фінанси на майбутнє, враховуючи попередні тенденції та прогнозовані доходи і витрати. Завдяки цій функції, користувачі можуть отримати більш чітке уявлення про те, якими можуть бути їхні фінансові можливості у майбутньому. Вони можуть побачити, наскільки реалістичними є їхні фінансові цілі та плани, і внести відповідні зміни до свого бюджету. Ця функція є важливим

інструментом для фінансового планування та допомагає користувачам бути більш впевненими в своїх фінансових рішеннях. Вона допомагає зменшити ризики та підвищити рівень контролю над своїми фінансами в майбутньому.

1.4 Постановка задач на проектування

Магістерська кваліфікаційна робота скерована на створення програмного забезпечення, яке буде відповідати потребам у моніторингу особистого бюджету, а також прогнозуванню майбутніх доходів та витрат.

Основна увага дослідження приділяється розробці алгоритмів та інтерфейсу даного програмного засобу. Зокрема, робота визначає вимоги до інтерфейсу, які повинні бути задоволені у процесі розробки.

- Інтуїтивно зрозумілий інтерфейс із приємною кольоровою гамою;
- Наявність кількох сторінок для зменшення нагромадження елементів на сторінці;

- Наявність діаграм, для кращої візуалізації інформації.

Для реалізації додатку, було розроблено наступні вимоги:

- Розробити алгоритм додавання витрат;
- Розробити алгоритм додавання надходження;
- Розробити алгоритм створення діаграми;
- Видалення записів;
- Розробити алгоритм прогнозування майбутніх доходів та витрат;
- Розробити програмний продукт, який вирішить поставлені проблеми та реалізує заплановані задачі;
- Провести тестування програмного продукту.

1.5 Висновки

Проведений глибокий аналіз сучасного стану проблематики підтвердив важливість теми обраного дослідження. У ході цього дослідження був проведений ретельний порівняльний аналіз існуючих аналогів розглядуваного продукту з метою виявлення можливостей для його подальшої оптимізації та вдосконалення з метою подальшої реалізації. Вивчення сильних та слабких сторін аналогів дозволило ідентифікувати ключові аспекти, які можна покращити у новому продукті.

Отримані відомості послужили основою для розробки детальних вимог до майбутнього продукту, а також для формулювання конкретних завдань та методології для ефективного проектування нового програмного продукту.

2 ВИБІР МЕТОДУ ПРОГНОЗУВАННЯ ТА РОЗРОБКА АРХІТЕКТУРИ МОБІЛЬНОГО ДОДАТКУ

2.1 Загальна характеристика методів моніторингу та прогнозування витрат

Моніторинг витрат та їх прогнозування є актуальною та важливою темою для найрізноманітніших сфер та галузей життя більшої частини суспільства. З метою проведення об'єктивного аналізу стану даної галузі проведемо порівняння та ознайомимось з методами та системами, що активно використовуються в актуальних аналогах та є конкурентно спроможними рішеннями для певних сфер життя.

Моніторинг витрат – це процес систематичного відстеження та контролю за фінансовими витратами. Це означає регулярне визначення, куди, на що і як були витрачені гроші. Основна мета моніторингу витрат полягає в тому, щоб краще розуміти свої фінанси, знаходити області для зменшення витрат, створювати бюджет та планувати фінансові цілі [6].

Існує різноманіття методів прогнозування витрат, які використовуються в практиці для прогнозування майбутніх фінансових витрат. Кожен з цих методів має свої переваги та недоліки і може бути використаний в залежності від конкретних потреб та характеристик даних. Деякі з найпоширеніших методів[7]:

- **Метод середнього.** Цей метод простий і полягає в обчисленні середнього значення попередніх витрат для прогнозування майбутніх. Хоча цей метод може бути ефективним для стабільних даних без трендів або сезонності, він може бути менш ефективним у випадку зміни умов.
- **Експоненційне згладжування.** Цей метод враховує зважене значення попередніх витрат, де вага кожного значення зменшується згідно з коефіцієнтом згладжування. Цей метод дозволяє краще реагувати на зміни в даних і може бути корисним для прогнозування майбутніх витрат.

- **Метод Holt-Winters.** Цей метод підходить для прогнозування часових рядів зі змінним трендом та сезонністю. Він враховує тренд та сезонність в даних і може бути корисним для прогнозування майбутніх витрат в умовах які змінюються.
- **Метод простого рухомого середнього.** Цей метод згладжує великий обсяг даних, використовуючи середнє значення попередніх періодів. Він простий у використанні та може бути ефективним для прогнозування майбутніх витрат.

Тож зважаючи на те, що видів моніторингу витрат існує багато, доцільно порівняти їх між собою та обрати найкращих із цих видів для реалізації.

2.2 Особливості застосування методу середнього для прогнозування витрат

Метод середнього – це один з найпростіших методів прогнозування, який використовується для оцінки майбутніх значень на основі середнього арифметичного попередніх даних. Основна ідея полягає в тому, щоб взяти середнє значення попередніх спостережень як прогноз майбутніх значень [7].

Щоб застосувати метод середнього, спочатку необхідно зібрати достатньо даних про попередні значення. Потім розраховується середнє арифметичне цих значень, яке використовується як прогноз майбутніх значень. Чим більше даних ми маємо, тим точніший може бути прогноз.

Наприклад, якщо ми маємо дані про щоденні витрати протягом останнього місяця, можна розрахувати середнє арифметичне цих витрат і використати його як прогноз на майбутній місяць. Обрахунок майбутніх витрат у цьому випадку буде здійснюватися за формулою:

$$C = \frac{\sum_{i=1}^n X_i}{n},$$

де C – сума майбутніх витрат (середнє значення),

X_i – кожне окреме значення у списку витрат,

n – кількість значень в списку витрат.

Метод середнього відзначається рядом переваг:

- Простота: Цей метод дуже простий у реалізації та розумінні. Він не потребує складних математичних розрахунків або спеціальних навичок для його використання.
- Швидкість: Розрахунок середнього значення досить швидкий, особливо при наявності великої кількості даних.
- Стабільність: У випадку стабільних даних без суттєвих змін або трендів, метод середнього може дати стабільний та надійний прогноз.
- Підходить для короткострокових прогнозів: Метод середнього особливо ефективний для короткострокових прогнозів, коли тренди та сезонність не є вирішальними факторами.
- Зручність у застосуванні: Цей метод можна легко використовувати для швидкого отримання прогнозу без необхідності у глибокому аналізі даних.
- Мінімальний обсяг даних: Для застосування методу середнього не потрібно великої кількості даних. Зазвичай, навіть невеликий набір даних може бути достатнім для створення прогнозу.

До недоліків можна віднести:

- Вразливість до випадкових аномалій: метод середнього може дати неточні результати, якщо вихідні дані містять великі випадкові коливання або аномалії.
- Не враховує динаміку змін: цей метод не враховує зміни трендів або сезонних впливів у даних, що може призвести до неточних прогнозів у довгостроковій перспективі.
- Використання усіх даних однаково: метод середнього припускає, що всі дані мають однакову вагу, що не завжди відповідає реальності, особливо якщо деякі дані більш важливі або репрезентативні, ніж інші.

- Витрати на зберігання та обробку даних: у випадку великої кількості даних метод середнього може потребувати значних обчислювальних ресурсів та часу на їх обробку.
- Не підходить для даних з великими коливаннями: у випадку даних зі значними коливаннями або нерегулярними змінами, метод середнього може дати неточні результати, оскільки він не враховує динаміку зміни значень.

2.3 Особливості застосування експоненційного згладжування для прогнозування витрат

Експоненційне згладжування – це метод аналізу часових рядів, який використовується для прогнозування майбутніх значень на основі попередніх даних. У цьому методі кожне нове значення вагується залежно від його відстані від поточного моменту часу, з присвоєнням більшої ваги ближнім значенням.

Експоненційне згладжування дозволяє згладжувати випадкові коливання та виявляти загальні тенденції в даних. Цей метод часто використовується для прогнозування витрат, прибутку або інших фінансових показників, особливо у випадках, коли дані мають тренди або сезонні коливання. Для застосування експоненційного згладжування необхідно обрати певний параметр згладжування, який визначає ступінь впливу нових даних на прогноз. Цей метод може бути ефективним для короткострокових прогнозів, але вимагає ретельного аналізу та налаштування параметрів для отримання найкращих результатів [8]. Нижче зображено формулу, за якою можна вираховувати майбутні витрати.

$$F_{t+1} = \alpha * Y_t + (1 - \alpha) * F_t,$$

де F_{t+1} – прогнозована витрата у майбутній період

Y_t – фактична витрата в поточному періоді

F_t – прогнозована витрата в попередньому періоді

α – коефіцієнт згладжування (вагомий коефіцієнт) де $0 < \alpha < 1$

Переваги експоненційного згладжування включають:

- Простота застосування: Метод відносно простий у реалізації та зрозумілий для аналітика без великої кількості спеціалізованого навчання.
- Ефективність для стаціонарних даних: Експоненційне згладжування добре працює зі стаціонарними даними, де відсутні явні тренди чи сезонні зміни.
- Швидкість прогнозування: Він може швидко здійснити прогноз на основі невеликої кількості даних, що робить його ефективним для короткострокових прогнозів.
- Добре адаптується до змін: Згладжування з урахуванням експоненції дозволяє швидко реагувати на зміни в даних, зберігаючи здатність прогнозувати майбутні значення.
- Можливість контролювати рівень згладжування: Параметр згладжування дозволяє аналітику контролювати вплив нових даних на прогноз, що дозволяє зберігати баланс між гладкістю прогнозу та реагуванням на зміни.
- Робота з великими обсягами даних: Метод може бути використаний з великими обсягами даних, і, навіть при цьому, зберігати ефективність та точність прогнозування.
- Невимогливість до структури даних: Він може працювати з даними різної структури, включаючи нерегулярні та пропущені значення, що робить його варіантом для широкого спектру застосувань.

Недоліки методу експоненційного згладжування включають:

- Чутливість до початкових значень: Прогнози, отримані за допомогою експоненційного згладжування, можуть бути чутливими до початкових значень даних. Невірні або неточні початкові значення можуть призвести до неточних прогнозів.
- Неспроможність прогнозувати зміни тренду: Цей метод не може ефективно прогнозувати або враховувати раптові зміни тренду в даних, що може призвести до неправильних прогнозів у разі виникнення таких змін.

- Неможливість врахування сезонності: Експоненційне згладжування не може ефективно враховувати сезонні коливання у даних, що може призвести до неточних прогнозів у випадку, якщо такі коливання присутні.
- Потреба у налаштуванні параметрів: Параметри експоненційного згладжування потребують налаштування, що може бути складним завданням і вимагати досвіду та експертизи.
- Неможливість врахування зовнішніх факторів: Цей метод не враховує зовнішніх факторів, які можуть впливати на дані, такі як економічні або соціальні зміни, що може призвести до неточних прогнозів у випадку виникнення таких факторів.

Метод ARIMA (Autoregressive Integrated Moving Average) є одним з найпоширеніших і ефективних методів прогнозування часових рядів. Він використовується для аналізу та прогнозування даних, які мають часову залежність, такі як фінансові дані, показники економічного зростання, демографічні тенденції тощо.

2.4 Особливості застосування метода Хольта-Вінтерса для прогнозування витрат

Метод Хольта-Вінтерса (Holt-Winters) – це статистичний метод для прогнозування часових рядів, який враховує тренд, сезонність та згладжування даних. Цей метод розвинувся від методу згладжування Хольта, що враховує тренд, до методу згладжування Хольта-Вінтерса, який також включає компонент сезонності. Основною ідеєю методу є те, що він здатен узагальнити попередні дані та використовувати їх для прогнозування майбутніх значень у часовому ряді.

Метод Хольта-Вінтерса складається з трьох основних компонентів: згладжування ряду, згладжування тренду та згладжування сезонності. Під час аналізу вхідних даних метод визначає стабільність тренду та сезонних коливань і використовує цю інформацію для прогнозування майбутніх значень.

Формула методу Хольта-Вінтерса включає три основні компоненти зображені нижче: згладжування ряду (level), згладжування тренду (trend) і згладжування сезонності (seasonality)

Згладжування ряду (Level):

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

Згладжування тренду (Trend):

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$$

Згладжування сезонності (Seasonality):

$$s_t = \gamma(y_t - l_t) + (1 - \gamma)s_{t-m}$$

Прогнозоване значення (Forecast):

$$\widehat{y_{t+h}} = l_t + h * b_t + s_{t-m+h}$$

Відповідно до формули, значення:

l_t – рівень (середнє значення) на момент часу

b_t – тренд на момент часу

s_t – сезонна компонента на момент часу

y_t – фактичне спостереження на момент часу

α, β^*, γ – параметри згладжування,

m – період сезонності,

h – кількість періодів вперед для прогнозування.

Переваги методу Хольта-Вінтерса включають:

- Здатність моделювати тренд і сезонні коливання: Метод Хольта-Вінтерса дозволяє врахувати як тренд, так і сезонні зміни в часовому ряді, що робить його ефективним для прогнозування в умовах наявності цих компонент.
- Гнучкість параметрів: Параметри методу Хольта-Вінтерса (такі як коефіцієнти згладжування) можуть бути налаштовані для відображення різних характеристик даних, що дозволяє збільшити точність прогнозів.
- Адаптивність до змін у структурі даних: Метод Хольта-Вінтерса може адаптуватися до змін у тренді, сезонності та рівні значень у часовому ряді, що дозволяє враховувати динаміку змін та підтримувати точність прогнозів навіть при зміні умов.

- Ефективність для короткострокових та довгострокових прогнозів: Метод Хольта-Вінтерса може бути ефективним як для прогнозування короткострокових, так і для довгострокових тенденцій у часових рядах, що робить його універсальним і корисним для різних потреб прогнозування.

- Доступність в багатьох статистичних програмах: Багато статистичних пакетів та програм для аналізу даних містять реалізації методу Хольта-Вінтерса, що робить його доступним для широкого кола користувачів.

- Добрі результати на практиці: Метод Хольта-Вінтерса широко використовується в практиці і показує добрі результати в багатьох галузях, включаючи фінанси, економіку, та інші сфери.

Недоліки методу Хольта-Вінтерса включають:

- Чутливість до початкових умов: Метод може бути досить чутливим до початкових значень параметрів, таких як початкові значення рівня (level), тренду (trend) і сезонності (seasonality). Неправильне визначення цих початкових умов може призвести до неточних прогнозів.

- Обмежена здатність моделювати складні зміни: Метод Хольта-Вінтерса має обмежену здатність враховувати складні зміни у часовому ряді, такі як стрибкоподібні зміни або неочікувані коливання. В таких ситуаціях можуть виникати неточності у прогнозах.

- Не враховує зовнішніх факторів: Метод Хольта-Вінтерса базується виключно на історичних даних і не враховує зовнішніх факторів, таких як економічні або соціальні зміни, що можуть вплинути на поведінку часового ряду.

- Потребує стабільності у структурі даних: Метод вимагає стабільності у структурі даних, включаючи стабільність тренду, сезонності та інших параметрів моделі. У випадку зміни структури даних, модель може втратити свою ефективність.

- Складність параметризації: Визначення оптимальних значень параметрів (таких як коефіцієнти згладжування) може бути складним завданням

і вимагати додаткового аналізу та експертного досвіду. Неправильний вибір параметрів може призвести до неточних прогнозів.

2.5 Особливості застосування метода простого рухомого середнього для прогнозування витрат

Метод простого рухомого середнього (Simple Moving Average, SMA) є одним із найбільш базових методів аналізу часових рядів. Він використовується для згладжування шумів та виявлення основних трендів у даних. Основна ідея полягає в тому, щоб розрахувати середнє значення вартості або іншого показника протягом певного періоду часу, переміщуючи вікно цього періоду по всьому часовому ряду.

Проста рухома середня може бути використана для згладжування шумів у даних, виявлення трендів та підтвердження змін напрямку. Також вона є важливим інструментом для технічного аналізу на фінансових ринках, де її часто використовують для прогнозування майбутніх цін на акції або інших фінансових інструментів [9]. Нижче формула цього методу:

$$SMA_t = \frac{X_{t-1} + X_{t-2} + \dots + X_{t-n}}{n}$$

Де $X_{t-1} + X_{t-2} + \dots + X_{t-n}$ – значення даних за n періодів (зазвичай це дні, тижні або інші відрізки часу)

n – кількість періодів, за яку обчислюється середнє значення (розмір вікна)

Переваги методу простої рухомої середньої (SMA) включають:

- Простота розрахунку: SMA легко розуміти та обчислити. Вона не вимагає складних математичних розрахунків і може бути легко реалізована у більшості програмних платформ.
- Згладжування шумів: SMA дозволяє згладити випадкові коливання в даних, допомагаючи виділити основний тренд і виявити довгострокові зміни.
- Використання для прогнозування: SMA може бути використана для прогнозування майбутніх значень на основі здійснених спостережень. Вона

надає засновані на даних підстави для прийняття рішень щодо майбутніх тенденцій.

- Придатність для технічного аналізу: У фінансовому аналізі, SMA часто використовується для виявлення трендів на ринку. Вона допомагає інвесторам та трейдерам у прийнятті рішень щодо входу та виходу з позицій.
- Універсальність застосування: SMA може бути застосована в різних областях, таких як фінанси, економіка, наука про дані та інші, для аналізу та прогнозування часових рядів даних.
- Робота з неперервними даними: SMA добре підходить для роботи з неперервними часовими рядами даних, де дані збираються відповідно до фіксованого періоду часу (наприклад, щоденно, щотижнево і т. д.).
- Інтерпретація: SMA має просту інтерпретацію. Її значення вказує на середнє значення даних протягом певного періоду часу, що дозволяє легко зрозуміти тенденції і зміни у даних.

Недоліки методу простої рухомої середньої включають:

- Затримка від реального часу: Одним з головних недоліків SMA є те, що вона використовує попередні значення для обчислення поточного значення. Це може призвести до затримки в оцінці та реакції на зміни в даних.
- Чутливість до вибору періоду: Вибір періоду для SMA може суттєво вплинути на результати аналізу. Короткий період може призвести до більшої чутливості до шумів, тоді як довгий період може призвести до більшої затримки у виявленні трендів.
- Неадекватність для швидко змінюваних даних: SMA не добре пристосована для даних з великими коливаннями або швидкими змінами. Вона може бути менш ефективною у виявленні і прогнозуванні трендів в таких умовах.
- Ризик втрати інформації: Оскільки SMA використовує лише попередні значення для обчислення, вона може пропустити важливу

інформацію, яка виникає в поточний момент, особливо при великій затримці в даних.

- Неадекватність для неперервних даних: SMA не завжди добре пристосована для роботи з неперервними часовими рядами даних, де відсутній фіксований період збору даних.
- Недостатня чутливість до змін у тренді: У деяких випадках SMA може бути повільною у виявленні змін у тренді, особливо коли тренд змінюється дуже швидко.
- Неявність урахування ваги для останніх значень: У простій рухомій середній всі значення мають однакову вагу, що може призвести до недооцінки значення останніх даних.

2.6 Обґрунтування вибору методу прогнозування

На основі аналізу різних методів прогнозування, експоненційне згладжування виявилось найбільш відповідним варіантом для використання у програмному додатку. Ось деякі висновки по порівнянню з іншими методами:

- Простий у реалізації: Експоненційне згладжування відзначається простотою в реалізації, що робить його привабливим варіантом для включення у програмний код.
- Добре пристосований для наближення трендів: Цей метод добре пристосований для прогнозування трендів у часових рядах, що може бути корисним для визначення майбутніх витрат користувачів.
- Зменшена чутливість до шуму: Експоненційне згладжування може забезпечити меншу чутливість до випадкових коливань у даних, що може покращити точність прогнозів.
- Можливість простого керування параметрами: Цей метод має лише декілька параметрів, які можна легко налаштувати, що спрощує процес розробки та налаштування програмного забезпечення.

Отже, на основі цих факторів, експоненційне згладжування може бути найбільш підходящим методом для реалізації в програмному додатку для прогнозування майбутніх витрат.

2.7 Розробка архітектури мобільного додатку

Перед початком розробки програмного додатку потрібно реалізувати діаграму роботи додатку. Цей етап дозволяє візуально представити структуру та взаємозв'язки між його компонентами. Такий підхід сприяє кращому розумінню розробниками того, як організувати код та функціонал додатку. Він також допомагає виявити ключові функції програми та уточнити вимоги до неї. Розглядаючи діаграму, можна виявити можливі надлишкові компоненти або функції, що дозволяє оптимізувати використання ресурсів та прискорювати розробку. Крім того, створення діаграми полегшує внесення змін у додаток під час його розробки, що є важливим для подальшого оптимізованого розвитку програмного продукту.

UML (Unified Modeling Language) діаграми є стандартним інструментом для візуалізації, специфікації, конструювання та документування компонентів програмного забезпечення. Вони допомагають розробникам, аналітикам і іншим зацікавленим сторонам зрозуміти структуру та поведінку системи [11].

Діаграма послідовності – це вид діаграми уніфікованої мови моделювання (UML), яка відображає послідовність взаємодії між різними об'єктами або компонентами в системі. Вона показує, які повідомлення передаються між цими об'єктами та в якому порядку вони виконуються [11].

Зазвичай діаграми послідовності використовуються для моделювання процесів взаємодії між об'єктами в системі в рамках конкретного сценарію або функціональної діяльності. Вони можуть бути корисним інструментом для візуалізації та аналізу взаємодії компонентів програмної системи або взаємодії користувача з системою.

Діаграми послідовності відображають об'єкти в системі у вигляді вертикальних ліній, а повідомлення між цими об'єктами у вигляді стрілок, які вказують напрямок взаємодії та порядок виконання операцій.

Цей тип діаграм допомагає розуміти логіку взаємодії між компонентами системи та ідентифікувати можливі проблеми або вузькі місця в процесі. Вони є корисним інструментом для аналізу та проектування систем, а також для документування взаємодії між об'єктами в проекті програмного забезпечення.

Діаграма, що представлена на рисунку 2.1, описує взаємодію користувача з різними сторінками додатку під час певних операцій.

Користувач починає з головної сторінки, де відбувається завантаження додатку і отримання даних про стан бюджету. Після цього користувач може додати нову транзакцію, що спрямовує його на сторінку створення нової транзакції. Тут він вводить необхідну інформацію і створює транзакцію, після чого повертається на головну сторінку або переходить на сторінку зі статистикою витрат.

На сторінці зі статистикою користувач може переглядати детальну інформацію про свої витрати. Архітектура додатку включає кілька основних компонентів: користувацький інтерфейс з різними сторінками та логіку додатку, яка обробляє дії користувача. Це забезпечує просту та зрозумілу систему для моніторингу особистих фінансів, зосереджену на відображенні стану бюджету, додаванні транзакцій і перегляді статистики.

Діаграма діяльності є видом структурної діаграми, що відображає послідовність дій, виконаних об'єктами або системами в рамках конкретного процесу або сценарію. Цей графічний інструмент використовується для моделювання діяльності, взаємодії та потоків даних між об'єктами або учасниками в системі. Діаграми діяльності допомагають візуалізувати робочі процеси, а також виявляти потенційні проблеми та можливості для оптимізації системи або додатку. Вони часто використовуються при аналізі, проектуванні та розробці програмного забезпечення, бізнес-процесів, а також в інших областях, де потрібно уявити та зрозуміти послідовність дій.

Створення діаграми діяльності для додатку має кілька переваг, які раціоналізують використання цього інструменту в процесі розробки програмного забезпечення.

Діаграма діяльності дозволяє зображати послідовність операцій та взаємодію об'єктів у додатку, що полегшує розуміння принципу його роботи. Такий інструмент дозволяє виявити можливі проблеми у проекті заздалегідь та уникнути їх у подальшій розробці. Діаграма діяльності може стати корисним документом для комунікації між учасниками проекту, сприяючи обміну ідеями та уточненню вимог.

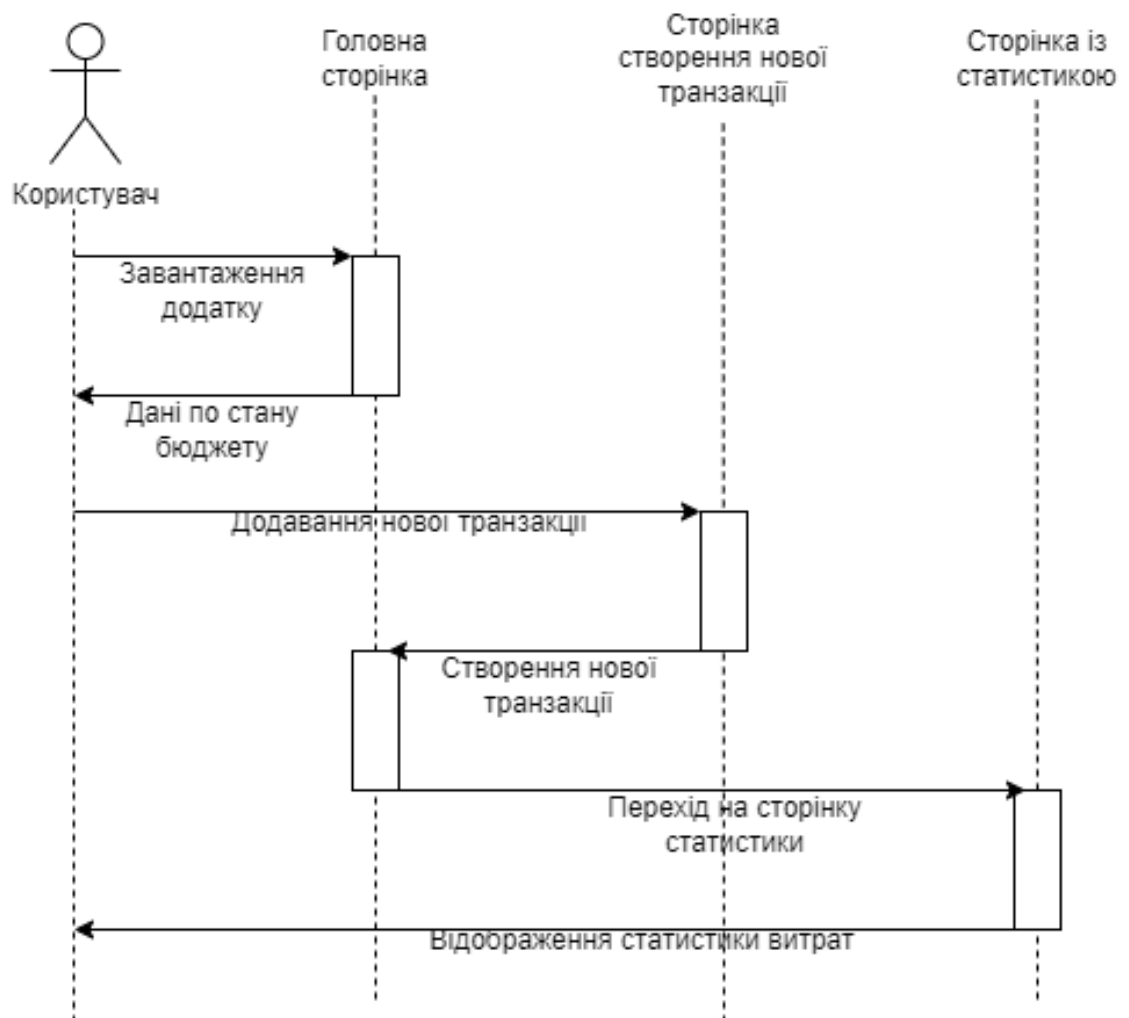


Рисунок 2.1 – Діаграма послідовності додатку

Діаграма діяльності, представлена на рисунку 2.2, відображає процеси використання додатку для моніторингу особистого бюджету, підкреслюючи

ключові етапи взаємодії користувача з додатком. Вона доповнює попередню діаграму зображену на рисунку 2.1, розширюючи розуміння архітектури додатку.

Спочатку користувач відкриває додаток, після чого потрапляє на головну сторінку, яка відображає історію транзакцій та фінансові дані. З головної сторінки користувач може обрати один із двох основних шляхів: створити нову транзакцію або переглянути статистику.

При виборі опції створення нової транзакції, користувач вводить необхідну інформацію, що призводить до оновлення історії транзакцій. Після цього користувач повертається на головну сторінку з оновленими даними.



Рисунок 2.2 – Діаграма діяльності додатку

Якщо ж користувач обирає перегляд статистики, він переходить на сторінку, де може переглянути детальну інформацію про свої витрати. Після перегляду статистики, користувач може знову повернутися на головну сторінку для подальших дій.

Архітектура додатку включає такі основні компоненти:

- Головна сторінка: Відображає історію транзакцій та поточний стан фінансів, а саме сума надходжень, сума витрат, а також сума, що залишилася у користувача на поточний момент часу. Є початковою точкою взаємодії після відкриття додатку.
- Функціональність створення нової транзакції: Дозволяє користувачу додавати нові транзакції, після чого історія транзакцій оновлюється.
- Функціональність перегляду статистики: Надає користувачу можливість переглядати статистичні дані про витрати відповідно до обраного періоду часу.

Ця діаграма ілюструє основний робочий процес користувача у додатку, включаючи відкриття додатку, навігацію між головною сторінкою, створенням нових транзакцій та переглядом статистики. Це підкреслює простоту та зручність використання додатку для моніторингу особистого бюджету, що забезпечує ефективне управління фінансами.

Користувач починає з відкриття додатку, де він потрапляє на головну сторінку, що надає огляд поточного фінансового стану. З головної сторінки користувач може легко переходити до створення нових транзакцій, вводячи доходи чи витрати, що автоматично оновлює загальний бюджет. Крім того, користувач має можливість переглядати детальну статистику, яка відображає фінансові тренди та аналітику, допомагаючи краще розуміти та керувати своїми фінансами.

Інтуїтивно зрозумілий інтерфейс дозволяє користувачу швидко знаходити необхідну інформацію та здійснювати фінансові операції без зайвих зусиль. Цей процес підтримується простими та логічними навігаційними елементами, що

робить додаток доступним для широкого кола користувачів, включаючи тих, хто не має спеціальних фінансових знань.

2.8 Висновки

У другому розділі було проведено аналіз математичних методів, які можна використати для прогнозування майбутніх витрат. На основі проведеного аналізу метод експоненційного згладжування було обрано як найбільш ефективний метод прогнозування, оскільки воно забезпечує достатню точність та адаптується до змін в часових рядах.

Запропоновано архітектуру інтегрованої інформаційної системи, яка об'єднує в собі функції моніторингу витрат сімейного бюджету і прогнозування майбутніх доходів та витрат, забезпечує автоматизовану обробку фінансової інформації, що дозволяє користувачам точно передбачати свої майбутні фінансові потреби.

Результатом дослідження стала розробка структури інтерфейсу, яка забезпечує зручний та інтуїтивно зрозумілий інструмент для користувачів. Ретельно продумана система візуального модулювання роботи додатку забезпечує зручність та ефективність використання.

3. РОЗРОБКА МОДУЛІВ ПРОГРАМНОГО ЗАСОБУ

3.1 Варіантний аналіз та обґрунтування вибору засобів реалізації програмного продукту

У всьому світі кількість користувачів смартфонів досягла вражаючих 6,7 мільярда. Цей значний приріст є показником стрімкого зростання, особливо з огляду на те, що наприкінці 2016 року було менше ніж 2 мільярди активних користувачів смартфонів. На сьогоднішній день, 98% смартфонів працюють на двох основних операційних системах, і це, безумовно, Android та iOS.

Однією з ключових причин є те, що Google є власником операційної системи Android, тоді як Apple володіє iOS; обидві компанії є трільйонними корпораціями з вражаючими ресурсами. Проте, їхній успіх, безумовно, не обмежується лише фінансовими можливостями. Як iOS, так і Android завжди опереджають технічні досягнення та пропонують користувачам передові можливості. Обидві операційні системи також регулярно оновлюються та вдосконалюються для забезпечення безперервної роботи користувачів, що робить їх надзвичайно популярними. Вони просто примушують своїх користувачів закохуватися у їх зручність використання та функціональність. Тому не дивно, що конкуренти не мають значущих шансів протистояти цим двом гігантам операційних систем [12].

У третьому кварталі 2021 року споживачі витратили практично \$34 мільярди на преміальні додатки, покупки в межах додатків та першу інсталяцію програм на платформах Android та iOS. Незважаючи на те, що частка ринку iOS на 50% менша, ніж у Android, дохід від iOS склав \$21,5 мільярда, що практично вдвічі перевищує дохід від Android, який склав \$12,1 мільярда.

iOS має значну перевагу на розвинених ринках, що, зокрема, дозволяє йому здобувати вищі доходи, оскільки користувачі на цих ринках більш схильні витратити більше на програми.

Отже, коли йдеться про частку світового ринку мобільних операційних систем, на ринку відсутні треті гравці, окрім Android та iOS. З поточних статистичних даних, які ми тут бачимо, очевидно, що політика Android полягає в максимальному проникненні на ринок та приверненні максимальної кількості користувачів. У той час, як iOS має тенденцію зберігати присутність на розвинених ринках, орієнтовану на дохід. Android перемагає у грі за частку ринку між iOS та Android, але iOS вирівнюється, маючи значно кращий потік доходів [12].

Спираючись на тенденцію збільшення користувачів iOS операційною системою, можна розглянути можливість використання кросплатформеної розробки. Кросплатформена розробка, порівняно з нативною, може бути кращим вибором з кількох причин [13]:

- Економічна вигода: Розробка кросплатформених додатків зазвичай обходиться дешевше, оскільки розробники можуть використовувати один кодову базу для створення додатків для різних платформ. Це зменшує витрати на розробку, тестування і підтримку.
- Швидкість розробки: Кросплатформена розробка дозволяє швидше виводити продукт на ринок, оскільки розробники можуть одночасно працювати над версіями для різних платформ, уникнувши необхідності писати окремий код для кожної.
- Можливість покриття більшої аудиторії: Кросплатформені додатки дозволяють досягти більшої аудиторії, оскільки вони можуть бути запуснені на різних платформах, таких як Android і iOS, а також на веб-платформах. Це полегшує розповсюдження та доступність додатка для більшої кількості користувачів.
- Простота підтримки: Коли додаток розробляється для кількох платформ з використанням однієї кодової бази, підтримка та виправлення помилок стають простішими і менш часовитратними, оскільки зміни потрібно робити лише в одному місці.

- Можливість використання одноразових інструментів та технологій: Кросплатформні фреймворки, такі як React Native, Flutter або Xamarin, дозволяють використовувати відомі інструменти та технології для розробки, такі як JavaScript, Dart або C#, що полегшує перехід для розробників з іншого досвіду.

Отже, на підставі обґрунтувань, наведених вище, вибір кросплатформної розробки може бути переважним у багатьох випадках. Кросплатформні рішення надають економічну вигоду, прискорюють процес розробки, дозволяють досягти більшої аудиторії та спрощують підтримку та покращення додатку.

Розробка кросплатформних додатків дозволяє компаніям ефективно використовувати ресурси, зменшуючи витрати та час, необхідний для розробки та підтримки додатків для різних платформ. Крім того, вона надає можливість більш швидкого виведення продукту на ринок і досягнення більшого кола користувачів.

Таким чином, вибір кросплатформної розробки може бути обґрунтованим з точки зору ефективного використання ресурсів, швидкості розробки та досягнення більшої аудиторії.

Для більш докладного аналізу можливостей, які можуть вплинути на розробку мобільного додатку, було проведено аналіз трьох провідних мов програмування у сфері кросплатформної розробки мобільних додатків: JavaScript, Dart та C#. У цьому порівняльному дослідженні були розглянуті різноманітні аспекти кожної мови програмування та технологій, такі як мова програмування, рівень продуктивності, а також унікальні та спільні характеристики, з метою визначення, яка з них найкраще підходить для вирішення потреб розробки поточного проекту [13].

JavaScript:

- Мова програмування: JavaScript є широко використовуваною мовою програмування, особливо в сфері веб-розробки.
- Фреймворки: Для кросплатформної розробки мобільних додатків, основним фреймворком є React Native.

- Швидкодія: Зазвичай, додатки, розроблені з використанням React Native, можуть виявити незначне сповільнення порівняно з нативними додатками.

Dart:

- Мова програмування: Dart розроблена компанією Google та використовується з фреймворком Flutter для розробки кросплатформених додатків.

- Фреймворки: Головним фреймворком є Flutter, що забезпечує швидку та ефективну розробку мобільних додатків.

- Швидкодія: Flutter відомий своєю високою швидкодією та відмінною продуктивністю, що робить його популярним вибором для кросплатформенної розробки.

C#:

- Мова програмування: C# використовується разом з фреймворком Xamarin для кросплатформенної розробки.

- Фреймворки: Основний фреймворк – Xamarin, дозволяє створювати додатки для різних платформ на мові C#.

- Швидкодія: Додатки, розроблені з використанням Xamarin, відомі своєю хорошою продуктивністю та швидкодією, близькою до нативних додатків.

Порівнявши мову програмування Dart та фреймворк Flutter з іншими технологіями кросплатформенної розробки, можна прийти до висновку на користь цієї комбінації з наступних причин:

- Швидкодія та продуктивність: Dart та Flutter відомі своєю високою швидкодією та продуктивністю. Завдяки використанню компіляції в машинний код та вбудованому готелю віртуальної машини Dart (VM), додатки, розроблені з використанням Flutter, виконуються швидко та ефективно.

- Гнучкість та кросплатформенність: Dart та Flutter дозволяють створювати додатки для різних платформ, включаючи Android, iOS, веб та навіть

настільні операційні системи. Це дозволяє розробникам максимально охопити аудиторію користувачів та розширити потенційну аудиторію свого продукту.

- Багаторазове використання коду та компонентів: Dart та Flutter пропонують механізми для перевикористання коду та компонентів між різними платформами. Це дозволяє розробникам ефективно використовувати ресурси та прискорює процес розробки.

- Модульність та простота розробки інтерфейсу користувача: Flutter використовує декларативний підхід до створення інтерфейсу користувача, що спрощує розробку та підтримку UI. Це дозволяє створювати красиві та динамічні інтерфейси з меншими зусиллями.

Отже, комбінація мови програмування Dart та фреймворку Flutter видається вигідною вибором для розробки мобільних додатків, забезпечуючи високу швидкодію, гнучкість, багаторазове використання коду та простоту розробки інтерфейсу користувача.

3.2 Розробка структури інтерфейсу для мобільного додатку

Дизайн інтерфейсу описує процес створення вигляду та взаємодії програмного забезпечення або пристрою з користувачем. Це охоплює естетичне візуальне оформлення, методи взаємодії та способи представлення інформації, що зрозумілі та доступні для користувача. Головна мета дизайну інтерфейсу - створити продукт, який легко користуватися, ефективно виконує свої функції та забезпечує задоволення від використання [14].

Дизайн інтерфейсу є ключовим елементом будь-якого програмного продукту або пристрою, оскільки він визначає, як користувачі будуть сприймати та взаємодіяти з ним. Він орієнтований на створення зручного та інтуїтивно зрозумілого середовища, яке допомагає користувачам легко виконувати свої завдання без зайвих перешкод або плутанини. Гарний дизайн інтерфейсу сприяє покращенню користувацького досвіду та підвищенню задоволення від використання продукту.

Крім того, дизайн інтерфейсу також впливає на репутацію продукту та його конкурентоспроможність на ринку. Привабливий та ефективно спроектований інтерфейс може привернути нових користувачів і збільшити їх лояльність, тоді як неякісний дизайн може відлякати потенційних клієнтів і призводити до негативних відгуків. Важливо також враховувати потреби та переваги цільової аудиторії при розробці дизайну інтерфейсу, щоб забезпечити максимальну користь для кінцевих користувачів [14].

Дизайн інтерфейсу включає в себе розробку елементів, таких як кнопки, меню, поля введення, іконки та графічні елементи, які спільно створюють зручне та логічне середовище для взаємодії з користувачем. Ефективний дизайн дозволяє спростити навігацію, робить інформацію зрозумілою та доступною, та допомагає зменшити час, необхідний для виконання завдань. Також, важливо враховувати адаптивність дизайну для різних пристроїв та розмірів екранів, щоб забезпечити однаково зручний доступ до продукту на різних платформах.

Необхідно також враховувати вимоги до доступності та універсальності дизайну, щоб забезпечити користування продуктом для всіх категорій користувачів, включаючи людей з обмеженими можливостями. Отже, дизайн інтерфейсу включає в себе не лише естетичні аспекти, але й функціональність, доступність та універсальність, спрямовані на створення оптимального користувацького досвіду для всіх користувачів.

На рисунку 3.1 зображено структуру інтерфейсу головної сторінки додатку для моніторингу витрат з прогнозуванням майбутніх доходів та витрат.

На головній сторінці розміщена карта користувача із найважливішими даними. На карті відображається загальний бюджет користувача, який є на поточний момент. Також зображена загальна сума надходжень поточного місяця та загальна сума витрат поточного місяця.

Також на головній сторінці розміщені вітання та інформація про користувача, а саме ім'я та прізвище, для того щоб більше персоналізувати додаток та зробити його більш дружельним.



Рисунок 3.1 – Структура головної сторінки додатку

На головній сторінці також розміщений блок із історією витрат та надходжень із детальним описом. У кожній транзакції зазначаються сума, візуальна помітка (кольором) типу транзакції (надходження витрата). Також у кожній транзакції є іконка відповідного типу транзакції. Всього є шість видів транзакцій: їжа, навчання, переказ коштів, розваги, транспорт та інше.

У нижній частині екрану розташоване меню навігації, для переходу на інші сторінки додатку.

При натисненні кнопки додати транзакцію відкривається нова сторінка для додавання нової транзакції (рисунок 3.2).

Повернутися назад

Вид операції

Примітка до витрати

Сума

Тип (надходження/витрата)

Дата

Зберегти

Рисунок 3.2 – Структура сторінки додавання транзакції

На сторінці розміщений випадний список із видами операцій. Всього є шість видів: їжа, навчання, переказ коштів, розваги, транспорт та інше. У кожного елемента випадного списку присутня картинка та текст.

Нижче розміщене поле для введення опису. Це поле призначене того, щоб користувач мав можливість додати уточнення, якщо, наприклад, наявних видів витрат не достатньо, щоб описати поточну транзакцію.

Наступне поле – це поле для введення суми. Нижче розміщений випадний список, для того щоб користувач міг обрати тип транзакції. Чи ця транзакція є надходженням чи це витрата. Останнє поле – це селектор дати. При натисненні на це поле, відкривається селектор із календарем, для вибору дати, транзакції.

Також на сторінці присутні дві кнопки. Кнопка зберегти, яка створює нову транзакцію, та кнопка повернутися назад, яка відмінює всі внесені дані та повертає користувача на попередньо відкриту сторінку.

Із головної сторінки можна перейти на сторінку статистики, структура якої зображена на рисунку 3.3

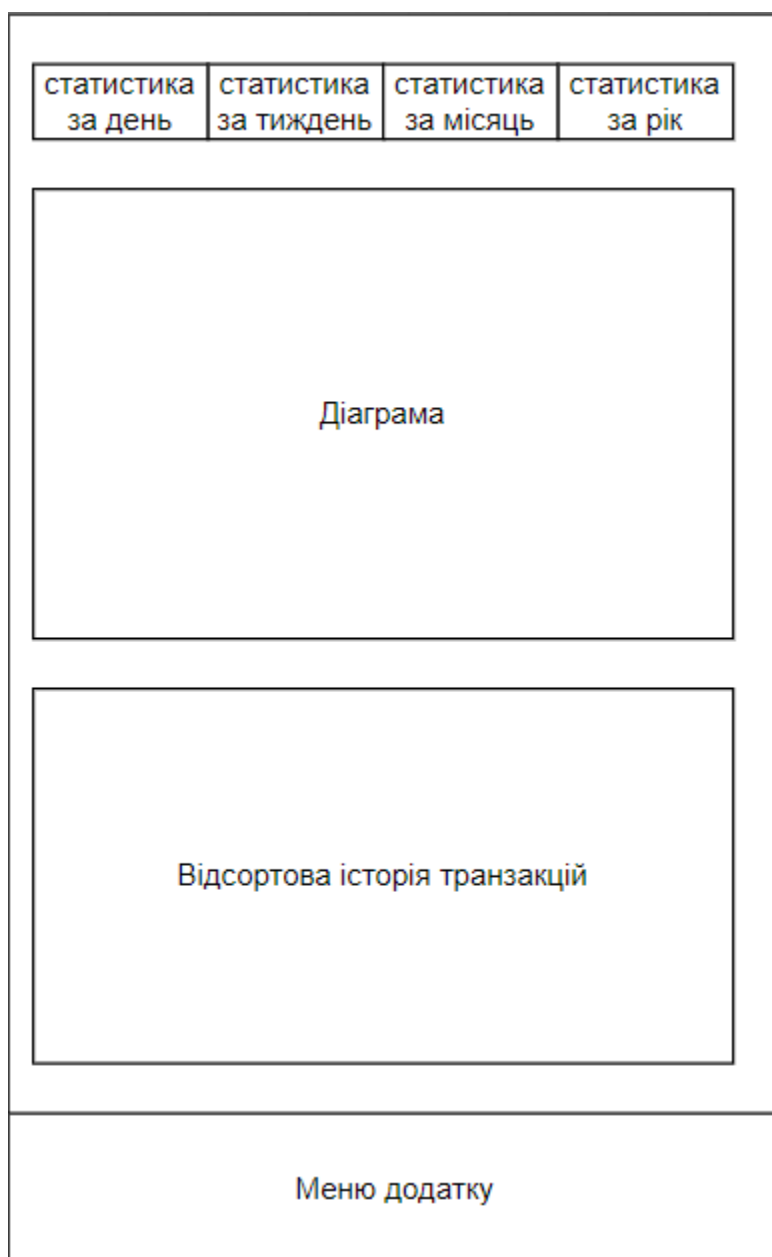


Рисунок 3.3 – Структура сторінки статистики

Сторінка статистики є однією із головних частин додатку на якій розташовується графічно звіт про витрати на надходження користувача.

У верхній частині екрану розміщені кнопки для перемикання між періодами за який створюється графік. Всього є чотири варіанта. Статистика за день, показує статистику витрат за поточний день, будуючи графік відповідно до даних за поточний день. Статистика за тиждень будує графік на основі даних введених за тиждень. Статистика за місяць відображає графічні дані, опираючись на історію надходжень і витрат поточного місяця. Статистика за рік будує графік витрат відповідно до поточного року.

Нижче розташований сам графік витрат. Дані на графіку оновлюються відповідно до даних які вносить користувач. Також вигляд графіку залежить від обраного періоду.

Під графіком розташовано історію витрат, які були зроблені у відповідний період, який користувач обрав. Тобто відображатися витрати будуть не всі, а тільки ті, які відповідають обраному періоду.

У самому низу екрану розташоване меню навігації, за допомогою якого, користувач може перемикатися на інші сторінки додатку

3.3 Розробка програмних модулів мобільного додатку

Flutter – сучасний відкритий фреймворк, розроблений для створення мобільних, веб- та десктоп-додатків. Унікальні характеристики та особливості цього фреймворку гарантують не лише швидкість та ефективність розробки, але й гнучкість у конфігурації, що дозволяє розробникам створювати додатки, які відповідають найсучаснішим стандартам. Завдяки оптимізованій архітектурі, Flutter забезпечує високий рівень продуктивності, що робить його одним з перших виборів для багатьох розробників.

Flutter вирізняється своїм унікальним підходом до розробки, пропонуючи використання мови програмування Dart як для клієнтської, так і для серверної частини застосунку. Ця єдність мови програмування на обох етапах розробки робить процес значно простішим, оскільки відпадає необхідність переходити між різними технологіями та мовами. Крім того, це дає розробникам можливість

краще ознайомитися з особливостями Dart, розвиваючи свої професійні навички та прискорюючи процес розвитку проектів.

Flutter справляє враження своєю багатофункціональністю, пропонуючи широкий спектр готових віджетів, які спрощують процес розробки. Більш того, завдяки своїй багат шаровій архітектурі, Flutter забезпечує систематизацію коду, роблячи його більш структурованим, зрозумілим і легко підтримуваним. Ціла концепція Flutter базується на ідеї, що кожен елемент є віджетом, що призводить до формування додатку у вигляді дерева віджетів, що спрощує його управління та розширення. [15].

Важливими елементами додатку можна вважати:

- Головна сторінка додатку;
- Сторінка внесення витрати чи надходження коштів;
- Сторінка статистики витрат та надходжень;
- Сторінка прогнозованих витрат та надходжень;
- Сторінка налаштувань.

Завантаживши додаток, користувач потрапляє на головну сторінку, що містить інформацію про загальний баланс, надходження та витрати користувача. Також на головній сторінці відображено історію транзакцій, створених на сторінці внесення витрат чи надходжень користувача (див. рисунок 3.4). Нижня панель головної сторінки створена для навігації сторінками додатку, вона наявна на кожній зі сторінок та включає в себе можливість навігації між сторінками додатку та можливість внесення нових змін балансу користувача – витрат та надходжень фінансового ресурсу вказуючи додаткові дані для подальшої обробки. Частина коду головної сторінки додатку зображена на рисунку 3.5. Головна сторінка перебдачена як стартова сторінка додатку з відображенням поточних даних та фінансового стану користувача. Весь код додатку вказаний в додатку В.

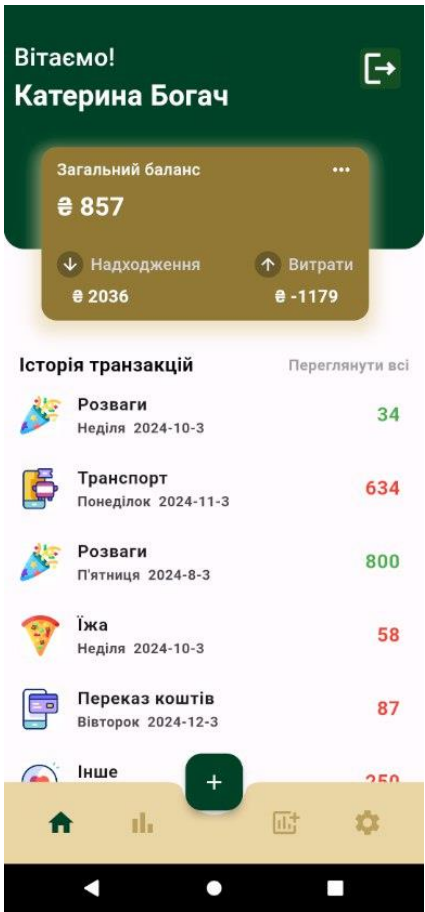


Рисунок 3.4 – Головна сторінка додатку

```
27 @override
28 widget build(BuildContext context) {
29   return Scaffold(
30     body: SafeArea(
31       child: ValueListenableBuilder(
32         valueListenable: box.listenable(),
33         builder: (context, value, child) {
34           return CustomScrollView(
35             slivers: [
36               SliverToBoxAdapter(
37                 child: SizedBox(height: 340, child: _head()),
38               ), // SliverToBoxAdapter
39               SliverToBoxAdapter(
40                 child: Padding(
41                   padding: const EdgeInsets.symmetric(horizontal: 15),
42                   child: Row(
43                     mainAxisAlignment: MainAxisAlignment.spaceBetween,
44                     children: [
45                       Text(
46                         'Історія транзакцій',
47                         style: TextStyle(
48                           fontWeight: FontWeight.w600,
49                           fontSize: 19,
50                           color: Colors.black,
51                         ), // TextStyle
52                       ), // Text
53                       Text(
54                         'Переглянути всі',
55                         style: TextStyle(
56                           fontWeight: FontWeight.w600,
57                           fontSize: 15,
58                           color: Colors.grey,
59                         ), // TextStyle
60                       ), // Text
61                     ], // Row
62                   ), // Padding
63               ), // SliverToBoxAdapter
64               SliverList(
65                 delegate: SliverChildBuilderDelegate(
66                   (context, index) {
67                     history = box.values.toList()[index];
68                     return getList(history, index);
69                   },
70                   childCount: box.length,
71                 ), // SliverChildBuilderDelegate
72             ), // SliverList
73           ], // CustomScrollView
74         ), // ValueListenableBuilder // SafeArea
75       ), // Scaffold
76     );
77   }
78 }
```

Рисунок 3.5 – Частина коду головної сторінки додатку

Перехід на сторінку внесення змін балансу здійснюється натисканням на відповідну центральну кнопку на нижній панелі з підписом «+». Після переходу

на сторінку внесення нового запису про зміни балансу користувач має ввести дані про тематику витрати чи надходження коштів, опис запису, вказати суму коштів та тип запису – була попередньо вказана сума коштів витрачена чи надійшла на рахунок. Вказання тематики передбачене випадаючим списком розповсюджених категорій, для більшої точності є можливість ввести додатковий опис. Після заповнення вище описаних полів, користувач вказує дату, коли саме відбулась фінансова операція. Можливість вказання дати самостійно дає можливість внесення даних навіть про операції що відбулись у минулому чи мають обов’язково виконані в певний час у майбутньому, що допоможе підвищити точність майбутнього прогнозування. Вказавши дату проведення нарахування чи списання коштів користувач натискає кнопку «Зберегти», зберігаючи внесені дані. На рисунку 3.6 було зображено сторінку внесення нового запису. Код сторінки внесено в додаток В.

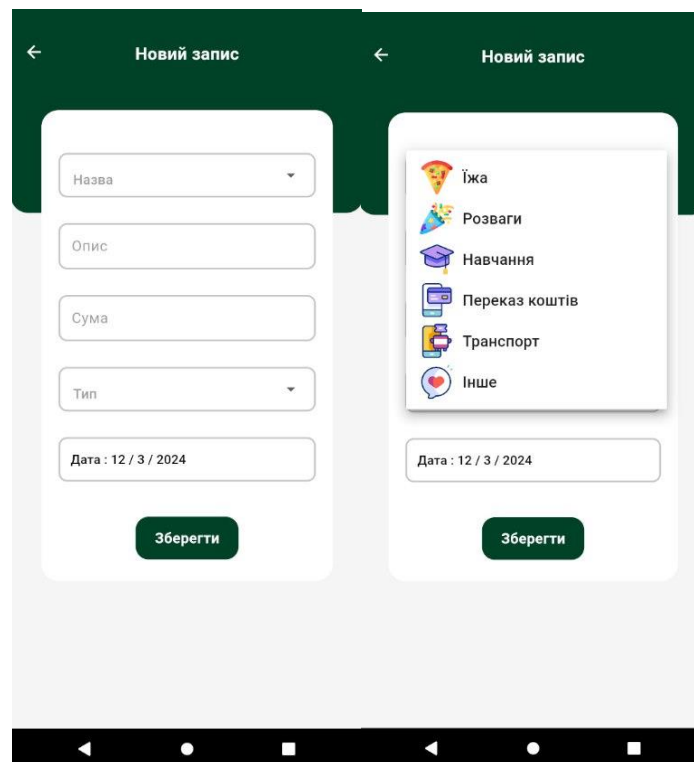


Рисунок 3.6 – Зображення сторінки внесення нового запису про зміну балансу користувача

Використовуючи панель навігації користувач має можливість перейти на сторінку статистики та переглянути актуальну статистику витрат та надходжень коштів за обраний проміжок часу. Користувач має змогу переглянути статистику за поточний день, за тиждень, за місяць та за рік внесення даних. Статистичні дані відображені у вигляді графіку з вертикальним вектором сум коштів, в межах яких відбувались операції та горизонтальним вектором часу, що зображує дні використання. Нижче зображеного графіку винесено список фінансових операцій, що відбулись за обраний для візуалізації у графіку статистики час. На рисунку 3.7 було зображено сторінку статистики за тиждень та за місяць, враховуючи надходження та витрати за відповідний проміжок часу. Частину коду описаної сторінки було зображено на рисунку 3.8. Код сторінки внесено в додаток В.

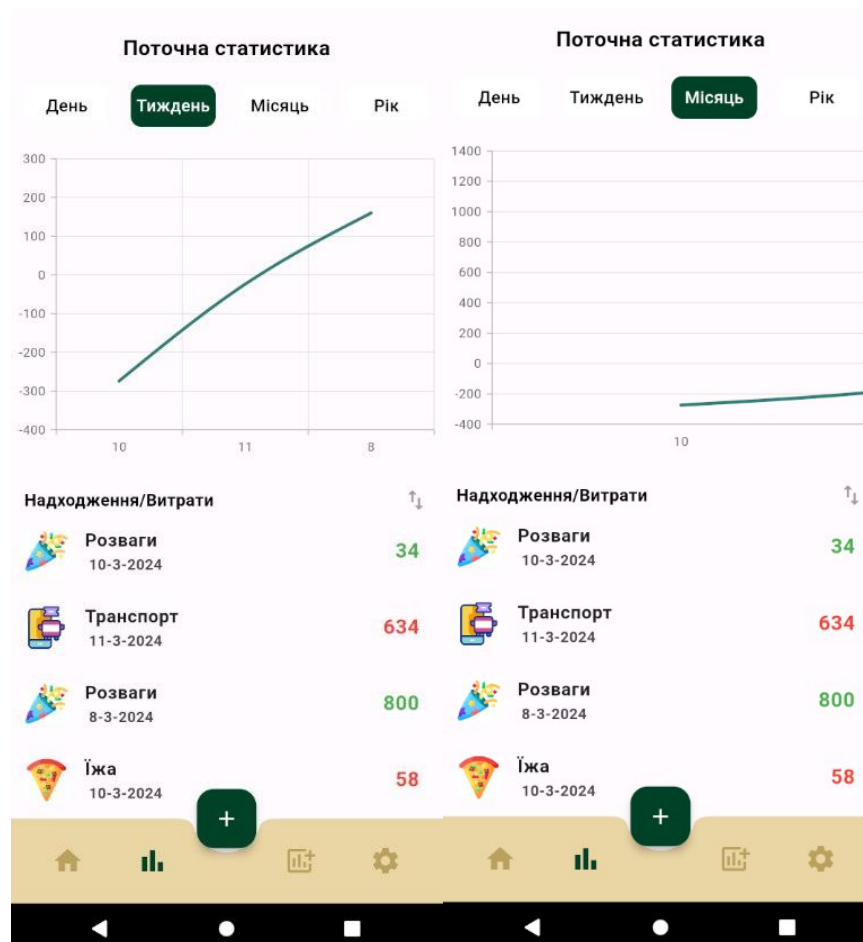


Рисунок 3.7 – Зображення сторінки поточної статистики надходження та витрат коштів

```

39 CustomScrollView custom() {
40   return CustomScrollView(
41     slivers: [
42       SliverToBoxAdapter(
43         child: Column(
44           children: [
45             SizedBox(height: 20),
46             Text(
47               'Поточна статистика',
48               style: TextStyle(
49                 color: Colors.black,
50                 fontSize: 20,
51                 fontWeight: FontWeight.w700,
52               ), // TextStyle
53             ), // Text
54             SizedBox(height: 20),
55             Padding(
56               padding: const EdgeInsets.symmetric(horizontal: 15),
57               child: Row(
58                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
59                 children: [
60                   ...List.generate(
61                     4,
62                     (index) {
63                       return GestureDetector(
64                         onTap: () {
65                           setState(() {
66                             index_color = index;
67                             kj.value = index;
68                           });
69                         }, // BoxDecoration
70                         alignment: Alignment.center,
71                         child: Text(
72                           day[index],
73                           style: TextStyle(
74                             color: index_color == index
75                               ? Colors.white
76                               : Colors.black,
77                             fontSize: 16,
78                             fontWeight: FontWeight.w500,
79                           ), // TextStyle
80                         ), // Text
81                       ), // Container
82                     ); // GestureDetector
83                   ), // List.generate
84                 ], // Row
85               ), // Padding
86             ), // Chart
87             SizedBox(height: 20),
88             Padding(
89               padding: const EdgeInsets.symmetric(horizontal: 15),
90               child: Row(
91                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
92                 children: [
93                   Text(

```

Рисунок 3.8 – Зображення частини коду сторіки поточної статистики надходження та витрат коштів

Крім поточної статистики користувач має змогу перейти на сторінку прогнозованої статистики та переглянути прогнозовану статистику, яка є результатом використання обраного та описаного вище методу. Сторінка відображає графік прогнозованих витрат за поточний день, за тиждень, за місяць та за рік внесення даних. На рисунку 3.9 було зображено сторінку прогнозованої статистики. Статистичні дані відображені у вигляді графіку з вертикальним вектором сум коштів, в межах яких відбувались операції та горизонтальним вектором часу, що зображує дні використання. Графік відображає час прогнозованих надходжень та витрат згідно до даних, що було вказано протягом минулих проміжків часу. Нижче зображеного графіку користувач може переглянути орієнтовні суми майбутніх надходжень та витрат з урахуванням можливих статистичних похибок. Для отримання розрахунків користувачу потрібно натиснути на кнопку «Провести прогнозування», що розміщена нижче на сторінці. Код сторінки внесено в додаток В.

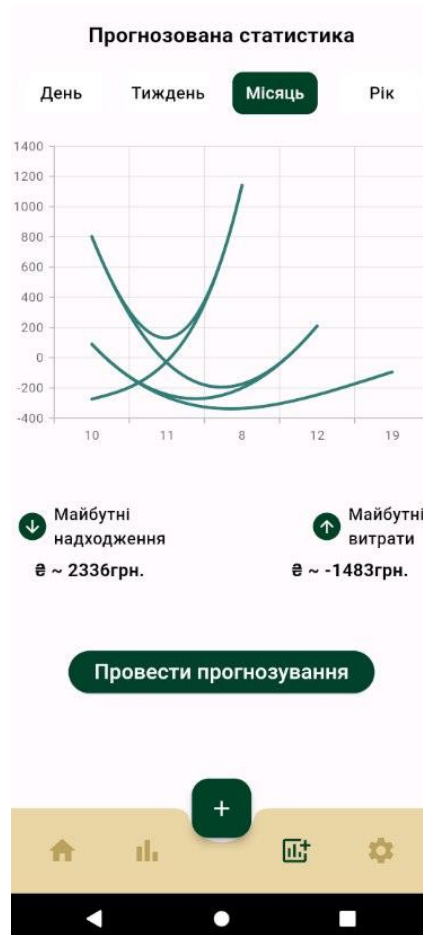


Рисунок 3.9 – Зображення сторінки прогнозованої статистики

Сторінка налаштувань слугує для зміни даних користувача та активації прогнозованої статистики. У випадку якщо користувач не бажає бачити прогнозовані витрати на надходження є можливість відключити відображення прогнозування. Для збереження внесених змін є відповідна кнопка. Сторінку налаштування було зображено на рисунку 3.10 а також код описаної сторінки винесено у додаток В.

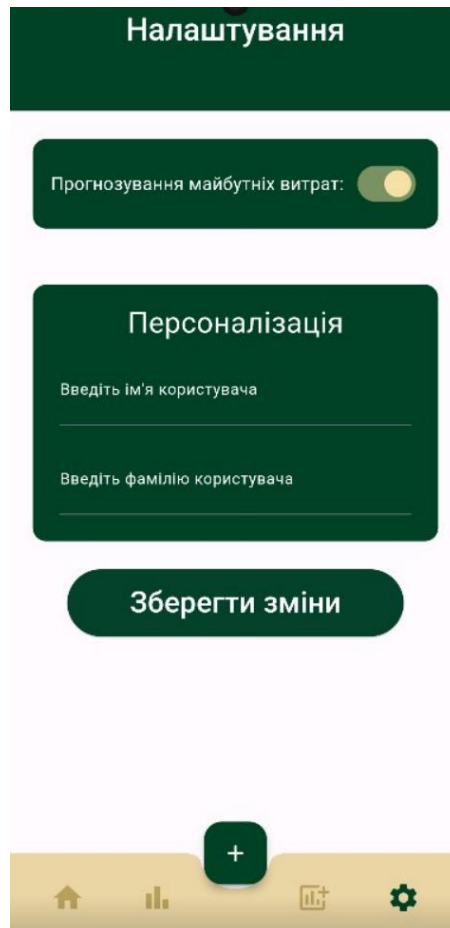


Рисунок 3.10 – Зображення сторінки налаштувань

3.4 Реалізація методу прогнозування експоненційним згладжуванням

Експоненційне згладжування – це метод аналізу часових рядів, який використовується для прогнозування майбутніх значень на основі попередніх даних. Цей метод припускає, що майбутні значення залежать від попередніх значень ряду, але вага кожного попереднього значення зменшується експоненційно, залежно від його віддаленості в часі.

Цей метод математичного перетворення зазвичай застосовується при прогнозуванні часових рядів. Свою назву він отримав через те, що при кожній наступній ітерації враховуються всі попередні значення ряду, але ступінь врахування зменшується за експоненційним законом. Нижче зображено формулу експоненційного згладжування:

$$F_{t+1} = \alpha * Y_t + (1 - \alpha) * F_t$$

Для прогнозування майбутніх витрат та надходжень коштів метод експоненційного згладжування застосований наступним чином. Спочатку, користувач вносить дані витрат та надходжень коштів вказуючи дату фінансової операції, що є історичними даними про витрати та надходження коштів за певний проміжок часу для проведення прогнозування за день, тиждень, місяць та рік. Після отримання даних від користувача визначається параметр згладжування, який вказує на те, наскільки швидко вага попередніх значень зменшується з часом.

Після цього, використовуючи формулу експоненційного згладжування, проводяться розрахунки прогнозованих значень обраних періодів часу. Ця формула враховує поточне спостереження та попередні прогнози, з кожним новим періодом оновлюючи прогноз на основі актуальних даних.

Отримані таким чином прогнози використовуються для користувача та використовуються для планування бюджету, управління фінансами та прийняття рішень щодо інвестицій чи витрат. Частина реалізації обраного методу мовою Dart було наведено на малюнку 3.12 до імплементації у додаток.

```
import 'dart:math';

// Метод експоненційного згладжування
List<double> exponentialSmoothing(List<double> data, double alpha) {
  List<double> smoothedData = [];
  smoothedData.add(data.first); // Початкове значення зберігається без згладжування

  for (int i = 1; i < data.length; i++) {
    double newValue = alpha * data[i] + (1 - alpha) * smoothedData[i - 1];
    smoothedData.add(newValue);
  }

  return smoothedData;
}

void main() {
  List<double> expenses = [100.0, 110.0, 120.0, 115.0, 105.0, 125.0, 130.0];
  double alpha = 0.2; // Параметр згладжування

  List<double> smoothedExpenses = exponentialSmoothing(expenses, alpha);

  print('Витрати: $expenses');
  print('Експоненційне згладжування: $smoothedExpenses');
}
```

Рисунок 3.12 – Зображення методу експоненційного згладжування

У наведеному коді, дані про витрати передаються у вигляді списку `expenses`, а параметр згладжування `alpha` встановлюється на значення 0.2. Функція `exponentialSmoothing` обчислює нові значення за допомогою методу експоненційного згладжування, а результати виводяться на екрані. Реалізований код використаного методу зображено у додатку В.

3.5 Висновки

У третьому розділі дипломної кваліфікаційної роботи, було проведено варіантний аналіз який показав, що вибір технологій і засобів реалізації програмного продукту є ключовим для досягнення високої продуктивності, масштабованості та зручності використання. Серед розглянутих варіантів були оцінені різні фреймворки, мови програмування та платформи, які відповідають вимогам проєкту. Обґрунтування вибору базувалося на критеріях, таких як сумісність з існуючими системами, продуктивність, легкість підтримки та оновлення, а також наявність активної спільноти підтримки. Як результат, були обрані оптимальні засоби, що забезпечують баланс між потужністю та ефективністю реалізації програмного продукту. Був обраний для реалізації метод прогнозування експоненційним згладжуванням через його здатність ефективно аналізувати тимчасові ряди даних і надавати точні прогнози.

Також було розроблено структури інтерфейсу мобільного додатку які були спрямовані на створення інтуїтивно зрозумілого, зручного та естетично привабливого дизайну. Була проведена глибока аналітика потреб користувачів, що дозволило визначити ключові елементи та функціональні зони додатку. Основна увага приділялась забезпеченню простоти навігації та доступності всіх функцій додатку.

4 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАСОБУ

4.1 Аналіз методів тестування

Програмні системи стали невід’ємною частиною нашого повсякденного життя. Більшість людей стикалися з програмним забезпеченням, яке не працювало так, як очікувалося. Неналежна робота програм може призвести до значних проблем, включаючи втрату грошей, часу, ділової репутації, а в крайніх випадках навіть до травм або смерті.

Тестування програмного забезпечення оцінює його якість і допомагає знизити ризик збоїв під час експлуатації. Це комплекс заходів для виявлення дефектів і оцінки якості програмних компонентів, відомих під час тестування як тестові об’єкти. Поширена помилка полягає в тому, що тестування складається лише з виконання тестів (тобто запуску програмного забезпечення та перевірки результатів). Однак тестування також включає інші види діяльності і має бути узгоджене з життєвим циклом розробки програмного забезпечення.

Інше поширене непорозуміння щодо тестування полягає в тому, що воно повністю зосереджене на перевірці тестового об’єкта. Тестування передбачає не лише верифікацію, тобто перевірку відповідності системи заданим вимогам, але й валідацію, тобто перевірку відповідності системи потребам користувачів та інших зацікавлених сторін у реальному робочому середовищі.

Тестування може бути динамічним і статичним. Динамічне тестування передбачає виконання програмного забезпечення, тоді як статичне тестування – ні. Статичне тестування включає огляди і статичний аналіз. Динамічне тестування використовує різні методики та підходи для створення тестових випадків. Тестування – це не лише технічна діяльність. Воно потребує ретельного планування, управління, оцінки, контролю та моніторингу [17].

Методи тестування допомагають тестувальнику аналізувати, що слід тестувати, і розробляти, як це тестувати. Вони дозволяють систематично створювати відносно невеликий, але достатній набір тестів. Крім того, методи

тестування допомагають визначати умови тестування, ідентифікувати елементи покриття та обирати тестові дані під час аналізу та розробки тестів.

Методи тестування чорної скриньки (також відомі як методи на основі специфікацій) ґрунтуються на аналізі заданої поведінки тестового об'єкта без врахування його внутрішньої структури. Тестові випадки, таким чином, не залежать від реалізації програмного забезпечення, тому залишаються корисними навіть при зміні реалізації, якщо поведінка системи залишається незмінною.

Методи тестування білої скриньки (також відомі як методи на основі структури) базуються на аналізі внутрішньої структури та обробки тестового об'єкта. Оскільки тестові випадки залежать від реалізації програмного забезпечення, вони можуть бути створені лише після проектування або реалізації об'єкта тестування.

Методи тестування на основі досвіду використовують знання та досвід тестувальників для розробки та виконання тестових випадків. Ефективність цих методів значною мірою залежить від навичок тестувальника. Вони можуть виявити дефекти, які залишаються непоміченими за допомогою методів чорної та білої скриньки, тому методи на основі досвіду є важливим доповненням до інших методів тестування [17].

На основі вищезазначеної інформації можна зробити висновок, що для повного покриття тестами розробленого додатку необхідно застосувати різноманітні техніки тест дизайну. Це дозволить забезпечити всебічне тестування програмного забезпечення та виявити потенційні дефекти на різних рівнях його функціонування. Техніки тест дизайну – це методи, що допомагають у створенні тестових випадків для перевірки програмного забезпечення на відповідність вимогам. Вони дозволяють тестувальникам систематично й ефективно підходити до тестування, забезпечуючи високу якість продукту [18].

Техніки тестування чорної скриньки базуються на аналізі специфікацій і вимог без врахування внутрішньої структури програми. До них належать [19]:

- Еквівалентне розбиття: розподіл вхідних даних на рівнозначні класи, для кожного з яких передбачається однаковий результат.
- Аналіз граничних значень: тестування меж еквівалентних класів для виявлення помилок, які можуть виникнути на границях діапазонів.
- Таблиця прийняття рішень: використання таблиць для опису логіки прийняття рішень і відповідних результатів на основі комбінацій вхідних умов.
- Тестування стану переходів: перевірка системи на основі можливих станів і переходів між ними відповідно до діаграми станів.
- Тестування на основі сценаріїв використання (Use Case Testing): тестування шляхом моделювання реальних сценаріїв використання програмного забезпечення користувачами.

Техніки тестування на основі досвіду використовують знання і досвід тестувальників для створення тестових випадків [19]:

- Аналіз помилок (Error Guessing): створення тестових випадків на основі передбачуваних помилок, які можуть виникнути в системі.
- Тестування на основі чек-листів (Checklist-based Testing): використання попередньо підготовлених списків можливих дефектів і типових помилок.
- Адаптивне тестування (Exploratory Testing): динамічне тестування, де тестувальник досліджує систему без чітко визначених тест-кейсів, щоб виявити нові дефекти.

Використовуючи описані техніки тест дизайну, було прийнято рішення створити тестову документацію, включаючи тест-кейси та чек-листи, щоб повністю покрити додаток тестами. Це дозволить мінімізувати ризик виникнення дефектів під час подальшого використання додатку користувачами, що в свою чергу позитивно вплине на їхнє враження про додаток.

Тестова документація — це набір документів, які описують процес тестування програмного забезпечення та його результати. Вона є критично важливою для забезпечення якості програмного продукту і включає різні типи

документів, що допомагають планувати, проводити, контролювати та оцінювати тестування [19].

Тест-кейс – це окремий документ, який описує конкретний тестовий сценарій для перевірки певного аспекту функціональності програмного забезпечення. Він містить детальну інформацію про умови, кроки, вхідні дані та очікувані результати, необхідні для проведення тестування. Тест-кейси є основним інструментом, що використовується тестувальниками для систематичної перевірки та верифікації роботи програмного продукту. Тест-кейси є важливим інструментом у процесі тестування, забезпечуючи ретельну перевірку програмного забезпечення, мінімізуючи ризик помилок і підвищуючи загальну якість продукту.

Чек-лист (або список перевірок) – це інструмент, який використовується для систематизації і фіксації перевірок під час тестування програмного забезпечення. Він являє собою перелік елементів, критеріїв або дій, які необхідно виконати для забезпечення повного і якісного тестування продукту. Чек-листи є важливим інструментом для організації і контролю тестування програмного забезпечення, забезпечуючи послідовність і повноту перевірок, що сприяє підвищенню загальної якості продукту [19].

4.2 Тестування розробленого програмного продукту

Перед початком самого процесу тестування необхідно створити детальну тестову документацію. Цей крок є критично важливим для забезпечення систематичного і ретельного підходу до перевірки програмного забезпечення. Тестова документація слугує основою для всього процесу тестування, забезпечуючи чіткі інструкції та критерії для оцінки якості продукту.

Щоб заощадити час на створення документації, було вирішено скласти чек-лист, який охоплює весь функціонал додатку. Чек-лист є ефективним інструментом для швидкої та зручної перевірки основних аспектів додатку, гарантуючи, що жодна важлива функція не буде пропущена під час тестування.

Використання чек-листа дозволяє швидко ідентифікувати основні дефекти і забезпечує базовий рівень контролю якості.

Для кращого покриття тестами і мінімізації ризику пропуску багів, на основні функціональні частини додатку було додатково вирішено написати тест-кейси. Тест-кейси забезпечують більш детальне і глибоке тестування, дозволяючи виявити більш складні та приховані дефекти, які могли б бути не помічені при використанні лише чек-листа. Таким чином, комбінація чек-листів і тест-кейсів забезпечує всебічну перевірку програмного забезпечення, що сприяє підвищенню його якості і надійності.

В додатку Г наведено чек-лист для проведення тестування додатку.

Згідно з результатами створеного чек-листа, найбільше перевірок припадає на функцію додатку: додавання нового запису. Це свідчить про те, що ця функція є найбільш критичними для коректної роботи додатку і потребує особливої уваги під час тестування. Забезпечення її бездоганного функціонування є ключовим для загальної стабільності та якості продукту.

Оскільки додавання нового запису є однією з найважливіших функцій нашого додатку, було прийнято рішення забезпечити цей функціонал більш глибоким покриттям тестами шляхом створення детальних тест-кейсів. Тест-кейси дозволять провести ретельну перевірку всіх можливих сценаріїв використання цієї функції, виявити потенційні дефекти та мінімізувати ризики некоректної роботи. Це особливо важливо для функцій, які мають безпосередній вплив на основну діяльність користувачів.

Таким чином, поєднання чек-листів і тест-кейсів для ключових функцій, таких як додавання нового запису, сприятиме значному підвищенню якості тестування. Такий підхід забезпечить всебічну перевірку, що дозволить виявити і усунути більшість можливих проблем на ранніх етапах розробки. В результаті, користувачі отримають більш надійний і стабільний додаток, що позитивно вплине на їхній досвід і задоволення від використання продукту.

Тож у таблиці Г наведено тест-кейси для функції додавання нового запису в додатку.

Описана вище тестова документація була розроблена з використанням технік тест дизайну чорної скриньки, що дозволило оптимізувати кількість тестів, забезпечивши при цьому значне покриття функціональності додатку. Це допомогло ефективно виявляти можливі проблеми та забезпечувати якість без потреби у великій кількості тестів.

З використанням створеної тестової документації було проведено комплексне тестування всіх функцій мобільного додатку. В результаті ретельного тестування жодних дефектів не було виявлено. Це свідчить про високу якість розробленого продукту і підтверджує ефективність використаних методик тестування. Таким чином, додаток працює стабільно і відповідає всім заданим вимогам, що забезпечить позитивний досвід користувачів у подальшому.

4.3 Оцінювання достовірності майбутніх витрат

Провівши порівняння методів прогнозування у розділі 2.1 було проведено оцінювання достовірності майбутніх витрат шляхом проведення порівняння результатів обчислення кожного з методів та обчислення середньої абсолютної помилки. В якості вхідних даних для обрахунків було використано набір тестових даних про витрати за 6 місяців користування, що були зображені в таблиці 4.3.

Таблиця 4.1 – Тестові дані витрат

Місяць	1	2	3	4	5	6
Сума коштів у грн.	12451	14325	13646	13962	15235	12435

Метод середнього передбачає використання середнього значення всіх попередніх спостережень для прогнозування майбутніх значень.

Маючи дані про витрати за 6 місяців можна переходити до вирахування середнього за формулою:

$$\text{Середнє} = \frac{\sum_{i=1}^n A_i}{n},$$

де: A_i - фактичні витрати за i -ий місяць, а n – кількість місяців

Провівши обчислення отримано:

$$\text{Середнє} = \frac{12451 + 14325 + 13646 + 13962 + 15235 + 12435}{6} = 13675.67$$

Прогнозовані витрати методом середнього для кожного місяця становлять 13675.67 гривень.

Для обчислення середньої абсолютної помилки необхідно знайти абсолютні різниці між фактичними значеннями і прогнозованими значеннями, а потім знайти їх середнє.

1. Вирахування абсолютних різниць:

$$|12451 - 13675.67| = 1224.67 \text{ грн}$$

$$|14325 - 13675.67| = 649.33 \text{ грн}$$

$$|13646 - 13675.67| = 29.67 \text{ грн}$$

$$|13962 - 13675.67| = 286.33 \text{ грн}$$

$$|15235 - 13675.67| = 1559.33 \text{ грн}$$

$$|12435 - 13675.67| = 1240.67 \text{ грн}$$

2. Сумма абсолютних різниць:

$$1224.67 + 649.33 + 29.67 + 286.33 + 1559.33 + 1240.67 = 4989 \text{ грн}$$

3. Середня абсолютна похибка:

$$MAE = \frac{1}{6} * 4989 = 831.5$$

Середня абсолютна помилка (MAE) для прогнозу витрат методом середнього становить 831.5 гривень, в середньому прогнозовані витрати відрізняються від фактичних на 831.5 гривень.

Експоненційне згладжування – це метод прогнозування, який дозволяє враховувати минулі дані для оцінювання майбутніх значень, надаючи більшу вагу останнім спостереженням.

Для оцінювання достовірності майбутніх витрат методом експоненційного згладжування було сформовано п'ятикроковий підхід для використання методу:

Першим етапом було проведено збір даних про витрати за попередні періоди.

Другий етап був вибором параметра згладжування(α). Параметр α це параметр згладжування, який знаходиться в діапазоні від 0 до 1. Високе значення α надає більшу вагу останнім спостереженням, тоді як низьке значення α надає більшу вагу минулим даним.

Третій етап це розрахунок початкового прогнозу. Початковий прогноз було взято як перше спостереження (початкове значення).

Четвертий етап це розрахунок прогнозу для кожного періоду: Використовуючи формулу експоненційного згладжування:

$$F_t = \alpha * A_{t-1} + (1 - \alpha) * F_{t-1},$$

де:

F_t - прогнозоване значення для часу t ;

A_{t-1} - фактичне значення для часу $t-1$;

F_{t-1} - прогнозоване значення для часу $t-1$;

Оцінка точності прогнозу була виконана за допомогою виконання обчислення середньої абсолютної помилки (Mean Absolute Error, MAE).

Маючи теоритичні дані, вказані в таблиці 4.3, за півроку використання додатку та показник α можна провести прогнозування витрат.

Оптимальне значення коефіцієнта згладжування α для проведення прогнозування майбутніх витрат методом експоненційного згладжування, маючи набір даних за 6 місяців, дорівнює 1.0. Це значення призвело до мінімальної середньої абсолютної помилки (MAE), яка дорівнює 0.0. Але $\alpha=1.0$ означає, що при згладжуванні ми повністю покладемося на останнє значення даних, ігноруючи попередні значення. Це може бути корисним в умовах, де найсвіжіші дані є найбільш релевантними. У випадку при показнику $\alpha=0.3$, нові дані отримують більшу вагу (30%), ніж старі дані (70%). Це дозволяє моделі швидше реагувати на зміни в даних, одночасно зберігаючи згладжений тренд. Зважаючи на це було обрано показник $\alpha=0.3$, як оптимальне рішення.

Початковий прогноз для першого місяця прогноз дорівнює фактичним витратам: $F_t = 12451$ грн.

Прогноз для другого місяця:

$$F_2 = \alpha * A_1 * (1 - \alpha) * F_1 = 0.3 * 12451 + 0.7 * 12451 = 12451 \text{грн}$$

Прогноз для третього місяця:

$$F_3 = \alpha * A_2 * (1 - \alpha) * F_2 = 0.3 * 14325 + 0.7 * 12451 = 13013.20 \text{грн}$$

Прогноз для четвертого місяця:

$$F_4 = \alpha * A_3 * (1 - \alpha) * F_3 = 0.3 * 13646 + 0.7 * 13013.2 = 13148.44 \text{грн}$$

Прогноз для п'ятого місяця:

$$F_5 = \alpha * A_4 * (1 - \alpha) * F_4 = 0.3 * 13962 + 0.7 * 13148.44 = 13381.71 \text{грн}$$

Прогноз для шостого місяця:

$$F_6 = \alpha * A_5 * (1 - \alpha) * F_5 = 0.3 * 15235 + 0.7 * 13381.71 = 13937.70 \text{грн}$$

Прогноз для сьомого місяця:

$$F_7 = \alpha * A_6 * (1 - \alpha) * F_6 = 0.3 * 12435 + 0.7 * 13937.70 = 13486.89 \text{грн}$$

Оцінка точності прогнозу була виконана за допомогою виконання обчислення середньої абсолютної помилки (Mean Absolute Error, MAE) за формулою:

$$MAE = \frac{1}{n} \sum_{t=1}^n |A_t - F_t|,$$

де:

n - кількість спостережень;

A_t - фактичне значення у момент часу t;

F_t - прогнозоване значення у момент часу t.

Обчисливши абсолютні помилки було отримано результати:

$$|A_2 - F_2| = |14325 - 12451| = 1874$$

$$|A_3 - F_3| = |13646 - 13013.2| = 632.8$$

$$|A_4 - F_4| = |13962 - 13148.44| = 813.56$$

$$|A_5 - F_5| = |15235 - 13381.71| = 1853.29$$

$$|A_6 - F_6| = |12435 - 13937.70| = -1502.7$$

Вирахувавши абсолютні помилки за певні проміжки часу було вирахувано середню абсолютну помилку:

$$MAE = \frac{1}{5} (1874 + 632.8 + 813.56 + 1853.29 + (-1502.7)) = \frac{1}{5} * 3670.95 = 734.19$$

Середня абсолютна помилка для цього випадку складає 734.19 гривень, в середньому прогноз помиляється на 734.19 гривень від фактичних витрат.

Метод простого рухомого середнього (Simple Moving Average, SMA) полягає у взятті середнього значення за певний період часу. Для підрахунку трьохмісячного періоду було використано середнє значення трьох останніх місяців. Було використано три місяці для обчислення простого рухомого середнього.

Для здійснення оцінки прогнозованих витрат було обчислено прогнозовані значення та середню абсолютну помилку.

1. Обчислені прогнозовані значення (SMA):

- Для місяця 4: середнє за перші три місяці (12451, 14325, 13646).
- Для місяця 5: середнє за другі три місяці (14325, 13646, 13962).
- Для місяця 6: середнє за треті три місяці (13646, 13962, 15235).

2. Обчислено середню абсолютну помилку (MAE):

Прогнозовані значення:

- Прогноз на 4 місяць: $\frac{(12451+14325+13646)}{3} = 13474$ грн
- Прогноз на 5 місяць: $\frac{(14325+13646+13962)}{3} = 13977.66$ грн
- Прогноз на 6 місяць: $\frac{(13646+13962+15235)}{3} = 14281$ грн

Обчисливши абсолютні помилки було отримано результати:

$$|A_4 - F_4| = |13962 - 13474| = 488$$

$$|A_5 - F_5| = |15235 - 13977.66| = 1257.34$$

$$|A_6 - F_6| = |12435 - 14281| = 1846$$

Обчислено середню абсолютну помилку за формулою:

$$MAE = \frac{1}{n} \sum_{t=1}^n |A_t - F_t|$$

$$MAE = \frac{1}{3}(488 + 1257.34 + 1846) = \frac{1}{3} * 3591.34 = 1197.11 \text{грн}$$

Середня абсолютна помилка для цього випадку складає 1197.11 гривень, в середньому прогноз помиляється на 1197.11 гривень від фактичних витрат.

Результатом порівняння методів прогнозування є оцінка достовірності майбутніх витрат. Шляхом проведення порівняння результатів обчислення майбутніх витрат методами середнього, простого рухомого середнього та експоненційного згладжування та обчислення середньої абсолютної помилки було підтверджено, що використання методу експоненційного згладжування є оптимальним з обраних до порівняння методів.

4.4 Інструкція користувача

Відкривши мобільний додаток, користувач потрапляє на головну сторінку, де зібрана вся необхідна інформація. У верхньому лівому куті відображається прізвище та ім'я користувача, що допомагає ідентифікувати активний профіль.

Нижче розміщена "Карта користувача", яка містить дані про загальну кількість коштів. Цей баланс розраховується на основі суми надходжень і витрат: надходження додаються до балансу, а витрати віднімаються. Під загальним балансом розташовані сума надходжень та сума витрат, що дозволяє швидко оцінити фінансовий стан.

Під "Картою користувача" знаходиться історія транзакцій, що містить детальну інформацію про кожну фінансову операцію. Кожна транзакція супроводжується іконкою, яка асоціюється з певним типом витрат, і назвою цього типу (одна з шести категорій: Їжа, Розваги, Навчання, Переказ коштів, Транспорт та Інше). Нижче назви типу транзакції зазначається дата її здійснення. Великі цифри відображають суму транзакції, яка може бути зеленого кольору (для надходжень) або червоного (для витрат), що допомагає швидко орієнтуватися у типі транзакції.

Головна сторінка додатку дозволяє користувачу легко переглядати та аналізувати свою фінансову активність. Прокручуючи список історії транзакцій, можна швидко знайти потрібну операцію, використовуючи іконку та назву типу витрати для ідентифікації категорії. Дата та сума транзакції допомагають зрозуміти, коли і скільки коштів було витрачено або отримано. Загальний баланс, сума надходжень та витрат забезпечують повний контроль над фінансами, допомагаючи користувачеві ефективно планувати бюджет і управляти своїми грошима.

Для створення нового запису в додатку, користувач повинен перейти на відповідну сторінку, яка представляє собою зручну форму для введення необхідних даних. Перше поле у формі під назвою "Назва" є випаданим списком, що містить перелік категорій: їжа, розваги, навчання, переказ коштів, транспорт та інше. Користувач має можливість обрати лише одну з цих категорій, що дозволяє точно класифікувати транзакцію.

Наступним є поле для опису транзакції, яке допоможе користувачу в майбутньому зрозуміти, на що конкретно були витрачені кошти або за що вони надійшли. Це поле є важливим для структурування витрат та надходжень, забезпечуючи кращий контроль над фінансами. Після цього користувач бачить поле для введення суми транзакції. У це поле можна вписувати лише числові значення, що відповідають конкретним сумах коштів, тим самим забезпечуючи точність введених даних.

Далі знаходиться випаданий список для вибору типу транзакції, що містить два варіанти: Надходження та Витрата. Це поле є ключовим, оскільки від типу транзакції залежить, чи буде введена сума додана до загального балансу або віднята з нього. Останнє поле у формі призначене для вибору дати. Користувач може обрати дату з календаря, яка може бути не лише поточною, але й будь-якою попередньою або майбутньою датою, що дозволяє вводити як минулі дані, так і плановані витрати.

Після заповнення всіх полів, користувач натискає кнопку "Зберегти". Після цього відбувається розрахунок, а користувач автоматично повертається на

головну сторінку додатку. Нова транзакція додається до списку транзакцій, а суми на карті користувача оновлюються відповідно до внесених даних, відображаючи актуальний баланс з урахуванням нових надходжень або витрат.

Для користування функцією "Поточна статистика" у додатку, користувачеві необхідно виконати наступні кроки. Відкривши додаток, користувач переходить на сторінку "Поточна статистика". На цій сторінці відображається візуалізація транзакцій у вигляді графіка. Користувач має можливість відфільтрувати витрати за чотирма різними фільтрами: день, тиждень, місяць та рік. Для вибору потрібного фільтра користувачу слід натиснути на відповідну кнопку на екрані.

Після вибору фільтра, графік витрат оновиться відповідно до обраного періоду. Це дозволяє користувачу побачити, в які саме дні, тижні чи місяці було найбільше витрат, а де, навпаки, було більше надходжень. Така візуалізація допомагає краще зрозуміти фінансові звички користувача та аналізувати, коли витрати були найбільшими, що може бути корисним для планування бюджету на майбутнє.

Під графіком розташований список витрат за обраний період. Цей список надає швидкий доступ до детальної інформації про кожну транзакцію, здійснену протягом вибраного часу. Користувач може переглядати цей список, щоб детальніше ознайомитися з конкретними витратами і надходженнями, що сприяє кращому розумінню фінансової ситуації користувача.

Для користування функцією "Прогнозована статистика" у додатку, користувачу необхідно виконати наступні кроки. На сторінці "Прогнозована статистика" користувач має можливість вирахувати свої потенційні витрати на наступний день, тиждень, місяць та рік. Для точнішого прогнозування необхідно мати принаймні кілька місяців попередніх записів із надходженнями та витратами. Хоча для початку прогнозування достатньо мати щонайменше два попередніх місяці заповнених інформацією, більш тривалі записи дозволяють отримати точніший прогноз.

Користувач може обрати фільтр для періоду, на який він хоче зробити прогноз (день, тиждень, місяць, рік). Однак, варто зазначити, що функція працюватиме коректно лише за наявності двох попередніх заповнених періодів. Після вибору необхідного терміну, користувачу потрібно натиснути кнопку "Провести прогнозування". Через кілька секунд сторінка "Прогнозована статистика" оновиться, і графік прогнозованих витрат та надходжень завантажиться.

Нижче графіку будуть зазначені окремо суми майбутніх надходжень та майбутніх витрат. Це дозволяє користувачу швидко оцінити свій фінансовий стан у вибраний період і приймати обґрунтовані рішення щодо витрат і заощаджень. Завдяки цій функції, користувач може ефективніше планувати бюджет, запобігати непередбаченим витратам та краще управляти своїми фінансами в майбутньому.

Для користування функцією на сторінці "Налаштування" у додатку, користувачу необхідно виконати наступні кроки. Відкривши додаток, перейдіть на сторінку "Налаштування". На цій сторінці користувач має можливість керувати різними параметрами свого облікового запису та функціональністю додатку.

Спершу, користувач може відімкнути функцію прогнозування, якщо вона йому не потрібна. Для цього потрібно знайти відповідний перемикач і натиснути його. Відімкнення цієї функції може бути корисним, якщо користувач не бажає отримувати прогнозовані дані про свої витрати та надходження.

Далі користувач може змінити свої персональні дані, такі як прізвище та ім'я. Для цього необхідно ввести нові значення в відповідні поля. Важливо зазначити, що ці поля мають валідацію: мінімальна кількість символів для імені та прізвища становить 2 символи, а максимальна — 30 символів. Після внесення змін, користувач повинен натиснути кнопку "Зберегти", щоб застосувати нові дані. Зміни набудуть чинності одразу, і користувач побачить оновлену інформацію на відповідних сторінках додатку. Це дозволяє легко і швидко

оновлювати персональні дані, забезпечуючи актуальність і точність інформації у профілі користувача.

4.5 Висновки

У четвертому розділі було досліджено теоретичні аспекти тестування програмного забезпечення. Було обґрунтовано важливість тестування для забезпечення надійності та безпеки програмного забезпечення, а також про різні методи та підходи до тестування, такі як тест-дизайн та техніки тестування.

Було створено тест-кейси та чек-лист, що дозволяє систематизувати процес тестування та забезпечити наближення до вичерпності перевірок функціональності програми. Ці документи допомагають ефективно перевірити різні аспекти програмного забезпечення, виявити можливі дефекти та забезпечити високу якість продукту.

Було написано детальну інструкцію для користувача. Чіткі та зрозумілі інструкції допомагають користувачам ефективно використовувати програму, зменшують ймовірність помилок та покращують загальне враження від взаємодії з додатком. Таким чином, створення документації та інструкцій для користувачів є важливим етапом у забезпеченні успішного впровадження та використання програмного забезпечення.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Проведення комерційного аудиту науково-технічної розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Результатом магістерської кваліфікаційної роботи «Інформаційна система моніторингу особистого бюджету з прогнозуванням майбутніх доходів і витрат» є програмний продукт у вигляді мобільного додатку для керування фінансовими витратами та надходженнями користувача.

Для проведення технологічного аудиту залучено трьох незалежних експертів: Чорноволик Галина Олександрівна (к.т.н. доц. кафедри ПЗ ВНТУ), Кравецький Вячеслав Валерійович (к.е.н. доц. кафедри ЕП ВНТУ), Ткаченко Олександр Миколайович (к.т.н. доц. кафедри ПЗ ВНТУ).

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 5.1 [20].

Таблиця 5.1 - Критерії оцінювання комерційного потенціалу розробки
бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 5.1

Кри-тер.	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 5.2.

Таблиця 5.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Ткаченко О.М.	2 – Чорноволик Г.О.	3 – Кравецький В.В.
	Бали, виставлені експертами:		
1	4	3	4
Ринкові переваги (недоліки):			
2	3	2	3
3	4	4	4
4	4	3	4
5	3	4	3
Ринкові перспективи			
6	2	3	3
7	3	3	3
Практична здійсненність			
8	4	4	4
9	4	4	4
10	3	3	4
11	4	4	4
12	3	4	3
Сума балів	СБ ₁ =43	СБ ₂ =43	СБ ₃ =43

Продовження таблиці 5.2

Середньоарифметична сума балів $\overline{СБ}$	43
---	----

За даними таблиці 5.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 5.3.

Таблиця 5.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 43 бали, що відповідає рівню «високий».

В якості аналога для розробки було обрано сервіси CoinKeeper та GoodBudget. Основними недоліками аналогів є блокування частини функціоналу через наявність платного контенту та преміум версій а також відсутність прогнозування майбутніх витрат. Можливість планування витрат та надходжень коштів шляхом прогнозування на основі минулих проміжків часу методом експоненційного згладжування не зустрічалось в порівнюваних аналогах, що вважається провідними у своїй галузі. Деякі функції CoinKeeper можуть бути обмеженими без доступу до Інтернету, що може стати проблемою в умовах обмеженого чи відсутнього інтернет-з'єднання. Також до недоліків можна віднести вартість використання CoinKeeper, що може бути ключовим фактором при виборі сервісу моніторингу витрат, особливо якщо користувач знаходиться в умовах обмеженого бюджету.

У розробці дані проблеми вирішується використанням бібліотек та фреймворків, які мають на меті забезпечити оптимізацію роботи розроблюваного продукту. Також програмний продукт випереджає аналог за такими параметрами як мінімізація інтенсивності використання ресурсів шляхом оптимізації коду та ресурсозберігаючих технік та добре спроектований інтерфейс з фокусом на інтуїтивність та легкість використання. У таблиці 5.4 наведені основні технічні показники аналога і нового програмного продукту.

Таблиця 5.4 - Основні технічні показники аналога і нового програмного продукту

Показники	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	90%	90%	1
Надійність	80%	80%	1
Сумісність	50%	70%	1,4
Супровід	70%	70%	1
Економія ресурсів і часу	50%	80%	1,6
Простота використання	60%	80%	1,3

Нова розробка може використовуватися як з метою простого моніторингу надходжень та витрат шляхом демонстрації статистики а також з метою прогнозування майбутніх витрат та надходжень. Безпосереднім споживачем є будь-яка особа, що має на меті контролювати власний потік фінансів та прагне оптимізувати та автоматизувати управління фінансовими ресурсами через використання цієї системи.

Технічна готовність складає 94%. Була розроблена архітектура програмного продукту, розроблено прототип, який частково покриває необхідний функціонал. На даний момент існує декілька зацікавлених сторін з комерціалізації розробки.

5.2 Розрахунок витрат на здійснення науково-технічної розробки.

Проведемо розрахунок витрат на здійснення науково-технічної розробки. Обчислимо основну заробітну Z_o розробника, за формулою нижче:

$$Z_o = \frac{M}{T_p} \cdot t[\text{грн}],$$

де M – місячний посадовий оклад конкретного розробника, грн;

T_p – число робочих днів в місяці – 24;

t – число днів роботи розробника.

Обчислимо заробітну плату розробника з місячним окладом 17000 грн а також дизайнера з місячним окладом в 15000 грн. Обидва працівники мають оданкову кількість робочих днів у місяці – 24.

$$Z_o = \frac{17000}{24} \cdot 72 = 51000,00(\text{грн}).$$

Результати розрахунків зведемо до таблиці 5.5.

Таблиця 5.5 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Розробник	17000,00	708,33	72	51000,00
Дизайнера	15000,00	625,00	72	45000,00
Всього				96000,00

Обчислимо додаткову заробітну плату розробників Z_d . Розрахуємо її як 10% від основної заробітної плати за формулою нижче:

$$Z_d = Z_o \cdot 0,11 [\text{грн}].$$

$$Z_d = 96000 \cdot 0,11 = 10560(\text{грн}).$$

Згідно діючого законодавства нарахування на заробітну плату (Єдиний соціальний внесок) складають 22% від суми основної та додаткової заробітної плати, як подано формулою нижче:

$$H_z = (Z_{op} + Z_d) \cdot 0,22 \text{ [грн]}.$$

$$H_z = (96000 + 10560) \cdot 0,22 = 98323,2 \text{ (грн)}.$$

Обчислимо амортизацію обладнання, що використовувались для розробки. В спрощеному вигляді амортизаційні відрахування розраховується за формулою наведеною нижче:

$$A = \frac{Ц \cdot T}{12 \cdot T_B} \text{ [грн]},$$

де $Ц$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, грн;

T – фактична тривалість використання, міс;

T_B – термін використання обладнання, приміщень тощо, роки.

Розрахуємо амортизаційні витрати на ноутбук та графічний планшет. Балансова вартість ноутбука становить 16000 грн, а термін його використання – 6 років, а фактична тривалість використання 3 місяці.

Балансова вартість графічного планшета становить 8000 грн, а термін його використання – 6 років, а фактична тривалість використання 3 місяці

$$A = \frac{16000 \cdot 3}{12 \cdot 6} = 666,66 \text{ (грн)}.$$

Величина амортизаційних відрахувань наведена в таблиці 5.6.

Таблиця 5.6 – Величина амортизаційних відрахувань

№	Найменування	Балансова вартість, грн	Термін використання, р	Фактична тривалість використання, міс.	Величина амортизаційних відрахувань, грн
1	Ноутбук	16000	6	3	666,66
2	Графічний планшет	8000	5	3	400
Всього:					1066,66

Обчислимо витрати на силову електроенергію за формулою наведеною нижче:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i},$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 3,45$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,4 \cdot 504,0 \cdot 3,45 \cdot 0,95 / 0,97 = 681,17 \text{ грн.}$$

Під час розробки програмного продукту використовувались лише безкоштовні програмні засоби.

Інші витрати (B_{in}) охоплюють: витрати на Інтернет, управління організацією, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, тощо. Інші витрати приймаємо як 100% від суми основної заробітної плати розробників.

Величину інших витрат розрахуємо за формулою нижче:

$$B_{in} = Z_o \cdot 50\% [\text{грн}].$$

$$B_{in} = 51000 \cdot 0,5 = 25500 \text{ (грн)}.$$

Обчислимо витрати на виконання даної роботи, що є сумою всіх попередніх витрат.

$$B = Z_o + Z_p + Z_d + H_3 + A + B_e + B_{in} [\text{грн}].$$

$$B = 51000 + 10560 + 98323,2 + 1066,66 + 681,17 + 25500 = 187131,03 (\text{грн}).$$

Виконаємо прогнозування загальних витрат ($ЗВ$) на виконання та впровадження результатів виконаної науково-технічної роботи за формулою наведеною нижче:

$$ЗВ = B_{\text{заг}} / \beta \text{ [грн]}$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Якщо розробка знаходиться на стадії впровадження, то $\beta \approx 0,9$.

$$ЗВ = 187131,03/0,9=207923.36 \text{ (грн)}.$$

Витрати на виконання наукової роботи та впровадження її результатів становитиме 207923.36грн.

5.3 Розрахунок економічної ефективності та обґрунтування економічної доцільності комерціалізації науково-технічної розробки.

Узагальнюючим результатом, який може отримати потенційний інвестор від комерціалізації науково-технічної розробки, є щорічне збільшення величини чистого прибутку. Зростання чистого прибутку надає можливість інвестору отримувати додаткові кошти, покращувати фінансові результати діяльності.

Виконання даної наукової роботи та впровадження її результатів складає приблизно до 1 року. Позитивні результати від впровадження розробки очікуються вже в перші місяці після впровадження.

Обчислимо збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

Збільшення чистого прибутку у потенційного інвестора протягом декількох років розраховується за формулою наведеною нижче:

$$\Delta\Pi_i = (\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

де $\Delta\Pi_0$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

N – основний кількісний показник, який визначає величину попиту на аналогічні розробки у році до впровадження результатів нової науково-технічної розробки;

Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 – основний якісний показник, який визначає ціну реалізації існуючої науково-технічної розробки у році до впровадження результатів;

ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту. $\rho = 0,3$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у $\vartheta = 18\%$.

В результаті впровадження наукової розробки покращується якість певного продукту, що дозволяє підвищити ціну його реалізації на 600 грн. Кількість користувачів збільшиться протягом першого року на 600, другого – 950, третього – 1200. Реалізація продукції до впровадження результатів наукової розробки складала 1 користувача, а її ціна становила – 7500 грн.

Розрахуємо показник прибутку впродовж трьох років відносно базового.

Збільшення чистого прибутку $\Delta\Pi_1$ протягом першого року складе:

$$\Delta\Pi_1 = (550 \cdot 29\,500 + (7500 + 550) \cdot 600) \cdot 0,8333 \cdot 0,3 \cdot (1 - 18/100) = 4316102,35 \text{ (грн)}.$$

Обчислимо збільшення чистого прибутку $\Delta\Pi_2$ протягом другого року:

$$\Delta\Pi_2 = (550 \cdot 29\,500 + (7500 + 550) \cdot (600+950)) \cdot 0,8333 \cdot 0,3 \cdot (1 - 18/100) = 5883777,15(\text{грн}).$$

Збільшення чистого прибутку $\Delta\Pi_3$ протягом третього року становитиме:

$$\Delta\Pi_3 = (550 \cdot 29\,500 + (7500 + 550) \cdot (600+950+1200)) \cdot 0,8333 \cdot 0,3 \cdot (1 - 18/100) = 7863997,93(\text{грн}).$$

Отже, розрахунки показують, що комерційний ефект від впровадження розробки виражається у значному збільшенні чистого прибутку.

Далі розраховуємо величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot 3B,$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. $K_{\text{інв}} = 2$.

$$PV = 2 \cdot 207923.36 = 415846,72(\text{грн}).$$

Чистий приведений дохід для інвестора розраховуємо за формулою:

$$\Pi\Pi = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t} [\text{грн}],$$

де $\Delta\Pi_i$ - збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

t – період часу, протягом якого виявляються результати впровадженої роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

$$ПП = \frac{4316102,35}{(1 + 0,1)^1} + \frac{5883777,15}{(1 + 0,1)^2} + \frac{7863997,93}{(1 + 0,1)^3} = 17694693.20(\text{грн}).$$

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$ за формулою:

$$E_{абс} = (ПП - PV), [\text{грн}],$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = 3B$, грн.

$$E_{абс} = 17694693.20 - 415846,72 = 17278846.48(\text{грн}).$$

Так як $E_{абс} > 0$, то результат від проведення наукових досліджень та їх впровадження принесе прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту.

На п'ятому етапі розрахуємо відносну (щорічну) ефективність вкладених у наукову розробку інвестицій E_v . Для цього використаємо формулу:

$$E_v = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1,$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV теперішня вартість інвестицій $PV = 3B$, грн;

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$E_v = \sqrt[3]{1 + \frac{17278846.48}{415846.72}} - 1 = 2,49 \text{ або } 249\%.$$

Розраховану величину E_g порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\min}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладати не вигідно. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування визначається за формулою:

$$\tau = d + f,$$

$$T_{ок} = \frac{1}{2,49} = 0,40 \text{ (роки)},$$

Термін окупності становить 0,40 року ($T_{ок} < 3 \dots 5$ років), що свідчить, що фінансування даної наукової розробки є доцільним.

5.4 Висновки

У п'ятому розділі було виконано оцінювання комерційного потенціалу розробки. Проведено комерційний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки є високим.

Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти, що підтверджує перспективність розробки. Програмний продукт має кращі функціональні показники, а тому є конкурентоспроможним товаром, а існуючі переваги нової розробки дозволять швидко поширити її на ринку.

Згідно із розрахунками всіх статей витрат на виконання науково-технічної роботи загальні витрати на розробку складають 207923.36 грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 17278846.48 грн свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 249%, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 19%.

Це означає потенційну зацікавленість інвесторів у комерціалізації нового програмного продукту.

Отже, розрахунок ефективності вкладених інвестицій та періоду їх окупності показав, що фінансування розробки є доцільним, адже інвестиції буде повернуто в термін до 1 року.

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи було розроблено мобільний додаток для моніторингу витрат особистого бюджету з прогнозуванням майбутніх доходів та витрат. Основною метою магістерської кваліфікаційної роботи є підвищення фінансової грамотності користувачів мобільного додатку, який допоможе їм уникати нерациональних витрат і сприятиме зростанню їх добробуту.

Було проведено аналіз аналогів на ринку моніторингу особистих витрат. На основі результату порівняння було виокремлено як їх переваги, які потрібно було удосконалити, так і недоліки, яких потрібно було уникнути під час розробки програмного продукту. Проаналізовано доступні програмні забезпечення для реалізації мобільного застосунку та обрано найбільш підходящу. Розроблено інтерфейс мобільного додатку.

У процесі створення програмного продукту було розроблено діаграми діяльності та станів на основі яких було запропоновано архітектуру інтегрованої інформаційної системи, яка об'єднує в собі функції моніторингу витрат сімейного бюджету і прогнозування майбутніх доходів та витрат.

Мобільний додаток розроблений використовуючи кросплатформенну мову програмування Dart, використовуючи фреймворк Flutter.

Тестування програмного продукту показало, що метод експоненційного згладжування забезпечує найменшу похибку прогнозування, значення якої склало 734.19 гривень, або 5% від фактичних витрат.

Виконано розрахунки та економічний аналіз в результаті якого було підтверджено доцільність, а також економічну вигідність для інвесторів та економічних установ.

Таким чином, мету магістерської кваліфікаційної роботи досягнуто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 К.С. Богач «Про необхідність ведення особистого бюджету для прогнозування майбутніх витрат» - ВНТУ, 2024. Молодь в науці: дослідження, проблеми, перспективи (МН-2024).
- 2 Goodbudget: Budget & Finance [Електронний ресурс]. – <https://play.google.com/store/apps/details?id=com.dayspringtech.envelopes> (дата звернення 05.05.2024) – Goodbudget: Budget & Finance
- 3 CoinKeeper: spending tracker [Електронний ресурс]. – <https://play.google.com/store/apps/details?id=com.disrapp.coinkeeper.material&hl=uk&gl=US> (дата звернення 05.05.2024) – CoinKeeper: spending tracker
- 4 Moneygraph+ [Електронний ресурс]. – <https://apps.microsoft.com/detail/9nblgggz5815?activetab=pivot%3Aoverviewtab&hl=uk-ua&gl=UA> (дата звернення 05.05.2024) – Moneygraph+
- 5 Моніторинг витрат [Електронний ресурс]. – <https://simeinyi-budzhet.ua/personaltaxes/%D1%84%D1%96%D0%BD%D0%B0%D0%BD%D1%81%D0%BE%D0%B2%D0%B8%D0%B9-%D0%BC%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3-%D1%82%D0%B0-%D1%87%D0%BE%D0%BC%D1%83-%D0%B1%D0%BB%D0%BE%D0%BA%D1%83%D1%8E/> (дата звернення 05.05.2024) – Фінансовий моніторинг та чому блокуються рахунки брокерами, банками та фінансовими установами
- 6 Види моніторингу витрат [Електронний ресурс]. – <https://fingal.com.ua/content/view/204/39/1/17/> (дата звернення 10.05.2024) – КОНТРОЛЬ ВИТРАТ І СТИМУЛЮВАННЯ ЕКОНОМІЇ РЕСУРСІВ
- 7 Метод експоненційного згладжування [Електронний ресурс]. – <https://buklib.net/books/31392/> (дата звернення 10.05.2024) – Сутність методу прогнозування методом експоненціального згладжування

- 8 Метод простого рухомого середнього [Електронний ресурс]. – <http://um.co.ua/10/10-7/10-71168.html> (дата звернення 10.05.2024) – Метод рухомого (змінного) середнього
- 9 Метод експоненційного згладжування [Електронний ресурс]. – <https://otexts.com/fpp3/> (дата звернення 10.05.2024) – Forecasting: Principles and Practice
- 10 Діаграма послідовності [Електронний ресурс]. – <https://dou.ua/forums/topic/40575/> (дата звернення 14.05.2024) – Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти
- 11 Про порівняння іос і андроїд [Електронний ресурс]. – <https://root-nation.com/ua/news-ua/it-news-ua/ua-android-ios-statistika-2022/> (дата звернення 15.05.2024) – Частка ринку Android та iOS: оприлюднено статистику 2022 року
- 12 Крос платформена розробка [Електронний ресурс]. – <https://whileweb.com/uk/blog/najkrashi-movi-programuvannya-dlya-rozrobki-mobilnih-dodatviv/> (дата звернення 15.05.2024) – НАЙКРАЩІ МОВИ ПРОГРАМУВАННЯ ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ
- 13 Що таке дизайн інтерфейсу [Електронний ресурс]. – https://en.wikipedia.org/wiki/User_interface_design (дата звернення 15.05.2024) – User interface design
- 14 Що таке Flutter [Електронний ресурс]. – https://aws.amazon.com/what-is/flutter/?nc1=h_ls (дата звернення 16.05.2024) – What is Flutter?
- 15 Метод експоненційного згладжування [Електронний ресурс]. – <https://buklib.net/books/31392/> (дата звернення 17.05.2024) – Сутність методу прогнозування методом експоненціального згладжування
- 16 ISTQB [Електронний ресурс]. – <https://www.istqb.org/> (дата звернення 18.05.2024) – Welcome to ISTQB
- 17 Що таке техніки тест дизайну [Електронний ресурс]. – <https://www.testlearning.net/en/posts/definition-and-examples-test-design->

[techniques](#) (дата звернення 18.05.2024) – The definition and examples of Test Design Techniques

- 18 Техніки тест дизайну [Електронний ресурс]. – <https://dou.ua/forums/topic/44882/> (дата звернення 18.05.2024) – Техніки тест-дизайну для «чайників»
- 19 Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А. Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
д.т.н., проф. О. Н. Романюк
"12" березня 2024 р.

Технічне завдання
на магістерську кваліфікаційну роботу
«Інформаційна система моніторингу витрат особистого бюджету з
прогнозуванням майбутніх доходів і витрат»
за спеціальністю 121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:

к.т.н., доц. каф. Ткаченко О.М.

Амз

"11" березня 2024 р.

Виконав:

студент гр. ПІ-22мз Богач К.С.

Богач

"11" березня 2024 р.

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Інформаційна система моніторингу витрат особистого бюджету з прогнозуванням майбутніх доходів і витрат».

Галузь застосування – програмне забезпечення використання для моніторингу особистих надходжень та витрат.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ № 81 від 11 березня 2024 року ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Підвищення фінансової грамотності користувачів мобільного додатку, який допоможе їм уникати нераціональних витрат і сприятиме зростанню їх добробуту.

Призначення роботи – розробка методів і засобів підвищення ефективності використання особистого бюджету користувача.

3 Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1 Види моніторингу витрат [Електронний ресурс]. – <https://fingal.com.ua/content/view/204/39/1/17/> (дата звернення 10.05.2024) – КОНТРОЛЬ ВИТРАТ І СТИМУЛЮВАННЯ ЕКОНОМІЇ РЕСУРСІВ

2 Метод експоненційного згладжування [Електронний ресурс]. – <https://buklib.net/books/31392/> (дата звернення 10.05.2024) – Сутність методу прогнозування методом експоненціального згладжування

3 Метод простого рухомого середнього [Електронний ресурс]. – <http://um.co.ua/10/10-7/10-71168.html> (дата звернення 10.05.2024) – Метод рухомого (змінного) середнього

4 Метод експоненційного згладжування [Електронний ресурс]. – <https://otexts.com/fpp3/> (дата звернення 10.05.2024) – Forecasting: Principles and Practice

5 Діаграма послідовності [Електронний ресурс]. – <https://dou.ua/forums/topic/40575/> (дата звернення 14.05.2024) – Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти

4. Технічні вимоги

Отримання результатів прогнозування майбутніх надходжень та витрат у вигляді діаграми, а також окремо прописаної суми

5. Конструктивні вимоги.

Конструкція пристрою повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

6. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

7. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

8. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Обґрунтування вибору методу розробки та постановка задач дослідження	12.03.2024- 28.03.2024
2	Вибір методу прогнозування та розробка архітектури мобільного додатку	29.03.2024- 11.03.2024
3	Розробка модулів програмного засобу	12.03.2024- 12.04.2024
4	Тестування розробленого програмного засобу	13.04.2024- 05.06.2024
5	Економічна частина	26.04.24- 10.05.24

9. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Інформаційна система моніторингу витрат особистого бюджету з прогнозуванням майбутніх доходів і витрат

Тип роботи: магістерська кваліфікаційна робота

Підрозділ : кафедра програмного забезпечення, ФІТКІ, ПІ – 22мз

Unichек	
Оригінальність	
Схожість	97.8%
	2.2%

Науковий керівник: к.т.н., доцент кафедри ПЗ Ткаченко О.М

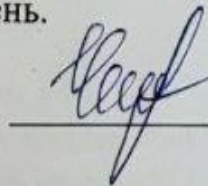
Аналіз звіту подібності

■ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

□ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

□ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку



Черноволик Г. О.

Опис прийнятого рішення: допустити до захисту

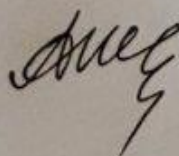
Знайдомлені з повним звітом подібності, який був згенерований системою Unichек

Автор роботи



Богач К.С.

Керівник роботи



Ткаченко О.М

Додаток В Лістинг розробленого додатку додатку

Лістинг модуля main.dart

```
import 'package:flutter/material.dart';
import 'package:CostTracker/Screens/home.dart';
import 'package:CostTracker/Screens/statistics.dart';
import 'package:CostTracker/widgets/bottomnavigationbar.dart';
import 'package:hive_flutter/hive_flutter.dart';

import 'data/model/add_date.dart';

void main() async {
  await Hive.initFlutter();
  Hive.registerAdapter(AdddataAdapter());
  await Hive.openBox<Add_data>('data');
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Bottom(),
    );
  }
}
```

Лістинг модуля програми home.dart

```
import 'package:flutter/material.dart';
import 'package:hive/hive.dart';
import 'package:hive_flutter/adapters.dart';
import 'package:CostTracker/Screens/authorization.dart';
import 'package:CostTracker/data/model/add_date.dart';
import 'package:CostTracker/data/utlity.dart';

class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  var history;
```

```

final box = Hive.box<Add_data>('data');
final List<String> day = [
  'Понеділок',
  'Вівторок',
  'Середа',
  'Четвер',
  "П'ятниця",
  'Субота',
  'Неділя'
];
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: ValueListenableBuilder(
      valueListenable: box.listenable(),
      builder: (context, value, child) {
        return CustomScrollView(
          slivers: [
            SliverToBoxAdapter(
              child: SizedBox(height: 340, child: _head()),
            ),
            SliverToBoxAdapter(
              child: Padding(
                padding: const EdgeInsets.symmetric(horizontal: 15),
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: [
                    Text(
                      'Історія транзакцій',
                      style: TextStyle(
                        fontWeight: FontWeight.w600,
                        fontSize: 19,
                        color: Colors.black,
                      ),
                    ),
                    Text(
                      'Переглянути всі',
                      style: TextStyle(
                        fontWeight: FontWeight.w600,
                        fontSize: 15,
                        color: Colors.grey,
                      ),
                    ),
                  ],
                ),
            ),
          ],
        ),
        SliverList(
          delegate: SliverChildBuilderDelegate(
            (context, index) {

```

```

        history = box.values.toList()[index];
        return getList(history, index);
    },
    childCount: box.length,
  ),
)
],
);
}),
);
}

Widget getList(Add_data history, int index) {
  return Dismissible(
    key: UniqueKey(),
    onDismissed: (direction) {
      history.delete();
    },
    child: get(index, history));
}

ListTile get(int index, Add_data history) {
  return ListTile(
    leading: ClipRRect(
      borderRadius: BorderRadius.circular(5),
      child: Image.asset('images/${history.name}.jpg', height: 40),
    ),
    title: Text(
      history.name,
      style: TextStyle(
        fontSize: 17,
        fontWeight: FontWeight.w600,
      ),
    ),
    subtitle: Text(
      '${day[history.datetime.weekday - 1]} ${history.datetime.year}-'
      '${history.datetime.day}-${history.datetime.month}',
      style: TextStyle(
        fontWeight: FontWeight.w600,
      ),
    ),
    trailing: Text(
      history.amount,
      style: TextStyle(
        fontWeight: FontWeight.w600,
        fontSize: 19,
        color: history.IN == 'Надходження' ? Colors.green : Colors.red,
      ),
    ),
  );
}

```

```

}

Widget _head() {
  return Stack(
    children: [
      Column(
        children: [
          Container(
            width: double.infinity,
            height: 240,
            decoration: BoxDecoration(
              color: Color.fromARGB(255, 0, 66,
                37), //backgroundColor:
Theme.of(context).colorScheme.background,255, 0, 66, 37
              borderRadius: BorderRadius.only(
                bottomLeft: Radius.circular(20),
                bottomRight: Radius.circular(20),
              ),
            ),
          child: Stack(
            children: [
              Positioned(
                top: 35,
                left: 340,
                child: ClipRRect(
                  borderRadius: BorderRadius.circular(7),
                  child: IconButton(
                    onPressed: () {
                      Navigator.push(
                        context,
                        MaterialPageRoute(
                          builder: (context) => const Auth_Screen()));
                    },
                  ),
                icon: Container(
                  decoration: BoxDecoration(
                    color: Color.fromARGB(194, 26, 85, 32),
                    borderRadius: BorderRadius.circular(5),
                  ),
                  child: Icon(
                    Icons.logout_outlined,
                    size: 40,
                    color: Colors.white,
                  ),
                ),
              ),
            ),
          ),
          /*Container(
            height: 40,
            width: 40,
            color: Color.fromRGBO(250, 250, 250, 0.1),
            child: Icon(

```



```

    ),
  ],
  color: Color.fromARGB(255, 146, 121, 53),
  borderRadius: BorderRadius.circular(15),
),
child: Column(
  children: [
    SizedBox(height: 10),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 15),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Text(
            'Загальний баланс',
            style: TextStyle(
              fontWeight: FontWeight.w500,
              fontSize: 16,
              color: Colors.white,
            ),
          ),
          Icon(
            Icons.more_horiz,
            color: Colors.white,
          ),
        ],
      ),
    ),
    SizedBox(height: 7),
    Padding(
      padding: const EdgeInsets.only(left: 15),
      child: Row(
        children: [
          Text(
            '\u20b4 ${total()}',
            style: TextStyle(
              fontWeight: FontWeight.bold,
              fontSize: 25,
              color: Colors.white,
            ),
          ),
        ],
      ),
    ),
    SizedBox(height: 25),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 15),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [

```

```

Row(
  children: [
    CircleAvatar(
      radius: 13,
      backgroundColor: Color.fromARGB(255, 106, 89, 43),
      child: Icon(
        Icons.arrow_downward,
        color: Colors.white,
        size: 19,
      ),
    ),
    SizedBox(width: 7),
    Text(
      'Надходження',
      style: TextStyle(
        fontWeight: FontWeight.w500,
        fontSize: 16,
        color: Color.fromARGB(255, 216, 216, 216),
      ),
    ),
  ],
),
Row(
  children: [
    CircleAvatar(
      radius: 13,
      backgroundColor: Color.fromARGB(255, 106, 89, 43),
      child: Icon(
        Icons.arrow_upward,
        color: Colors.white,
        size: 19,
      ),
    ),
    SizedBox(width: 7),
    Text(
      'Витрати',
      style: TextStyle(
        fontWeight: FontWeight.w500,
        fontSize: 16,
        color: Color.fromARGB(255, 216, 216, 216),
      ),
    ),
  ],
),
],
),
),
SizedBox(height: 6),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 30),

```

```

        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            Text(
              '\₴ ${income()}',
              style: TextStyle(
                fontWeight: FontWeight.w600,
                fontSize: 17,
                color: Colors.white,
              ),
            ),
            Text(
              '\₴ ${expenses()}',
              style: TextStyle(
                fontWeight: FontWeight.w600,
                fontSize: 17,
                color: Colors.white,
              ),
            ),
          ],
        ),
      ],
    ),
  ],
);
}
}

```

Лістинг модуля програми settings_page.dart

```

import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

class SettingsPage extends StatefulWidget {
  const SettingsPage({super.key});

  @override
  State<SettingsPage> createState() => _SettingsPageState();
}

class _SettingsPageState extends State<SettingsPage> {
  bool light = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: PreferredSize(

```

```

preferredSize: Size.fromHeight(100.0),
child: AppBar(
  backgroundColor: Color.fromARGB(255, 0, 66, 37),
  title: Center(
    child: Container(
      child: const Text(
        "Налаштування",
        style: TextStyle(
          fontWeight: FontWeight.w500,
          fontSize: 28,
          color: Color.fromARGB(255, 255, 255, 255),
        ),
      ),
    ),
  ),
),
),
body: Column(
  children: [
    Container(
      decoration: BoxDecoration(
        color: Color.fromARGB(255, 0, 66, 37),
        borderRadius: BorderRadius.circular(12),
      ),
      padding: const EdgeInsets.all(16),
      margin: const EdgeInsets.all(25),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          //second theme
          const Text(
            "Прогнозування майбутніх витрат:",
            style: TextStyle(
              fontWeight: FontWeight.w400,
              fontSize: 16,
              color: Color.fromARGB(255, 255, 255, 255),
            ),
          ),
          Switch(
            // This bool value toggles the switch.
            value: light,
            activeColor: Color.fromARGB(255, 246, 225, 166),
            onChanged: (bool value) {
              // This is called when the user toggles the switch.
              setState(() {
                light = value;
              });
            },
          ),
        ],
      ),
    ],
  ),
),

```

```

    ),
  ),
  Container(
    decoration: BoxDecoration(
      color: Color.fromARGB(255, 0, 66, 37),
      borderRadius: BorderRadius.circular(12),
    ),
    padding: const EdgeInsets.all(16),
    margin: const EdgeInsets.all(25),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        //second theme
        const Text(
          "Персоналізація",
          style: TextStyle(
            fontWeight: FontWeight.w400,
            fontSize: 26,
            color: Color.fromARGB(255, 255, 255, 255),
          ),
        ),
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: TextFormField(
            decoration: const InputDecoration(
              border: UnderlineInputBorder(),
              label: Text(
                'Введіть ім\`я користувача',
                style: TextStyle(color: Colors.white, fontSize: 14),
              ),
            ),
          ),
        ),
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: TextFormField(
            decoration: const InputDecoration(
              border: UnderlineInputBorder(),
              label: Text(
                'Введіть фамілію користувача',
                style: TextStyle(color: Colors.white, fontSize: 14),
              ),
            ),
          ),
        ),
      ],
    ),
  ),
  SizedBox(
    width: 300.0,

```

```

        height: 60.0,
        child: ElevatedButton(
          onPressed: () {},
          child: Text(
            'Зберегти зміни',
            style: TextStyle(
              fontWeight: FontWeight.w500,
              fontSize: 26,
              color: Color.fromARGB(255, 255, 255, 255),
            ),
          ),
          style: ElevatedButton.styleFrom(
            backgroundColor: Color.fromARGB(255, 0, 66, 37)),
        ),
      ],
    ),
  );
}
}

```

Лістинг модуля програми authorization.dart

```

import 'package:flutter/material.dart';

class Auth_Screen extends StatefulWidget {
  const Auth_Screen({super.key});

  @override
  State<Auth_Screen> createState() => _Add_ScreenState();
}

class _Add_ScreenState extends State<Auth_Screen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey.shade100,
      body: SafeArea(
        child: Stack(
          alignment: AlignmentDirectional.center,
          children: [
            background_container(context),
            Positioned(
              top: 120,
              child: main_container(),
            ),
          ],
        ),
      ),
    ),
  ),
}

```

```

    );
}

Container main_container() {
  return Container(
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(20),
      color: Colors.white,
    ),
    height: 550,
    width: 340,
    child: Column(
      children: [
        SizedBox(height: 30),
        explain(),
        SizedBox(height: 30),
        amount(),
        Spacer(),
        save(),
        SizedBox(height: 25),
      ],
    ),
  );
}

GestureDetector save() {
  return GestureDetector(
    onTap: () {},
    child: Container(
      alignment: Alignment.center,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(15),
        color: Color.fromARGB(255, 0, 66, 37),
      ),
      width: 120,
      height: 50,
      child: Text(
        'Bxid',
        style: TextStyle(
          fontFamily: 'f',
          fontWeight: FontWeight.w600,
          color: Colors.white,
          fontSize: 17,
        ),
      ),
    ),
  );
}

```

```

Padding amount() {

```

```

return Padding(
  padding: const EdgeInsets.symmetric(horizontal: 20),
  child: TextField(
    keyboardType: TextInputType.number,
    decoration: InputDecoration(
      contentPadding: EdgeInsets.symmetric(horizontal: 15, vertical: 15),
      labelText: 'Пароль',
      labelStyle: TextStyle(fontSize: 17, color: Colors.grey.shade500),
      enabledBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(10),
        borderSide: BorderSide(width: 2, color: Color(0xffC5C5C5)),
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(10),
        borderSide:
          BorderSide(width: 2, color: Color.fromARGB(255, 0, 66, 37))),
    ),
  ),
);
}

```

```

Padding explain() {
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 20),
    child: TextField(
      decoration: InputDecoration(
        contentPadding: EdgeInsets.symmetric(horizontal: 15, vertical: 15),
        labelText: 'Email',
        labelStyle: TextStyle(fontSize: 17, color: Colors.grey.shade500),
        enabledBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide: BorderSide(width: 2, color: Color(0xffC5C5C5)),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide:
            BorderSide(width: 2, color: Color.fromARGB(255, 0, 66, 37))),
      ),
    ),
  );
}

```

```

Column background_container(BuildContext context) {
  return Column(
    children: [
      Container(
        width: double.infinity,
        height: 240,
        decoration: BoxDecoration(
          color: Color.fromARGB(255, 0, 66, 37),
          borderRadius: BorderRadius.only(
            bottomLeft: Radius.circular(20),

```

```

        bottomRight: Radius.circular(20),
      ),
    ),
    child: Column(
      children: [
        SizedBox(height: 40),
        Padding(
          padding: EdgeInsets.symmetric(horizontal: 15),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              GestureDetector(
                onTap: () {
                  Navigator.of(context).pop();
                },
                child: Icon(Icons.arrow_back, color: Colors.white),
              ),
              Text(
                'Авторизація',
                style: TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.w600,
                  color: Colors.white),
              ),
              Icon(
                Icons.attach_file_outlined,
                color: const Color.fromARGB(255, 0, 66, 37),
              ),
            ],
          ),
        ),
      ],
    ),
  ],
);
}
}

```

Лістинг модуля програми statistics.dart

```

import 'package:flutter/material.dart';
import 'package:hive_flutter/hive_flutter.dart';
import 'package:CostTracker/data/utility.dart';
import 'package:CostTracker/widgets/chart.dart';

import '../data/model/add_date.dart';

```

```

import '../data/top.dart';

class Statistics extends StatefulWidget {
  const Statistics({Key? key}) : super(key: key);

  @override
  State<Statistics> createState() => _StatisticsState();
}

ValueNotifier kj = ValueNotifier(0);

class _StatisticsState extends State<Statistics> {
  List day = ['День', 'Тиждень', 'Місяць', 'Рік'];
  List f = [today(), week(), month(), year()];
  List<Add_data> a = [];
  int index_color = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: ValueListenableBuilder(
          valueListenable: kj,
          builder: (BuildContext context, dynamic value, Widget? child) {
            a = f[value];
            return custom();
          },
        ),
      ),
    );
  }

  CustomScrollView custom() {
    return CustomScrollView(
      slivers: [
        SliverToBoxAdapter(
          child: Column(
            children: [
              SizedBox(height: 20),
              Text(
                'Поточна статистика',
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 20,
                  fontWeight: FontWeight.w700,
                ),
              ),
              SizedBox(height: 20),
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 15),

```



```

        Text(
          'Надходження/Витрати',
          style: TextStyle(
            color: Colors.black,
            fontSize: 16,
            fontWeight: FontWeight.bold),
        ),
        Icon(
          Icons.swap_vert,
          size: 25,
          color: Colors.grey,
        ),
      ],
    ),
  ],
),
SliverList(
  delegate: SliverChildBuilderDelegate(
    (context, index) {
      return ListTile(
        leading: ClipRRect(
          borderRadius: BorderRadius.circular(5),
          child: Image.asset('images/${a[index].name}.jpg', height: 40),
        ),
        title: Text(
          a[index].name,
          style: TextStyle(
            fontSize: 17,
            fontWeight: FontWeight.w600,
          ),
        ),
        subtitle: Text(
          '${a[index].datetime.day}-${a[index].datetime.month}-'
          '${a[index].datetime.year}',
          style: TextStyle(
            fontWeight: FontWeight.w600,
          ),
        ),
        trailing: Text(
          a[index].amount,
          style: TextStyle(
            fontWeight: FontWeight.w600,
            fontSize: 19,
            color:
              a[index].IN == 'Надходження' ? Colors.green : Colors.red,
          ),
        ),
      );
    },
  ),
);

```

```

        },
        childCount: a.length,
      ))
    ],
  );
}
}

```

Лістинг модуля програми future_statistics.dart

```

import 'package:flutter/material.dart';
import 'package:hive_flutter/hive_flutter.dart';
import 'package:CostTracker/data/utility.dart';
import 'package:CostTracker/widgets/chart.dart';

import '../data/model/add_date.dart';
import '../data/top.dart';
import 'dart:math';

class FutureStatistics extends StatefulWidget {
  const FutureStatistics({Key? key}) : super(key: key);

  @override
  State<FutureStatistics> createState() => _StatisticsState();
}

ValueNotifier kj = ValueNotifier(0);

class _StatisticsState extends State<FutureStatistics> {
  List day = ['День', 'Тиждень', 'Місяць', 'Рік'];
  List f = [today(), week(), month(), year()];
  List<Add_data> a = [];
  int index_color = 0;
  int randNum = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: ValueListenableBuilder(
          valueListenable: kj,
          builder: (BuildContext context, dynamic value, Widget? child) {
            a = f[value];
            return custom();
          },
        ),
      ),
    );
  }
}

```

```

}

CustomScrollView custom() {
  return CustomScrollView(
    slivers: [
      SliverToBoxAdapter(
        child: Column(
          children: [
            SizedBox(height: 20),
            Text(
              'Прогнозована статистика',
              style: TextStyle(
                color: Colors.black,
                fontSize: 20,
                fontWeight: FontWeight.w700,
              ),
            ),
            SizedBox(height: 20),
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 15),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  ...List.generate(
                    4,
                    (index) {
                      return GestureDetector(
                        onTap: () {
                          setState(() {
                            index_color = index;
                            kj.value = index;
                          });
                        },
                        child: Container(
                          height: 40,
                          width: 80,
                          decoration: BoxDecoration(
                            borderRadius: BorderRadius.circular(10),
                            color: index_color == index
                              ? Color.fromARGB(255, 0, 66, 37)
                              : Colors.white,
                          ),
                          alignment: Alignment.center,
                          child: Text(
                            day[index],
                            style: TextStyle(
                              color: index_color == index
                                ? Colors.white
                                : Colors.black,
                              fontSize: 16,

```



```

        child: Text(
          'Провести прогнозування',
          style: TextStyle(
            fontWeight: FontWeight.w500,
            fontSize: 20,
            color: Color.fromARGB(255, 255, 255, 255),
          ),
        ),
      style: ElevatedButton.styleFrom(
        backgroundColor: Color.fromARGB(255, 0, 66, 37)),
    ),
  ],
),
],
);
}
}

int getRandomInt(int min, int max) {
  final random = Random();
  return min + random.nextInt(max - min + 1);
}

```

Лістинг модуля програми add.dart

```

import 'package:flutter/material.dart';
import 'package:CostTracker/data/model/add_date.dart';
import 'package:hive_flutter/hive_flutter.dart';

class Add_Screen extends StatefulWidget {
  const Add_Screen({super.key});

  @override
  State<Add_Screen> createState() => _Add_ScreenState();
}

class _Add_ScreenState extends State<Add_Screen> {
  final box = Hive.box<Add_data>('data');
  DateTime date = new DateTime.now();
  String? selctedItem;
  String? selctedItem1;
  final TextEditingController expalin_C = TextEditingController();
  FocusNode ex = FocusNode();
  final TextEditingController amount_c = TextEditingController();
  FocusNode amount_ = FocusNode();
  final List<String> _item = [
    'Їжа',
    "Розваги",
    "Навчання",

```

```

    "Переказ коштів",
    "Транспорт",
    "Інше",
  ];
  final List<String> _itemei = [
    'Надходження',
    "Витрати",
  ];
  @override
  void initState() {
    // TODO: implement initState
    super.initState();
    ex.addListener(() {
      setState(() {});
    });
    amount_.addListener(() {
      setState(() {});
    });
  }

  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey.shade100,
      body: SafeArea(
        child: Stack(
          alignment: AlignmentDirectional.center,
          children: [
            background_container(context),
            Positioned(
              top: 120,
              child: main_container(),
            ),
          ],
        ),
      ),
    );
  }

  Container main_container() {
    return Container(
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        color: Colors.white,
      ),
      height: 550,
      width: 340,
      child: Column(
        children: [
          SizedBox(height: 50),
          name(),
        ],
      ),
    );
  }

```

```

        SizedBox(height: 30),
        explain(),
        SizedBox(height: 30),
        amount(),
        SizedBox(height: 30),
        How(),
        SizedBox(height: 30),
        date_time(),
        Spacer(),
        save(),
        SizedBox(height: 25),
      ],
    ),
  );
}

```

```

GestureDetector save() {
  return GestureDetector(
    onTap: () {
      var add = Add_data(
        selctedItem!, amount_c.text, date, expalin_C.text, selctedItem!);
      box.add(add);
      Navigator.of(context).pop();
    },
    child: Container(
      alignment: Alignment.center,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(15),
        color: Color.fromARGB(255, 0, 66, 37),
      ),
      width: 120,
      height: 50,
      child: Text(
        'Зберігти',
        style: TextStyle(
          fontFamily: 'f',
          fontWeight: FontWeight.w600,
          color: Colors.white,
          fontSize: 17,
        ),
      ),
    ),
  );
}

```

```

Widget date_time() {
  return Container(
    alignment: Alignment.bottomLeft,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),

```

```

        border: Border.all(width: 2, color: Color(0xffC5C5C5)),
width: 300,
child: TextButton(
  onPressed: () async {
    DateTime? newDate = await showDatePicker(
      context: context,
      initialDate: date,
      firstDate: DateTime(2020),
      lastDate: DateTime(2100));
    if (newDate == Null) return;
    setState(() {
      date = newDate!;
    });
  },
  child: Text(
    'Data : ${date.day} / ${date.month} / ${date.year}',
    style: TextStyle(
      fontSize: 15,
      color: Colors.black,
    ),
  ),
),
);
}

```

```

Padding How() {
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 15),
    child: Container(
      padding: EdgeInsets.symmetric(horizontal: 15),
      width: 300,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        border: Border.all(
          width: 2,
          color: Color(0xffC5C5C5),
        ),
      ),
      child: DropdownButton<String>(
        value: selctedItem,
        onChanged: ((value) {
          setState(() {
            selctedItem = value!;
          });
        }),
        items: _itemei
          .map((e) => DropdownMenuItem(
            child: Container(
              alignment: Alignment.center,
              child: Row(

```

```

        children: [
          Text(
            e,
            style: TextStyle(fontSize: 18),
          )
        ],
      ),
    ),
    value: e,
  ))
  .toList(),
selectedItemBuilder: (BuildContext context) => _itemei
  .map((e) => Row(
    children: [Text(e)],
  ))
  .toList(),
hint: Padding(
  padding: const EdgeInsets.only(top: 12),
  child: Text(
    'Тип',
    style: TextStyle(color: Colors.grey),
  ),
),
dropdownColor: Colors.white,
isExpanded: true,
underline: Container(),
),
),
);
}

Padding amount() {
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 20),
    child: TextField(
      keyboardType: TextInputType.number,
      focusNode: amount_,
      controller: amount_c,
      decoration: InputDecoration(
        contentPadding: EdgeInsets.symmetric(horizontal: 15, vertical: 15),
        labelText: 'Сума',
        labelStyle: TextStyle(fontSize: 17, color: Colors.grey.shade500),
        enabledBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide: BorderSide(width: 2, color: Color(0xffC5C5C5)),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide: BorderSide(
            width: 2, color: Color.fromARGB(255, 120, 33, 114))),
      ),
    ),
  ),

```

```

    ),
  );
}

```

```

Padding explain() {
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 20),
    child: TextField(
      focusNode: ex,
      controller: expalin_C,
      decoration: InputDecoration(
        contentPadding: EdgeInsets.symmetric(horizontal: 15, vertical: 15),
        labelText: 'Опис',
        labelStyle: TextStyle(fontSize: 17, color: Colors.grey.shade500),
        enabledBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide: BorderSide(width: 2, color: Color(0xffC5C5C5)),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide:
            BorderSide(width: 2, color: Color.fromARGB(255, 0, 66, 37))),
      ),
    ),
  );
}

```

```

Padding name() {
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 15),
    child: Container(
      padding: EdgeInsets.symmetric(horizontal: 15),
      width: 300,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        border: Border.all(
          width: 2,
          color: Color(0xffC5C5C5),
        ),
      ),
      child: DropdownButton<String>(
        value: selctedItem,
        onChanged: ((value) {
          setState(() {
            selctedItem = value!;
          });
        }),
        items: _item
          .map((e) => DropdownMenuItem(
            child: Container(
              alignment: Alignment.center,

```

```

        child: Row(
          children: [
            Container(
              width: 40,
              child: Image.asset('images/${e}.jpg'),
            ),
            SizedBox(width: 10),
            Text(
              e,
              style: TextStyle(fontSize: 18),
            )
          ],
        ),
      ),
      value: e,
    ))
    .toList(),
  selectedItemBuilder: (BuildContext context) => _item
    .map((e) => Row(
      children: [
        Container(
          width: 42,
          child: Image.asset('images/${e}.jpg'),
        ),
        SizedBox(width: 5),
        Text(e)
      ],
    ))
    .toList(),
  hint: Padding(
    padding: const EdgeInsets.only(top: 12),
    child: Text(
      'Назва',
      style: TextStyle(color: Colors.grey),
    ),
  ),
  dropdownColor: Colors.white,
  isExpanded: true,
  underline: Container(),
),
),
);
}

```

```

Column background_container(BuildContext context) {
  return Column(
    children: [
      Container(
        width: double.infinity,
        height: 240,

```

```

decoration: BoxDecoration(
  color: Color.fromARGB(255, 0, 66, 37),
  borderRadius: BorderRadius.only(
    bottomLeft: Radius.circular(20),
    bottomRight: Radius.circular(20),
  ),
),
child: Column(
  children: [
    SizedBox(height: 40),
    Padding(
      padding: EdgeInsets.symmetric(horizontal: 15),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          GestureDetector(
            onTap: () {
              Navigator.of(context).pop();
            },
            child: Icon(Icons.arrow_back, color: Colors.white),
          ),
          Text(
            'Новий запис',
            style: TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.w600,
              color: Colors.white),
          ),
          Icon(
            Icons.attach_file_outlined,
            color: Color.fromARGB(255, 0, 66, 37),
          ),
        ],
      ),
    ],
  ),
),
],
);
}
}

```

Додаток Г Таблиця із чек-лістом та тест-кейсами

Таблиця Г.1 – Чек-лист тестування мобільного додатку CostTracker+

№	Крок	Очікуваний результат	Статус
1	Відкрити додаток	Головна сторінка відображається	Успішно
2	Натиснути на «+»	Сторінка «Новий запис відображається»	Успішно
2.1	Заповнити всі обов'язкові поля валідними значеннями	Поля заповнюються, помилки не з'являються	Успішно
2.2	Натиснути «Зберегти»	Головна сторінка відображається новий запис з'являється в «Історія транзакцій»	Успішно
2.3	Залишити всі обов'язкові поля пустими	Помилки під обов'язковими полями відображаються. Новий запис не зберігається	Успішно
2.4	Ввести не валідні дані	Помилки під не валідними полями відображаються. Новий запис не зберігається	Успішно
2.5	У поля «Опис» та «Сума» ввести пробіл	Помилка під полями відображається. Новий запис не створюється	Успішно
3	Перейти на сторінку «Поточна статистика»	Сторінка із графіком, періодом та Надходженням, Витратами відображається	Успішно
3.1	Змінити сортування по періоду	Вигляд графіку змінюється, обраний період	Успішно

		підсвічується, список витрат та надходжень змінюється	
4	Перейти на сторінку «Прогнозована статистика»	Сторінка із графіком відображається. Фільтрація по періодах відображається, суми Прогнозованих витрат та надходжень. Кнопка для вирахування	Успішно
4.1	Натиснути на вирахувати	Суми за минулий обраний період вираховуються за формулою. Графік майбутніх витрат та надходжень відображається. Суми майбутніх витрат та надходжень відображаються	Успішно
5	Перейти на сторінку налаштувань	Сторінка налаштувань відображається. Перемикач прогнозування майбутніх витрат, персональні дані користувача та кнопка зберегти відображається	Успішно
5.1	Змінити ім'я та фамілію користувача	Дані в полях для введення змінюються	Успішно
5.2	Зберегти зміни	Головна сторінка відображається. У верхньому лівому куті додатку, введені дані відображаються	Успішно

5.3	Ввести менше 2 символів у поля Ім'я та фамілії	Помилка відображається, дані не змінюються	Успішно
5.4	Залишити поля пустими	Помилка відображається, дані не змінюються	Успішно
5.5	Ввести більше 30 символів	Помилка відображається, дані не змінюються	Успішно

Таблиця Г.2 – Тест-кейси функції додавання нового запису про витрати на надходження коштів

№	Назва	Передумови	Кроки	Результати	Статус
1	Створення нового запису про надходження	Відкрити додаток, перейти на сторінку «Новий запис»	1. Натиснути на випадний список «Назва»	*Випадний список відкривається *Доступні категорії в списку відображаються	Успішно
			2. Обрати одну із категорій	*Обрана категорія відображається	Успішно
			3.Написати опис (Принаймні один символ)	*Введені символи відображаються в полі для введення	Успішно

			4. Написати в поле Сума принаймні одну цифру	*Введені символи відображаються в полі для введення	Успішно
			5. Натиснути на випадний список типу операції	*Випадний список відкривається -Тип надходження -Тип Витрата	
			6. Обрати тип Надходження	*Обраний тип відображається в полі	Успішно
			7.Натиснути на пікер дати	*Нативний календар із датами відкривається	Успішно
			8.Обрати поточну дату. Натиснути «Ок»	*Пікер закривається *В полі дата відображається введена дата	Успішно
			9.Натиснути кнопку «Зберегти»	*Головна сторінка відображається *Додана транзакція відображається	Успішно

				у списку «Історія транзакцій» *Дані транзакції відповідають тим, що були введені при додаванні *Сума транзакції підсвічується зеленим *Поле «Надходження» збільшується на введену суму *Загальний баланс збільшується на введену суму	
2	Створення нового запису про витрату	Виконати передумову першого тест-кейсу. Виконати кроки 1 – 5 із першого тест-кейсу	1.Обрати тип «Витрати»	*Обраний тип відображається в полі	Успішно

			2. Натиснути на пікер дати	*Нативний календар із датами відкривається	Успішно
			3.Обрати поточну дату. Натиснути «Ок»	*Пікер закривається *В полі дата відображається введена дата	Успішно
			4. Натиснути кнопку «Зберегти»	*Головна сторінка відображається *Додана транзакція відображається у списку «Історія транзакцій» *Дані транзакції відповідають тим, що були введені при додаванні *Сума транзакції підсвічується червоним *Поле	Успішно

				«Витрати» збільшується на введену суму *Загальний баланс зменшується на введену суму	
3	Залишити всі поля пустими	Відкрити додаток. Перейти на сторінку «Новий запис»	1.Натиснути на кнопку зберегти	*Під пустими полями відображається повідомлення про помилку *Новий запис не додається	Успішно
4	В поле сума ввести символи	Скопіювати текст «e1». Відкрити додаток. Перейти на сторінку «Новий запис»	1.В поле Сума вставити скопійований текст	*Нічого не відбувається. Скопійований текст із не валідними символами не вставляється	Успішно
5	В поле опису написати лише один пробіл « »	Відкрити додаток. Перейти на сторінку «Новий запис»	1.В поле «Опис» написати один пробіл	* Пробіл вводиться в поле	Успішно

			2.Заповнити інші поля даними	*Поля заповнюються даними	Успішно
			3. Зберегти	*Поле «Опис» має підпис із помилкою *Нова транзакція не зберігається	Успішно
6	В поле сума написати лише один пробіл « »	Відкрити додаток. Перейти на сторінку «Новий запис»	1.В поле «Сума» написати один пробіл	* Пробіл вводиться в поле	Успішно
			2.Заповнити інші поля даними	*Поля заповнюються	Успішно
			3. Зберегти	*Поле «Опис» має підпис із помилкою *Нова транзакція не зберігається	Успішно
7	Повернутися на попередню сторінку не зберігаючи зміну	Відкрити додаток. Перейти на сторінку «Новий запис»	1.Заповнити всі поля валідними даними	*Поля заповнюються	Успішно

			2.Повернутися на попередню сторінку	*Попередньо відкрита сторінка відображається *Транзакція не зберігається *Всі поля очищуються на сторінці «Новий запис»	Успішно
8			1.Заповнити всі поля валідними даними	*Поля заповнюються	Успішно
			2.Повернутися на попередню сторінку	*Сторінка відображається *Транзакція не зберігається *Всі поля очищуються на сторінці «Новий запис»	Успішно

Додаток Г Ілюстративна частина

ІЛЮСТРАТИВНА ЧАСТИНА

Інформаційна система моніторингу витрат особистого бюджету з
прогнозуванням майбутніх доходів та витрат

Магістерська кваліфікаційна робота

Інформаційна система моніторингу витрат особистого
бюджету з прогнозуванням майбутніх доходів і витрат

ВИКОНАЛА: Богач Катерина Сергіївна (ПІ-22мз)

КЕРІВНИК: к.т.н, доцент Ткаченко О.М

РисунокГ.1 - Слайд презентації №1

Актуальність дослідження

- ▶ Ведення особистого бюджету є важливим елементом фінансової самодисципліни та контролю. Це допомагає людям краще розуміти свої фінансові потреби, здійснювати свідомі рішення щодо витрат та збереження коштів, а також планувати майбутні фінансові цілі. Ведення бюджету допомагає уникнути небажаних витрат та навіть зекономити фінанси.
- ▶ Перевага використання мобільного додатку для моніторингу бюджету полягає в його зручності та доступності. Порівняно з традиційними методами, такими як ведення записів на папері або в електронних таблицях, мобільний додаток завжди під рукою, що дозволяє швидко вносити фінансові транзакції в будь-який час та в будь-якому місці.

РисунокГ.2 - Слайд презентації №2

Мета та завдання

- *Метою магістерської кваліфікаційної роботи є підвищення фінансової грамотності користувачів мобільного додатку, який допоможе їм уникати нерациональних витрат і сприятиме зростанню їх добробуту.*

Основними задачами дослідження є:

- Провести аналітичний огляд сучасних програмних засобів, які застосовуються для планування особистого бюджету.
- Розробити математичну модель для прогнозування майбутніх витрат.
- Розробити алгоритми та програмні засоби для створення мобільного програмного продукту.
- Провести тестування розробленого програмного продукту.
- Прорахувати економічний ефект від впровадження розробленого програмного продукту.

РисунокГ.3 - Слайд презентації №3

Об'єкт, предмет та методи дослідження

- *Об'єктом дослідження є процеси моніторингу поточних та прогнозування майбутніх фінансових витрат.*
- *Предметом дослідження є методи прогнозування майбутніх витрат сімейного бюджету та засоби розробки програмних систем.*
- *Методи досліджень: методи математичного моделювання для вибору методу прогнозування майбутніх витрат; методи проектування програмного забезпечення для розробки архітектури системи, алгоритмів та програмного забезпечення; методи математичної статистики для аналізу результатів тестування програмного продукту.*

РисунокГ.4 - Слайд презентації №4

Наукова новизна

- **Вперше** запропоновано архітектуру інтегрованої інформаційної системи, яка об'єднує в собі функції моніторингу витрат сімейного бюджету і прогнозування майбутніх доходів та витрат, забезпечує автоматизовану обробку фінансової інформації, що дозволяє користувачам точно передбачати свої майбутні фінансові потреби.
- **Отримав** подальшого розвитку метод експоненційного згладжування, який відрізняється від існуючих значенням параметра згладжування α , що дозволило зменшити середнє значення абсолютної похибки прогнозу майбутніх витрат.



Рисунок І.5 - Слайд презентації №5

Практична цінність отриманих результатів

- Розроблено алгоритми та програмне забезпечення, яке дозволить користувачам ефективніше відслідковувати свої витрати, а також надасть можливість систематичного планування і прогнозування майбутніх витрат. Використання такого програмного засобу сприятиме формуванню стійких фінансових навичок та покращенню фінансової грамотності користувачів.



Рисунок І.6 - Слайд презентації №6

Існуючі рішення

- ▶ Goodbudget: Budget & Finance - це додаток для управління особистим бюджетом, який дозволяє користувачам вести облік своїх фінансів. Його ключовою особливістю є використання системи "енvelopів", що дозволяє розділити бюджет на категорії та призначити певну суму грошей кожній категорії.
- ▶ CoinKeeper: spending tracker - це додаток для ведення особистого бюджету та відстеження витрат, який допомагає користувачам керувати своїми фінансами. Основною його функціональністю є відстеження витрат, внесення доходів, створення бюджетів та аналіз фінансових потоків
- ▶ Moneygraph+ – це додаток для ведення особистого бюджету та аналізу фінансів, який надає користувачам інструменти для керування своїми фінансами. Основною метою додатку є допомога користувачам відстежувати свої доходи та витрати, а також аналізувати їх для прийняття обґрунтованих фінансових рішень

РисунокГ.7 - Слайд презентації №7

Недоліки існуючих рішень

- ▶ Goodbudget: Budget & Finance – обмежена безкоштовна версія, відсутність функції прогнозування надходжень та витрат.
- ▶ CoinKeeper: spending tracker – безкоштовна версія має свої обмеження, включаючи відсутність можливості прогнозування надходжень та витрат.
- ▶ Moneygraph+ – повністю відсутній безкоштовний доступ, а відсутність функції прогнозування надходжень та витрат

РисунокГ.8 - Слайд презентації №8

Діаграма послідовності додатку

Діаграма послідовності – це тип діаграми, який показує, як об'єкти взаємодіють один з одним у певному порядку.

1. користувач відкриває додаток та потрапляє на головну сторінку, яка відображає історію транзакцій та фінансові дані.
2. З головної сторінки користувач може обрати один із двох основних шляхів: створити нову транзакцію або переглянути статистику.
3. При виборі створення нової транзакції, користувач вводить необхідну інформацію та оновлюється історія транзакцій.
4. Користувач повертається на головну сторінку з оновленими даними.

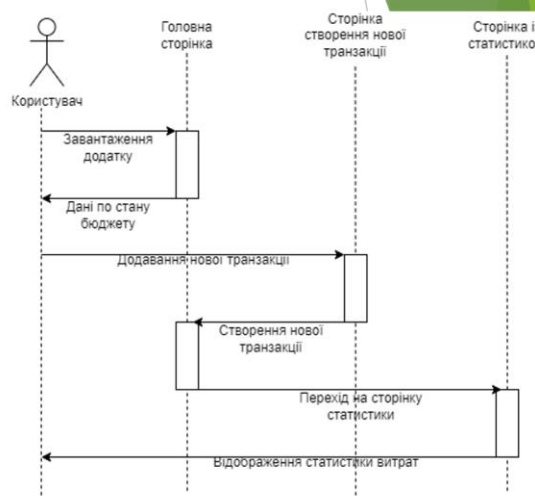


Рисунок І.9 - Слайд презентації №9

Діаграма діяльності додатку

- Діаграма діяльності відображає процеси використання розробленого мобільного додатку, підкреслюючи ключові етапи взаємодії користувача з додатком. Вона доповнює попередню діаграму зображену на попередньому слайді, розширюючи розуміння архітектури додатку.

Завдяки розробленій архітектурі інтегрованої інформаційної системи, забезпечується автоматичність прогнозування майбутніх доходів та витрат



Рисунок І.10 - Слайд презентації №10

Удосконалення методу прогнозування

- Обчисливши абсолютні помилки було отримано результати:

$$|A_2 - F_2| = |14325 - 12451| = 1874$$

$$|A_3 - F_3| = |13646 - 13013.2| = 632.8$$

$$|A_4 - F_4| = |13962 - 13148.44| = 813.56$$

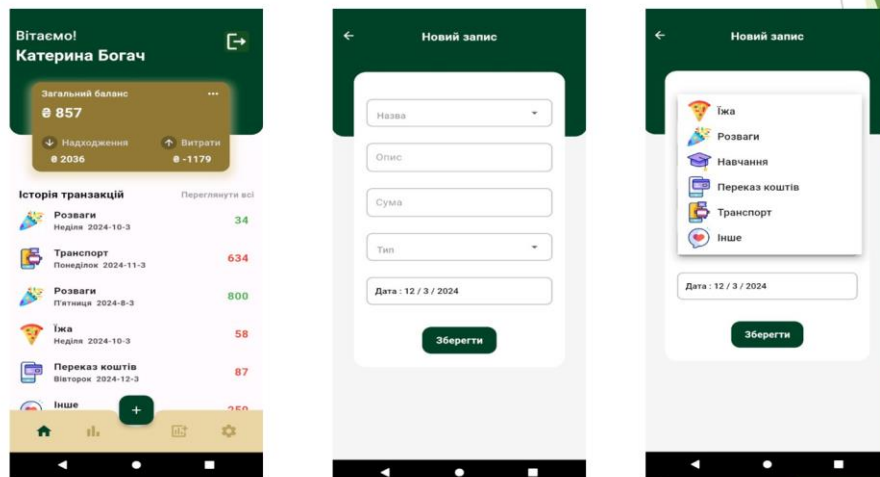
$$|A_5 - F_5| = |15235 - 13381.71| = 1853.29$$

$$|A_6 - F_6| = |12435 - 13937.70| = -1502.7$$

- $MAE = \frac{1}{5} (1874 + 632.8 + 813.56 + 1853.29 + (-1502.7)) = \frac{1}{5} * 3670.95 = 734.19$
- Також, підібраний показник α допомогло знизити значення абсолютної похибки. Похибка дорівнює 734.19 гривень, або 5% від фактичних витрат (якщо витрати становлять 15 500 гривень).

РисунокГ.11 - Слайд презентації №11

Демонстрація роботи додатку



РисунокГ.12 - Слайд презентації №12

Демонстрація роботи додатку

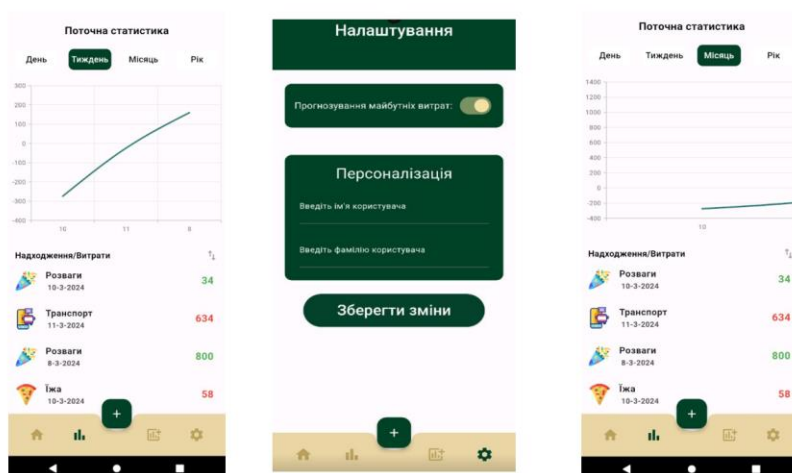


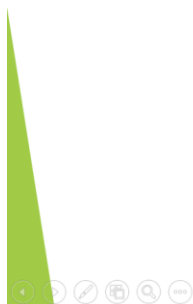
Рисунок І.13 - Слайд презентації №13

Висновки

- Під час виконання магістерської кваліфікаційної роботи було розроблено мобільний додаток для моніторингу витрат особистого бюджету з прогнозуванням майбутніх доходів та витрат. Мобільний додаток розроблений використовуючи кросплатформенну мову програмування Dart, використовуючи фреймворк Flutter.
- На основі аналізу аналогів визначено їх переваги та недоліки. Проаналізовано доступне програмне забезпечення для реалізації мобільного застосунку. Розроблено інтерфейс мобільного застосунку.
- Розроблено архітектуру інтегрованої інформаційної системи, яка об'єднує в собі функції моніторингу витрат сімейного бюджету і прогнозування майбутніх доходів та витрат.
- Тестування програмного продукту показало, що метод експоненційного згладжування забезпечує найменшу похибку прогнозування, значення якої склало 734.19 гривень від фактичних витрат.
- Виконано розрахунки та економічний аналіз в результаті якого було підтверджено доцільність, а також економічну вигідність для інвесторів та економічних установ.

Рисунок І.14 - Слайд презентації №14

Дякую за увагу



РисунокГ.15 - Слайд презентації №15