

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Інформаційна система для ведення обліку особистих витрат
Назва теми

КВРІСТ 101042.21.01.04 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»
Шифр, назва

Освітня програма «Інформаційні системи та технології»
Назва

Виконав: студент III курсу, група ІСТс-21-1

Підпис

І. О. Константінов
Ініціали, прізвище

Керівник

Підпис, дата

Є. Г. Гнатчук
Ініціали, прізвище

Нормоконтролер

Підпис, дата

І.О. Засорнова
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

Підпис

Т.О. Говорущенко
Ініціали, прізвище

«12» червня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 01 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Константінова Івана Олександровича

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система для ведення обліку особистих витрат

Керівник проекту (роботи) Гнатчук Є.Г., к.т.н., доцент.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____
Інформаційної системи для ведення обліку особистих витрат та постановка задачі щодо її
удосконалення

Проектування інформаційної системи для ведення обліку особистих витрат

Програмна реалізація інформаційної системи для ведення обліку особистих витрат

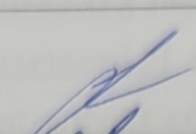
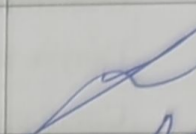
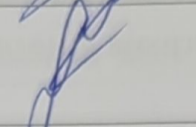
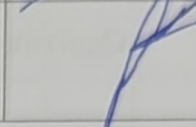
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Інтерфейс інформаційної системи

Діаграма класів інформаційної системи

Діаграма варіантів використання інформаційної системи

6. Консультанти розділів дипломного проекту (роботи)

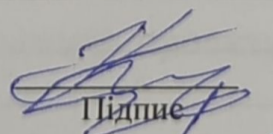
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|---|---|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Засорнова І.О., доцент кафедри КІС |  |  |
| Антиплагіат | Нічепорук А.О., доцент кафедри КІС |  |  |

7. Дата видачі завдання « 10 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

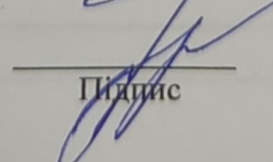
| №з/п | Назва етапів (розділів) дипломного проекту (роботи) | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1 | Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником | 10.01.2024 | виконано |
| 2 | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2024 | виконано |
| 3 | Робота над розділом 1 – аналіз предметної області та специфікація вимог до інформаційної системи | 01.03.2024 | виконано |
| 4 | Робота над розділом 2 – проектування інформаційної системи для ведення обліку особистих витрат | 01.04.2024 | виконано |
| 5 | Робота над розділом 3 – програмна реалізація інформаційної системи для ведення обліку особистих витрат | 29.04.2024 | виконано |
| 6 | Оформлення пояснювальної записки згідно вимог | 25.05.2024 | виконано |
| 7 | Попередній захист ВКР | 30.05.2024 | виконано |
| 8 | Захист ВКР на засіданні ЕК | Червень 2024 року | |

Студент



І. О. Константинов
Ініціали, прізвище

Керівник роботи



Є. Г. Гнатчук
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система для ведення обліку особистих витрат».

Автор роботи: Константинов Іван Олександрович.

Керівник роботи: Гнатчук Єлизавета Геннадіївна.

Пояснювальна записка: 56 с., 22 рис., 3 табл., 3 дод., 41 джерел.

Графічна частина: 3 креслення.

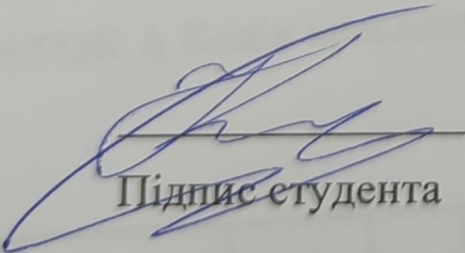
ІНФОРМАЦІЙНА СИСТЕМА, ОСОБИСТІ ФІНАНСИ, ОБЛІК ВИТРАТ, КАТЕГОРІЇ БЮДЖЕТУ, ДАНІ КОРИСТУВАЧІВ

Метою дипломної роботи є розробка та впровадження інформаційної системи для ведення обліку особистих витрат, яка дозволяє користувачам ефективно відслідковувати свої фінансові операції, аналізувати витрати за різними категоріями та покращувати управління особистим бюджетом.

Об'єктом дослідження є процеси ведення обліку особистих фінансів, що включають збір, зберігання та аналіз даних про витрати.

Предметом дослідження є інформаційна система, призначена для автоматизованого ведення обліку особистих витрат та управління фінансами.

Під час проведення даного дослідження були використані методи системного аналізу та проектування інформаційних систем, що дозволили розробити архітектуру системи, обрати відповідні технології для її реалізації та провести тестування функціональних можливостей. Система забезпечує зручний інтерфейс для користувачів, підтримує введення та категоризацію витрат, надає можливості для створення звітів та аналітичних графіків.



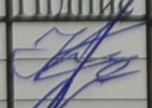
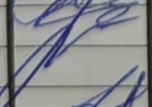
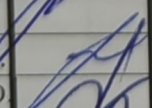

Підпис студента

30.05.2024

Дата

ЗМІСТ

| | |
|---|----|
| ВСТУП | 4 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ | 6 |
| 1.1 Аналіз предметної області | 6 |
| 1.2 Інформаційне забезпечення предметної області..... | 8 |
| 1.2.1 Існуючі інформаційні системи в предметній області..... | 10 |
| 1.2.2 Сучасні засоби створення інформаційних систем в предметній області | 16 |
| 1.3 Постановка задачі..... | 20 |
| 1.4 Висновки | 22 |
| 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ОБЛІКУ ОСОБИСТИХ ВИТРАТ | 23 |
| 2.1 Розробка архітектури інформаційної системи | 23 |
| 2.2. Функційні вимоги до інформаційної системи..... | 25 |
| 2.3. Нефункційні вимоги до інформаційної системи..... | 29 |
| 2.4. Користувацькі вимоги до інформаційної системи | 33 |
| 2.5 Висновки | 34 |
| 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ОБЛІКУ ОСОБИСТИХ ВИТРАТ | 35 |
| 3.1 Створення інформаційної системи за допомогою середовища розробки Visual Studio | 35 |
| 3.2 Опис процесу створення баз даних..... | 47 |
| 3.3. Висновки | 54 |
| ВИСНОВКИ | 55 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ | 57 |
| ДОДАТОК А Копія креслення «Інтерфейс інформаційної системи» | 62 |

| | | | | |
|---|------|-------------------|---|---------|
| КВРІСТ 101042.21.01.04 ПЗ | | | | |
| Зм. | Арк. | №докум. | Підпис | Дата |
| Виконав | | Константинов І.О. |  | |
| Перевір. | | Гнатчук Є.Г. |  | 12.06. |
| Н.контр. | | Засорнова І.О. |  | |
| Затвер. | | Говорущенко Т.О. |  | 12.06. |
| Інформаційна система для ведення обліку особистих витрат. | | | | |
| Пояснювальна записка | | | | |
| | | Літера | Аркуш | Аркушів |
| | | у | 2 | 56 |
| ХНУ ІСТс-21-1 | | | | |

ДОДАТОК Б Копія креслення «Діаграма класів інформаційної системи»... 63

ДОДАТОК В Копія креслення «Діаграма варіантів використання інформаційної системи» 64

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------|
| | | | | | КВРІСТ 101042.21.01.04 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 3 |

ВСТУП

У сучасному світі, де економічна стійкість та фінансова незалежність визначають успіх і комфорт життя, важливо мати ефективні інструменти для контролю та аналізу особистих фінансів. Вірно організований облік витрат не лише сприяє досягненню легкого контролю економічних витрат, а й показує недоліки та прогалини у використанні особистих фінансів.

Вірне використання обліку та аналіз особистих фінансів стає вирішальним елементом успішного управління фінансами. Відповідно до власних цілей і потреб, кожна особа має можливість побудувати свою фінансову стратегію, яка б відповідала її потребам. З моменту, коли особа починає контролювати свої витрати та доходи, вона має можливість уникнути боргів та непередбачених витрат, що створює фундамент для фінансового успіху.

Таким чином, фінансове планування допомагає не тільки зменшити ризики, але й досягти довгострокових фінансових цілей, таких як накопичення на пенсію або великих покупок. У світі, де цифрові технології швидко розвиваються, впровадження інноваційних рішень для фінансового обліку стає ще більш важливим, дозволяючи зберігати і аналізувати дані у зручній та доступній спосіб. Завдяки цьому користувачі можуть краще розуміти свої фінансові звички і вчасно вживати необхідних заходів для покращення свого фінансового стану.

Метою даного дипломного проекту є створення інформаційної системи для ефективного ведення обліку витрат. Заснована на принципах зручності та функціональності, ця програма має сприяти належному контролю за фінансами користувачів.

Серед основних критеріїв програми є зручність введення даних, можливість візуалізації статистики, планування та категоризація доходів і витрат, а також забезпечення захисту даних та синхронізація з різними пристроями. В процесі розробки передбачено вирішення таких завдань, як огляд існуючих програм для

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. |
| | | | | | | 4 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

обліку особистих фінансів, визначення необхідного функціоналу, тестування програмного забезпечення.

У даному проекті будуть використані методи аналізу та синтезу. Аналіз дозволить розглянути потреби користувачів та вимоги до програмного продукту, а синтез - об'єднати ці вимоги в єдине, функціональне рішення.

Використання сучасних технологій та методів розробки дозволить створити програмне забезпечення, що відповідає сучасним стандартам та вимогам ринку. Результатом даного дипломного проекту буде створення програмного забезпечення, яке стане надійним помічником у веденні обліку особистих фінансів. Практичне значення полягає у можливості користувачів ефективно керувати своїми фінансами, планувати видатки та забезпечувати стабільність фінансового стану.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 5 |

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Аналіз предметної області

Ведення обліку особистих витрат - це процес фіксації та аналізу грошових витрат особистого характеру, які здійснюються окремою особою чи домогосподарством [1]. Основна мета ведення обліку особистих витрат полягає в контролі за фінансовими ресурсами, плануванні бюджету та управлінні особистими фінансами.

У плануванні та управлінні бюджетом, найважливіше - це вміння думати довгостроково. Недостатність цього досвіду є коренем багатьох проблем [2]. З природи люди не дуже добре вміють планувати, оскільки звикли діяти на мить. Однак можна виправити це досить легко, оскільки наразі існує безліч готових, перевірених підходів та методик. Саме про них ми сьогодні й поговоримо.

Освоєння фінансової грамотності на особистому рівні дозволить ефективніше працювати у сфері управління. Найперше, слід зареєструвати всі заплановані прибутки та витрати протягом поточного місяця. Це сприятиме створенню загального фінансового зображення [22-24]. Детальний план витрат необхідно вести у письмовому вигляді, оскільки навіть у випадку гарної пам'яті важко зберігати усю інформацію в голові.

Спочатку створюємо таблицю з усіма надходженнями і витратами на папері (або в Excel, залежно від вподобань). Важливо ретельно обдумати всі аспекти життєдіяльності, щоб нічого не пропустити. На початковому етапі, звісно, можуть виникнути недоліки. Неможливо одразу здійснити чітке і правильне планування. При виявленні будь-яких недоліків, слід зафіксувати їх у нотатках [25]. Такий підхід допоможе вдосконалити схему ведення бюджету в наступному місяці. Таким чином, кожний наступний облік буде якіснішим за попередній.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|-----------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 6 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Для збереження такої інформації можна використовувати зошит, блокнот або спеціалізовані програми на смартфоні. Цей метод допоможе виявити непотрібні покупки і уникнути їх у майбутньому.

Для ефективного управління домашнім бюджетом необхідно дотримуватися 7 основних принципів:

- уникайте закредитованості. Намагайтеся уникати боргів, непланованих кредитів і невігідних кредитних карт, якщо це можливо;
- розумно накопичуйте і розподіляйте кошти. Не просто накопичуйте без мети. Краще обрати конкретну ціль, наприклад, придбання автомобіля. Така мета додасть мотивації і покращить продуктивність;
- забезпечте розуміння кожним членом сім'ї цілей і планів бюджету.
- плануйте розподіл доходів і витрат відповідно до поточних фінансових умов;
- регулярно заощаджуйте від 10% доходів на резервний фонд;
- в кінці певного періоду, наприклад, року, радуйте родину поїздкою, покупкою або розвагою;
- регулярно, найкраще щомісяця, ведіть планування і облік. Тільки систематичність допоможе досягти результатів.

Спосіб ведення бюджету варіюється в залежності від особистих вподобань і навичок. Для когось зручніше вести записи в зошиті, для когось – використовувати комп'ютер або смартфон [3].

Якщо немає можливості або бажання користуватися технологічними засобами, можна вести "комірну книгу" у паперовому форматі. Основна частина складається з таблиці, розділеної на три стовпчики (витрати, доходи, загальний підсумок) [27-29]. У перших двох ведеться облік надходжень і витрат, а третій стовпчик використовується для порівняння залишку коштів у кишенях.

Більш ефективним варіантом є використання спеціалізованих програм. Багато з них мають версії як для ПК, так і для смартфонів, що дозволяє синхронізувати дані та вносити їх у будь-який час доби [4]. Функціонал такого

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|-----------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 7 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

програмного забезпечення включає важливі опції, наприклад, автоматичне створення звітів, ведення обліку боргів та кредитів у зручній формі, планування фінансів на майбутнє, а також збереження даних про накопичення. Деякі програми також мають можливість додавання профілів користувачів, що дозволяє вести спільний бюджет з усіма членами сім'ї.

Процес ведення обліку особистих витрат включає в себе кілька етапів. По-перше, це фіксація всіх витрат, які здійснюються протягом певного періоду часу, зазвичай це може бути щотижневий, щомісячний або щорічний період. По-друге, це класифікація витрат на різні категорії, такі як їжа, житло, транспорт, розваги тощо. По-третє, аналіз та оцінка витрат з метою виявлення основних статей витрат, визначення найбільш витратних категорій та пошук можливостей для економії.

Після впровадження системи ведення обліку особистих витрат користувач може вести детальний облік своїх фінансів, виявляти зайві витрати та знаходити шляхи для економії. Аналіз такої інформації дозволяє користувачеві приймати обґрунтовані рішення щодо свого бюджету, планувати майбутні витрати та досягати фінансової стабільності.

1.2 Інформаційне забезпечення предметної області

Інформаційна система для ведення обліку особистих витрат є ключовим інструментом для досягнення фінансової стабільності та контролю над особистим бюджетом. Ця система повинна бути спроектована таким чином, щоб забезпечити ефективний збір, обробку та аналіз фінансових даних користувача.

Основною функцією інформаційної системи є збір та класифікація витрат на різні категорії, такі як їжа, житло, транспорт, розваги тощо. Для цього система повинна мати зручний інтерфейс користувача, який дозволяє легко вводити дані про витрати. Крім того, система повинна забезпечити можливість автоматичного виявлення та класифікації витрат на основі певних критеріїв, наприклад, категорії товарів чи місця покупки.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 8 |

Для аналізу фінансових даних інформаційна система повинна мати вбудовані інструменти для створення звітів та графіків, які дозволяють користувачам отримувати уявлення про свої витрати за певний період часу, виявляти тенденції та знаходити можливості для економії. Це може включати в себе аналіз витрат за місяць, квартал, рік, порівняння витрат за різні періоди, а також виявлення найбільш витратних категорій.

Для забезпечення безпеки та конфіденційності фінансових даних, інформаційна система повинна мати вбудовані заходи захисту, такі як автентифікація користувачів, шифрування даних та контроль доступу до інформації. Це дозволяє користувачам бути впевненими у тому, що їхні фінансові дані захищені від несанкціонованого доступу.

Крім того, інформаційна система може включати в себе додаткові функції, такі як нагадування про заплановані витрати, автоматичні розрахунки бюджету, інтеграцію з банківськими акаунтами для автоматичного виявлення та імпортування транзакцій.

Додатковою функціональністю, яка може бути включена до інформаційної системи для ведення обліку особистих витрат, є можливість прогнозування майбутніх витрат на основі історичних даних. Це дозволяє користувачам краще планувати свій бюджет та уникати непередбачених витрат. Наприклад, система може аналізувати попередні місячні витрати на певні категорії і автоматично рекомендувати бюджетні обмеження для майбутніх періодів.

Ще однією корисною функцією є можливість інтеграції з мобільними додатками та іншими сервісами, що дозволяють користувачам отримувати доступ до своїх фінансових даних у будь-який час та з будь-якого пристрою. Це забезпечує зручність і доступність використання системи навіть у русі.

Крім того, можливою функціональністю є можливість встановлення цілей бюджету та відстеження їх досягнення [5]. Наприклад, користувач може встановити ціль на заощадження певної суми грошей протягом певного періоду

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|-----------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 9 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

часу, і система буде автоматично відстежувати прогрес досягнення цієї цілі та надавати рекомендації щодо досягнення бюджетних цілей.

Нарешті, інформаційна система може включати в себе можливість аналізу та виявлення потенційних економічних нераціональностей або можливостей для економії. Шляхом аналізу витрат система може рекомендувати користувачеві оптимізувати свої витрати, наприклад, шляхом перегляду абонентських платежів або знаходження дешевших альтернатив за ті ж послуги чи товари.

Ці додаткові функції розширюють можливості інформаційної системи для ведення обліку особистих витрат та забезпечують користувачам більш повний контроль над їхніми фінансами.

Отже, інформаційна система для ведення обліку особистих витрат повинна бути спроектована таким чином, щоб забезпечити ефективний збір, обробку та аналіз фінансових даних користувача, забезпечуючи при цьому безпеку та конфіденційність цих даних. Така система допомагає користувачам керувати своїми фінансами, планувати бюджет та досягати фінансових цілей.

1.2.1 Існуючі інформаційні системи в предметній області

У сучасному світі роль інформаційних систем важлива у всіх сферах діяльності людини. Вони стали невід'ємною складовою для збору, обробки, зберігання та аналізу інформації з різних джерел для досягнення різноманітних цілей. Однак, вибір правильної інформаційної системи для ведення обліку особистих витрат є важливим завданням. Запровадження неефективної системи може призвести до невдачі та недосягнення поставлених фінансових цілей [33-34].

Аналіз існуючих інформаційних систем в області обліку особистих витрат дозволить визначити їх переваги, недоліки, а також потреби та очікування користувачів. Оцінка систем, таких як Mint, YNAB (You Need a Budget) та Wallet, забезпечить глибоке розуміння їх функціональності, зручності використання та ефективності для досягнення фінансової стабільності.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 10 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Аналіз конкурентів є ключовим етапом у встановленні вимог та можливостей програми, яку планується розробити. Порівняння сервісів Mint, YNAB та Wallet дозволить здійснити обґрунтований вибір оптимальної інформаційної системи для ведення обліку особистих витрат, що відповідає потребам та очікуванням користувачів.

Згадані системи несуть у собі різні можливості та особливості, які варто ретельно вивчити та порівняти, щоб обрати найбільш підходящу для конкретних потреб користувача.

Опис області інформаційної системи Mint - надання можливості автоматичного відстеження та категоризації витрат, створення бюджету, отримання фінансових порад [6]. Автоматичне відстеження витрат, категоризація транзакцій, створення та відстеження бюджету, надання фінансових порад.

Перевагами даної системи є:

- автоматичне відстеження та категоризація витрат;
- можливість створення персоналізованого бюджету;
- надання фінансових порад на основі аналізу витрат.

Недоліками даної системи є:

- можливість неякісної категоризації деяких транзакцій;
- обмежена можливість інтеграції з деякими банками;
- виявлення проблем та потреб користувачів;
- недоліки в категоризації транзакцій;
- обмеження в інтеграції з банками.

Основними потреби користувачів інформаційної системи є:

- оптимізована система категоризації транзакцій;
- розширені можливості інтеграції з банками.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 11 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Основною областю інформаційної системи є дозвіл користувачам відстежувати витрати та керувати своїми фінансами, категоризація транзакцій, планування бюджету, аналіз фінансових даних.

Переваги інформаційної системи:

- простий та інтуїтивно зрозумілий інтерфейс;
- можливість ручного внесення транзакцій.

Недоліки інформаційної системи

- обмежена функціональність порівняно з іншими системами;
- можливість недостатнього аналізу фінансових даних.

Виявлення проблем та потреб користувачів показали наступні результати система потребує покращень аналітичних можливостей, розширення функціоналу для кращого керування фінансами.

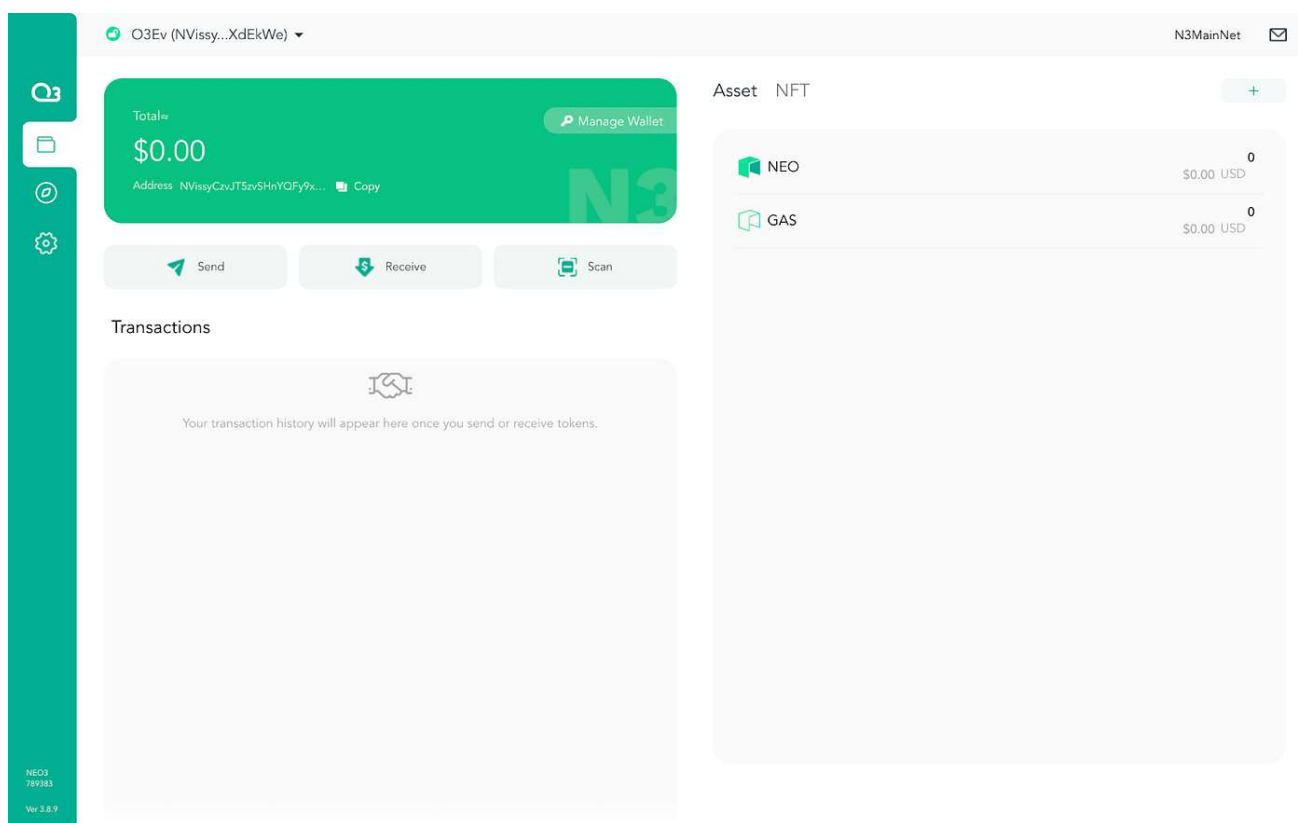


Рисунок 1.3 – Інтерфейс «Wallet» [11]

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КВРІСТ 101042.21.01.04 ПЗ | Арк. 14 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Цей аналіз дозволяє зрозуміти переваги та недоліки кожної інформаційної системи для ведення обліку особистих витрат, а також виявити потреби користувачів та можливі напрямки розвитку кожної з них.

Результати порівняння інформаційних систем відображені в таблиці 1.1.

Таблиця 1.1 - Порівняння інформаційних систем

| Параметр / Система | Mint | YNAB (You Need a Budget) | Wallet |
|--------------------|---|---|--|
| 1 | 2 | 3 | 4 |
| Функціонал | Можливість автоматичного відстеження та категоризації витрат, створення бюджету, отримання фінансових порад | Створення бюджету та планування витрат за принципом "кожен долар отриманої зарплати має бути спланованим" | Відстеження витрат, введення їх вручну або за допомогою автоматичних категорій |
| Зручність | Зручний інтерфейс, автоматичне відстеження витрат | Зручний інтерфейс, акцент на плануванні витрат | Зручний інтерфейс, можливість введення витрат вручну |
| Призначення | Підходить для тих, хто шукає автоматизований підхід до ведення бюджету та витрат | Підходить для тих, хто акцентує на плануванні кожного витраченого долара | Підходить для тих, хто віддає перевагу ручному введенню витрат |

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

Кінець таблиці 1.1 - Порівняння інформаційних систем

| 1 | 2 | 3 | 4 |
|----------|-------------------------|--------|-------------------------|
| Вартість | Безкоштовно / Платно | Платно | Безкоштовно / Платно |

Розглянемо тривимірний аналіз інформаційних систем Mint, YNAB (You Need a Budget) та Wallet для ведення особистих фінансів. Вони відрізняються та схожі за функціоналом, зручністю використання, призначенням, вартістю та приватністю. У всіх трьох системах є можливість відстежування витрат. Mint та YNAB автоматизовано категоризують витрати та дозволяють створювати бюджет, тоді як Wallet дозволяє вводити витрати вручну або автоматично.

Щодо зручності, всі три системи мають інтуїтивно зрозумілий інтерфейс. Проте Mint та YNAB акцентують на автоматизації, а Wallet дозволяє користувачам вводити дані самостійно. Призначення систем відрізняється, Mint для автоматизованого ведення бюджету, YNAB для детального планування кожного витраченого грошу, а Wallet для тих, хто віддає перевагу ручному вводу витрат.

Цінова політика також різниться, Mint та Wallet можуть бути безкоштовними або платними, в залежності від потреб користувача, тоді як YNAB є платною. Щодо конфіденційності, вона залежить від політики кожної системи, яку можна знайти на їхніх веб-сайтах чи в умовах користування.

Таким чином, кожна з цих систем має свої переваги та обмеження, які користувачі можуть врахувати при виборі тієї, яка найбільше відповідає їхнім потребам.

1.2.2 Сучасні засоби створення інформаційних систем в предметній області

Вибір належних інструментів для розробки інформаційних систем є ключовим аспектом у їх створенні. Серед таких інструментів варто відзначити мови програмування, фреймворки, бази даних, середовища розробки (IDE) та інші

технології. Вибір таких засобів залежить від численних факторів, у тому числі від вимог до проекту, наявних ресурсів та компетентності команди [32]. Поточною основною групою середовищ розробки є IntelliJ IDEA, Visual Studio та PyCharm.

JetBrains IntelliJ IDEA - це інноваційна система, що включає в себе широкий спектр інструментів для програмування, які тісно інтегровані між собою [12]. Серед цих інструментів - інтелектуальний редактор коду з передовими засобами автоматизації, потужні інструменти для рефакторингу коду, вбудована підтримка технологій J2EE, інтеграція з механізмами тестування Ant/JUnit та системами контролю версій, а також унікальний інструмент для оптимізації та перевірки якості коду. Особливий акцент робиться на інноваційному візуальному графічному дизайнері інтерфейсів.

JetBrains IntelliJ IDEA відбирає від програміста тягар рутинної роботи, допомагає вчасно виявляти та виправляти помилки, а також підвищує якість коду, піднімаючи продуктивність розробника на новий рівень.

IntelliJ IDEA, що розроблена компанією JetBrains, є інтегрованим середовищем розробки (IDE) [13]. Хоча вона призначена переважно для розробки на Java, IntelliJ IDEA також підтримує інші мови програмування. Вона забезпечує всі необхідні інструменти для розробників Java та Kotlin, включаючи розумний редактор коду, вбудовані інструменти розробника, підтримку фреймворків, баз даних та веб-розробку. IntelliJ IDEA спроектована з урахуванням максимізації продуктивності розробників, надаючи автоматизоване завершення коду, статичний аналіз коду та рефакторинг.

Переваги:

- велика кількість вбудованих інструментів для розробки та налагодження додатків;
- підтримка різних мов програмування;
- велика кількість доступних плагінів;
- інтеграція з системами контролю версій, такими як Git;
- інтеграція з іншими інструментами JetBrains.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 17 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Недоліки:

- високі вимоги до обсягу оперативної пам'яті та можливість зависання на слабкопотужних системах;
- складний початковий процес використання;
- платна версія з певними функціями.

Visual Studio, розроблена Microsoft, є ще одним інтегрованим середовищем розробки (IDE). Вона забезпечує розробникам широкий спектр інструментів для створення програмного забезпечення, включаючи редактор коду, візуальний дизайнер, інструменти для налагодження, компілятори, підтримку баз даних та інтеграцію з системами контролю версій.

Visual Studio - це інтегроване середовище розробки, яке забезпечує розробників потужним набором інструментів для виконання всього циклу розробки програмного забезпечення в одному місці [14]. Він дозволяє записувати, редагувати, налагоджувати та збирати код, а також розгортати програми. Окрім редагування та налагодження коду, Visual Studio включає в себе компілятори, інструменти автодоповнення коду, системи керування версіями, розширення та інші функції, що полегшують кожен етап розробки програмного забезпечення.

Інтегроване середовище розробки Visual Studio пропонує широкий спектр функцій, які спрощують процес написання та управління кодом з впевненістю [15]. Наприклад, завдяки засобам розробки, таким як GitHub Copilot і IntelliCode, можна швидко та точно створювати код, а також отримувати рекомендації щодо його покращення за допомогою лампочок [16]. Крім того, функція структурування дозволяє швидко розгортати або згортати блоки коду. Перегляд та організація коду здійснюється за допомогою Переглядача рішень, який дозволяє переглядати код, впорядкований за файлами, або відображати класи, що допомагає у вивченні коду за класами.

Переваги:

- підтримка різних мов програмування;
- автоматичне імпортування файлів та широкі можливості налагодження;

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 18 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

- інтеграція з системами контролю версій, такими як Git;
- інтеграція з іншими продуктами Microsoft.

Недоліки:

- іноді може відчуватися нестабільність та повільність роботи;
- складний початковий процес використання;
- час, необхідний для встановлення та оновлення, може бути значним.

PyCharm - інтегроване середовище розробки, що є продуктом компанії JetBrains, - це кросплатформний інструмент, який надає широкі можливості для зручної роботи з мовою програмування Python. Від автодоповнення коду до потужних інструментів відлагодження та рефакторингу, PyCharm створений для ефективної роботи розробника з Python.

PyCharm був розроблений компанією JetBrains, відомою своїми інноваційними інструментами для розробки програмного забезпечення. Ця міжнародна компанія спеціалізується на створенні інструментів для програмістів у різних мовах програмування, таких як Java, Kotlin, C#, F#, C++, Ruby, Python, PHP, JavaScript і багатьох інших [17]. Почавши як проект під назвою IntelliJ, вона стала провідним постачальником рішень для розробки програмного забезпечення, які використовуються професіоналами з усього світу.

PyCharm пропонує розробникам безліч ключових функцій, які істотно спрощують процес програмування. Ось лише кілька з них:

- налагодження коду, потужний налагоджувач дозволяє знаходити та виправляти помилки, встановлюючи точки зупинки та аналізуючи значення змінних;
- рефакторинг, інструменти рефакторингу допомагають змінювати структуру коду без втрати функціональності;
- підтримка систем контролю версій, інтеграція з Git, Mercurial та іншими системами дозволяє зручно працювати з історією змін;
- автодоповнення коду, інтелектуальне автодоповнення прискорює процес написання коду і знижує ймовірність помилок;

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 19 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

- інспектування коду, статичний аналіз коду допомагає виявляти потенційні помилки та підтримувати високу якість коду;
- інтеграція з віртуальними оточеннями, PyCharm надає зручний інтерфейс для роботи з віртуальними оточеннями Python, що дозволяє створювати та керувати різними оточеннями для кожного проекту.

Це лише огляд ключових функцій PyCharm. Завдяки цьому розмаїттю інструментів розробники можуть ефективніше працювати над проектами, покращувати якість свого коду та зосереджуватися на творчому боці програмування.

1.3 Постановка задачі

Провівши аналіз предметної області було виявлено – це область, що стосується керування фінансами і фінансовою діяльністю індивідуальної особи або домогосподарства. Основною метою такої систем повинно виступати систематичне відстеження та аналіз особистих витрат з метою керування бюджетом, ефективним управлінням фінансами та досягненням фінансових цілей.

Опираючись на аналіз, було сформовано наступний перелік основних вимог до інформаційної системи для ведення обліку особистих витрат, він складається з наступних пунктів:

- зручний та інтуїтивно зрозумілий інтерфейс, система повинна мати простий та зрозумілий інтерфейс, що дозволить користувачам легко вводити та переглядати інформацію про свої витрати;
- можливість автоматичного відстеження витрат система повинна мати можливість автоматичного відстеження та категоризації витрат з використанням банківських транзакцій або інших джерел фінансової інформації;
- створення та керування бюджетом користувачі повинні мати можливість створювати та налаштовувати особисті бюджети, встановлювати цілі та відстежувати їх досягнення;

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 20 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

– гнучкі засоби аналізу та звітності система повинна надавати користувачам засоби для аналізу своїх витрат, включаючи графіки, звіти та статистику;

– забезпечення конфіденційності і безпеки даних система повинна мати ефективні заходи безпеки для захисту особистих фінансових даних користувачів від несанкціонованого доступу.

Сформувавши вимоги до інформаційної системи та порівнявши системи для розробки інформаційних систем, були створені наступні вимоги для середовища розробки:

– доступність, система повинна мати можливість працювати як в онлайн, так і в офлайн режимах, щоб користувачі могли отримувати доступ до своїх фінансових даних у будь-який час та в будь-якому місці;

– ефективне зберігання та обробка даних, система повинна мати потужні сервери та бази даних для зберігання великого обсягу фінансової інформації та ефективної обробки запитів користувачів;

– безпека даних, система повинна використовувати сучасні методи шифрування та захисту даних для запобігання несанкціонованому доступу до особистої фінансової інформації користувачів;

– масштабованість та надійність система повинна бути масштабованою та надійною, щоб забезпечити продуктивну роботу при будь-якому обсязі витрат та кількості користувачів.

Опираючись на вищеперераховані вимоги, до середовища розробки, було обрано середовище від Microsoft, а саме Visual Studio. Для розробки інформаційної системи для ведення обліку особистих витрат, Visual Studio має декілька переваг:

– мова програмування, середовище надає широкі можливості для розробки за допомогою мови програмування C#, яка є ефективною та потужною для створення різноманітних програм, включаючи системи обліку особистих витрат;

– багатofункціональність, Visual Studio має широкий набір інструментів та функцій, що сприяють розробці складних програм. Його можливості

розширюються за допомогою плагінів та додаткових компонентів, що дозволяє забезпечити всі необхідні функції для ведення обліку особистих витрат;

– зручний інтерфейс розробки, система відома своїм зручним та інтуїтивно зрозумілим інтерфейсом, який дозволяє розробникам швидко орієнтуватися та працювати з проектами будь-якої складності;

– підтримка різних платформ, які дозволяють розробляти програмне забезпечення для Windows, Android, iOS та веб-додатки. Це дозволяє створювати універсальні та доступні інформаційні системи для ведення обліку особистих витрат.

Загалом, Visual Studio є потужним та високоефективним інструментом, який надає всі необхідні засоби та компоненти

1.4 Висновки

Загалом, керування власними фінансами є однією з найбільш критичних навичок у сучасному світі. Початково, більшість людей не мають такої навички, оскільки батьки та навіть система освіти часто не приділяють цьому достатньо уваги.[39-40] Отже, вивчення цих навичок покладається на вас. Проте, є надія, оскільки наразі існує багато методик, підходів і програм, які допомагають у цьому процесі. Разом з тим, розвиток цієї навички може відкрити широкі можливості для управління власними фінансами, покращити фінансове благополуччя і забезпечити більш стабільне майбутнє [41].

Проведено огляд існуючих напрацювань в області створення програмних засобів для ведення персональної бухгалтерії . Базуючись на цьому було поставлено завдання для створення власного програмного засобу, визначено варіанти використання програми та висунуто функціональні та нефункціональні вимоги.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 22 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ОБЛІКУ ОСОБИСТИХ ВИТРАТ

2.1 Розробка архітектури інформаційної системи

Архітектура програми - це структурне розташування компонентів програмного забезпечення та їх взаємодія, яке визначається на етапі проектування програми. [18] Вона описує загальну організацію програми, з урахуванням модулів, класів, функцій та їх взаємозв'язків.

Архітектура програми не лише визначає структуру програми, але й визначає основні принципи реалізації, взаємодії між компонентами, а також стратегії тестування та розгортання [19]. Основна мета архітектури програми - забезпечити ефективність, масштабованість, надійність та легкість супроводу програмного продукту.

Основні функції архітектури програми включають:

- структурування коду, вона допомагає розбити великі програми на менші, більш керовані модулі, що спрощує розробку та супровід програми;
- визначення взаємозв'язків, архітектура визначає, як компоненти взаємодіють між собою, що дозволяє забезпечити правильну роботу програми та уникнути проблем;
- забезпечення розширюваності та масштабованості, добре спроектована архітектура дозволяє легко додавати новий функціонал та масштабувати систему з ростом потреб;
- підтримка та супровід, архітектура допомагає зрозуміти структуру програми, що полегшує роботу команди розробників та підтримки програмного забезпечення;
- забезпечення безпеки та надійності вона дозволяє визначити механізми захисту даних та забезпечити стійкість програми до непередбачених ситуацій.

Після проведення ієрархічного аналізу вимог до програмного забезпечення були визначені групи, які відповідають окремим підсистемам, що будуть використовуватися для створення користувацького інтерфейсу, взаємодії з базою даних та реалізації класів у предметній області [26].

На зображенні у рисунку 2.1 демонструється діаграма архітектури програмного забезпечення для управління особистими фінансами фізичних осіб на мобільній платформі.



Рисунок 2.1 - Архітектура програмної частини для інформаційної системи

Виходячи з функції системи, встановлюємо наступні функціональні блоки системи, які співпадають з варіантами використання:

- пошук - «Search» - користувач повинен мати можливість швидко та зручно знаходити записи про витрати чи прибутки за певний період часу, за категорією, за ключовим словом або за іншими параметрами. Пошук повинен бути ефективним і швидким навіть при великій кількості записів у системі. Система повинна забезпечувати можливість фільтрації та сортування результатів пошуку;
- створити звіт - «Create_Report» - користувач повинен мати можливість створювати звіти про витрати та прибутки за певний період часу, за категорією, за джерелом тощо. Звіти повинні мати зручний формат та бути легко зрозумілі для користувача;

- внести прибутки- «Make_profits» - користувач повинен мати можливість додавати записи про отримані прибутки в систему. Система повинна дозволяти користувачу вказати суму прибутку, дату, джерело отримання тощо;
- внести витрати- «Make_costs» - користувач повинен мати можливість додавати записи про витрати в систему. Система повинна дозволяти користувачу вказати суму витрат, дату, категорію, метод платежу;
- створити категорію- «New_Category» - користувач повинен мати можливість створювати нові категорії для витрат та прибутків. Система повинна дозволяти користувачу вказати назву категорії, опис та, за необхідності, батьківську категорію. Можливість редагування та видалення існуючих категорій.

2.2 Функційні вимоги до інформаційної системи

Функційні вимоги до інформаційних систем - це конкретні функції, які має виконувати програмний продукт або система для задоволення потреб користувачів та досягнення мети проекту. Основна мета функційних вимог - описати поведінку системи та функціональність, яку вона має надавати [20]. Ці вимоги описують поведінку системи і сервіси (функції), які вона виконує, і залежать від типу розроблюваної системи і від потреб користувачів. Якщо функціональні вимоги оформлені як користувацькі, вони, як правило, описують системи в узагальненому вигляді [30]. На противагу цьому функціональні вимоги, оформлені як системні, описують систему максимально докладно, включаючи її вхідні і вихідні дані, виключення

Функціональні вимоги зазвичай мають такі характеристики [21]:

- специфіка, вони детальні та конкретні, залишаючи мало місця для двозначності. Окреслюють точні функції, входи та виходи системи;
- можливість перевірки, функціональні вимоги можна перевірити та підтвердити, щоб переконатися, що програмне забезпечення працює належним чином;

– орієнтація на користувача, вони тісно узгоджені з потребами та очікуваннями користувача, гарантуючи, що програмне забезпечення виконує заплановану мету;

– змінність, функціональні вимоги можуть змінюватися протягом проекту, оскільки відгуки користувачів і потреби бізнесу розвиваються.

В принципі специфікація функціональних вимог повинна бути комплексною і несуперечливою. Комплексність має на увазі опис (визначення) усіх системних сервісів. Несуперечність означає відсутність несумісних і взаємовиключних визначень сервісів.

Для системи для ведення обліку особистих витрат можна визначити наступні функційні вимоги:

- введення витрат - користувач повинен мати змогу вносити дані про свої витрати, такі як сума, дата та категорія витрати;
- аналіз витрат - система має забезпечувати можливість аналізу витрат за певний період часу або за категоріями, наприклад, щомісячний аналіз витрат на їжу, транспорт;
- створення звітів - система повинна дозволяти генерувати звіти про витрати, які можуть бути використані для податкової звітності або для внутрішнього аналізу;
- управління категоріями витрат - користувач повинен мати можливість додавати, редагувати та видаляти категорії витрат відповідно до своїх потреб;
- нагадування про витрати - система може надавати можливість налаштування нагадувань для користувачів щодо їх планованих витрат або платежів.

Вимоги будуть деталізовані за допомогою діаграми варіантів використання. Єдиний учасник у системі, взаємодіє з усіма варіантами використання. У модель включені асоціації, які визначають напрямки передачі інформації між учасником та варіантами використання. Проводиться розподіл варіантів використання по

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 26 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

пакетам і визначається їх взаємодія з учасником, що репрезентує діаграму варіантів використання у функціональній моделі.

На рисунку 2.4 наведена загальна діаграма варіантів використання розробленого програмного забезпечення.

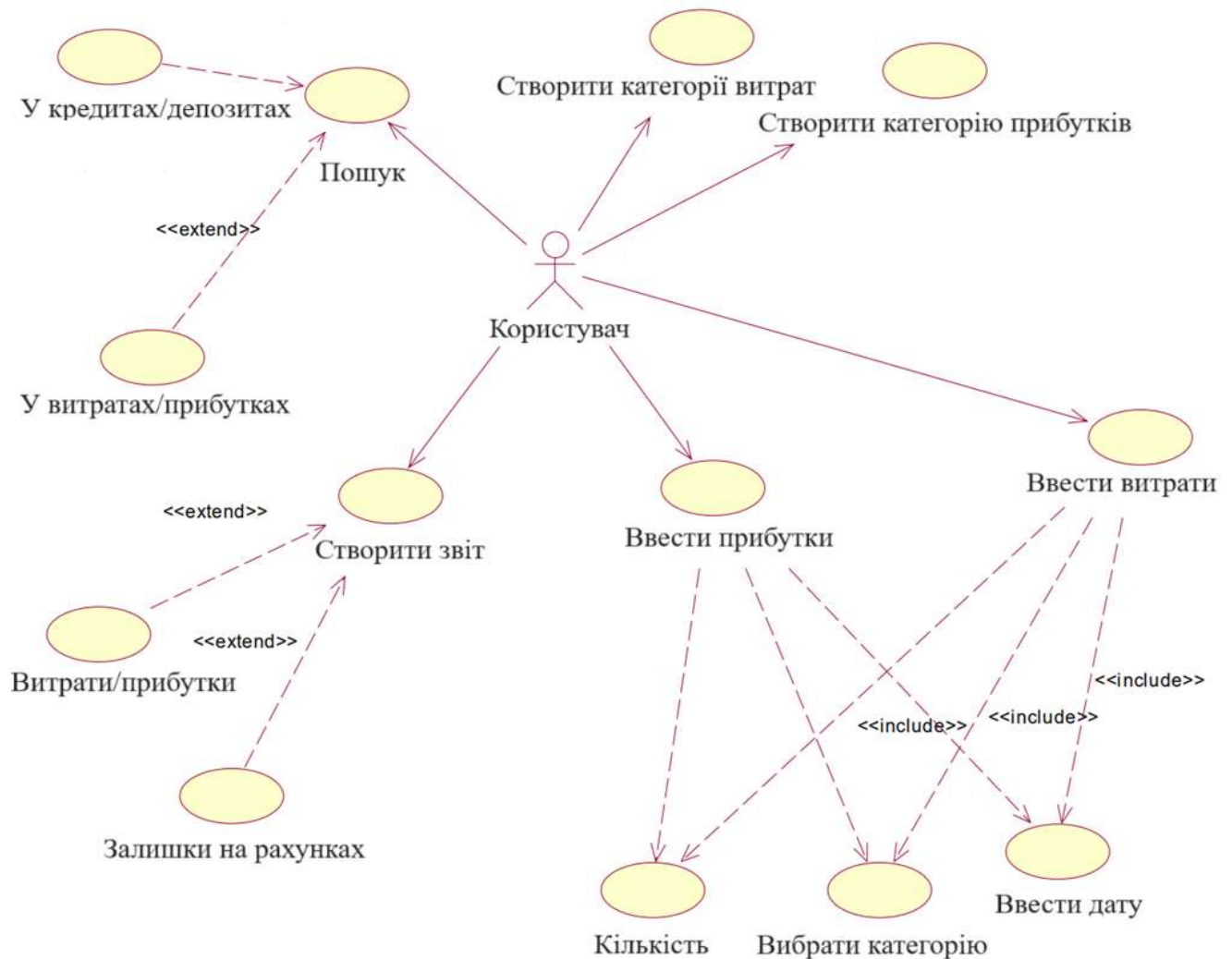


Рисунок 2.4 - Діаграма варіантів використання

Таблиця 2.1 - Функційні вимоги

| Назва вимог | Атрибути вимог | |
|-------------|----------------|------------|
| | Приоритет | Складність |
| 1 | 2 | 3 |
| Пошук | Обов'язкова | Середня |

Кінець таблиці 2.1 - Функційні вимоги

| 1 | 2 | 3 |
|---------------------|---------------|---------|
| Створення звіту | Обов'язкова | Середня |
| Внесення прибутків | Обов'язкова | Висока |
| Внесення витрат | Обов'язкова | Середня |
| Створення категорій | Рекомендована | Середня |

Таблиця містить важливі атрибути вимог до інформаційної системи. Кожна вимога має вказані два атрибути: пріоритет та складність.

Пошук - вимога має обов'язковий пріоритет, пошукова функція є критичною для користувача та повинна бути реалізована в системі. Складність середня, реалізація пошукового механізму потребує певного рівня зусиль.

Створення звіту - вимога також має обов'язковий пріоритет, оскільки звіти є важливою функцією для аналізу фінансових даних. Складність її реалізації середня, що вказує на наявність певних технічних викликів, але не надто складних.

Внесення прибутків - вимога також має обов'язковий пріоритет через важливість функції внесення прибутків в систему. Однак вона має високий рівень складності, реалізація цієї функції є складною і вимагає більше ресурсів.

Внесення витрат - вимога також має обов'язковий пріоритет, але її складність оцінюється як середня. Внесення витрат є важливою функцією, її реалізація може бути менш складною порівняно з внесенням прибутків.

Створення категорій - вимога має рекомендований пріоритет, що означає, що вона важлива, але не критична для користувача. Складність реалізації також оцінюється як середня, функція може бути корисною, вона не є надто складною для реалізації в інформаційній системі

2.3 Нефункційні вимоги до інформаційної системи

Нефункційні вимоги до інформаційних систем - це вимоги, які визначають якісні аспекти системи, такі як її надійність, продуктивність, безпека, зручність використання та інші характеристики, що не пов'язані безпосередньо з функціональністю програми [35]. Основна мета нефункційних вимог - забезпечити задоволення потреб користувачів та відповідати вимогам предметної області з точки зору якості та продуктивності системи [20].

Крім того, нефункціональні вимоги можуть визначати обмеження на систему, наприклад на пропускну здатність пристроїв введення-виведення, або формати даних, використовуваних у системному інтерфейсі.

Нефункціональні вимоги мають такі характеристики [21]:

- якість, на відміну від функціональних вимог, які зазвичай оцінюються кількісно, нефункціональні вимоги зосереджені на якісних аспектах, таких як продуктивність, надійність і безпека;
- глобальність, вони застосовуються до всієї системи та впливають на її загальну поведінку;
- стабільність, як правило, залишаються стабільними протягом усього життєвого циклу проекту, рідше піддаються змінам порівняно з функціональними вимогами;
- вимірність, хоча виміряти нефункціональні вимоги може бути складно, їх все одно можна оцінити та перевірити.

Багато нефункціональних вимог відносяться до системи в цілому, а не до окремих її засобів. Це означає, що вони більш значимі і критичні, чим окремі функціональні вимоги. Помилка, допущена у функціональній вимозі, може знизити якість системи, помилка в нефункціональних вимогах може зробити систему неприцездатною.

Разом з тим нефункціональні вимоги можуть відноситись не тільки до самої програмної системи: одні можуть відноситись до технологічного процесу

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 29 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

створення ПЗ, інші – містити перелік стандартів якості, що накладаються на процес розробки. Крім того, у специфікації нефункціональних вимог може бути зазначене, що проектування системи повинне виконуватися тільки певними CASE-засобами, і наведений опис процесу проектування, якому необхідно впливати.

Для системи для ведення обліку особистих витрат можна визначити наступні нефункційні вимоги:

- швидкодія та продуктивність: система повинна забезпечувати швидку реакцію на запити користувачів та ефективну обробку великого обсягу фінансових даних;
- надійність, система має бути стійкою до збоїв та забезпечувати безперебійну роботу, щоб уникнути втрати фінансової інформації;
- безпека, забезпечення конфіденційності та цілісності фінансових даних користувачів шляхом застосування механізмів аутентифікації, авторизації та шифрування;
- зручність використання, розробка інтуїтивного та зручного інтерфейсу користувача, що дозволяє легко та швидко виконувати операції з обліку особистих витрат;
- масштабованість та гнучкість, система повинна бути гнучкою та здатною до розширення, щоб вона могла враховувати змінні потреби користувачів та збільшувати свою функціональність.

В таблиці 2.2 зображено нефункційні вимоги для інформаційної системи для ведення обліку особистих витрат

Таблиця 2.2 – Специфікація нефункційних вимог для інформаційної системи

| Ідентифікатор вимог | Назва вимоги | Атрибути вимоги | |
|-----------------------------------|--------------------|-----------------|------------|
| | | Пріоритет | Складність |
| Застосовність | | | |
| 1 | Пошук | Обов'язкова | 20 |
| 2 | Створити звіт | Обов'язкова | 30 |
| 3 | Внести прибутки | Обов'язкова | 15 |
| 4 | Внести витрати | Обов'язкова | 15 |
| 5 | Створити категорію | Рекомендована | 20 |
| Надійність | | | |
| 1 | Пошук | Обов'язкова | 98 |
| 2 | Створити звіт | Обов'язкова | 98 |
| 3 | Внести прибутки | Обов'язкова | 98 |
| 4 | Внести витрати | Обов'язкова | 98 |
| 5 | Створити категорію | Рекомендована | 95 |
| Експлуатаційна придатність | | | |
| 1 | Пошук | Обов'язкова | 100 |
| 2 | Створити звіт | Обов'язкова | 100 |
| 3 | Внести прибутки | Обов'язкова | 100 |
| 4 | Внести витрати | Обов'язкова | 100 |
| 5 | Створити категорію | Рекомендована | 100 |

Таблиця містить інформацію про різні вимоги до інформаційної системи, розподілені за трьома аспектами: застосовність, надійність і експлуатаційна придатність

Застосовність - ця категорія визначає, наскільки добре система відповідає потребам та очікуванням користувачів і як ефективно вона виконує свої основні

функції. Застосовність включає в себе важливість функцій системи, їхню доступність та зручність для користувача. Вимоги до застосовності допомагають забезпечити, що система задовольняє потреби користувачів і допомагає їм досягати своїх цілей ефективно.

Надійність - ця категорія визначає, наскільки стійкою є система до випадкових або навмисних помилок, які можуть виникнути під час її роботи. Надійність системи визначається її здатністю до ефективної роботи протягом тривалого часу без виникнення ситуацій, коли система недоступна або функціонально несправна. Вимоги до надійності спрямовані на забезпечення стабільності та безпеки роботи системи.

Експлуатаційна придатність - ця категорія визначає, наскільки легко користувач може використовувати систему та як швидко він може навчитися користуватися всіма її функціями. Експлуатаційна придатність включає в себе такі аспекти, як інтуїтивність інтерфейсу користувача, наявність документації та навчальних матеріалів, а також підтримка системи та можливість швидко вирішувати проблеми, що виникають у користувачів. Вимоги до експлуатаційної придатності допомагають забезпечити, що система легко використовується та задовольняє потреби користувачів.

Таблиця 2.2 допомагає систематизувати та узагальнити вимоги до інформаційної системи. Кожна категорія відображає конкретний аспект функціонування системи, який важливий для забезпечення її ефективності, надійності та зручності в експлуатації. За допомогою цих категорій можна чітко визначити вимоги до системи, що дозволяє краще розуміти, як система повинна працювати та які функції та характеристики є найбільш важливими для користувачів. Такий підхід сприяє розробці більш якісної та зручної для використання інформаційної системи.

2.4 Користувацькі вимоги до інформаційної системи

Користувацькі вимоги до інформаційних систем - це вимоги, що визначають очікування, потреби та вимоги користувачів до функціональності, інтерфейсу та інших аспектів системи [20]. Ці вимоги визначаються з урахуванням потреб і вимог користувачів, що будуть використовувати систему, і вони слугують основою для розробки програмного продукту. Деякі проблеми, що виникають у процесі розробки вимог, породжені відсутністю чіткого розуміння відмінності між цими різними рівнями вимог. Щоб розрізнити вимоги різних рівнів, тут використовуються терміни користувацькі вимоги (user requirements) для позначення високорівневих узагальнених вимог.

Для системи для ведення обліку особистих витрат користувацькі вимоги можуть включати: простий та зрозумілий інтерфейс користувача: користувачі очікують, що інтерфейс системи буде інтуїтивно зрозумілим та легким у використанні, навіть для тих, хто не має великого досвіду в області обліку фінансів.

Можливість оперативного та зручного внесення витрат - користувачі виявляють бажання мати можливість швидко та просто вносити інформацію про свої витрати, щоб забезпечити актуальність даних про свої фінанси.

Аналітичні звіти та статистика - користувачі очікують можливості створювати різноманітні звіти та аналізувати свої витрати за різними категоріями, періодами часу та іншими параметрами.

Персоналізовані налаштування та нагадування - система повинна надавати можливість налаштовувати нагадування про заплановані витрати або платежі, а також персоналізовані налаштування, що відповідають індивідуальним потребам користувача.

Безпека та захист конфіденційності - користувачі очікують, що їх фінансові дані будуть захищені від несанкціонованого доступу та збережені у безпечному середовищі.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 33 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Мобільність та доступність - система має бути доступною для використання на різних пристроях та платформах, таких як комп'ютери, смартфони, планшети тощо, з метою забезпечення користувачам доступу до своїх фінансових даних у будь-який час та в будь-якому місці.

2.5 Висновки

У межах цього розділу проведено проектування та розробка архітектури інформаційної системи. Розглянуті функційні, не функційні вимоги а також користувацькі. В результаті основні функції архітектури будуть наступними:

- структурування коду;
- визначення взаємозв'язків;
- забезпечення розширюваності та масштабованості;
- підтримка та супровід;
- забезпечення безпеки та надійності.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 34 |

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ОБЛІКУ ОСОБИСТИХ ВИТРАТ

3.1 Створення інформаційної системи за допомогою середовища розробки Visual Studio

У цьому розділі було розглянуто етапи створення інформаційної системи для ведення обліку особистих витрат. Детально описано процес налаштування середовища розробки, вибір технологій і мов програмування, а також реалізацію основних функціональних блоків системи. Особлива увага буде приділена забезпеченню зручності використання, безпеки даних і оптимізації продуктивності.

Спочатку потрібно створити проект програми на мові програмування C#. Для цього потрібно обрати файл шаблонів, представлених у Visual Studio, це позбавить нас від зайвих зусиль при розробці інформаційної системи.

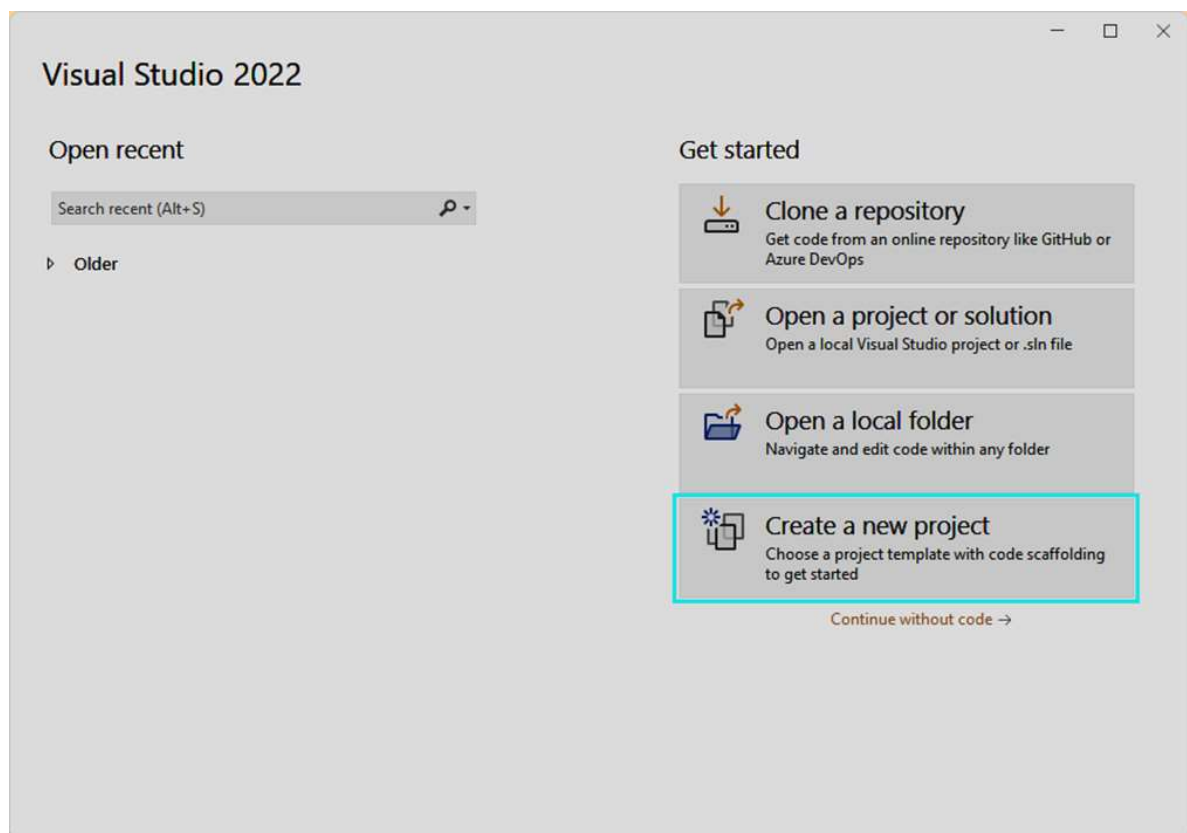


Рисунок 3.1 – Стартове вікно середовища Visual Studio

Далі у вікні створення проекту обираємо шаблон Програми Windows Forms.

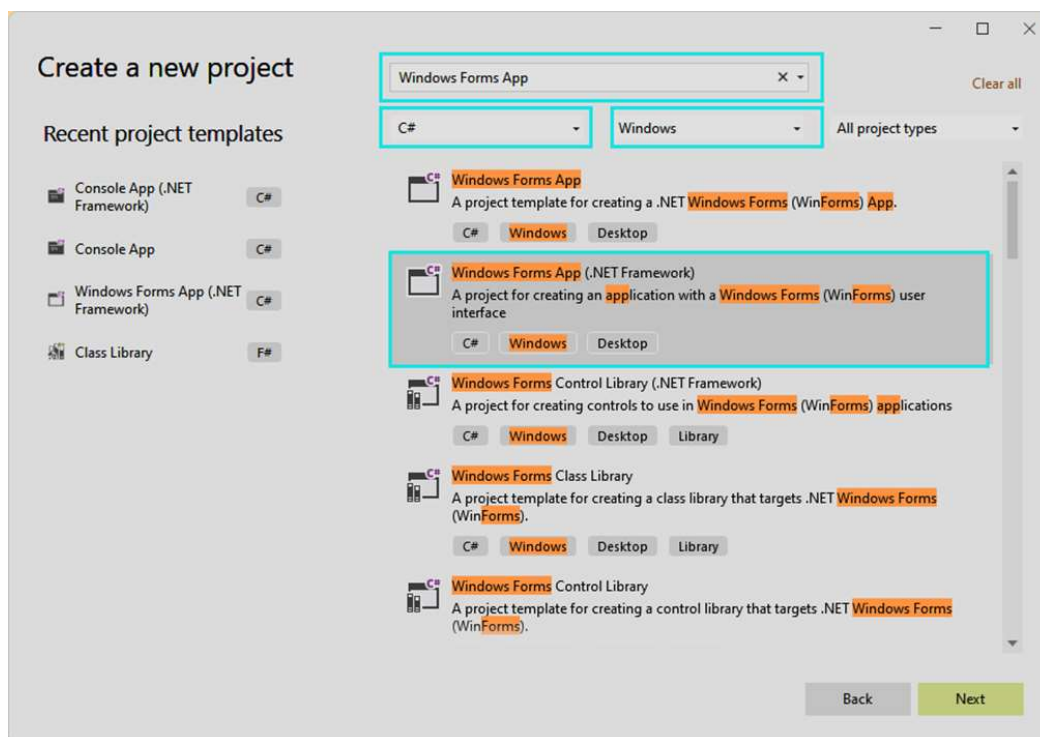


Рисунок 3.2 – Вікно створення проекту середовища Visual Studio

Наступним кроком буде налаштування імені проекту та вибір місця розташування проекту.

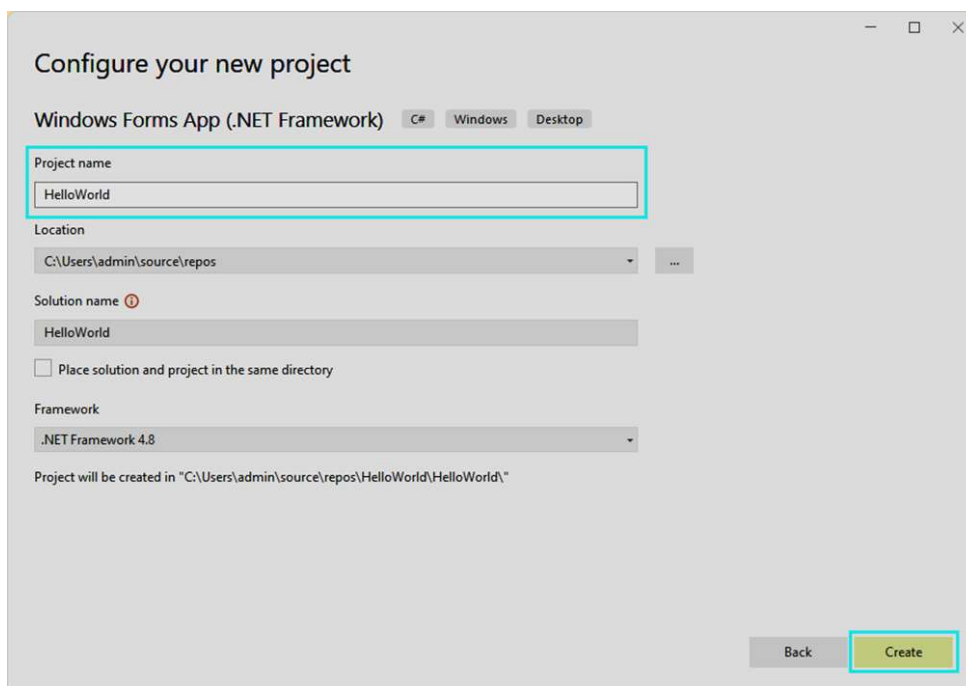


Рисунок 3.3 – Вікно конфігурації проекту

Після створення проекту виконаємо розкадровку вікон використання, щоб описати необхідні екранні форми, які застосовуються для реалізації функцій.

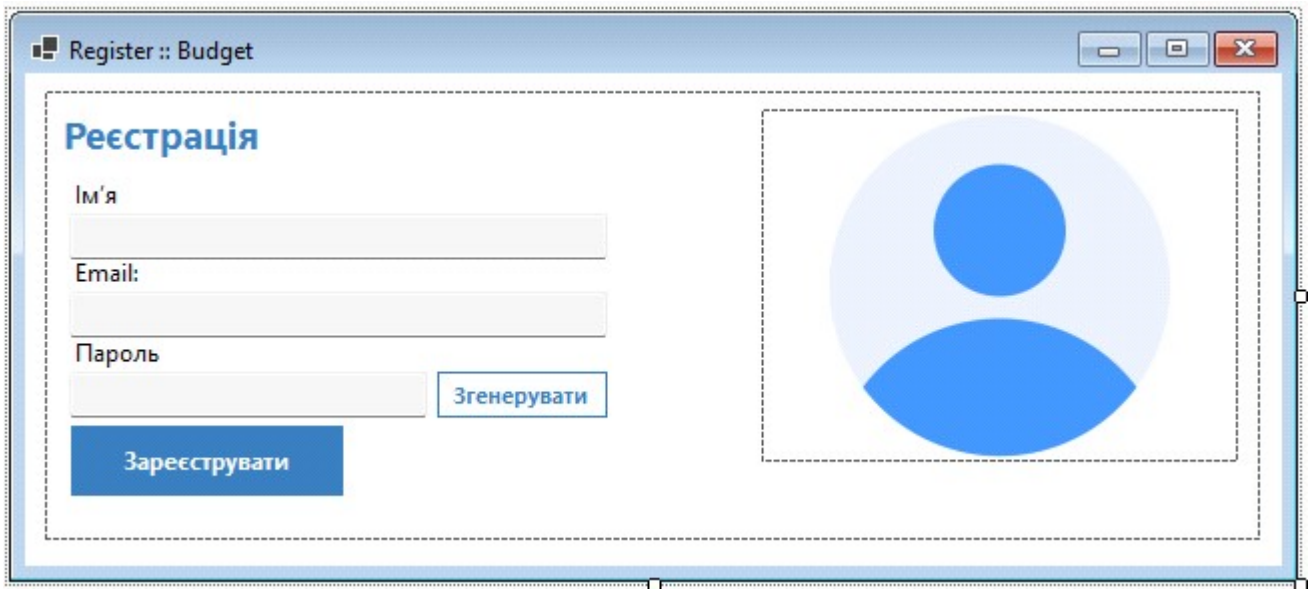


Рисунок 3.4 – Вікно реєстрації користувачів

Дане вікно програми представляє собою форму реєстрації користувача в інформаційній системі для ведення обліку особистих витрат. У вікні розміщені поля для введення імені, електронної пошти та пароля, а також кнопки для генерації пароля та реєстрації нового користувача. У конструкторі форми виконується ініціалізація компонентів форми.

Коли користувач натискає кнопку "Зареєструвати", виконується перевірка валідності введених даних. Якщо всі поля заповнені коректно, створюється новий об'єкт користувача з введеними параметрами (ім'я, електронна пошта, пароль). Цей новий користувач додається до бази даних, зберігаються зміни, і форма закривається з результатом DialogResult.OK.

Натискання кнопки "Згенерувати" викликає метод, який створює випадковий пароль довжиною 10 символів. Цей пароль складається з великих і малих літер, цифр та спеціальних символів. Згенерований пароль встановлюється в поле пароля.

При введенні тексту в поле імені, автоматично заповнюється поле електронної пошти, додаючи суфікс @gmail.com до введеного імені. Це полегшує

процес введення даних і зменшує можливість помилок при введенні адреси електронної пошти.

Також наявний метод валідації, який перевіряє, чи всі обов'язкові поля (ім'я, електронна пошта та пароль) заповнені. Для цього використовується клас, який перевіряє кожне поле і відображає повідомлення про помилки за допомогою об'єкта `err`, якщо поле залишилось порожнім або заповнене некоректно.

Форма реєстрації має за мету забезпечити зручний і безпечний спосіб створення нових користувачів в системі, забезпечуючи генерацію надійних паролів та перевірку коректності введених даних перед їх збереженням у базі даних.

Наступним вікном є форма для авторизації користувачів, вона зображена на рисунку 3.5.

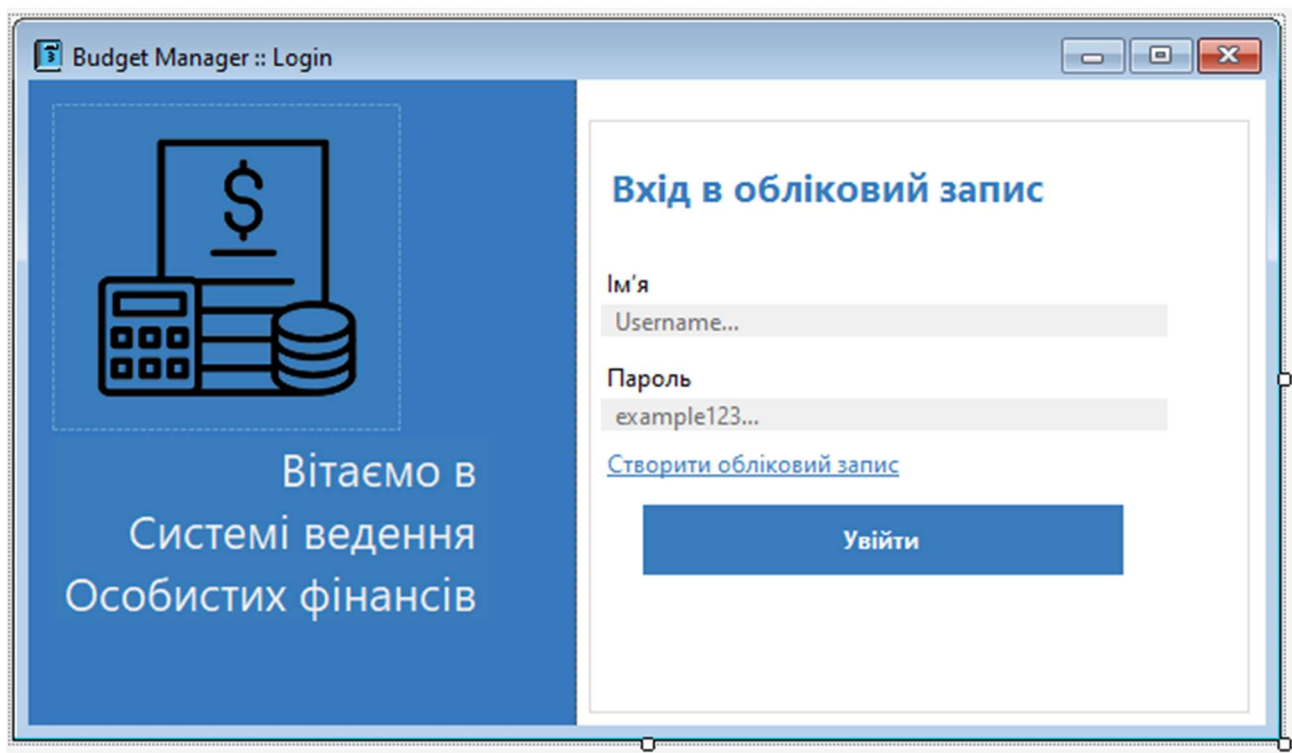


Рисунок 3.5 – Вікно авторизації користувачів

Це вікно програми представляє форму входу користувача в інформаційну систему. Вікно складається з полів для введення імені користувача та пароля, а також кнопки входу в систему і посилання для створення нового облікового запису. У конструкторі форми ініціалізуються компоненти форми, а також задаються

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 38 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

параметри вікна, такі як заборона зміни розмірів вікна і вимкнення кнопки максимізації

При натисканні кнопки "Увійти" виконується перевірка коректності введених даних за допомогою методу валідації. Якщо всі обов'язкові поля заповнені, виконується пошук користувача в базі даних за введеними іменем та паролем. Якщо користувач знайдений, поточне вікно приховується, отримується ідентифікатор користувача, і відкривається нове вікно форми, передаючи туди ідентифікатор користувача.

Якщо ж введені дані некоректні або користувач не знайдений, відображається повідомлення про помилку за допомогою `MessageBox.Show`, яке інформує користувача про неправильно введені дані.

Метод валідації перевіряє, чи обидва поля (ім'я користувача та пароль) заповнені. Для цього використовується клас, який перевіряє кожне поле і відображає повідомлення про помилки, якщо поле залишилось порожнім або заповнене некоректно.

При натисканні на посилання "Створити обліковий запис", відкривається форма реєстрації нового користувача, описана вище. Це дозволяє користувачу створити новий обліковий запис без закриття форми входу.

Ця форма входу забезпечує зручний і безпечний спосіб доступу користувачів до системи, перевіряючи коректність введених даних і надаючи можливість створення нових облікових записів.

Наступним вікном є основна форма інформаційної системи, де представлений весь функціонал програми для ведення обліку особистих витрат, вона зображена на рисунках 3.6-3.8.

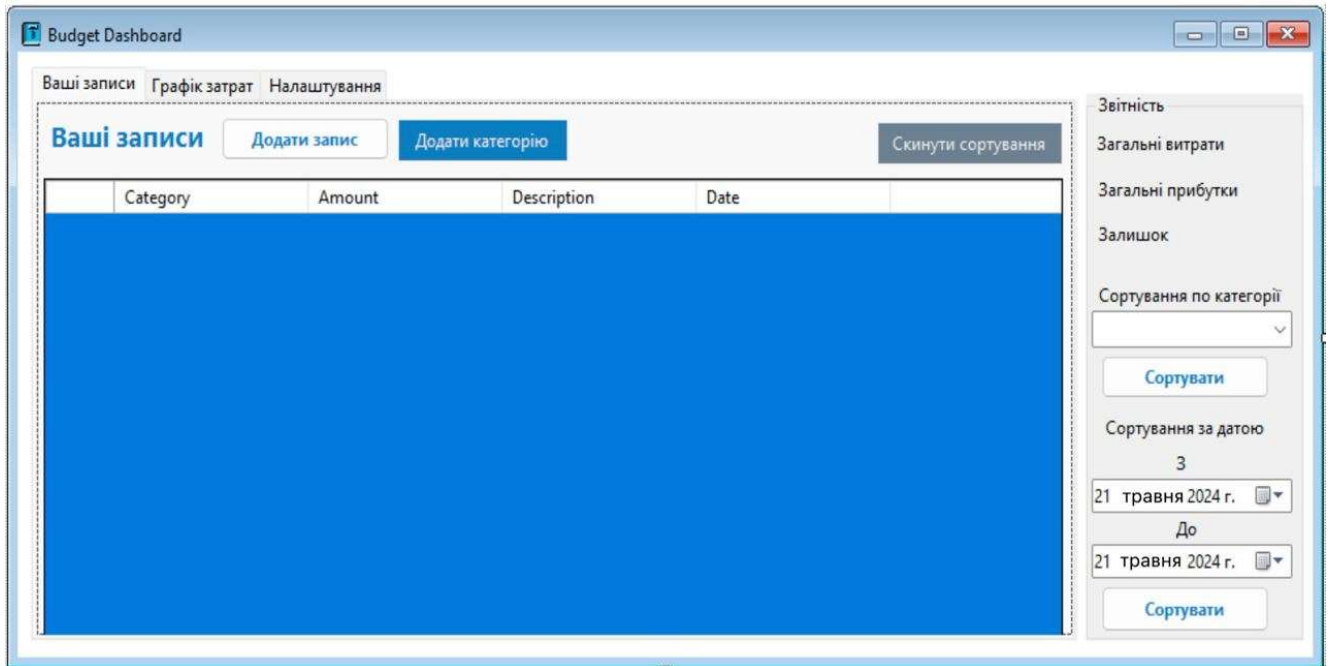


Рисунок 3.6 – Головна панель інформаційної системи вкладка Ваші записи

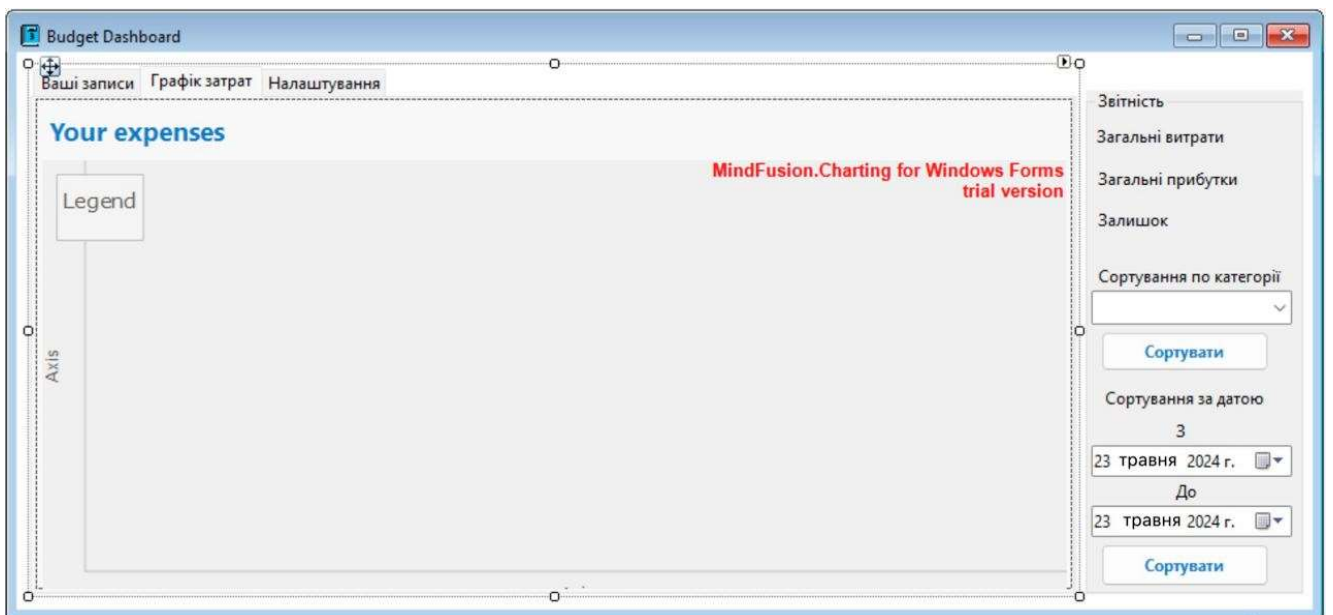


Рисунок 3.7 – Головна панель інформаційної системи вкладка Графік витрат

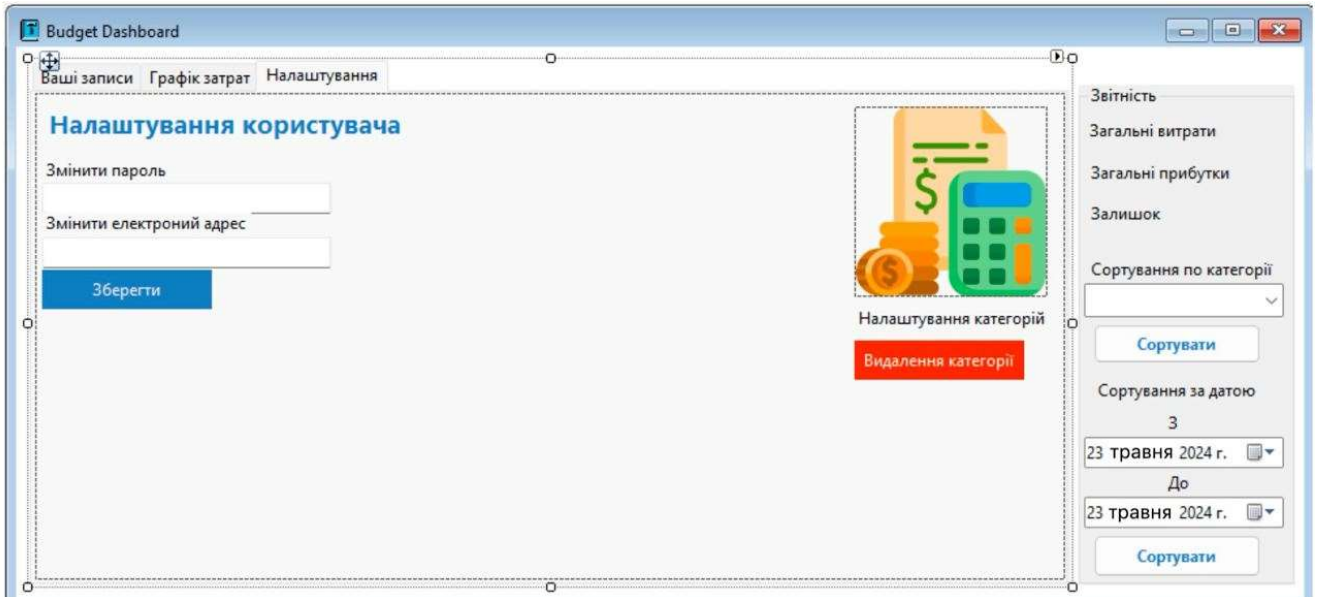


Рисунок 3.8 – Головна панель інформаційної системи вкладка Налаштування

Це вікно програми представляє головну панель користувача в системі ведення обліку особистих фінансів. Форма призначена для відображення детальної інформації про доходи і витрати користувача, а також надає інструменти для їх керування та аналізу. В конструкторі форми ініціалізуються компоненти форми, задається фіксований стиль вікна і забороняється максимізація, налаштовується таблиця, а також викликаються методи для завантаження даних.

Метод завантаження комбобоксів заповнює комбіноване вікно списком категорій з бази даних. Метод завантаження витрат завантажує і відображає загальні показники витрат і доходів користувача на відповідних мітках. Для цього використовуються запити до бази даних, щоб отримати суми витрат за різними категоріями, та обчислюється різниця між доходами і витратами, яка відображається на мітці.

Метод завантаження діаграми створює графік витрат за категоріями, використовуючи дані з таблиці. Для кожної категорії призначається випадковий колір, щоб забезпечити візуальну різноманітність графіка. Графік додається до контейнера, забезпечуючи інтерактивний і наочний спосіб аналізу витрат користувача.

Також наявний метод, який обробляє події кліків по коміркам таблиці. Якщо натиснута кнопка видалення, відображається підтвердження видалення витрати. Після підтвердження витрата видаляється з бази даних, і таблиця, графік та мітки оновлюються відповідно.

Метод завантаження головного вікна викликається при завантаженні форми і забезпечує первинне заповнення комбінованого вікна, таблиці, графіка та міток. Далі завантажує витрати користувача в таблицю, використовуючи дані з бази даних, і додає відповідні рядки до таблиці з інформацією про витрати, такі як дата, сума, опис та категорія.

Безумовно наявні методи, які обробляють події кліків на кнопках для додавання нових категорій, витрат або видалення існуючих. Після виконання відповідних дій форми, таблиця, графік та мітки оновлюються, щоб відобразити актуальні дані.

Метод збереження внесених змін дозволяє користувачу оновлювати свою інформацію, таку як електронна пошта або пароль. Після збереження змін в базі даних, користувачу відображається повідомлення про успішне оновлення, і таблиця, графік та мітки оновлюються для відображення нових даних.

Також можна побачити, що в інформаційній системі наявні засоби для фільтрації даних, вони забезпечують фільтрацію витрат в таблиці за категоріями або за датами. Це дозволяє користувачу легко аналізувати свої витрати, фільтруючи їх за певними критеріями.

Це вікно є центральною частиною системи, забезпечуючи користувача всіма необхідними інструментами для ефективного управління своїми фінансами, включаючи додавання, видалення, фільтрацію та аналіз витрат та доходів.

Наступним вікном є форма для створення категорій витрат, вона зображена на рисунку 3.9.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 42 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

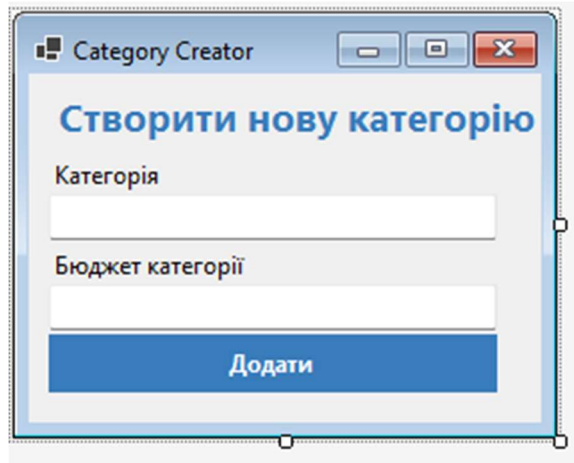


Рисунок 3.9 – Вікно для створення категорій витрат

Це вікно програми представляє форму для створення та додавання нових категорій бюджету. Форма `frmCategory` надає користувачам можливість ввести назву категорії та суму бюджету для цієї категорії. У конструкторі форми ініціалізуються компоненти форми, задається фіксований стиль вікна, що забороняє зміну розміру.

При додаванні категорії, викликається функція, що обробляє подію натискання кнопки для додавання нової категорії. Вона перевіряє введені дані за допомогою методу валідації, який здійснює аналіз введених значень у текстових полях. Якщо введені дані є валідними, здійснюється спроба перетворити введену суму бюджету у ціле число. Якщо перетворення вдається, створюється новий об'єкт з відповідними значеннями назви та суми бюджету, який потім додається до бази даних. Після успішного збереження нової категорії в базу даних, встановлюється результат діалогу DialogResult.OK, що сигналізує про успішне завершення операції. Якщо введена сума бюджету є недійсною, користувачу відображається повідомлення про помилку із проханням ввести коректне ціле число.

Метод валідації забезпечує валідацію введених даних у текстових полях за допомогою класу, перевіряючи наявність необхідних значень у полях для назви категорії та суми бюджету. Також наявний метод, який обробляє подію натискання клавіш у текстовому полі для суми бюджету. Він дозволяє вводити лише цифрові

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 43 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

значення та управляючі символи, забезпечуючи тим самим коректний формат введених даних.

Ця форма забезпечує простий і інтуїтивно зрозумілий інтерфейс для створення нових категорій бюджету, дозволяючи користувачам легко додавати нові категорії з відповідними бюджетними значеннями.

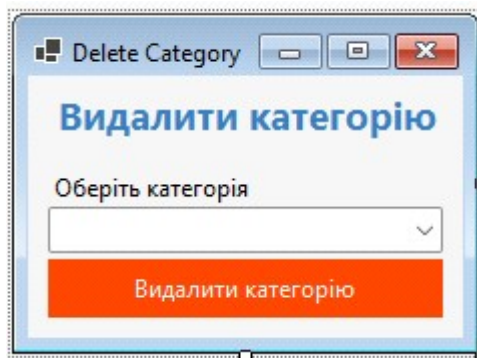


Рисунок 3.10 – Вікно для видалення категорій витрат

Це вікно програми представляє форму для видалення існуючих категорій бюджету. Форма надає користувачам можливість вибрати та видалити певну категорію зі списку доступних категорій бюджету. У конструкторі форми ініціалізуються компоненти форми, задається фіксований стиль вікна, що забороняє зміну розміру, а також вимикається можливість максимізації вікна. Під час ініціалізації заповнюється комбінований список існуючими категоріями бюджету з бази даних.

Далі система обробляє подію натискання кнопки видалення категорії. Спочатку перевіряється, чи вибрана категорія зі списку. Якщо категорія не вибрана, користувачу відображається попередження з проханням вибрати категорію для видалення. Якщо категорія вибрана, користувачу відображається підтвердження на видалення з відповідним повідомленням. У випадку підтвердження користувачем видалення, здійснюється спроба видалити вибрану категорію з бази даних. Якщо вибрана категорія є валідною, вона видаляється з бази даних, і зміни зберігаються. Після успішного видалення користувачу відображається повідомлення про

успішне видалення, а список категорій у комбінованому списку оновлюється для відображення актуальних даних.

У разі, якщо вибрана категорія є недійсною або під час видалення виникає помилка, користувачу відображається відповідне повідомлення про помилку із деталями.

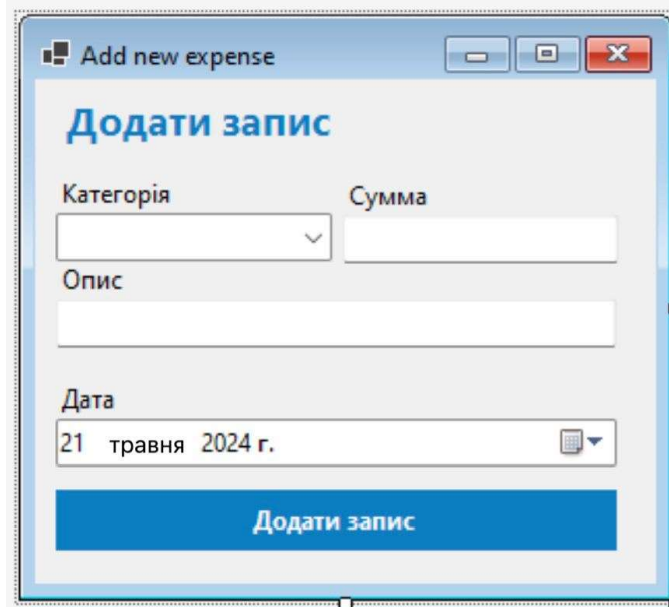


Рисунок 3.11 – Вікно для додавання витрат/доходів

Це вікно програми представляє форму для додавання нових витрат до системи бюджету. Форма дозволяє користувачам ввести інформацію про нову витрату, вибрати відповідну категорію та зберегти цю інформацію в базу даних. При створенні форми, конструктор приймає дані користувача, що ідентифікує його та аналізує хто додає витрату/дохід. Також під час ініціалізації форми заповнюється комбінований список доступними категоріями бюджету з бази даних.

Метод завантаження категорій, переносить список категорій бюджету при завантаженні форми, що дозволяє користувачу вибрати одну з існуючих категорій для нової витрати.

При натисненні конпки обробляється подія для додавання витрати. Спочатку здійснюється перевірка на валідність введених даних за допомогою методу, який перевіряє, чи заповнені всі необхідні поля. Якщо дані валідні, вибирається категорія витрати з комбінованого списку, а також перевіряється правильність

введеної суми витрати. Якщо сума є валідним числовим значенням, створюється новий об'єкт витрати з відповідними полями, після чого нова витрата додається в базу даних і зберігаються зміни. Якщо витрата успішно додана, діалогове вікно закривається з результатом `DialogResult.OK`.

У випадку, якщо введена сума витрати є недійсною (не числове значення), користувачу відображається повідомлення про помилку із закликом ввести коректну суму.

Усі представлені вікна програми працюють разом, щоб створити зручну та функціональну інформаційну систему для управління особистими фінансами. Вікно входу забезпечує автентифікацію користувача перед доступом до основного функціоналу. Головне вікно надає користувачу огляд його витрат і доходів, включаючи графічне представлення даних і можливість керування витратами. Вікно додавання категорій дозволяє користувачам створювати нові категорії бюджету, які потім можуть бути використані для класифікації витрат. Вікно видалення категорій, забезпечує можливість видалення небажаних категорій з бази даних. Вікно додавання витрат дозволяє користувачам вводити нові витрати, зберігаючи їх у базі даних з відповідними категоріями та описами.

3.2 Опис процесу створення баз даних.

База даних є структурованою сукупністю інформації. Ці дані можуть включати все, від простого списку покупок до повного каталогу експонатів художньої галереї чи величезної кількості інформації в корпоративній мережі. Для запису, вибірки та обробки даних, збережених у комп'ютерній базі даних, потрібна система управління базою даних.

MySQL є системою керування базами даних. У реляційній базі даних інформація зберігається не в одній суцільній структурі, а в окремих таблицях, що забезпечує швидкість і гнучкість доступу до даних. Таблиці пов'язані між собою за допомогою відносин, що дозволяє об'єднувати дані з різних таблиць під час

виконання запиту. SQL, як частина MySQL, можна охарактеризувати як мову структурованих запитів і найбільш поширену стандартну мову для доступу до баз даних.

Таблиця є основним об'єктом для зберігання даних у реляційній базі даних. Вона складається з рядків і стовпців, які містять дані, займає фізичний простір у базі даних і може бути як постійною, так і тимчасовою. Поле, яке також називають стовпцем у реляційній базі даних, є частиною таблиці, що має певний тип даних. Кожна таблиця в базі даних повинна містити принаймні один стовпець. Рядок даних, або запис, складається з полів, що містять дані одного запису таблиці.

Проектуємо архітектуру баз даних (БД). Всі архітектурні схеми представляються за допомогою нотації UML та, за необхідності, доповнюються текстовими описами. Під час проектування архітектури БД було обрано реляційну модель даних для її реалізації, і виділені відношення представлені на наступних рисунках.

Процес створення бази даних розпочинається з аналізу вимог користувачів. На цьому етапі важливо збирати та ретельно аналізувати інформацію щодо того, які дані потрібні та як вони будуть використовуватися. Після цього приходить час проектування бази даних, коли визначається структура таблиць, відносини між ними та індекси.

Нормалізація є наступним важливим кроком. Це процес оптимізації бази даних для зменшення дублювання даних та забезпечення ефективності. Потім розробляється схема бази даних, яка включає в себе структуру таблиць та обмеження цілісності даних.

Після цього вибирається система керування базами даних (СКБД), що найкращим чином відповідає вимогам проекту. Створюються таблиці та інші об'єкти бази даних, а потім їх заповнюють даними.

Тестування відіграє ключову роль у перевірці працездатності, цілісності та продуктивності бази даних. І, нарешті, база даних впроваджується в роботу, а потім

підтримується та модифікується відповідно до потреб користувачів та змін в бізнес-процесах.

Деталізуємо обрані варіанти використання і модулі системи з використанням діаграми модулів системи рисунок 3.12.

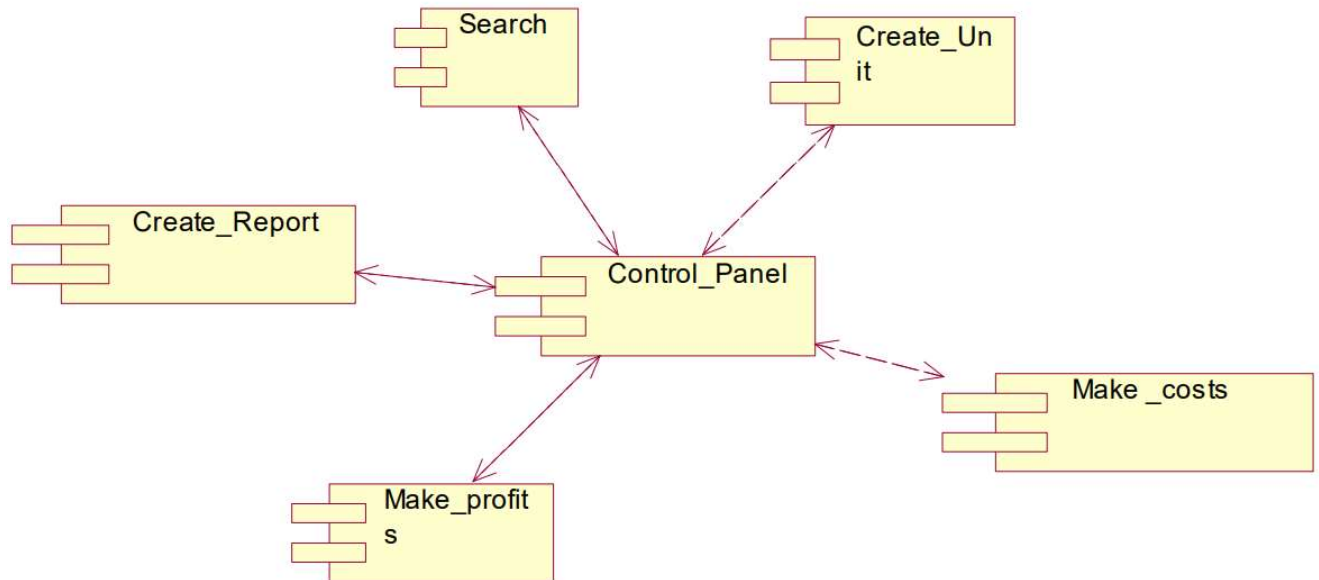


Рисунок 3.12 – Структурна схема інформаційної системи системи для ведення обліку особистих витрат

Для втілення обраної системи ми використовуємо об'єктно-орієнтований підхід у проектуванні та програмуванні. Таким чином, наступним етапом є визначення діаграми класів, яка представлена на рисунку 3.13.

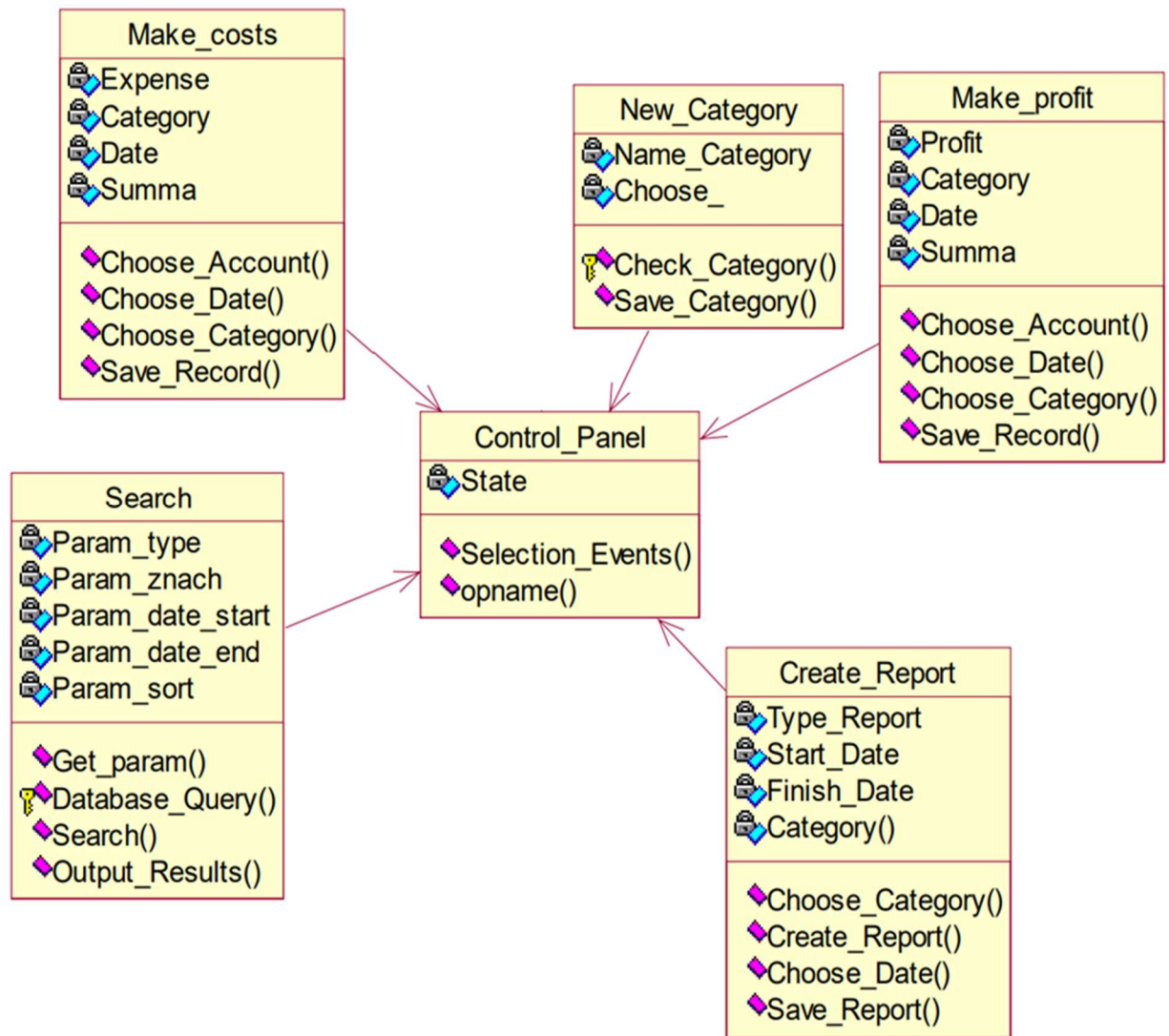


Рисунок 3.13 – Діаграма класів інформаційної системи

Проаналізувавши усі необхідні модулі і вимоги інформаційної системи було створено структуру бази даних зображену на рисунку 3.14.

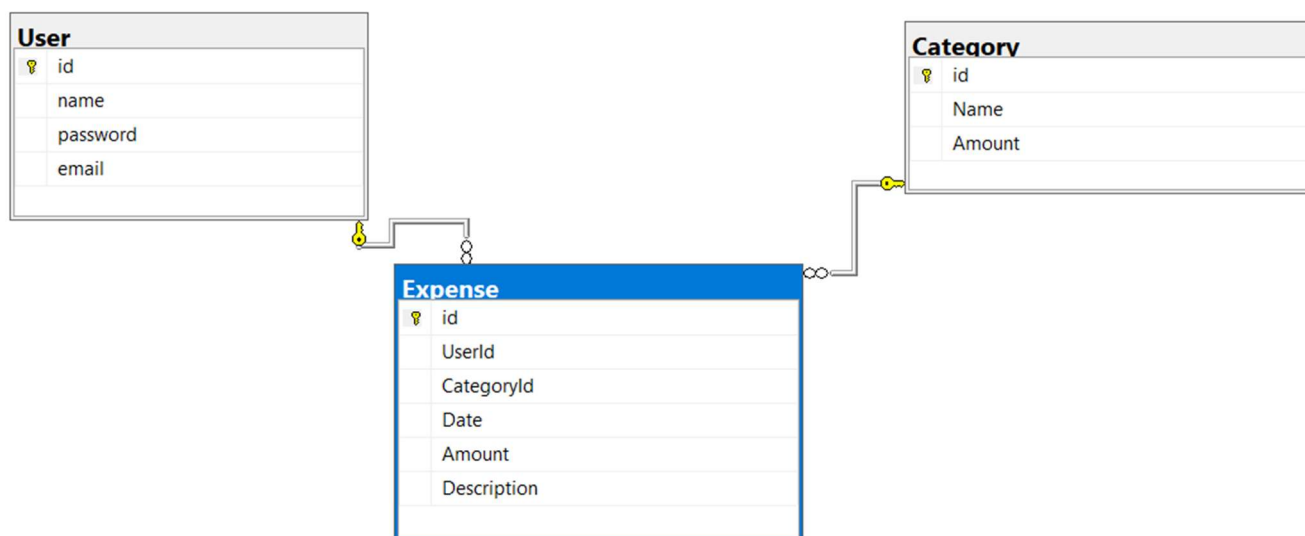


Рисунок 3.14 – Структура бази даних

Діаграма бази даних складається з трьох основних таблиць: User, Category і Expense.

User (Користувач):

- id (Primary Key) – унікальний ідентифікатор користувача;
- name – ім'я користувача;
- password – пароль користувача;
- email – електронна адреса користувача.

Category (Категорія):

- id (Primary Key) – унікальний ідентифікатор категорії;
- Name – назва категорії;
- Amount – сума бюджету, пов'язана з цією категорією;

Expense (Витрати):

- id (Primary Key) – унікальний ідентифікатор витрат;
- UserId (Foreign Key) – ідентифікатор користувача;
- CategoryId (Foreign Key) – ідентифікатор категорії витрат;
- Date – дата здійснення витрат;

- Amount – сума витрат;
- Description – опис витрат.

Між таблицями наявні наступні зв'язки, один до багатьох між User та Expense, тобто один користувач може мати багато записів про витрати, і цей зв'язок встановлено через атрибут UserId в таблиці Expense, який є зовнішнім ключем, що посилається на id в таблиці User.

Один до багатьох між Category та Expense, тобто одна категорія може бути використана у багатьох записах про витрати, і цей зв'язок встановлено через атрибут CategoryId в таблиці Expense, який є зовнішнім ключем, що посилається на id в таблиці Category.

Foreign Key Constraints забезпечують цілісність даних, гарантують, що кожен запис у таблиці Expense буде посилатися на існуючий запис у таблицях User та Category. Користувачі можуть реєструватися та входити в систему, додаючи свої витрати, які зберігаються в таблиці Expense. Категорії допомагають організувати витрати за різними типами, такими як харчування, транспорт, розваги тощо.

Витрати записуються з датою, сумою та описом, що дозволяє користувачам відстежувати свої фінанси. Наприклад, користувач додає новий запис витрат: обирає категорію, вводить суму витрат, дату та опис. Цей запис зберігається в таблиці Expense, де зберігається також посилання на користувача, який здійснив витрати, і на категорію витрат. Ця структура дозволяє ефективно організовувати та керувати даними про витрати користувачів, забезпечуючи зручність у відстеженні та аналізі особистих.

| | Column Name | Data Type | Allow Nulls |
|---|-------------|--------------|--------------------------|
| 🔑 | id | int | <input type="checkbox"/> |
| | UserId | int | <input type="checkbox"/> |
| | CategoryId | int | <input type="checkbox"/> |
| | Date | date | <input type="checkbox"/> |
| | Amount | int | <input type="checkbox"/> |
| ▶ | Description | nvarchar(50) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.15 – Структура таблиці Expense

Таблиця має наступну структуру:

- стовпчик id типу int не допускає значення NULL і є первинним ключем, що унікально ідентифікує кожен запис;
- стовпчик UserId типу int також не допускає значення NULL і є зовнішнім ключем, який посилається на ідентифікатор користувача, пов'язаного з цим записом витрат;
- стовпчик "CategoryId типу int не допускає значення NULL і є зовнішнім ключем, що посилається на ідентифікатор категорії витрат;
- стовпчик Date типу date не допускає значення NULL і містить дату здійснення витрат;
- стовпчик Amount типу int не допускає значення NULL і містить суму витрат;
- стовпчик Description типу nvarchar(50) не допускає значення NULL і містить опис витрат, обмежений до 50 символів.

Усі поля обов'язкові для заповнення. Таблиця призначена для зберігання інформації про витрати користувачів, включаючи суму, дату, опис, а також посилання на відповідного користувача та категорію витрат.

| | Column Name | Data Type | Allow Nulls |
|---|-------------|--------------|--------------------------|
| 🔑 | id | int | <input type="checkbox"/> |
| | Name | nvarchar(50) | <input type="checkbox"/> |
| ▶ | Amount | int | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.16 – Структура таблиці Category

Таблиця має наступну структуру:

- стовпчик id типу int не допускає значення NULL і є первинним ключем, що унікально ідентифікує кожен запис;
- стовпчик Name типу nvarchar(50) не допускає значення NULL і містить назву категорії, обмежену до 50 символів;
- стовпчик Amount типу int також не допускає значення NULL і містить суму, пов'язану з цією категорією.

Усі поля є обов'язковими для заповнення. Таблиця призначена для зберігання інформації про категорії витрат, включаючи їх назви та відповідні суми.

| | Column Name | Data Type | Allow Nulls |
|---|-------------|--------------|--------------------------|
| 🔑 | id | int | <input type="checkbox"/> |
| | name | nvarchar(50) | <input type="checkbox"/> |
| | password | nvarchar(50) | <input type="checkbox"/> |
| | email | nvarchar(50) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.17 – Структура таблиці User

Таблиця має наступну структуру:

- стовпчик id типу int не допускає значення NULL і є первинним ключем, що унікально ідентифікує кожен запис;
- стовпчик Name типу nvarchar(50) не допускає значення NULL і містить назву категорії, обмежену до 50 символів;

- стовпчик Password типу nvarchar(50) не допускає значення NULL і містить паролі користувачі, обмежену до 50 символів;
- стовпчик Email типу nvarchar(50) не допускає значення NULL і містить адреси електронної пошти користувачів, обмежену до 50 символів.

Усі поля є обов'язковими для заповнення. Таблиця призначена для зберігання інформації про користувачів, включаючи їх електронні адреси.

3.3 Висновки

Інформаційна система дозволяє користувачам додавати нові категорії витрат, перевіряючи валідність введених даних, таких як назва категорії та сума бюджету. Також надає інтерфейс для видалення існуючих категорій з підтвердженням дії, що запобігає випадковим видаленням, дозволяє додавати нові витрати, вибираючи відповідну категорію та вказуючи деталі витрат, такі як сума, опис та дата.

Таблиці бази даних User, Category та Expense структуровані для зберігання інформації про користувачів, категорії витрат та самі витрати.

Ця структура забезпечує ефективне зберігання та управління даними, дозволяючи користувачам додавати, видаляти та відслідковувати витрати, організовані за категоріями. Інтерфейси користувача забезпечують простоту та зручність взаємодії з системою, зберігаючи цілісність та валідність даних.

ВИСНОВКИ

Створення інформаційної системи для ведення обліку особистих витрат вимагає ретельного проектування та реалізації як інтерфейсів користувача, так і структури бази даних. Розробивши описані вікна програм та структуру бази даних, можна зробити декілька важливих висновків, які дозволяють зрозуміти, як ця система функціонує та забезпечує потреби користувачів.

Інтерфейси користувача грають ключову роль у забезпеченні ефективної взаємодії з системою. Кожне вікно програми виконує специфічні функції, необхідні для ведення обліку витрат.

frmCategory - це форма, яка дозволяє користувачам додавати нові категорії витрат. Вона забезпечує перевірку валідності введених даних, таких як назва категорії та сума бюджету. Якщо дані введено неправильно, користувач отримує повідомлення про помилку. Це дозволяє зберегти цілісність даних в базі.

frmDelete - форма для видалення існуючих категорій. Вона включає підтвердження дії, щоб запобігти випадковим видаленням. Якщо користувач намагається видалити категорію, з'являється запит на підтвердження, що забезпечує додатковий рівень захисту.

frmExpense - форма для додавання нових витрат. Користувач може вибрати відповідну категорію витрат та вказати деталі, такі як сума, опис та дата. Форма також перевіряє валідність введених даних, щоб уникнути помилок та невідповідностей.

Структура бази даних є критично важливою для забезпечення належного зберігання та управління даними. База даних складається з трьох основних таблиць: User, Category та Expense.

User - таблиця для зберігання інформації про користувачів. Вона містить поля для ідентифікатора користувача (id), його імені (name), пароля (password) та електронної пошти (email). Це дозволяє ідентифікувати кожного користувача та забезпечити безпеку доступу до системи.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 55 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Category - таблиця для зберігання категорій витрат. Вона включає поля для ідентифікатора категорії (id), назви категорії (Name) та суми бюджету (Amount). Це дозволяє класифікувати витрати та встановлювати бюджетні обмеження.

Expense - таблиця для зберігання даних про витрати. Вона містить поля для ідентифікатора витрати (id), ідентифікатора користувача (UserId), ідентифікатора категорії (CategoryId), дати витрати (Date), суми витрати (Amount) та опису (Description). Це забезпечує детальний облік кожної витрати, включаючи інформацію про те, хто, коли і на що витратив кошти.

Загалом, інформаційна система для ведення обліку особистих витрат забезпечує зручний та інтуїтивний інтерфейс для користувачів, а також надійну структуру бази даних для зберігання та управління даними. Форми для додавання, видалення та обліку витрат дозволяють користувачам ефективно управляти своїми фінансами, а ретельна перевірка введених даних гарантує їх валідність та цілісність. Взаємозв'язок між таблицями бази даних забезпечує зберігання повної інформації про витрати, включаючи дані про користувачів та категорії. Така система дозволяє легко відстежувати фінансові потоки, планувати бюджет та приймати обґрунтовані фінансові рішення.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. 56 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Сервіси для контролю особистих фінансів. URL: <https://kosht.media/servisy-dlia-kontroliu-osobystykh-finansiv/> (дата звернення: 20.04.2024).

2. Як правильно вести свої фінанси. URL: <https://blog.agrokebety.com/yak-pravyлно-vesty-svoy-finansy> (дата звернення: 22.04.2024).

3. Особисті фінанси. URL: <https://www.ukraine-lifehacker.com/osobysti-finansy-tse> (дата звернення: 17.04.2024).

4. Контроль особистих фінансів: обговорення. URL: <https://dou.ua/forums/topic/41422/> (дата звернення: 13.05.2024).

5. Як заощадити під час війни: поради. URL: <https://www.epravda.com.ua/publications/2023/02/22/697326/> (дата звернення: 8.05.2024).

6. Emma: UK alternative to Mint. URL: <https://emma-app.com/blog/2017/07/02/uk-alternative-to-mint/> (дата звернення: 11.05.2024).

7. Mint: Budget & Bills, Finance. URL: https://apps.mob.org.ua/apps/mint_budget_bills_finance.html (дата звернення: 2.05.2024).

8. Fintech: Як використовувати сучасні фінансові технології. URL: <https://www.epravda.com.ua/projects/fintech/2019/01/28/644615/> (дата звернення: 20.04.2024).

9. You Need A Budget (YNAB): Review. URL: <https://www.pcmag.com/reviews/ynab> (дата звернення: 25.04.2024).

10. Block21 Wallet. URL: <https://block21-wallet.en.aptoide.com/app> (дата звернення: 25.04.2024).

11. BudgetBakers: Manage your budget online. URL: <https://web.budgetbakers.com/login> (дата звернення: 25.04.2024).

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КВРІСТ 101042.21.01.04 ПЗ | Арк. 57 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

12. JetBrains IntelliJ IDEA: Product description. URL: <https://itpro.ua/product/jetbrains-intellij-idea/?tab=description> (дата звернення: 25.04.2024).

13. Discover IntelliJ IDEA: User guide. URL: <https://www.jetbrains.com/help/idea/discover-intellij-idea.html> (дата звернення: 01.05.2024).

14. Microsoft Visual Studio: Documentation. URL: <https://learn.microsoft.com/uk-ua/visualstudio/windows/?view=vs-2022&preserve-view=true> (дата звернення: 01.05.2024).

15. Основи роботи з Visual Studio. URL: <https://learn.ztu.edu.ua/mod/page/view.php?id=9974> (дата звернення: 05.05.2024).

16. Структура інформаційних систем. URL: <https://studfile.net/preview/5994722/page:7/> (дата звернення: 05.05.2024).

17. PyCharm: Інструмент для розробки на Python. URL: <https://foxminded.ua/pycharm-tse/> (дата звернення: 07.05.2024).

18. Лекція 1: Архітектура програмного забезпечення. URL: https://org2.knuba.edu.ua/pluginfile.php/28591/mod_resource/content/1/ЛЕКЦІЯ_1_АП%20ПЗ_Архітектура_ПЗ.pdf (дата звернення: 10.05.2024).

19. Конспект лекцій: Інформаційні системи. URL: <https://dspace.wunu.edu.ua/jspui/bitstream/316497/24194/1/опорний%20конспект%20лекцій.pdf> (дата звернення: 10.05.2024).

20. Основи розробки інформаційних систем. URL: <https://studfile.net/preview/4452595/page:8/> (дата звернення: 19.04.2024).

21. Управління вимогами: функціональні та нефункціональні вимоги. URL: <https://visuresolutions.com/uk/requirements-management-traceability-guide/functional-vs-non-functional-requirements/> (дата звернення: 19.04.2024).

22. Дубовик В. В. Інформаційні системи і технології у фінансовому менеджменті. Київ: КНЕУ, 2019. 238 с.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КВРІСТ 101042.21.01.04 ПЗ | Арк. 58 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

23. Kim Y. Efficient Financial Tracking Systems. Cambridge: Cambridge University Press. 2019. 260 p.
24. Johnson L. Managing Personal Expenses in the Digital Age. Oxford: Oxford University Press. 2019. 298 p.
25. Brown J. Mastering Personal Finances with Technology. London: Springer. 2021. 295 p.
26. Карпенко О. І., Білик, С. В. Інформаційні системи та бази даних. Тернопіль: ТНЕУ, 2020. 244 с.
27. Савченко П. П. Основи економічної інформатики. Одеса: ОНУ, 2018. 226 с.
28. Grant S. Personal Budgeting and Expense Management. New York: Palgrave Macmillan. 2021. 282 p.
29. Левченко О. М. Сучасні інформаційні технології в обліку та аудиті. Харків: ХНЕУ, 2020. 272 с.
30. Smith R. Financial Management Systems: A Practical Guide. Chicago: McGraw-Hill. 2020. 320 p.
31. Murphey D. Digital Personal Finance. New York: Routledge. 2019. 276 p.
32. Іванов Д. О., Коваль, Ю. В. Вступ до комп'ютерних наук. Харків: ХНУРЕ, 2020. 312 с.
33. Петренко Л. М. Аналіз і управління витратами підприємства. Київ: КНЕУ, 2019. 254 с.
34. Allen P. The Future of Personal Finance Management. Edinburgh: Elsevier. 2020. 240 p.
35. Романенко І. А. Сучасні технології в управлінні фінансами. Дніпро: ДНУ, 2021. 198 с.
36. Михайленко С. В. Управління особистими фінансами. Київ: НТУ, 2021. 220 с.
37. Davis K. Automated Personal Expense Tracking. Boston: MIT Press. 2021. 275 p.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КВРІСТ 101042.21.01.04 ПЗ | Арк. 59 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

38. Синявський В. А. Інформаційні системи та технології. Київ: КНЕУ, 2019. 256 с.
39. Ellis C. Expense Tracking: Simplifying Your Finances. San Francisco: No Starch Press. 2020. 310 p.
40. Zehnder, C. Personal Finance for Dummies. Hoboken: Wiley. 2018. 384 p.
41. Пономаренко В. С. Інформаційні системи в економіці. Львів: ЛНУ, 2021. 284 с.
42. Печерська І. І. Інформаційні системи в управлінні фінансами. Київ: НАУ, 2019. 245 с.
43. Самойленко А. С. Автоматизація обліку витрат підприємства. Харків: ХНЕУ, 2020. 216 с.
44. Ткаченко М. П. Основи фінансового менеджменту. Львів: ЛНУ, 2021. 234 с.
45. Долгих О. В. Інформаційні технології в економіці. Одеса: ОНУ, 2020. 253 с.
46. Савченко П. П. Основи економічної інформатики. Одеса: ОНУ, 2018. 226 с.
47. Іванов В. В. Управління фінансовими потоками. Дніпро: ДНУ, 2019. 201 с.
48. Караванова К. В. Інформаційні системи та технології в економіці. Київ: КНЕУ, 2020. 287 с.
49. Пономаренко В. С. Інформаційні системи в обліку та аудиті. Львів: ЛНУ, 2019. 267 с.
50. Гончарук В. П. Сучасні інформаційні технології в управлінні фінансами. Київ: НАУ, 2021. 222 с.
51. Петренко Л. М. Аналіз і управління витратами підприємства. Київ: КНЕУ, 2019. 254 с.
52. Левченко О. М. Сучасні інформаційні технології в обліку та аудиті. Харків: ХНЕУ, 2020. 272 с.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------------|
| | | | | | КВРІСТ 101042.21.01.04 ПЗ | Арк. 60 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

53. Дубовик В. В. Інформаційні системи і технології у фінансовому менеджменті. Київ: КНЕУ, 2019. 238 с.

| | | | | | | |
|-----|------|----------|--------|------|---------------------------|------|
| | | | | | КвРІСТ 101042.21.01.04 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 61 |

Додаток А (обов'язковий)

Копія креслення «Інтерфейс інформаційної системи»

Вітаємо в Системі ведення Особистих Фінансів

Вхід в обліковий запис

Ім'я

Пароль

Створити обліковий запис

Увійти

Додати нову категорію

Категорія

Бюджет категорії

Додати

Додати запис

Категорія

Опис

Сума

Дата

Додати запис

Видалити категорію

Оберіть категорію

Видалити категорію

Ваші записи

Category Amount Description Date

| | | | |
|---------------|--|--|--|
| [Empty Table] | | | |
|---------------|--|--|--|

Налаштування користувача

Змінити пароль

Змінити електронний адрес

Зберегти

Your expenses

Звітність

Загальні витрати

Загальні прибутки

Залишок

Сортування по категорії

Сортувати

Сортування за датою

23 травень 2024 г. До

23 травень 2024 г. Сортувати

Налаштування категорій

Загальні витрати

Загальні прибутки

Залишок

Сортування по категорії

Сортувати

Сортування за датою

23 травень 2024 г. До

23 травень 2024 г. Сортувати

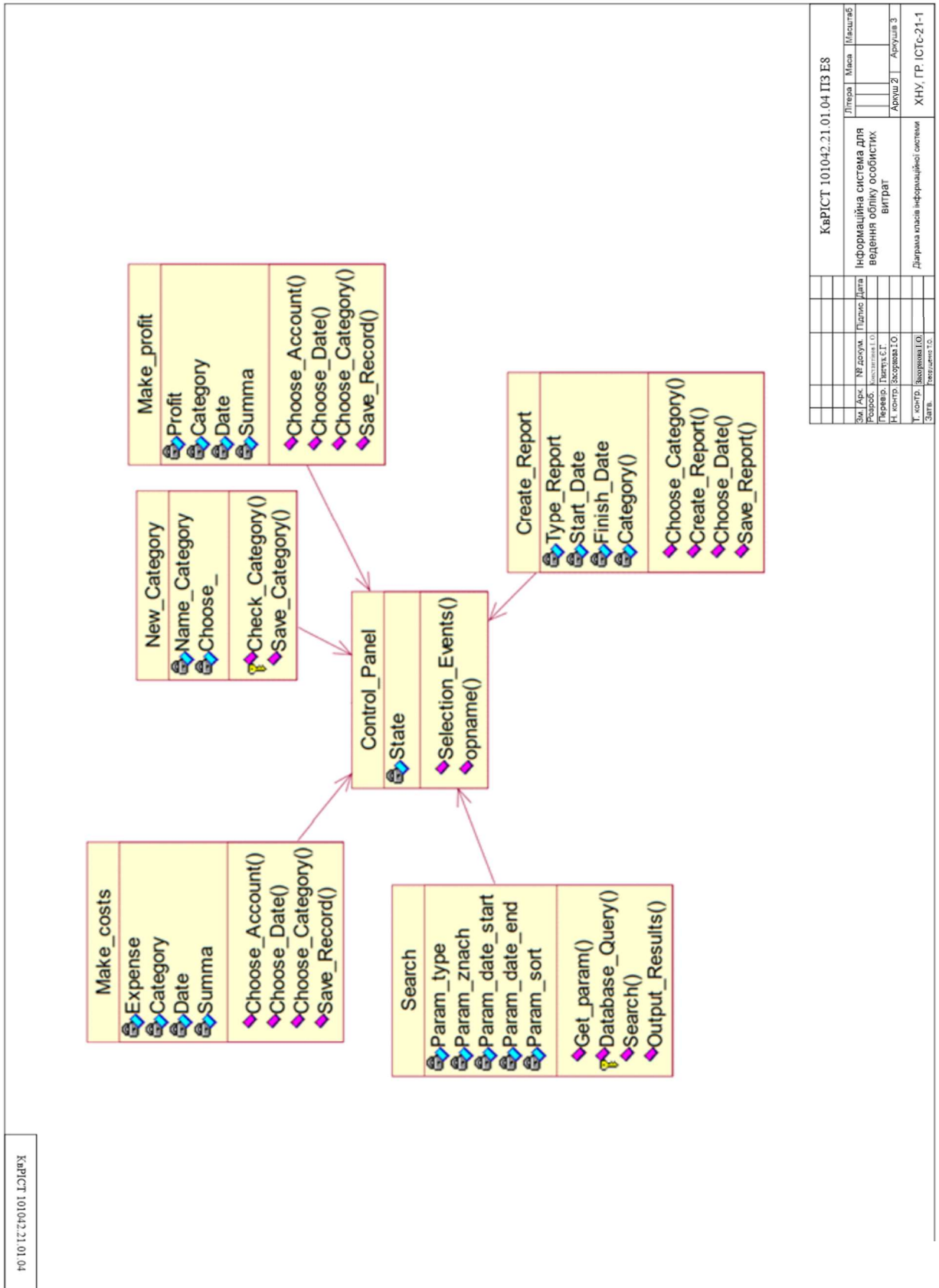
КАРІСТ 101042.21.01.04

КАРІСТ 101042.21.01.04 ES

| | | | |
|--|---------|-----------|---------|
| Інформаційна система для ведення обліку особистих витрат | Літера | Маса | Масштаб |
| Інтерфейс програми | ХНУ, ГР | ІСТС-21-1 | |

Додаток Б (обов'язковий)

Копія креслення «Діаграма класів інформаційної системи»

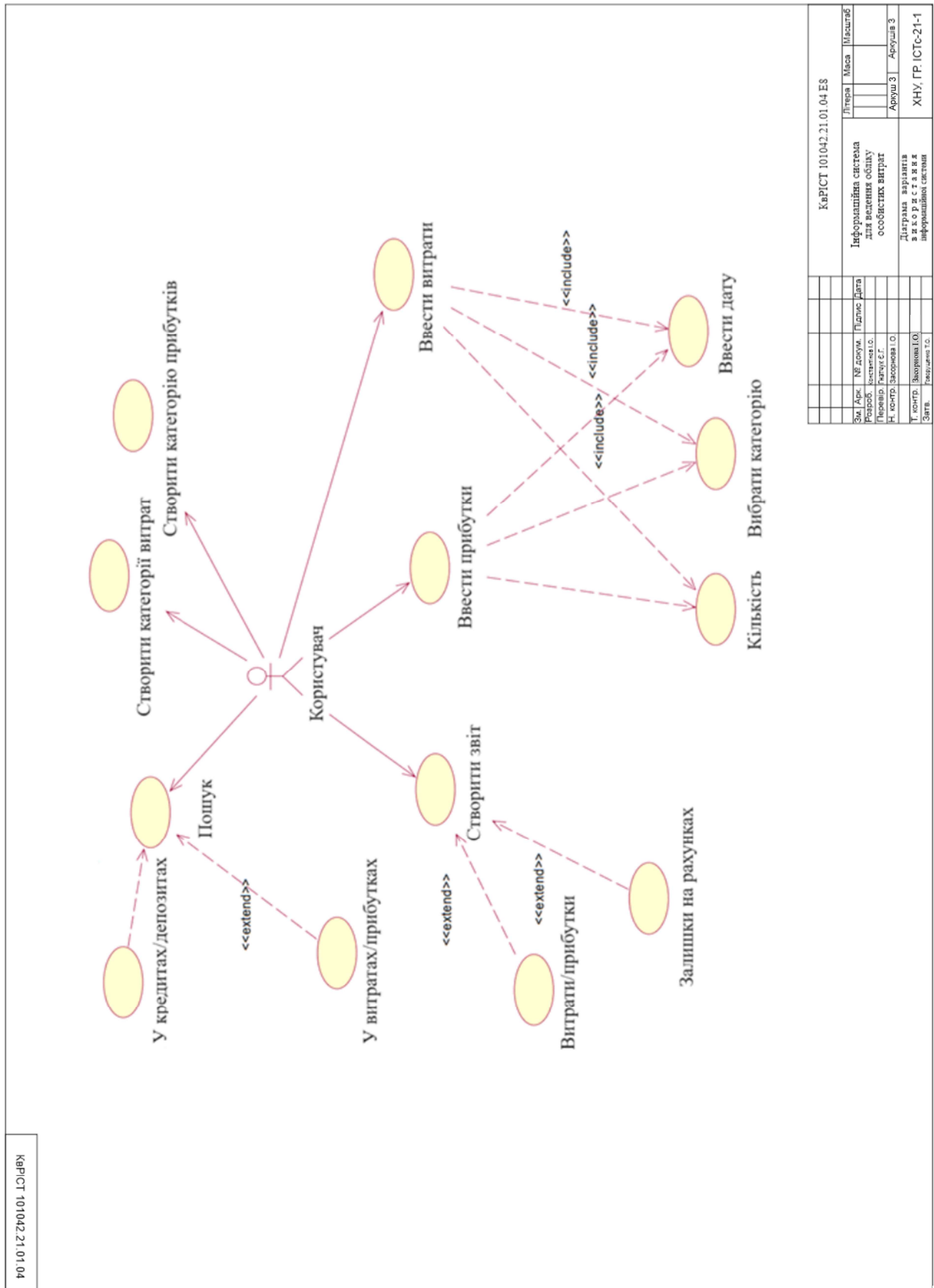


КвРІСТ 101042.21.01.04

| | | | | | |
|--|-----------------|---------|------|--|--|
| КвРІСТ 101042.21.01.04 ПЗ ЕБ | | | | | |
| Літера | Місяц | Масштаб | | | |
| Зм. Арк. | № докум. | Підпис | Дата | | |
| Розроб. | Виконавець І.О. | | | | |
| Перевір. | Твердить С.Г. | | | | |
| Н. контр. | Зборює І.О. | | | | |
| Т. контр. | Виконує І.О. | | | | |
| Затв. | Розглядає І.О. | | | | |
| Інформаційна система для ведення обліку особистих витрат | | | | | |
| Діаграма класів інформаційної системи | | | | | |
| ХНУ, ГР. ІСТС-21-1 | | | | | |

Додаток В (обов'язковий)

Копія креслення «Діаграма варіантів використання інформаційної системи»



КВРІСТ 101042.21.01.04

| КВРІСТ 101042.21.01.04 ES | | | |
|--|---------------|---------|-------------------|
| Літера | Маса | Масштаб | |
| | | | |
| Інформаційна система для ведення обліку особистих витрат | | | |
| Зм. / Арк. | № докум. | Підпис | Дата |
| Розроб. | Виконавець | | |
| Перевір. | Розробч. ЗГ | | |
| Р. констр. | Затверджен. О | | |
| У констр. | Визначено. Ю | | |
| Затв. | Розробч. ІС | | |
| | | | ХНУ, ГР. ІСТ-21-1 |

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016333085

Дата перевірки:
07.06.2024 21:04:04 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
07.06.2024 21:53:06 EEST

ID користувача:
100005591

Назва документа: **Константинов_Інформаційна система для ведення обліку особистих витрат**

Кількість сторінок: 68 Кількість слів: 9985 Кількість символів: 80975 Розмір файлу: 1.51 MB ID файлу: 1016133153

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

7.89% Схожість

Найбільша схожість: 2.37% з Інтернет-джерелом (<http://dspace.kntu.kr.ua/jspui/retrieve/16942/%D0%A2%D0%9F%D0%9>).

6.92% Джерела з Інтернету 204 Сторінка 70

2.28% Джерела з Бібліотеки 105 Сторінка 71

0.79% Цитат

Цитати 3 Сторінка 72

Посилання 1 Сторінка 72

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 1

Підозріле форматування 13 сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилко в документах: 6%**

| | | | | |
|--|----------|---------|-----------------------------|---------|
| ID: 129101 Назва: БКР Інформаційна система для ведення обліку особистих витрат Додано в БД: 2024-06-07 Автора: І. О. Константинов Керівники: Є. Г. Гнатчук Консультанти: Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 70973 | 610 | 2287 (3%) | 31 (5%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |
| | | | |

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Константинов Іван Олександрович

Тема: Інформаційна система для ведення обліку особистих витрат

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 61

1. Короткий зміст роботи та прийнятих рішень: Розроблено інформаційну систему для ведення обліку особистих витрат, яка забезпечує зручний та інтуїтивний інтерфейс для користувачів, а також надійну структуру бази даних для зберігання та управління даними. Форми для додавання, видалення та обліку витрат дозволяють користувачам ефективно управляти своїми фінансами, а ретельна перевірка введених даних гарантує їх валідність та цілісність. Взаємозв'язок між таблицями бази даних забезпечує зберігання повної інформації про витрати, включаючи дані про користувачів та категорії. Така система дозволяє легко відстежувати фінансові потоки, планувати бюджет та приймати обґрунтовані фінансові рішення.

2. Висновок про відповідність роботи дипломному завданню Дипломна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі було проаналізовано предметну область що стосується керування фінансами і фінансовою діяльністю індивідуальної особи або домогосподарства. Основною метою такої систем виступає систематичне відстеження та аналіз особистих витрат з метою керування бюджетом, ефективним управлінням фінансами та досягненням фінансових цілей. Опираючись на аналіз, було сформовано перелік основних вимог до інформаційної системи для ведення обліку особистих витрат.

В другому було проведено проектування та розроблена архітектура інформаційної системи. Розглянуті функційні, нефункційні, а також користувацькі вимоги.

В третьому розділі розроблено інтерфейс для представленої інформаційної системи та представлена структура, що забезпечує ефективне зберігання та управління даними, дозволяючи користувачам додавати, видаляти та відслідковувати витрати, організовані за категоріями. Інтерфейси користувача забезпечують простоту та зручність взаємодії з системою, зберігаючи цілісність та валідність даних.

4. Позитивні сторони роботи: Інформаційна система дозволяє користувачам додавати нові категорії витрат, перевіряючи валідність введених даних, таких як назва категорії та сума бюджету.

5. Негативні сторони роботи: В роботі не приділено достатньої уваги можливостям використання розробленої системи на мобільних пристроях.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому професійному рівні

8. Інші зауваження: =

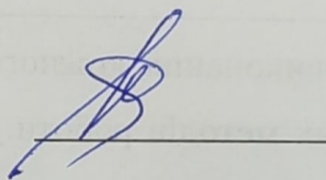
9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Гашенко О. М., доцент кафедри ІТЗ/ХЖУ

"12" 06 2024р.



Завідувачу кафедри КІПС
д-р.техн.наук, проф. Говорущенко Т. О.

Константінова Івана Олександровича

ГІВ здобувача вищої освіти

ФІТ, 3 курсу, групи ІСТс-21-1

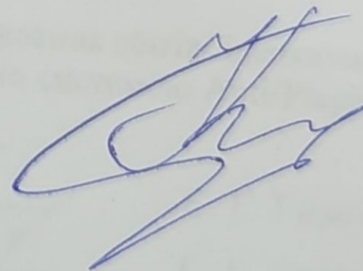
ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2024 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система для ведення обліку особистих витрат

Автор: Константинов Іван Олександрович

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Гнатчук Є.Г., к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

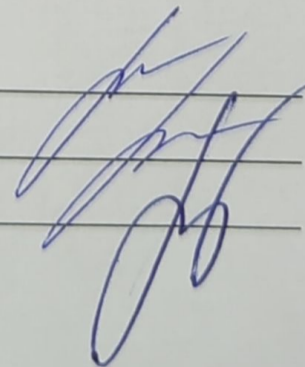
- 1) запозичення, розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості Unicheck, складає 7.89% і адресується до 204 першоджерела; та системою Anti-Plagiarism складає 1%.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Є. Г. Гнатчук

Є. Г. Гнатчук

Т. О. Говорущенко