



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки Кафедра інформаційних систем та
технологій

Лабораторна робота №2

із дисципліни *«Технології розроблення програмного забезпечення»*

**Тема: «ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЙ
ВАРІАНТІВ ВИКОРИСТАННЯ.
ДІАГРАМИ UML. ДІАГРАМИ
КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»**

Виконав:

студент групи ІА-23
Пожар Д. Ю.

Перевірив:

Мягкий М.Ю.

Київ 2024

Тема лабораторних робіт:

Варіант 27

Особиста бухгалтерія (state, prototype, decorator, bridge, flyweight, SOA)

Програма повинна бути наочним засобом для ведення особистих фінансів: витрат і прибутку; з можливістю встановлення періодичних витрат / прибутку (зарплата і орендна плата); введення сканованих чеків з відповідними статтями витрат; побудова статистики; експорт/імпорт в Excel, реляційні джерела даних; різні рахунки; ведення єдиного фонду на всі рахунки (всією сім'єю) - на особливі потреби (ремонт, автомобіль, відпустка); можливість введення вкладів / кредитів для контролю банківських рахунків (звірка нарахованих відсотків з необхідними і т.д.).

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторної.
3. Намалюйте діаграму класів для реалізованої частини системи.
4. Виберіть 3 прецеденти і напишіть на їх основі прецеденти.
5. Розробити основні класи і структуру системи баз даних.
6. Класи даних повинні реалізувати шаблон Репозиторію для взаємодії з базою даних.
7. Підготувати звіт про хід виконання лабораторних робіт. Звіт, що подається повинен містити: діаграму прецедентів, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних.

Зміст

<u>Аналіз теми та шаблонів проектування</u>	<u>3</u>
<u>Створення діаграми прецедентів</u>	<u>5</u>
<u>Опис трох прецедентів</u>	<u>11</u>
<u>Створення діаграми класів для частини майбутньої системи</u>	<u>14</u>
<u>Представлення схеми таблиць бази даних</u>	<u>15</u>
<u>Висновок</u>	<u>16</u>

Хід роботи:

Крок 1. Аналіз теми та шаблонів проектування.

Перед початком роботи над лабораторною роботою, варто проаналізувати надані шаблони проектування та визначитись з варіантами їх використання.

Шаблон	Суть	Варіант застосування
State	Дає змогу об'єктам змінювати поведінку в залежності від їхнього стану. Ззовні створюється враження, ніби змінився клас об'єкта.	Стан кредитів та вкладів: Активний, Прострочений, Погашений
Prototype	Дозволяє копіювати об'єкти, не вдаючись у подробиці їхньої реалізації.	Створення копії вже існуючих шаблонів транзакцій або рахунків для швидкого введення даних без повторного їх створення
Decorator	Дозволяє динамічно додавати об'єктам нову функціональність, використовуючи обгортки.	Функціональність надання інформації про залишок, кредитний ліміт або спеціальні умови.

Bridge	Розділяє один або кілька класів на дві окремі ієрархії — абстракцію та реалізацію, дозволяючи змінювати код в одній гілці класів, незалежно від іншої.	Абстракція — це звітність (рахунки, витрати), а реалізація — різні способи її відображення (графічна, текстова, PDF або Excel)
Flyweight	Дає змогу вмістити більшу кількість об'єктів у відведеній оперативній пам'яті. Він заощаджує пам'ять, розподіляючи спільний стан об'єктів між собою, замість зберігання однакових даних у кожному об'єкті.	Управління однотипними транзакціями або категоріями витрат
SOA	Архітектурний стиль, який розбиває програму на незалежні сервіси, що взаємодіють через чітко визначені інтерфейси.	Можна окремо реалізувати сервіси для управління рахунками, статистикою, транзакціями, які можуть взаємодіяти через API

Крок 2. Створення діаграми прецедентів.

Діаграма прецедентів системи представлена на рис. 1.

Акторами є користувачі системи: незареєстрований (гість), зареєстрований (користувач).

Зареєстрованому користувачу доступна уся функціональність, що і незареєстрованому, а також можливість управління рахунками, реєстрації транзакцій, управління категоріями, управління фондами, управління вкладками та кредитами, отримання звітності та статистики. Детально усі сценарії використання описані нижче.



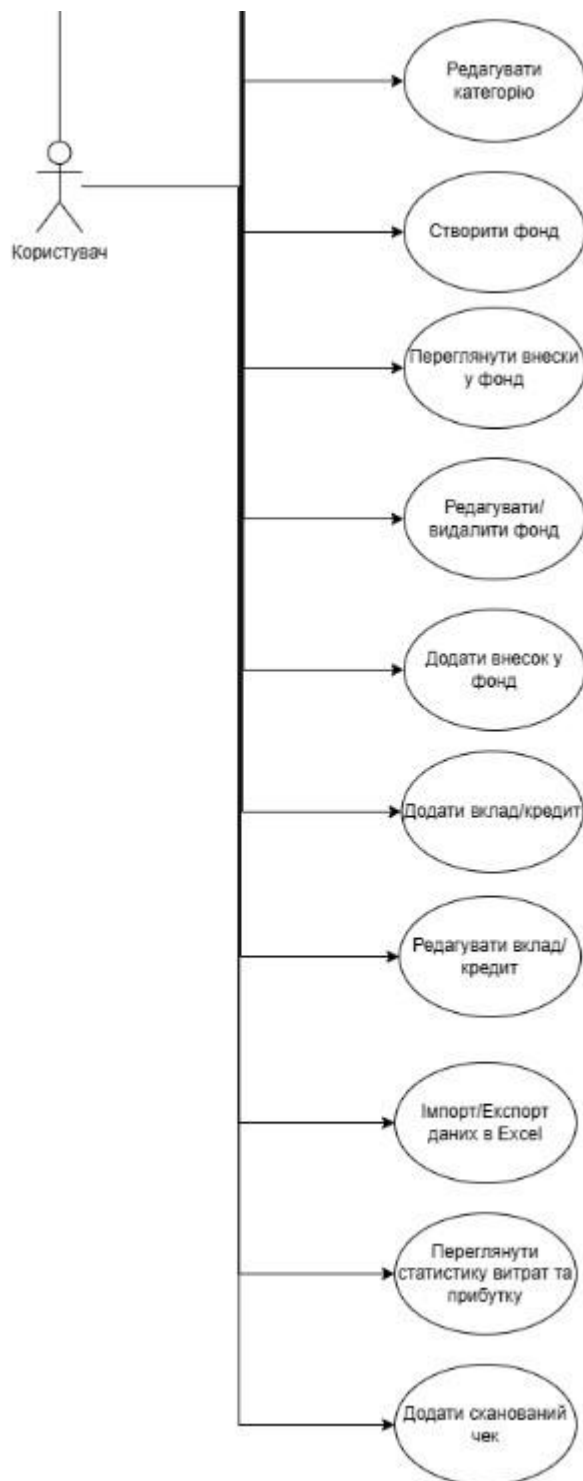


Рисунок 1 – Діаграма прецедентів

Посилання на повну діаграму прецедентів:
<https://drive.google.com/file/d/1p9epDIWTbSqUPyhhGOxLhW44WPia1MR/view?usp=sharing>

Опис сценаріїв використання системи

Управління обліковим записом

1) Реєстрація

- Користувач може створити новий акаунт, вказавши основну інформацію (ім'я, пошта, пароль), яка зберігається у базі даних.

2) Автентифікація

- Користувач надає дані для входу (пошта або нікнейм, пароль), і система надає доступ до фінансових даних.

Управління рахунками

1) Створення рахунку:

- Користувач може створити новий рахунок (наприклад, дебітовий, кредитний, накопичення) з визначеною назвою та початковим балансом. Ця інформація зберігається в базі даних, і система відображає рахунок у загальному списку доступних користувачу рахунків.

2) Оновлення рахунку:

- Користувач може змінити інформацію про рахунок (назву). Зміни зберігаються в базі даних, а оновлені дані відображаються у профілі користувача.

3) Перегляд балансу:

- Користувач може переглянути поточний баланс для всіх своїх рахунків.

Система обчислює загальну суму з усіх рахунків і відображає цю інформацію.

Реєстрація транзакцій

1) Додавання транзакції:

- Користувач може створити нову транзакцію (витрату або прибуток), вказавши суму, категорію, опис і дату. Ця інформація зберігається в базі даних і оновлює баланс рахунку.

2) Завантаження сканованого чека:

- Користувач може завантажити зображення чека, пов'язаного з транзакцією, у систему. Система зберігає дані зображення, що спрощує управління транзакціями.
- ### **3) Категоризація транзакції:**

- Користувач може прив'язати транзакцію до певної категорії (наприклад, їжа, розваги). Категорія допомагає в організації та аналізі витрат та прибутків.

4) Редагування транзакції:

- Користувач може змінити деталі існуючої транзакції, зокрема суму, категорію, опис і дату. Зміни зберігаються у базі даних і оновлюють баланс рахунку.

Управління категоріями

1) Створення категорії:

- Користувач може створити нову категорію витрат або доходів (наприклад, навчання, продукти). Категорія зберігається в базі даних і стає доступною для використання в майбутніх транзакціях.

2) Редагування категорії:

- Користувач може змінити назву або видалити існуючу категорію. Зміни відображаються у всіх пов'язаних транзакціях.

Управління фондами (накопиченнями)

1) Створення фонду:

- Користувач може створити фонд для накопичення на певні потреби (наприклад, відпустка, ремонт).

2) Перегляд внесків у фонд:

- Користувач може переглянути внески, які були додані до фонду. Система відображає всі внески, зроблені на цю мету.

3) Додавання внеску у фонд:

- Користувач може додати певну суму до конкретного фонду. Система автоматично оновлює баланс фонду.

4) Редагування фонду:

- Користувач може змінити ціль або суму фонду.

Управління вкладами та кредитами

1) Додавання вкладу/кредиту:

- Користувач може зареєструвати вклад або кредит, вказавши суму, відсоток, термін. Система зберігає цю інформацію для подальшого моніторингу.

2) Редагування вкладу/кредиту:

- Користувач може змінити параметри вкладу чи кредиту. Зміни зберігаються в базі даних і враховуються в розрахунках.

Звітність та статистика

1) Перегляд статистики витрат та прибутків:

- Користувач може переглядати звіти за обраний період, побачити категорії з найбільшими витратами або доходами. Система генерує статистику, спираючись на усі транзакції.

2) Імпорт/Експорт в Excel:

- Користувач може експортувати звіт у формат Excel для аналізу або архівування. Система генерує файл із потрібними даними.

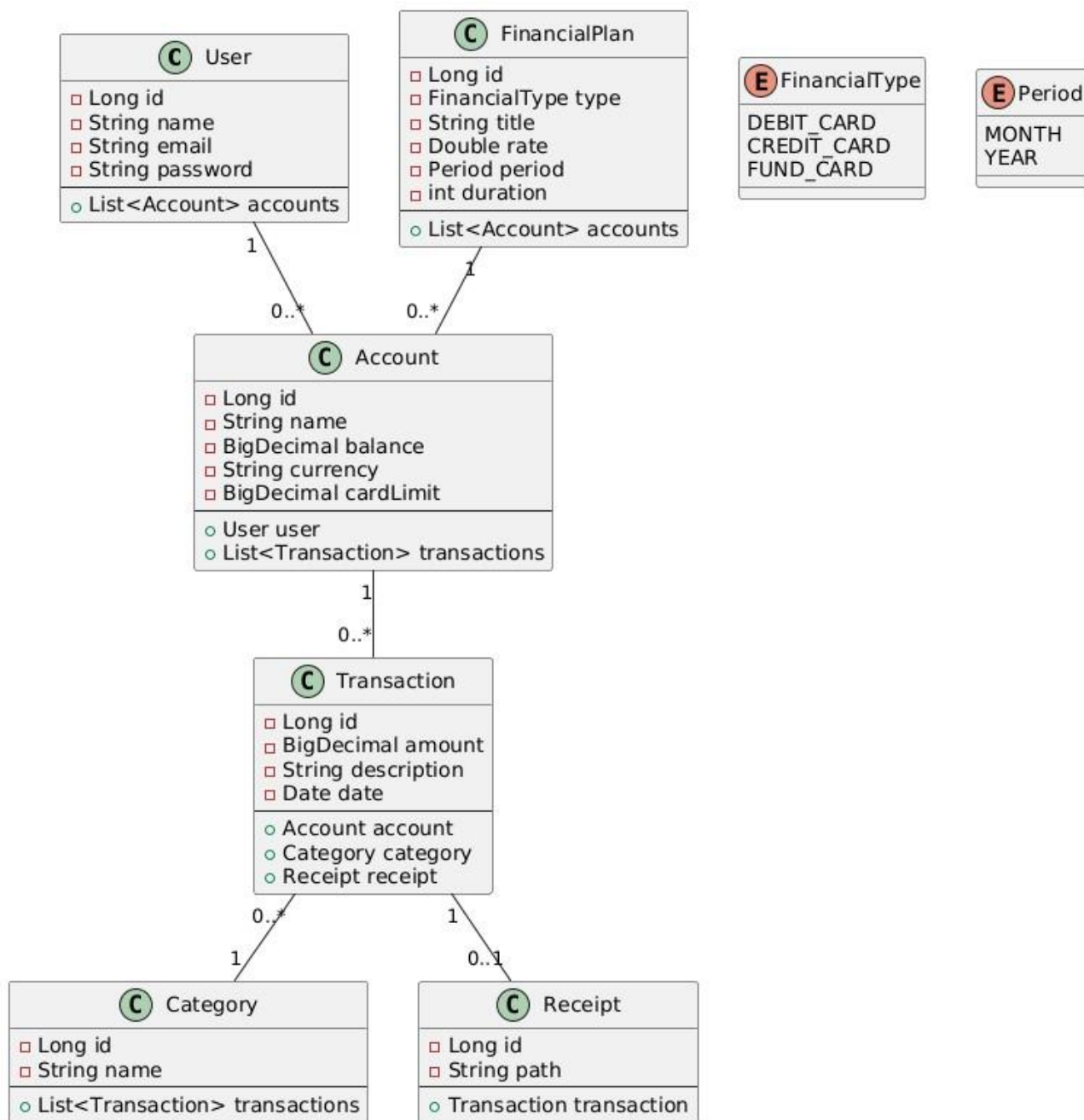
Крок 3. Опис трьох прецедентів.

Назва	Додавання транзакції
ID	1
Опис	Користувач може додати нову транзакцію витрати або доходу, щоб слідкувати за своїм фінансовим станом
Актори	Користувач
Вигоди компанії	Забезпечує можливість для користувача контролювати свої фінанси, що підвищує залученість та задоволеність користувачів
Частота користування	Часто
Тригери	Користувач натискає кнопку "Додати нову транзакцію"
Передумови	Користувач увійшов у систему та має принаймні один рахунок
Постумови	Транзакція зберігається в базі даних, і баланс оновлюється
Основний розвиток	<p>Користувач натискає на кнопку "Додати транзакцію"</p> <p>Вибирає тип транзакції (витрата чи дохід), вказує суму, дату, категорію та коментарі</p> <p>У Користувача є можливість додати чек разом з транзакцією</p> <p>Система зберігає транзакцію та оновлює баланс.</p>
Альтернативні розвитку	<ul style="list-style-type: none"> • Користувач натискає кнопку "Скасувати" на будь-якому етапі додавання транзакції. • Система відображає повідомлення про некоректні дані та пропонує виправити помилку.

Назва	Експорт даних у Excel
ID	2
Опис	Користувач може експортувати свої фінансові дані у файл Excel для подальшого аналізу або архівування
Актори	Користувач
Вигоди компанії	Надає користувачам можливість зберігати фінансову історію в зручному форматі, що сприяє лояльності користувачів до системи
Частота користування	Іноді
Тригери	Користувач натискає кнопку "Експорт у файл"
Передумови	Користувач увійшов у систему і має фінансові дані, доступні для експорту
Постумови	Система генерує файл з обраними даними, який користувач може завантажити
Основний розвиток	Користувач відкриває модуль експорту даних Вибирає період та формат експорту(Excel) Система генерує файл і пропонує користувачу його завантажити
Альтернативні розвитку	<ul style="list-style-type: none"> Через технічні проблеми файл Excel не створюється. Система відображає повідомлення про помилку і пропонує спробувати пізніше.

Назва	Створення фонду
ID	3
Опис	Користувач може створити фонд, вказавши мету (напр., відпустка, ремонт) і додати до нього кошти
Актори	Користувач
Вигоди компанії	Система стає більш привабливою для використання, оскільки сприяє накопиченню коштів на важливі потреби
Частота користування	Часто
Тригери	Користувач натискає кнопку "Створити новий фонд"
Передумови	Користувач увійшов у систему і має доступ до функції фонду
Постумови	Створюється фонд із визначеною метою, і він стає доступним для поповнення
Основний розвиток	Користувач вибирає функцію " Створити новий фонд " Вказує мету та суму фонду Система зберігає фонд і додає його до списку доступних для поповнення
Альтернативні розвитку	<ul style="list-style-type: none"> Користувач натискає "Скасувати" під час заповнення даних про фонд. Система відображає повідомлення про некоректні дані та пропонує виправити помилку.

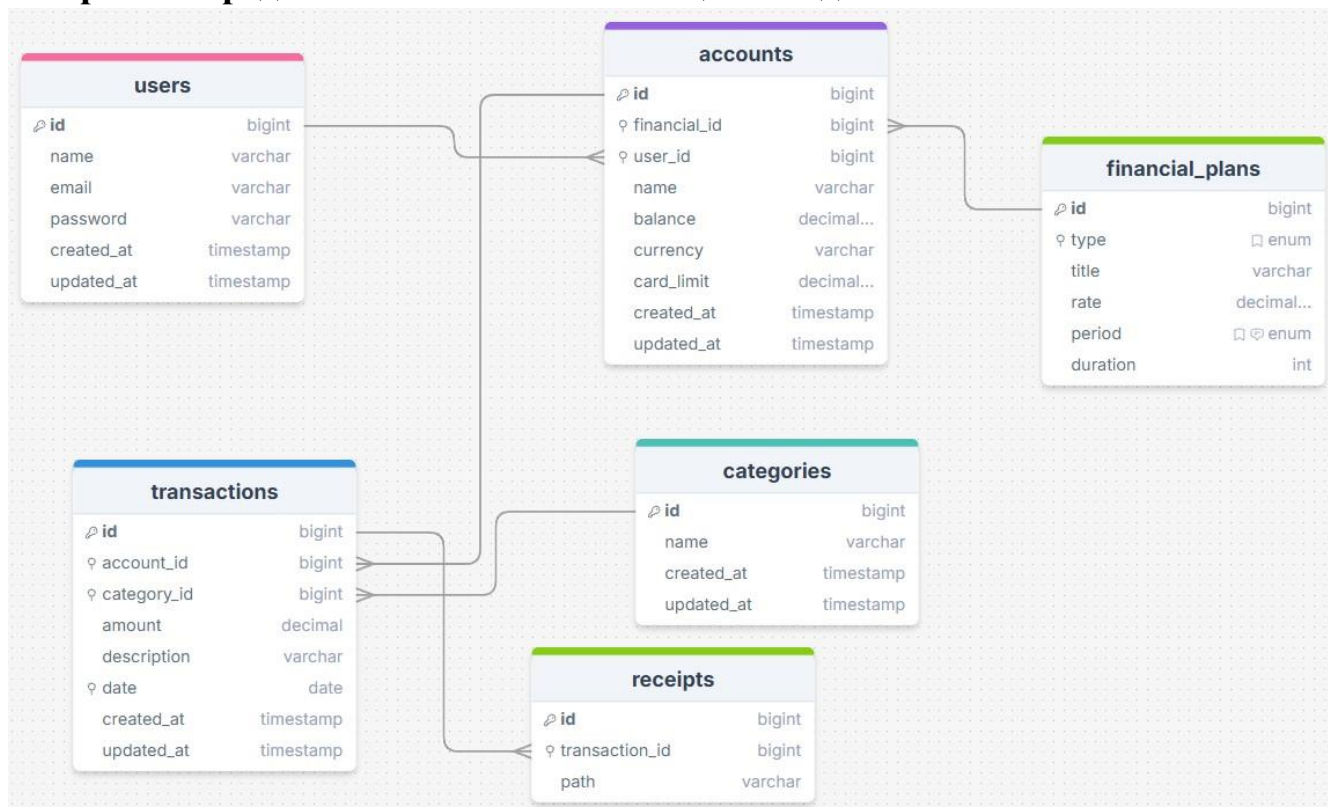
Крок 4. Створення діаграми класів для частини майбутньої системи.



Код реалізації частини системи можна знайти на гітхабі за посиланням

(https://github.com/Maphin/personal_accounting_app)

Крок 5. Представлення схеми таблиць бази даних.



Опис зв'язків:

- Users – Accounts: 1 : N (один користувач може мати багато рахунків)
- Accounts – Transactions: 1 : N (один рахунок може мати багато транзакцій)
- Financial_plans – Accounts: 1 : N (один фінансовий план може бути використаний у багатьох рахунках)
- Categories – Transactions: 1 : N (багато транзакцій можуть належати одній категорії)
- Transactions – Receipts: 1 : N (одна транзакція може мати багато чеків)

Висновок

Під час виконання даної лабораторної роботи я ознайомився з основними теоретичними поняттями, що стосуються моделювання та проектування програмних систем. На основі обраної теми я створив діаграму прецедентів, яка ілюструє основні функціональні можливості системи, а також описав обрані прецеденти, що дозволяють зрозуміти послідовність взаємодії користувача із системою.

Було розроблено діаграму класів, яка відображає основні класи, їх атрибути, методи, а також зв'язки між ними. Це допомогло структурувати систему та визначити ключові об'єкти й їхні ролі в проєкті. Крім того, я створив структуру бази даних, яка забезпечує зберігання інформації, необхідної для роботи системи.