



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

Технології розроблення програмного забезпечення

Тема: «Особиста бухгалтерія»

Виконала:

Студентка групи ІА-34

Дригант А.С

Перевірив:

Мягкий Михайло Юрійович

Тема: Основи проектування

Мета: Обрати зручну систему побудови UML-діаграм та навчитися будувати діаграми варіантів використання для системи що проєктується, розробляти сценарії варіантів використання та будувати діаграми класів предметної області.

Тема роботи:

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.).

2.1. Теоретичні відомості

UML та його діаграми

UML (Unified Modeling Language) - уніфікована мова візуального моделювання, що використовується для аналізу, проєктування та документування програмних систем. UML дозволяє описувати систему на різних рівнях: від концептуального до фізичного.

Основні діаграми UML

- Діаграма варіантів використання (Use Case Diagram) - показує вимоги до системи та взаємодію користувачів із нею.
- Діаграма класів (Class Diagram) - описує статичну структуру системи: класи, їх атрибути, методи та зв'язки.

Діаграма варіантів використання

Діаграма use case відображає функціональність системи з точки зору користувача.

Основні елементи:

- Актори (Actor) - користувачі або зовнішні системи.
- Варіанти використання (Use Case) - дії або послуги, які система надає актору (наприклад: вхід, перегляд даних, створення транзакції).

Типи відносин:

- Асоціація - прямий зв'язок актора з варіантом використання.
- Include - один сценарій завжди включає інший (обов'язковий).
- Extend - сценарій може бути розширений додатковим (необов'язковим).
- Узагальнення - спадкування ролей або функціоналу.

Для уточнення роботи системи складаються сценарії використання (use case scenarios), які описують:

- передумови та постумови;
- учасників;
- короткий опис;
- основний перебіг подій;
- винятки.

Діаграма класів

Діаграма класів показує структуру системи: класи, їх атрибути, методи та зв'язки між ними.

Клас містить:

- назву;
- атрибути (дані);
- методи (операції).

Види зв'язків:

- Асоціація - загальний зв'язок між класами.
- Узагальнення (успадкування) - зв'язок між батьківським і дочірнім класом.
- Агрегація - відношення «ціле-частина», де частини можуть існувати окремо.
- Композиція - сильне відношення «ціле-частина», де частини не існують без цілого.

Логічна структура бази даних

Проектування бази даних часто виконується на основі діаграми класів.

Виділяють:

- Фізичну модель - організація файлів і способів зберігання.
- Логічну модель - таблиці, атрибути, ключі, зв'язки.

Щоб уникнути надмірності даних застосовують нормалізацію:

- 1НФ - кожен атрибут має лише одне атомарне значення.
- 2НФ - усі неключові атрибути залежать від усього первинного ключа.
- 3НФ - немає транзитивних залежностей (атрибутів, що залежать від інших неключових атрибутів).
- НФ Бойса-Кодда (BCNF) - посилена форма 3НФ, кожна залежність визначається ключем.

Хід роботи

1. Розробка діаграми варіантів використання (Use Case Diagram)

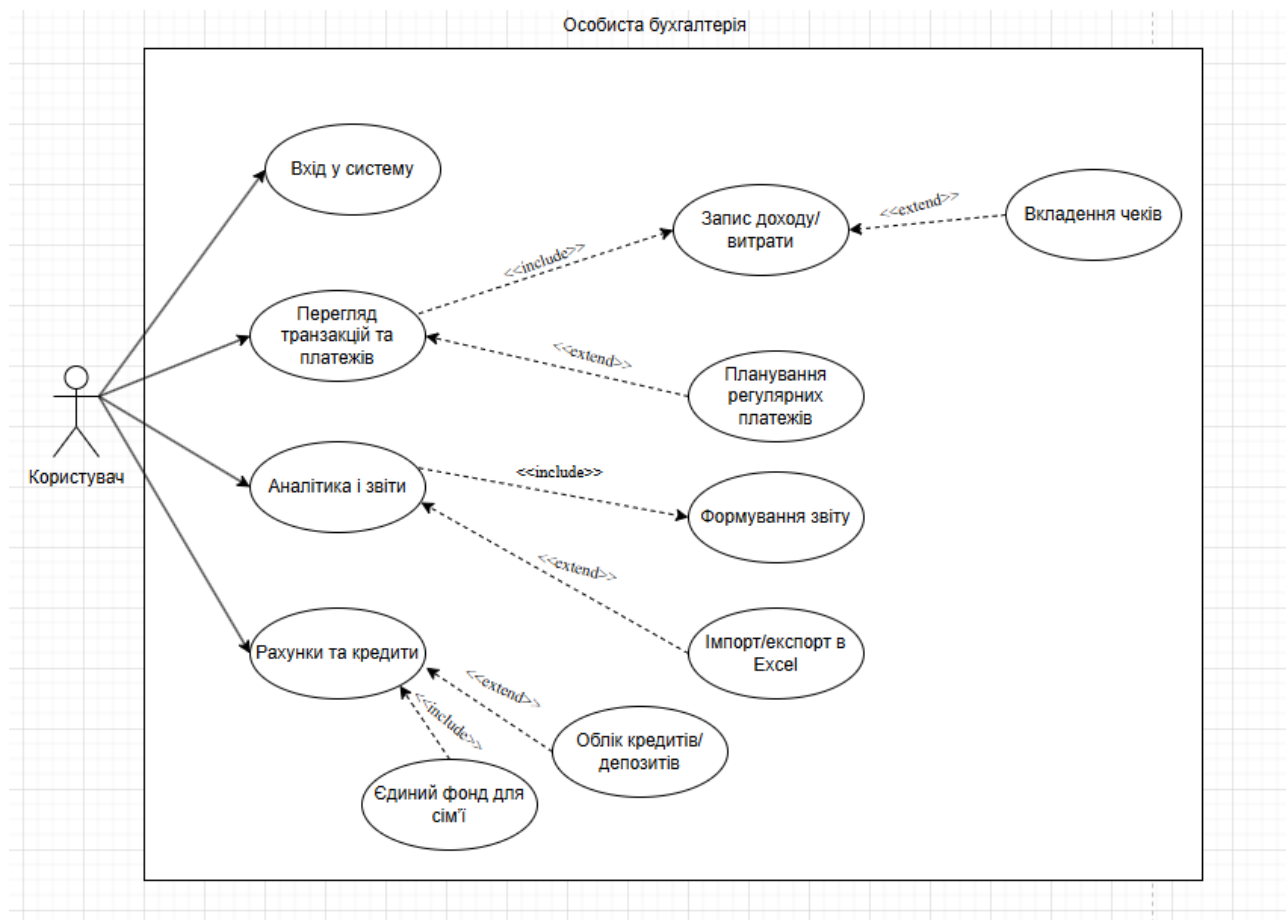


Рисунок 1 – Use case діаграма для звичайного користувача

2. Розробка діаграми класів предметної області

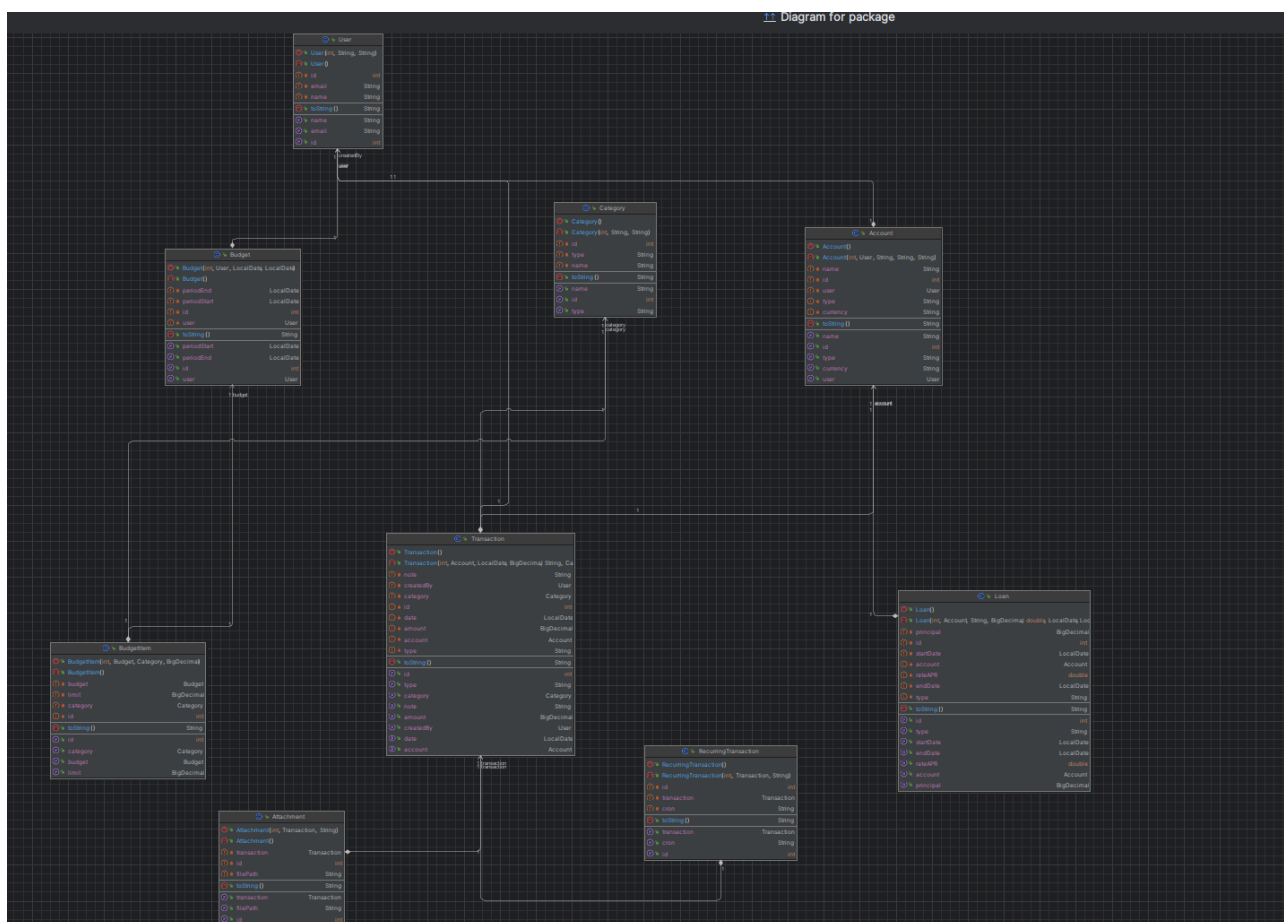


Рисунок 2 –Діаграма класів

3 варіанти сценарії використання

Сценарій 1: Запис доходу/витрати

Передумови:

- Користувач автентифікований у системі
- Існує хоча б один рахунок і одна категорія транзакцій

Постумови:

- У системі створено нову транзакцію.
- Баланс рахунку оновлено

Взаємодіючі сторони:

Користувач, Система.

Короткий опис:

Цей варіант використання описує процес додавання доходу або витрати користувачем у систему.

Основний потік подій:

1. Користувач обирає функцію «Додати транзакцію»
2. Система пропонує ввести дані: рахунок, категорію, дату, суму, тип (дохід/витрата), примітку
3. Користувач вводить усі необхідні дані
4. Система створює нову транзакцію, оновлює баланс рахунку та повідомляє про успішне збереження

Винятки:

- Якщо користувач не вказав категорію то система просить вибрати або створити нову
- Якщо валюта рахунку не збігається з введеною транзакцією, то система виконує конвертацію або видає повідомлення про помилку

Примітки:

Можливе додавання вкладення (скан чеку).

Сценарій 2: Планування регулярних платежів

Передумови:

- Користувач увійшов у систему
- Існує хоча б один рахунок і категорія

Постумови:

- У системі збережено нове правило періодичної транзакції

- У зазначений час будуть автоматично створюватися відповідні транзакції

Взаємодіючі сторони:

Користувач, Система.

Короткий опис:

Цей варіант описує створення періодичного платежу (наприклад, щомісячної оплати оренди).

Основний потік подій:

1. Користувач обирає функцію «Створити періодичну транзакцію»
2. Система пропонує ввести дані: рахунок, категорію, суму, періодичність, дату початку
3. Користувач вводить дані й підтверджує створення
4. Система зберігає правило і планує першу дату виконання

Винятки:

- Якщо користувач вводить некоректну періодичність то система видає повідомлення і просить повторити введення
- Якщо на рахунку немає достатньо коштів у момент виконання то система позначає транзакцію як «очікує підтвердження»

Примітки:

Для регулярних доходів (наприклад, зарплата) процес аналогічний.

Сценарій 3: Експорт даних у Excel

Передумови:

- Користувач увійшов у систему.

- У базі є хоча б одна транзакція.

Постумови:

- Згенеровано файл Excel із даними про транзакції.

Взаємодіючі сторони:

Користувач, Система.

Короткий опис:

Цей варіант описує процес експорту фінансових даних у формат Excel для подальшого аналізу.

Основний потік подій:

1. Користувач обирає функцію «Експортувати дані»
2. Система пропонує вибрати період (наприклад, за місяць)
3. Користувач підтверджує вибір
4. Система формує Excel-файл та надає можливість його завантажити

Винятки:

Якщо за вибраний період немає даних → система показує повідомлення «Немає даних для експорту»

Примітки:

Експорт може виконуватися і у формат CSV

Основні класи та структура бази даних системи:

Сервис		
Показать или скрыть		
Схема данных Categories Loans RecurringTransaction:		
Имя поля	Тип данных	
ID	Счетчик	
Name	Короткий текст	
Email	Короткий текст	

Сервис		
Показать или скрыть		
Схема данных Categories Loans RecurringTransaction:		
Имя поля	Тип данных	
ID	Счетчик	
UserID	Числовой	
PeriodStart	Дата и время	
PeriodEnd	Дата и время	

Сервис		
Показать или скрыть		
Схема данных Categories Loans RecurringTransaction:		
Имя поля	Тип данных	
ID	Счетчик	
BudgetID	Числовой	
CategoryID	Числовой	
Limit	Денежный	

Сервис		
Показать или скрыть		
События полей, записи		
BudgetItems Attachments Схема данных RecurringTransaction		
Имя поля	Тип данных	
ID	Счетчик	
TransactionId	Числовой	
Cron	Короткий текст	

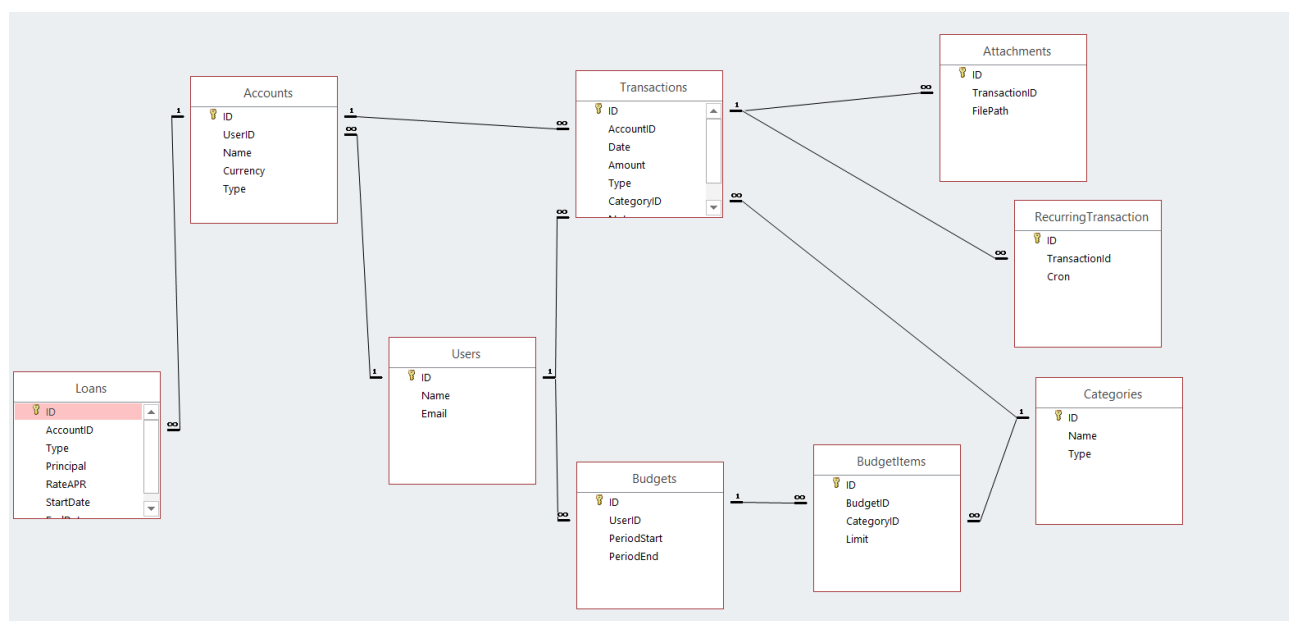
Loans		
	Имя поля	Тип данных
🔑	ID	Счетчик
	AccountID	Числовой
	Type	Короткий текст
	Principal	Денежный
	RateAPR	Числовой
	StartDate	Дата и время
	EndDate	Дата и время

Transactions		
	Имя поля	Тип данных
🔑	ID	Счетчик
	AccountID	Числовой
	Date	Дата и время
	Amount	Денежный
	Type	Короткий текст
	CategoryID	Числовой
	Note	Короткий текст
	CreatedBy	Числовой

Categories		
	Имя поля	Тип данных
🔑	ID	Счетчик
	Name	Короткий текст
	Type	Короткий текст

Budgets	
Имя поля	Тип данных
ID	Счетчик
UserID	Числовой
PeriodStart	Дата и время
PeriodEnd	Дата и время

BudgetItems Attachments	
Имя поля	Тип данных
ID	Счетчик
TransactionID	Числовой
FilePath	Короткий текст



Діаграма класів реалізованої частини системи:



Рисунок 3 – Діаграма класів реалізованої частини системи

На діаграмі зображено сутності предметної області (User, Account, Transaction, Category та інші), сервіси бізнес-логіки (FinanceService, RecurringService, ExportService) та інтерфейси репозиторіїв (UserRepository, AccountRepository, TransactionRepository), які забезпечують доступ до даних. Діаграма демонструє основну архітектуру системи: сервіси взаємодіють із репозиторіями, а репозиторії оперують сутностями.

Висновок: у ході роботи я ознайомилась з основами проектування програмних систем. Було розроблено діаграму варіантів використання, діаграму класів предметної області та реалізованої частини системи «Особиста бухгалтерія». На основі побудованої моделі створено структуру бази даних у MS Access та встановлено зв'язки між таблицями. Отримані результати демонструють вміння застосовувати UML та принципи проектування для організації даних і бізнес-логіки системи.