Лабораторна робота №3

Кохан Денис

Тема: UML - Unіfіеd Modеlіng Lаnguаgе. Потоки подій.

Мета: Навчитися створювати потоки подій для діаграми прецедентів

Завдання 1: Використовуючи діаграму прецедентів умовної предметної області «Підприємство по збірці та продажу комп'ютерів» (Рис. 7) створити описи потоків подій прецедентів (варіантів використання) та побудувати відповідну схему потоку подій прецеденту UC. Описати, які типи зв’язків та відношень формуються між об’єктами діаграми. Номер прецедентна для опису розподіляється між студентами за 5 варіантами.

**1. Опис потоку подій UC1: «Управління інформацією про клієнта»**

**1.1. Основний (успішний) потік подій**

**Ініціація операції**  
**Опис:** Актор (наприклад, оператор або адміністратор) запускає модуль управління клієнтами через інтерфейс користувача.  
**Результат:** Система переходить у режим перегляду даних клієнтів та пропонує список записів, а також основні операції.

**Вибір операції**  
**Опис:** Користувач вибирає тип операції: додавання, редагування або видалення інформації про клієнта.  
**Результат:** Залежно від вибору, система відображає відповідну форму для введення нових даних або редагування існуючих.

**Введення або модифікація даних**  
**Опис:** Користувач заповнює форму введення або редагування даних.  
**Результат:** Система проводить попередню перевірку даних (формат, обов’язковість полів) та готує дані до збереження.

**Підтвердження операції**  
**Опис:** Після вводу даних користувач підтверджує операцію (натискає кнопку «Зберегти» або «Підтвердити»).  
**Результат:** Система зберігає зміни в базі даних і відображає повідомлення про успішне виконання операції.

**Завершення процесу**  
**Опис:** Система повертає користувача на основний екран або оновлює список клієнтів.  
**Результат:** Процес управління інформацією завершено, і дані клієнта оновлено.

**1.2. Альтернативні потоки та виняткові ситуації**

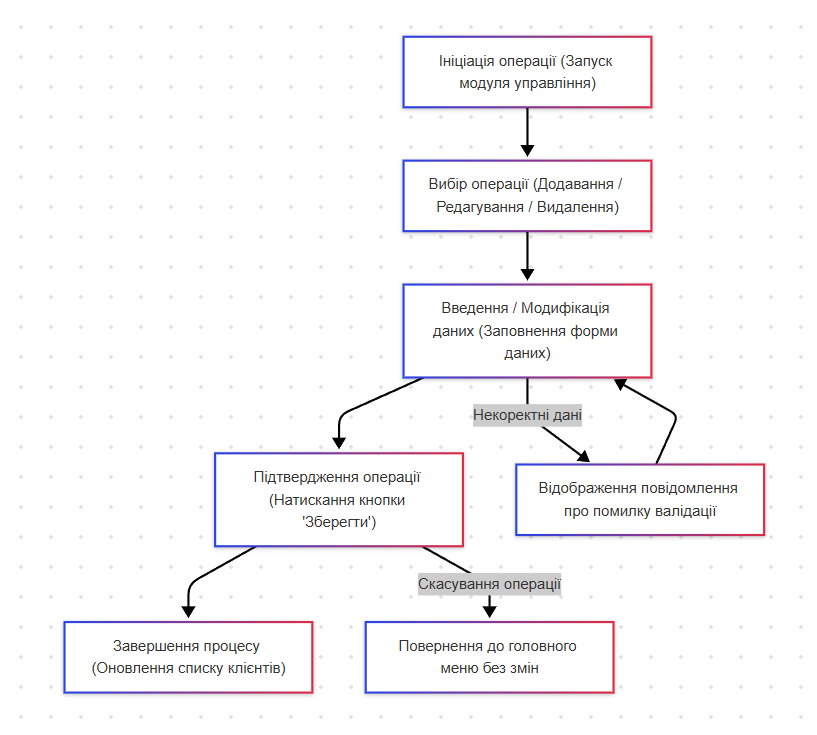
**Помилка валідації даних:**  
Якщо користувач вводить некоректні або неповні дані, система відображає повідомлення про помилку і просить повторно ввести інформацію.  
*Відновлення:* Користувач виправляє дані та повторює спробу підтвердження.

**Аварійне завершення операції:**  
При виникненні системної помилки або втрати з’єднання з базою даних система повідомляє користувача та пропонує спробувати операцію пізніше.  
*Відновлення:* Користувач повертається до попереднього етапу або завершення процесу.

**Скасування операції:**  
Якщо користувач вирішує скасувати дію, він може натиснути кнопку «Відміна», після чого операція переривається і система повертає попередній стан.  
*Результат:* Дані не зберігаються, а система повертається до головного екрану без змін.

**2. Схема потоку подій для UC1**

**UC1: Управління інформацією про клієнта**:



3. Таблиця опису потоків подій для UC1

**UC1: Управління інформацією про клієнта**:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Дія користувача | Подія в системі | Результат |
| 1 | Ініціювання | Запуск модуля управління клієнтами | Система відкриває головний екран модуля |
| 2 | Вибір операції | Вибір операції | Система відображає відповідну форму |
| 3 | Заповнення форми | Введення даних про клієнта | Система проводить валідацію введених даних |
| 4 | Підтвердження | Натискання кнопки «Зберегти» | Дані зберігаються, відображається повідомлення про успішну операцію |
| 5 | Завершення | Завершення роботи з модулем | Система повертається до головного меню або оновлює список клієнтів |

Завдання 2:  
from graphviz import Digraph

diagram = Digraph('UC1\_Client\_Management', format='png')

diagram.attr(rankdir='LR', size='8,5')

diagram.attr('node', shape='box', style='filled', color='lightblue')

diagram.node('Start', 'Початок')

diagram.node('Input', 'Введення/вибір дії (додати/переглянути/редагувати/видалити)')

diagram.node('Add', 'Додавання нового клієнта')

diagram.node('View', 'Перегляд інформації про клієнта')

diagram.node('Edit', 'Редагування даних клієнта')

diagram.node('Delete', 'Видалення клієнта')

diagram.node('End', 'Кінець')

diagram.edge('Start', 'Input')

diagram.edge('Input', 'Add', label='якщо "додати"')

diagram.edge('Input', 'View', label='якщо "переглянути"')

diagram.edge('Input', 'Edit', label='якщо "редагувати"')

diagram.edge('Input', 'Delete', label='якщо "видалити"')

diagram.edge('Add', 'Input')

diagram.edge('View', 'Input')

diagram.edge('Edit', 'Input')

diagram.edge('Delete', 'Input')

diagram.edge('Input', 'End', label='вийти')

diagram.render('uc1\_client\_flow', cleanup=True)

print("[OK] Діаграма збережена як uc1\_client\_flow.png")