**Міністерство освіти і науки України**

**Тернопільський національний технічний університет ім. І. Пулюя**

*Кафедра комп’ютерних наук*

ЗВІТ

про виконання лабораторної роботи №3

з дисципліни «Технологія створення програмних продуктів» на тему «Моделювання предметної області з використанням UML»

Виконав:  
 Студент групи СТ-21 Кальві Е.

Прийняв: Палка О. В.

Тернопіль 2025

**Тема роботи**: Моделювання предметної області з використанням UML.

**Мета роботи**: Розробити UML-діаграми, які відображають структуру та поведінку системи, а також допомагають у формалізації вимог до програмного забезпечення.

**Хід виконання лабораторної роботи**:

1. Визначити свій варіант.

У таблиці 1 наведено варіант лабораторної роботи.

Таблиця 1 – Варіант лабораторної роботи

|  |  |
| --- | --- |
| 8 | UML-діаграми для сервісу доставки їжі з функцією оцінки страв користувачами |

2. Зберіть функціональні та нефункціональні вимоги до проєкту.

**Функціональні вимоги**

1. Реєстрація та аутентифікація користувачів.
2. Управління меню:
   * Додавання, редагування, видалення страв.
   * Категоризація страв за типами кухонь або категоріями (піца, салати тощо).
3. Оформлення замовлень:
   * Створення замовлення.
   * Вибір способу оплати та доставки.
4. Відстеження статусу замовлення (нове, готується, у дорозі, доставлено).
5. Оцінювання страв користувачами:
   * Додавання оцінки (зірки/бали).
   * Написання відгуків.
6. Управління профілем користувача:
   * Оновлення контактної інформації.
   * Перегляд історії замовлень.
7. Повідомлення:
   * Сповіщення про зміну статусу замовлення.
8. Пошук та фільтрація страв за різними критеріями (ціна, рейтинг, категорія).

**Нефункціональні вимоги**

1. Швидке завантаження сторінок навіть при великій кількості користувачів.
2. Безпечне збереження особистих даних та платіжної інформації.
3. Шифрування паролів користувачів.
4. Підтримка масштабованості сервісу (розширення при збільшенні кількості замовлень).
5. Адаптивний дизайн для різних пристроїв (смартфони, планшети, ПК).
6. Зручний та інтуїтивно зрозумілий інтерфейс.
7. Багатомовність додатку (українська, англійська, інші мови).
8. Висока доступність системи (мінімізація часу простою).

3. Створення UML-діаграм.

**Діаграма варіантів використання**

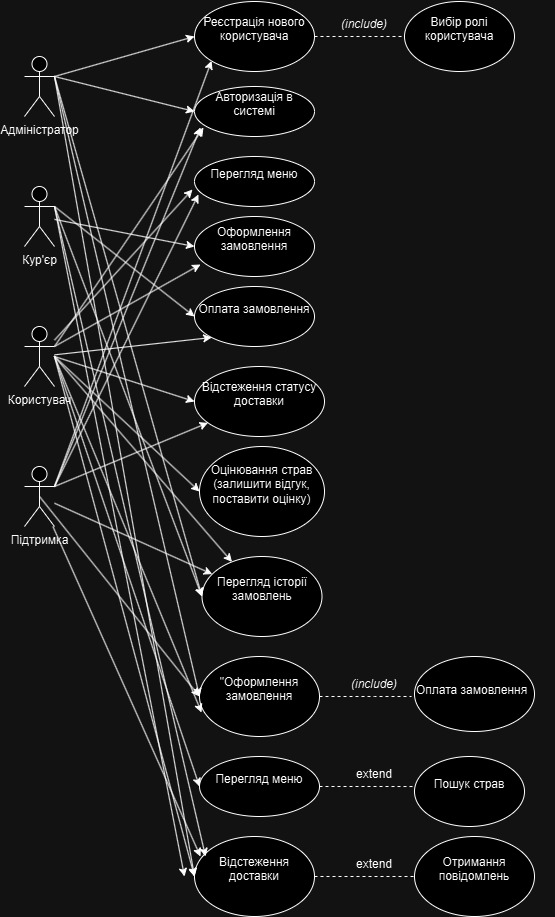


Рисунок 1 – Use Case діаграма

**Діаграму класів**

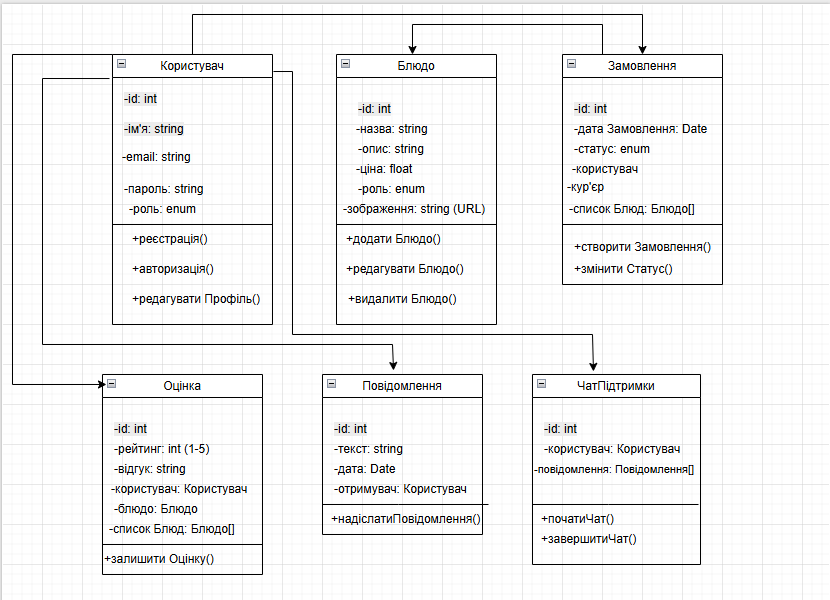
****

Рисунок 2 – Діаграма класів

**Діаграма активностей**

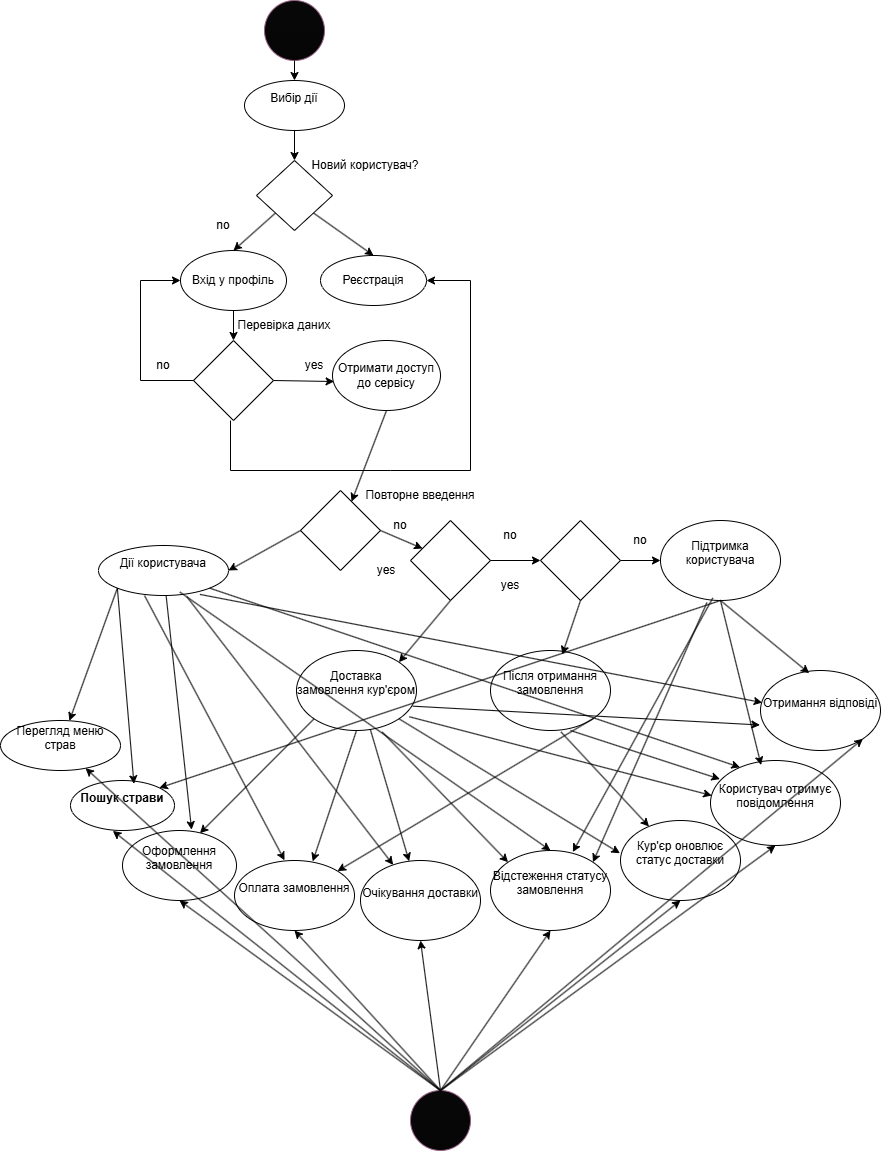


Рисунок 3 – Діаграма активності

**Діаграму станів**

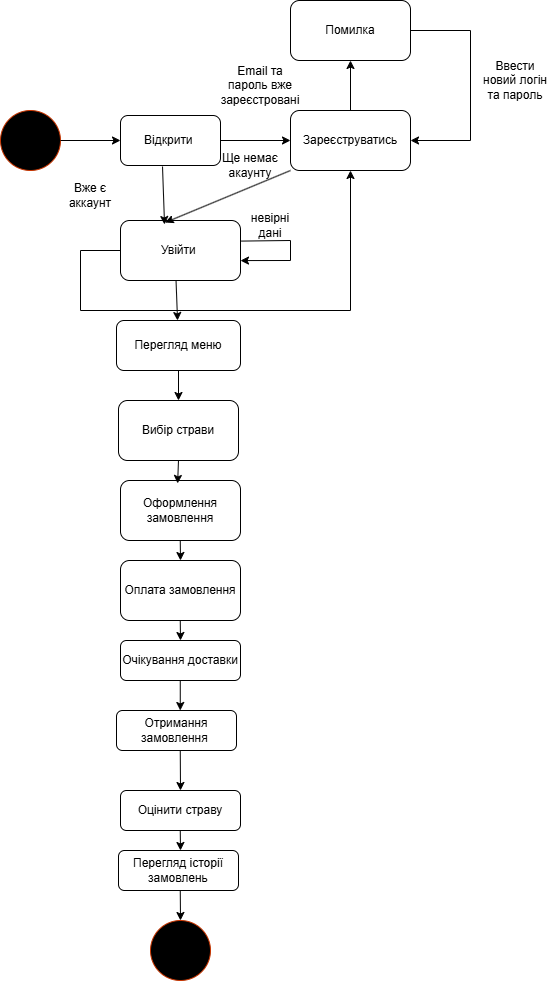


Рисунок 4 – Діаграма станів(1 користувач)

**Висновок:** розроблено UML-діаграми, які відображають структуру та поведінку системи, а також допомагають у формалізації вимог до програмного забезпечення.