

Назва закладу вищої освіти

Факультет/Інститут

Кафедра

Курсова робота

**Інформаційна система ательє**

З дисципліни: Об'єктно-орієнтоване проектування

Курсова робота: Інформаційна система ательє

Виконав(ла):

студент(ка) Група ХХ-ХХ

Прізвище Ім'я По батькові

Керівник:

Науковий керівник, ст. посада, ПІБ

Місто 2025

# ЗМІСТ

Перелік умовних позначень, скорочень . . . . .	iii
<b>1 Вступ . . . . .</b>	<b>1</b>
<b>2 Аналіз предметної області та вимог . . . . .</b>	<b>1</b>
2.1 Опис предметної області . . . . .	1
2.2 Зацікавлені сторони . . . . .	1
2.3 Функціональні вимоги . . . . .	2
2.4 Нефункціональні вимоги . . . . .	2
<b>3 Проєктування системи . . . . .</b>	<b>2</b>
3.1 Діаграма прецедентів . . . . .	2
3.2 Діаграма активності: обробка замовлення . . . . .	4
3.3 Діаграма класів . . . . .	5
3.4 Діаграма послідовностей: оформлення замовлення . . . . .	5
3.5 ER-діаграма бази даних . . . . .	6
3.6 Діаграма компонентів (структури) . . . . .	6
<b>4 Реалізація . . . . .</b>	<b>6</b>
4.1 Фрагменти коду C# . . . . .	6
4.2 SQL (SQLite) схеми та запити . . . . .	10
<b>5 Тестування . . . . .</b>	<b>11</b>
5.1 Статистичні графіки . . . . .	11
<b>6 Висновки . . . . .</b>	<b>11</b>
<b>Список використаних джерел . . . . .</b>	<b>12</b>

# Перелік умовних позначень, скорочень

API — прикладний програмний інтерфейс; БД — база даних; UML — мова уніфікованого моделювання.

# 1. Вступ

Метою курсової роботи є розроблення та документування інформаційної системи ательє, що підтримує процеси прийому замовлень, обліку клієнтів, ведення номенклатури послуг та матеріалів, планування завантаження майстрів і розрахунку вартості робіт.

Актуальність теми обумовлена потребою малих підприємств сфери побутового обслуговування в цифровізації обліку та процесів, підвищенні прозорості взаємодії з клієнтами та оптимізації ресурсів.

Об'єкт дослідження — процеси діяльності ательє з пошиття та ремонту одягу. Предмет дослідження — методи аналізу вимог і проектування програмних систем із використанням нотацій UML і ER.

Структура роботи відповідає вимогам ДСТУ та включає аналіз предметної області, формалізацію вимог, проектування (діаграми прецедентів, активності, класів, компонентів, послідовностей, ER-діаграму), стислий опис реалізації, підходи до тестування та висновки.

## 2. Аналіз предметної області та вимог

### 2.1. Опис предметної області

Ательє надає послуги з пошиття та ремонту виробів. Основні сутності: клієнт, замовлення, виріб, послуга, матеріал, майстер, платіж. Замовлення має статуси: нове, у роботі, готове, видане, скасоване.

### 2.2. Зацікавлені сторони

- Клієнт — оформлює замовлення, отримує виріб, здійснює оплату;
- Адміністратор — приймає замовлення, веде клієнтську базу, формує рахунки;
- Майстер — виконує роботи, фіксує етапи та витрати матеріалів;
- Власник — аналізує звітність, встановлює прайс-листи.

## **2.3. Функціональні вимоги**

- Реєстрація клієнтів та їх контактних даних;
- Оформлення замовлень з переліком робіт і матеріалів;
- Калькуляція вартості; облік платежів (передплата, остаточний розрахунок);
- Планування та відстеження етапів виконання робіт;
- Формування звітів (виручка, завантаженість майстрів, популярні послуги).

## **2.4. Нефункціональні вимоги**

- Зручність інтерфейсу для настільних браузерів;
- Збереження даних у реляційній БД; резервне копіювання;
- Аудит змін замовлень та платежів;
- Ролі та доступи (адміністратор, майстер, власник).

# **3. Проєктування системи**

## **3.1. Діаграма прецедентів**

На рис. 1 подано узагальнену діаграму прецедентів.

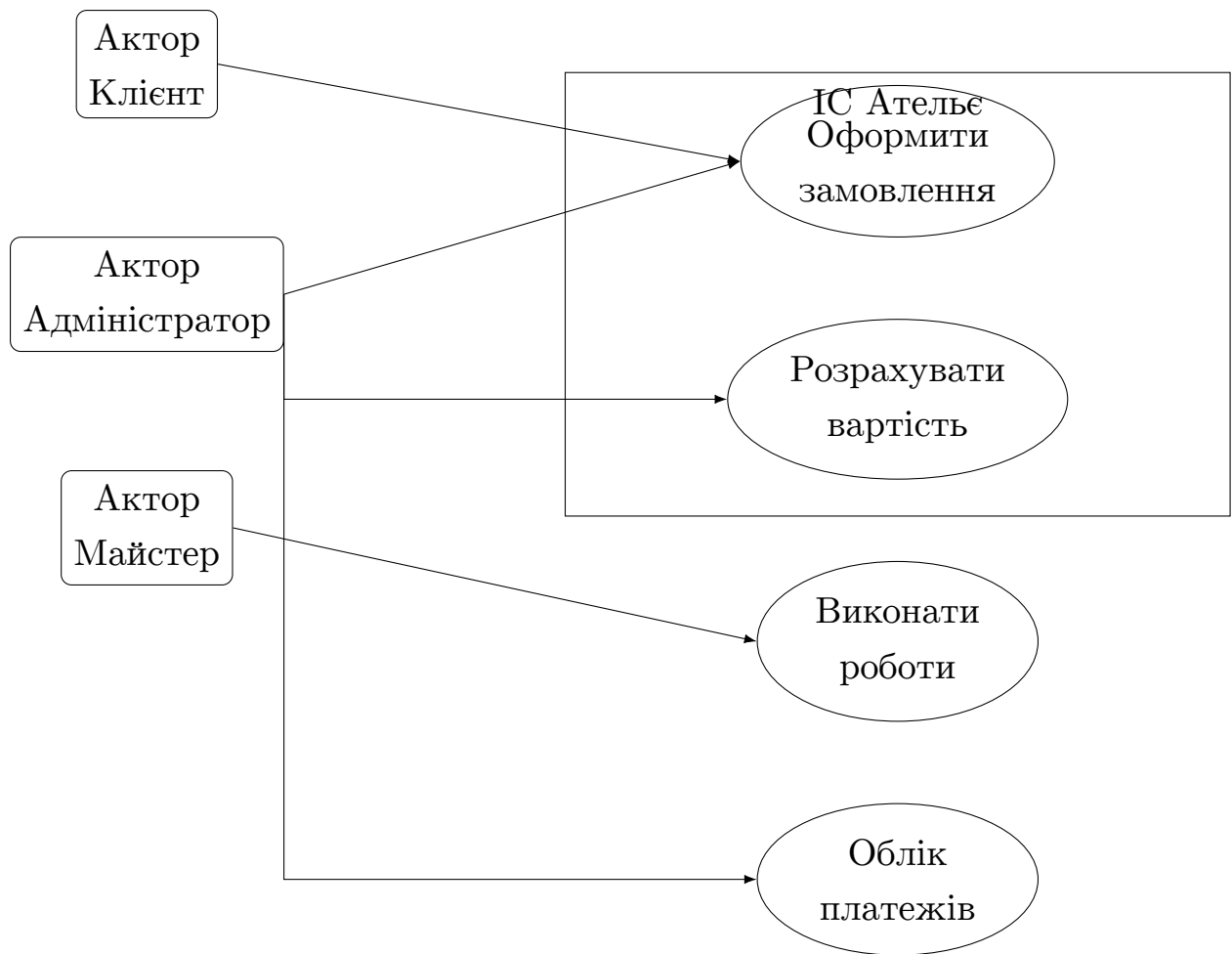


Рис. 1: Діаграма прецедентів для ІС ательє

### 3.2. Діаграма активності: обробка замовлення



Рис. 2: Діаграма активності процесу замовлення

### 3.3. Діаграма класів

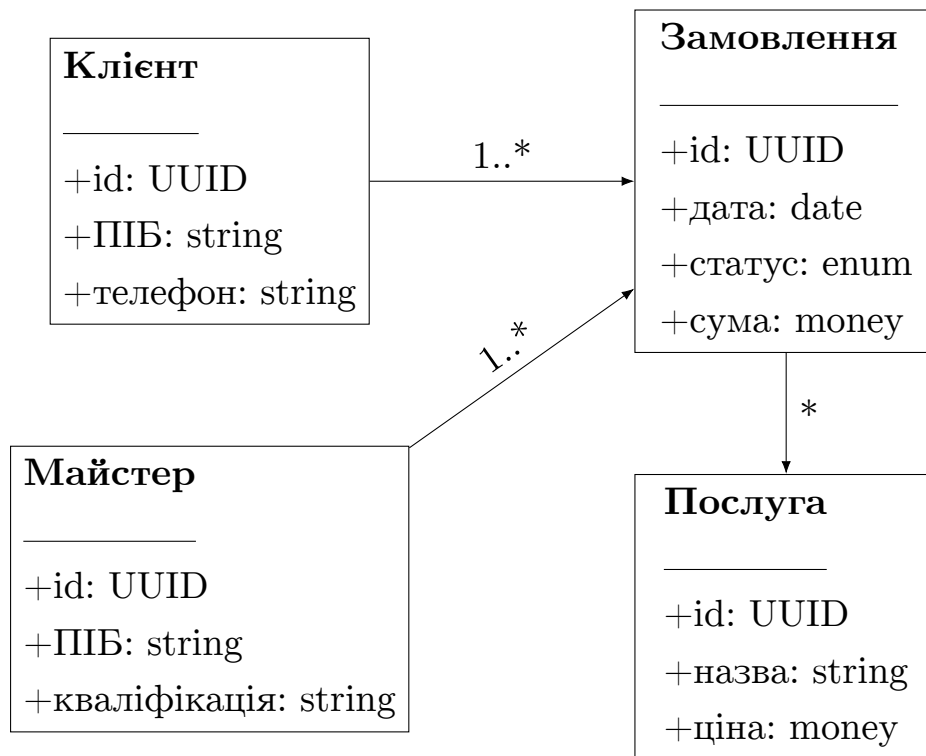


Рис. 3: Спрощена діаграма класів

### 3.4. Діаграма послідовностей: оформлення замовлення

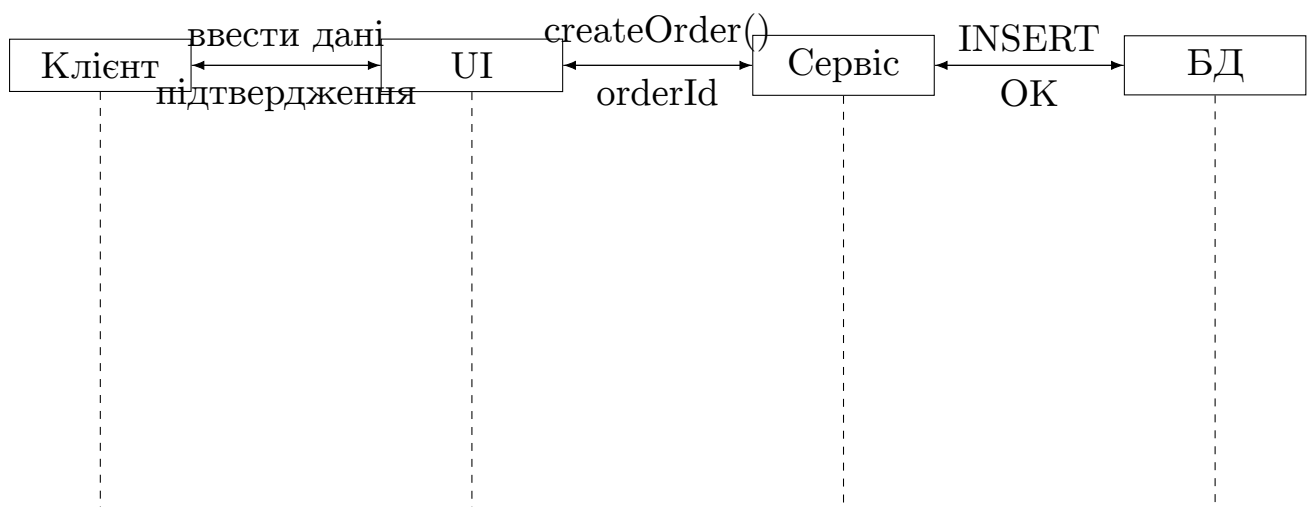


Рис. 4: Діаграма послідовностей



### 3.5. ER-діаграма бази даних

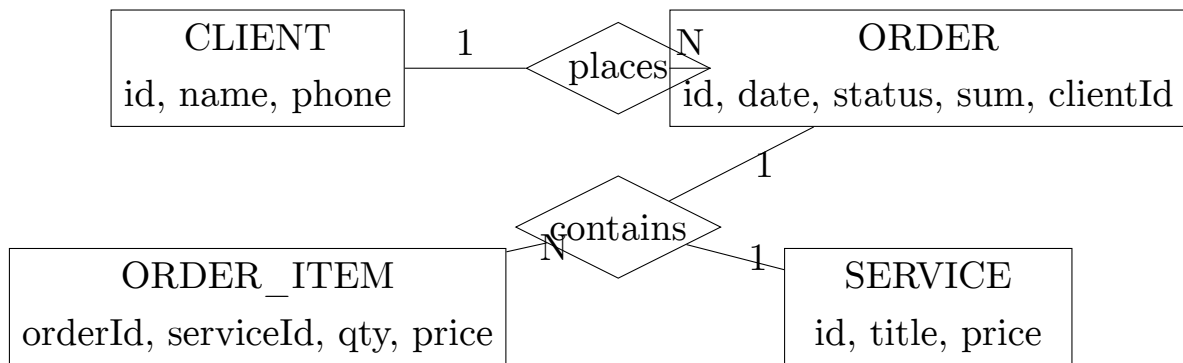


Рис. 5: ER-діаграма БД

### 3.6. Діаграма компонентів (структури)



Рис. 6: Діаграма компонентів

## 4. Реалізація

Програмна реалізація може бути виконана у вигляді веб-додатка з архітектурою клієнт-сервер. Клієнтська частина (наприклад, React) взаємодіє з серверним REST API (наприклад, Node.js/Express, Java Spring або Python FastAPI), дані зберігаються у реляційній СУБД (PostgreSQL/MySQL). Аутентифікація може базуватися на JWT, авторизація — на ролях.

### 4.1. Фрагменти коду C#

DTO та моделі

Listing 1: Моделі домену для ательє

```
1 public sealed class Client
2 {
3     public Guid Id { get; init; }
4     public string FullName { get; set; } = string.Empty;
```

```

5     public string Phone { get; set; } = string.Empty;
6 }
7
8 public enum OrderStatus
9 {
10     New,
11     InProgress,
12     Ready,
13     Delivered,
14     Cancelled
15 }
16
17 public sealed class Order
18 {
19     public Guid Id { get; init; }
20     public DateOnly Date { get; init; } = DateOnly.FromDateTime(DateTime
        .UtcNow);
21     public OrderStatus Status { get; set; } = OrderStatus.New;
22     public decimal Total { get; set; }
23     public Guid ClientId { get; init; }
24 }
25
26 public sealed class Service
27 {
28     public Guid Id { get; init; }
29     public string Title { get; set; } = string.Empty;
30     public decimal Price { get; set; }
31 }

```

## Робота зі SQLite (ADO.NET)

Listing 2: Ініціалізація SQLite та запити

```

1 using System.Data;
2 using Microsoft.Data.Sqlite; // dotnet add package Microsoft.Data.Sqlite
3
4 const string ConnectionString = "Data Source=atelier.db";
5
6 using var connection = new SqliteConnection(ConnectionString);
7 await connection.OpenAsync();

```

```

8
9 // Створення таблиць
10 var createSql = @"
11 CREATE TABLE IF NOT EXISTS Client (
12     Id TEXT PRIMARY KEY,
13     FullName TEXT NOT NULL,
14     Phone TEXT NOT NULL
15 );
16 CREATE TABLE IF NOT EXISTS Service (
17     Id TEXT PRIMARY KEY,
18     Title TEXT NOT NULL,
19     Price REAL NOT NULL
20 );
21 CREATE TABLE IF NOT EXISTS [Order] (
22     Id TEXT PRIMARY KEY,
23     Date TEXT NOT NULL,
24     Status INTEGER NOT NULL,
25     Total REAL NOT NULL,
26     ClientId TEXT NOT NULL REFERENCES Client(Id)
27 );
28 ";
29 await new SqliteCommand(createSql, connection).ExecuteNonQueryAsync();
30
31 // Вставка клієнта параметризовано
32 var insertClient = new SqliteCommand(
33     "INSERT INTO Client (Id, FullName, Phone) VALUES ($id, $name, $phone
34         )", connection);
35 insertClient.Parameters.AddWithValue("$id", Guid.NewGuid().ToString());
36 insertClient.Parameters.AddWithValue("$name", "Іван Петренко");
37 insertClient.Parameters.AddWithValue("$phone", "+380501112233");
38 await insertClient.ExecuteNonQueryAsync();
39
40 // Вибір казми піงому об'єкту
41 var clients = new List<Client>();
42 var select = new SqliteCommand("SELECT Id, FullName, Phone FROM Client",
43     connection);
44 using (var reader = await select.ExecuteReaderAsync(CommandBehavior.
45     CloseConnection))
46 {
47     while (await reader.ReadAsync())

```

```

45     {
46         clients.Add(new Client
47         {
48             Id = Guid.Parse(reader.GetString(0)),
49             FullName = reader.GetString(1),
50             Phone = reader.GetString(2)
51         });
52     }
53 }

```

## Мінімальний Web API (.NET)

Listing 3: Мінімальний ендпоінт для замовлень

```

1 var builder = WebApplication.CreateBuilder(args);
2 var app = builder.Build();
3
4 app.MapGet("/orders", async () =>
5 {
6     using var connection = new SqlConnection("Data Source=atelier.db"
7         );
8     await connection.OpenAsync();
9     using var cmd = new SqlCommand("SELECT Id, Date, Status, Total,
10         ClientId FROM [Order]", connection);
11     var list = new List<Order>();
12     using var reader = await cmd.ExecuteReaderAsync();
13     while (await reader.ReadAsync())
14     {
15         list.Add(new Order
16         {
17             Id = Guid.Parse(reader.GetString(0)),
18             Date = DateOnly.Parse(reader.GetString(1)),
19             Status = (OrderStatus)reader.GetInt32(2),
20             Total = (decimal)reader.GetDouble(3),
21             ClientId = Guid.Parse(reader.GetString(4))
22         });
23     }
24     return Results.Ok(list);
25 });

```

```
25 app.Run();
```

## 4.2. SQL (SQLite) схеми та запити

Listing 4: DDL і базові запити для SQLite

```
1  -- Створення таблиць
2  CREATE TABLE IF NOT EXISTS Client (
3      Id TEXT PRIMARY KEY,
4      FullName TEXT NOT NULL,
5      Phone TEXT NOT NULL
6  );
7
8  CREATE TABLE IF NOT EXISTS Service (
9      Id TEXT PRIMARY KEY,
10     Title TEXT NOT NULL,
11     Price REAL NOT NULL
12 );
13
14 CREATE TABLE IF NOT EXISTS [Order] (
15     Id TEXT PRIMARY KEY,
16     Date TEXT NOT NULL,
17     Status INTEGER NOT NULL,
18     Total REAL NOT NULL,
19     ClientId TEXT NOT NULL REFERENCES Client(Id)
20 );
21
22 -- Топ-5 популярних послуг
23 SELECT s.Title, COUNT(*) AS Times
24 FROM [Order] o
25 JOIN OrderItem oi ON oi.OrderId = o.Id
26 JOIN Service s ON s.Id = oi.ServiceId
27 GROUP BY s.Title
28 ORDER BY Times DESC
29 LIMIT 5;
```

## 5. Тестування

Запропоновано рівні тестування: модульне (сервісні методи калькуляції, валідації), інтеграційне (взаємодія API та БД), системне (сценарії оформлення замовлення, оплати), приймальне (з користувачами). Для нефункціональних аспектів — навантажувальне тестування ключових ендпойнтів.

### 5.1. Статистичні графіки

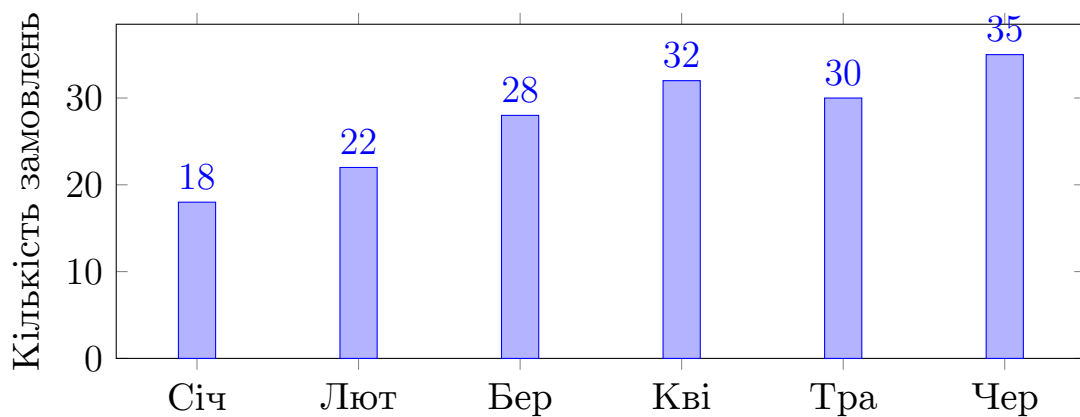


Рис. 7: Динаміка замовлень за півріччя

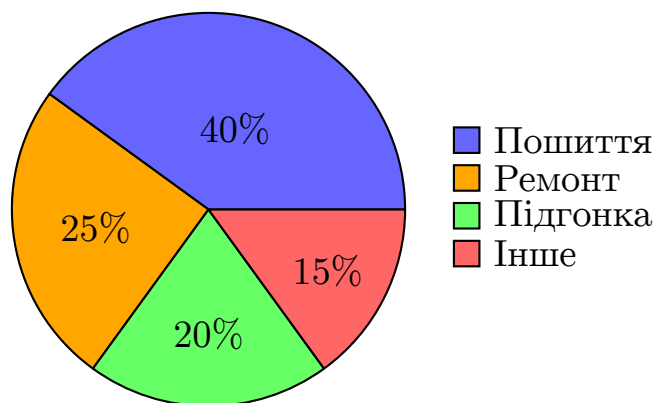


Рис. 8: Розподіл типів послуг

## 6. Висновки

Було проаналізовано предметну область ательє, уточнено вимоги та розроблено модель системи за допомогою UML і ER-діаграм. Запро-

поновано архітектуру рішення та підходи до тестування. Результати можуть бути використані як основа для подальшої детальної реалізації.

## Список використаних джерел

### Література

- [1] М. Фаулер. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, 2003.
- [2] Т. Конноллі, К. Бегг. Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson, 2014.
- [3] Вимоги до оформлення кваліфікаційних робіт за ДСТУ. Доступ: <https://example.edu/dstu-guidelines> (дата звернення: 24.10.2025).