

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
1 Вступ	1
2 Аналіз предметної області та вимог	1
2.1 Опис предметної області	1
2.2 Зацікавлені сторони	1
2.3 Функціональні вимоги	2
2.4 Нефункціональні вимоги	2
3 Проєктування системи	2
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
4 Формалізація вимог	6
4.1 Функціональні вимоги	6
4.2 Нефункціональні вимоги	8
4.3 Опис ключових прецедентів	9
5 Модель даних і словник даних	10
5.1 Нормалізація та обмеження цілісності	11
6 Додаткові діаграми	12
6.1 Діаграма станів замовлення	12
6.2 Діаграма розгортання	12
7 План-графік робіт	13
8 Аналітика та візуалізація	14
8.1 Виручка за місяцями	14

8.2	Завантаженість майстрів (стек-бар)	14
9	Реалізація	15
9.1	Фрагменти коду C#	15
9.2	SQL (SQLite) схеми та запити	18
10	Тестування	19
10.1	Статистичні графіки	20
11	Висновки	20
	Список використаних джерел	21

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.

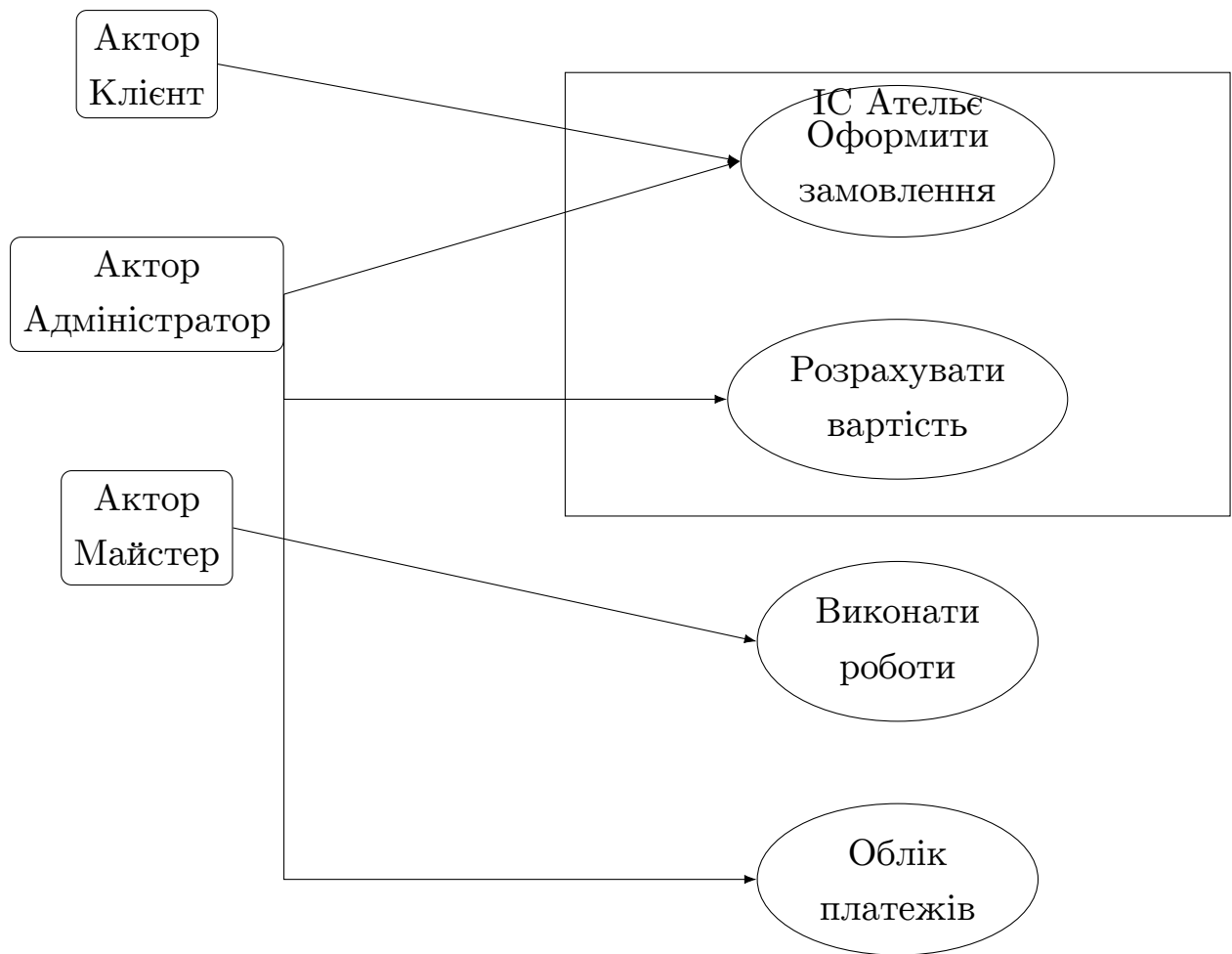


Рис. 1: Діаграма прецедентів для ІС ательє

3.2. Діаграма активності: обробка замовлення

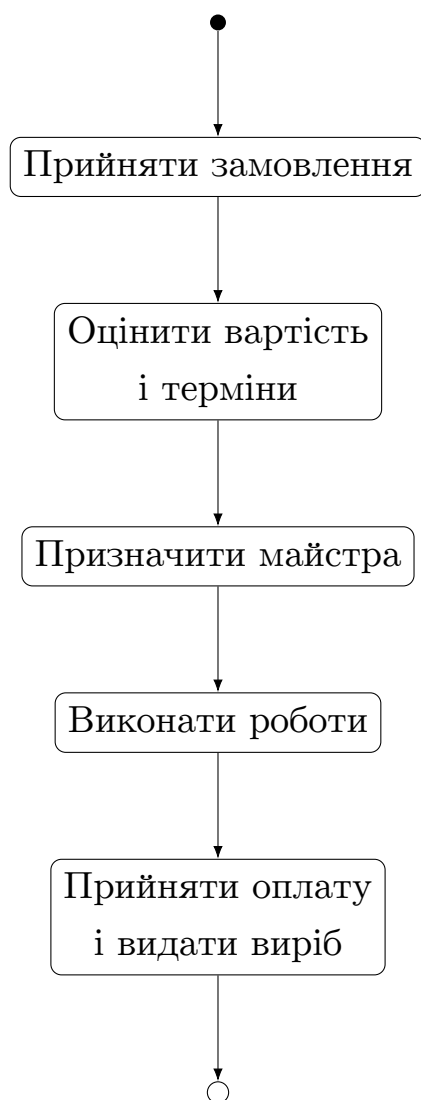


Рис. 2: Діаграма активності процесу замовлення

3.3. Діаграма класів

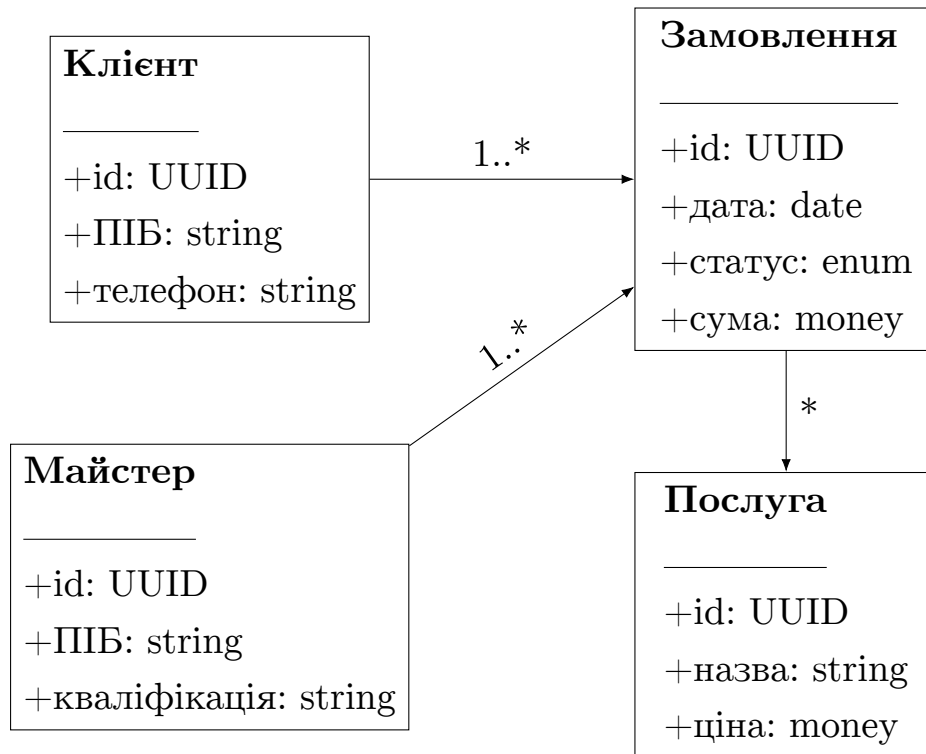


Рис. 3: Спрощена діаграма класів

3.4. Діаграма послідовностей: оформлення замовлення

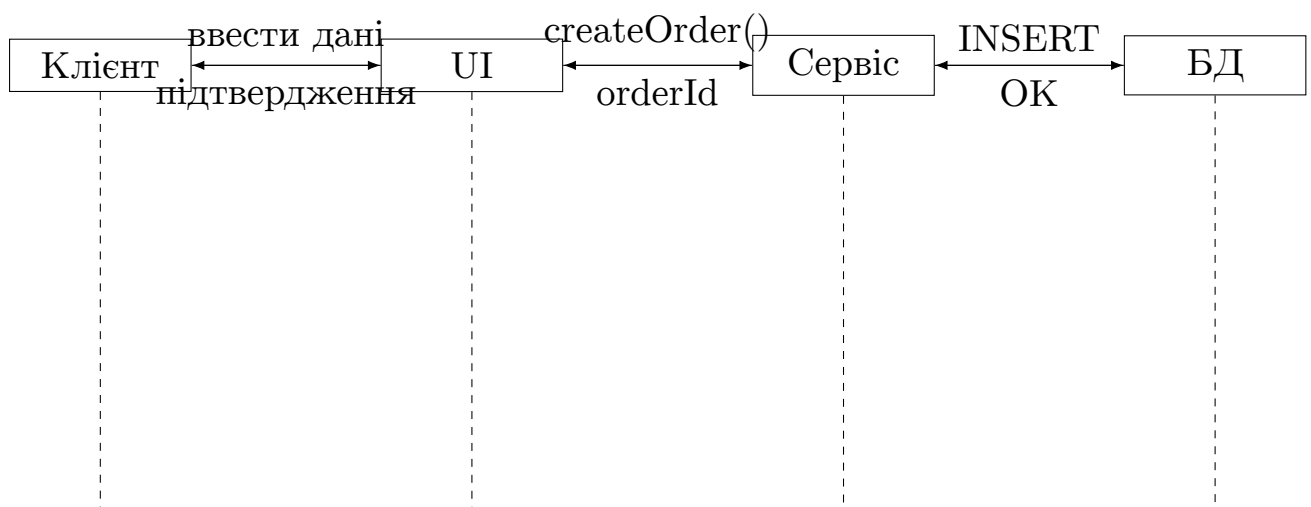


Рис. 4: Діаграма послідовностей

3.5. ER-діаграма бази даних

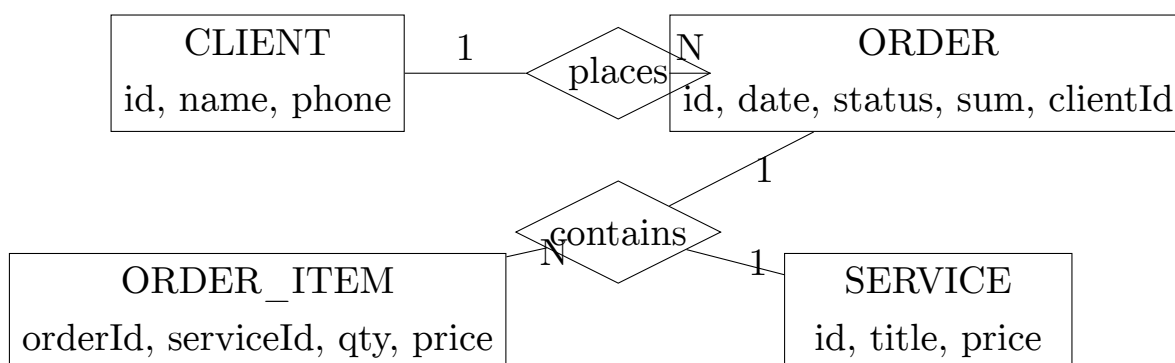


Рис. 5: ER-діаграма БД

3.6. Діаграма компонентів (структури)



Рис. 6: Діаграма компонентів

4. Формалізація вимог

4.1. Функціональні вимоги

Нижче наведено перелік основних функціональних вимог (FR). Для кожної вимоги вказано короткий опис, пріоритет і критерії приймання.

ID	Назва	Опис	Пріор.	Критерії
FR-1	Реєстрація клієнта	Система дозволяє створити клієнта з ПІБ і телефоном.	Високий	Після збереження; валідація
FR-2	Пошук клієнтів	Пошук клієнтів за ПІБ/телефоном з частковим збігом.	Високий	Пошук по частковій назві; записи
FR-3	Створення замовлення	Оформлення нового замовлення для обраного клієнта.	Високий	Присвоєння ID; статус=Ново.

ID	Назва	Опис	Пріор.	Критері
FR-4	Додавання послуг	Додавання позицій послуг до замовлення з ціною і кількістю.	Високий	Позиції з загальною сума на.
FR-5	Калькуляція вартості	Розрахунок загальної вартості (роботи + матеріали).	Високий	Загальна вартість з то
FR-6	Редагування замовлення	Зміна складу, термінів, виконавця до старту робіт.	Середній	Зміни логічного обмеження
FR-7	Призначення майстра	Призначення відповідального майстра із графіка завантаження.	Високий	Врахування завантаження; мує задачі
FR-8	Етапи виконання	Фіксація етапів: прийом, кроїння, пошиття, примірка, видача.	Середній	Хід від початку до кінця шкалі проектування.
FR-9	Облік матеріалів	Списання матеріалів з партій та складу.	Середній	Залишки матеріалів шуються; від'ємний баланс
FR-10	Облік платежів	Фіксація передплати й остаточного розрахунку.	Високий	Сальдо=сума дебетових дачі; квітень закінчується.
FR-11	Скасування замовлення	Скасування з причиною; відкат бронювань, повернення коштів.	Середній	Лог змін; історія; сові пров
FR-12	Видача виробу	Завершення робіт, зміна статусу на Delivered.	Високий	Статус змін; дачі зафі
FR-13	Звіти по виручці	Формування звіту по місяцях і послугах.	Середній	Дані від клієнтів; жам і ста
FR-14	Звіт по завантаженню	Звіт по зайнятості майстрів у періоді.	Середній	Відсоток зайнятості; ння об'єкту; задачі
FR-15	Прайс-лист	Управління переліком послуг і тарифами.	Високий	Історія змін; ться.

ID	Назва	Опис	Пріор.	Критері
FR-16	Резервне копіювання	Експорт БД до файлу; імпорт з резерву.	Низький	Резерв с новлення стентний
FR-17	Ролі та доступ	Ролі: адміністратор, майстер, власник.	Високий	Доступи триці пра
FR-18	Аудит	Лог змін замовлень і платежів.	Середній	Видно хт нив.
FR-19	Повідомлення клієнту	Сповіщення (email/SMS) про етапи й готовність.	Низький	Шаблони ження до
FR-20	Примірки	Планування і фіксація результатів примірок.	Середній	Коментар історія до
FR-21	Фото виробу	Збереження фото дефектів/-етапів для історії.	Низький	Фото пов нням; вид
FR-22	Знижки та промо	Застосування знижок і промокодів.	Низький	У чеку п тотал кор
FR-23	Податки	Нарахування ПДВ за ставкою.	Низький	Сума ПД й відобра
FR-24	Повернення коштів	Оформлення повернення з підставою.	Низький	Коректні тус плате
FR-25	Експорт звітів	Експорт у CSV/XLSX основних звітів.	Низький	Формат в ння UTF

4.2. Нефункціональні вимоги

ID	Атрибут якості	Опис	П
NFR-1	Продуктивність	95-й перцентиль відповіді API < 300 мс для типових операцій.	В
NFR-2	Доступність	99.5% на місяць; відновлення після збою < 15 хв.	В
NFR-3	Безпека	Паролі хешуються (Argon2/BCrypt); JWT; ролева авторизація.	В

ID	Атрибут якості	Опис	П
NFR-4	Надійність даних	Транзакції з ACID; регулярні резервні копії.	В
NFR-5	Масштабованість	Горизонтальне масштабування API; БД — вертикальне.	С
NFR-6	Супровідність	Документація OpenAPI; логування та тра- сування.	С
NFR-7	UX	Послідовний дизайн; доступність WCAG 2.1 AA.	С
NFR-8	Сумісність	Підтримка останніх версій Chrome/Firefox/Edge.	С
NFR-9	Локалізація	Українська локаль за замовчуванням; підтримка i18n.	Н
NFR-10	Аудит	Історія змін критичних сутностей (замовлення, платежі).	В

4.3. Опис ключових прецедентів

UC-1: Оформити замовлення Мета — створити замовлення для клієнта, визначити перелік робіт, розрахувати вартість і терміни. Основний потік: вибір клієнта → введення виробу й послуг → розрахунок → збереження. Винятки: невалідні дані; недоступні матеріали.

UC-2: Виконати роботи Мета — забезпечити виконання етапів з фіксацією часу та витрат. Потік: призначення майстра → етапи → контроль якості → готовність.

UC-3: Прийняти оплату Мета — облік передплати/остаточного розрахунку. Потік: генерація рахунку → прийом платежу → оновлення сальдо.

5. Модель даних і словник даних

Наведено словник даних (Data Dictionary) для основних таблиць. Типи подано для SQLite; у промисловій СУБД можуть відрізнятися.

Таблиця	Поле	Тип	Null	За замовч.	Опис
Client	Id	TEXT (UUID)	Hi	—	Ідентифікатор клієнта
Client	FullName	TEXT	Hi	—	ПІБ клієнта
Client	Phone	TEXT	Hi	—	Контактний номер телефону клієнта
Service	Id	TEXT (UUID)	Hi	—	Ідентифікатор послуги
Service	Title	TEXT	Hi	—	Назва послуги
Service	Price	REAL	Hi	0	Базова ціна послуги
[Order]	Id	TEXT (UUID)	Hi	—	Ідентифікатор замовлення
[Order]	Date	TEXT (ISO)	Hi	now	Дата створення замовлення
[Order]	Status	INTEGER	Hi	0	Статус замовлення
[Order]	Total	REAL	Hi	0	Загальна сума замовлення
[Order]	ClientId	TEXT	Hi	—	Посилання на клієнта
OrderItem	OrderId	TEXT	Hi	—	Посилання на замовлення
OrderItem	ServiceId	TEXT	Hi	—	Посилання на послугу
OrderItem	Qty	REAL	Hi	1	Кількість послуги
OrderItem	Price	REAL	Hi	—	Ціна за одиницю послуги
Payment	Id	TEXT (UUID)	Hi	—	Ідентифікатор платежу
Payment	OrderId	TEXT	Hi	—	Посилання на замовлення
Payment	Date	TEXT (ISO)	Hi	now	Дата платежу
Payment	Amount	REAL	Hi	—	Сума платежу
Payment	Method	TEXT	Hi	cash	Метод платежу

Таблиця	Поле	Тип	Null	За замовч.	Опис
Master	Id	TEXT (UUID)	Hi	—	Ідентифі
Master	FullName	TEXT	Hi	—	ПІБ май
Master	Skill	TEXT	Так	—	Кваліфік
Material	Id	TEXT (UUID)	Hi	—	Ідентифі
Material	Title	TEXT	Hi	—	Назва ма
Material	Qty	REAL	Hi	0	Залишок
Material	Unit	TEXT	Hi	pcs	Одиниця
AuditLog	Id	INTEGER	Hi	autoinc	Первинн
AuditLog	Entity	TEXT	Hi	—	Назва су
AuditLog	EntityId	TEXT	Hi	—	Ідентифі
AuditLog	Action	TEXT	Hi	—	Дія (CRI
AuditLog	ChangedAt	TEXT (ISO)	Hi	now	Час змін
AuditLog	UserId	TEXT	Так	—	Виконав
					ний).

5.1. Нормалізація та обмеження цілісності

Схема відповідає щонайменше 3НФ: довідники винесено окремо, Order-OrderItem — зв'язок $1:N$, суми та ПДВ — похідні поля. Референційна цілісність забезпечується зовнішніми ключами; тотал замовлення перераховується тригерами після змін позицій.

6. Додаткові діаграми

6.1. Діаграма станів замовлення

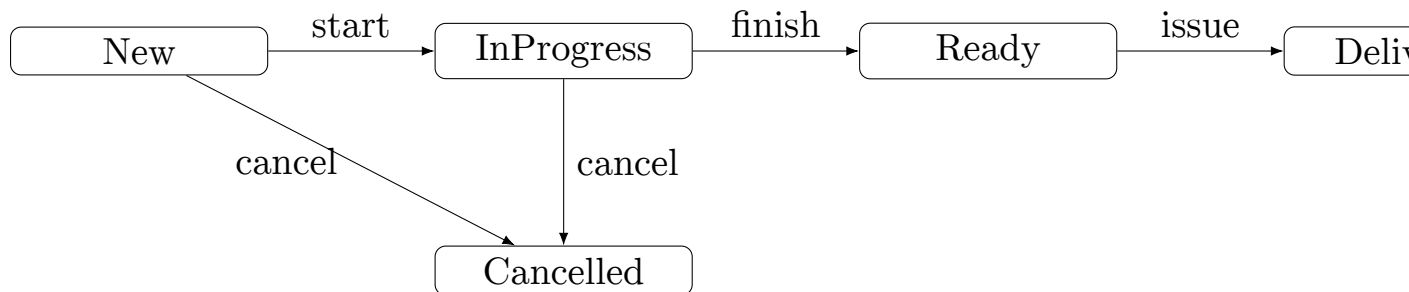


Рис. 7: Станова діаграма життєвого циклу замовлення

6.2. Діаграма розгортання

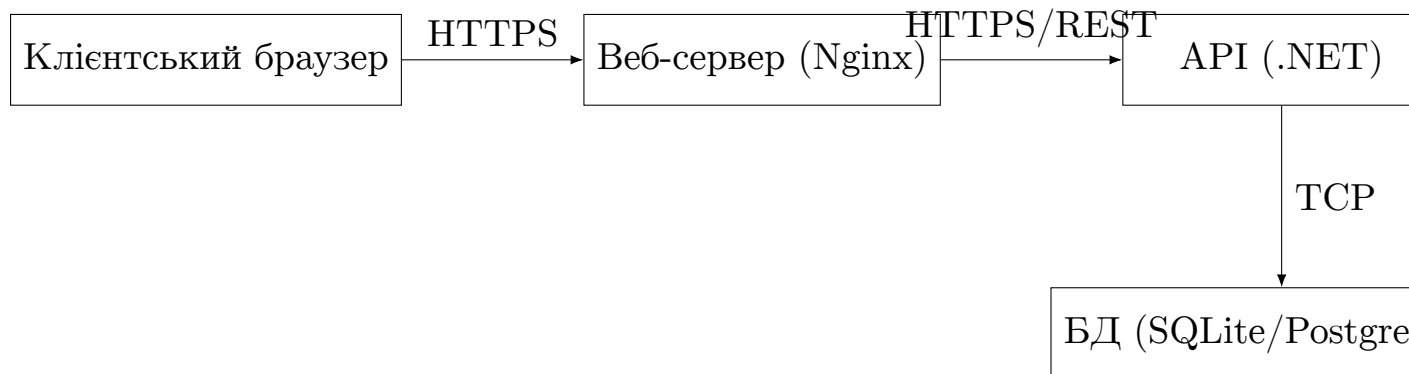


Рис. 8: Схема розгортання компонентів

7. План-графік робіт

	2005
	07
Аналіз вимог	
Проектування	
Реалізація	
Тестування	
Впровадження	
-01-06	

8. Аналітика та візуалізація

8.1. Виручка за місяцями

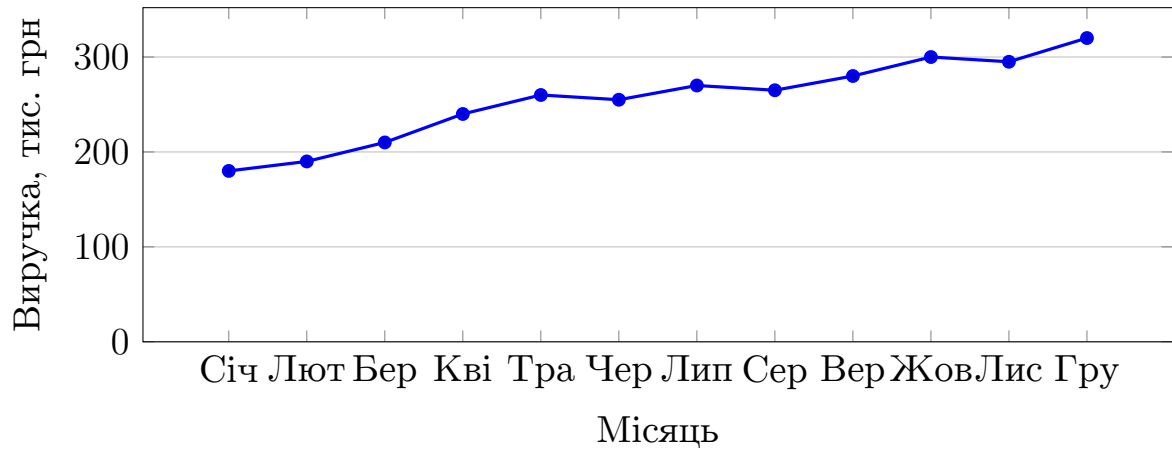


Рис. 9: Місячна виручка протягом року

8.2. Завантаженість майстрів (стек-бар)

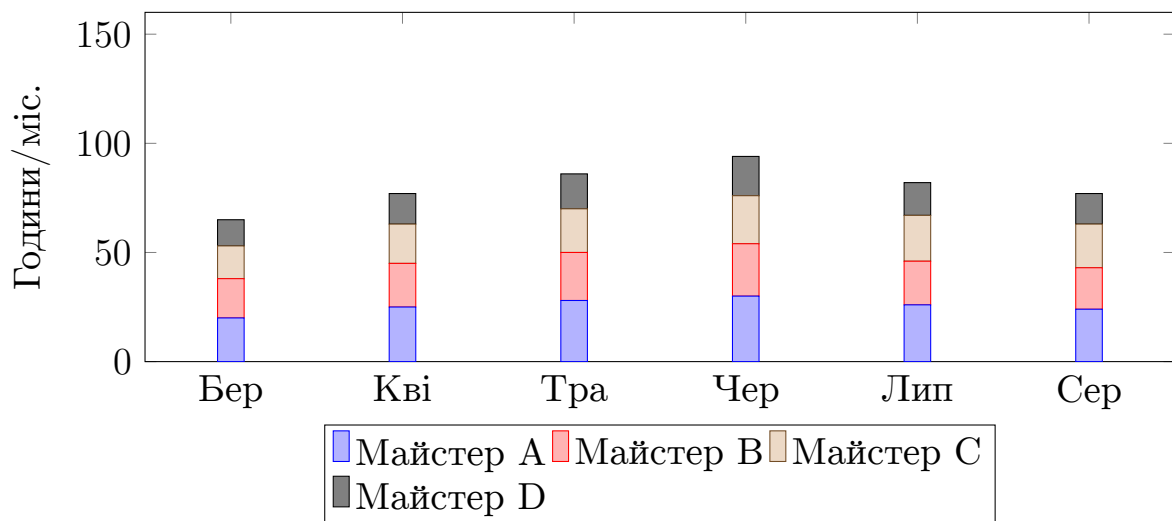


Рис. 10: Навантаження майстрів по місяцях

9. 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.

9.1. 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 SqlCommand(createSql, connection).ExecuteNonQueryAsync();
30
31 // Вставка клієнта параметризовано
32 var insertClient = new SqlCommand(
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 каз мапінгу об'єкти
41 var clients = new List<Client>();
42 var select = new SqlCommand("SELECT Id, FullName, Phone FROM Client",
43     connection);
44 using (var reader = await select.ExecuteReaderAsync(CommandBehavior.
45     CloseConnection))
46 {
47     while (await reader.ReadAsync())
48     {
49         clients.Add(new Client
50         {
51             Id = Guid.Parse(reader.GetString(0)),
52             FullName = reader.GetString(1),
53             Phone = reader.GetString(2)
54         });
55     }
56 }

```

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 SqliteConnection("Data Source=atelier.db"
7          );
8      await connection.OpenAsync();
9      using var cmd = new SqliteCommand("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  });
26
27  app.Run();

```

9.2. 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;

```

10. 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.

10.1. Статистичні графіки

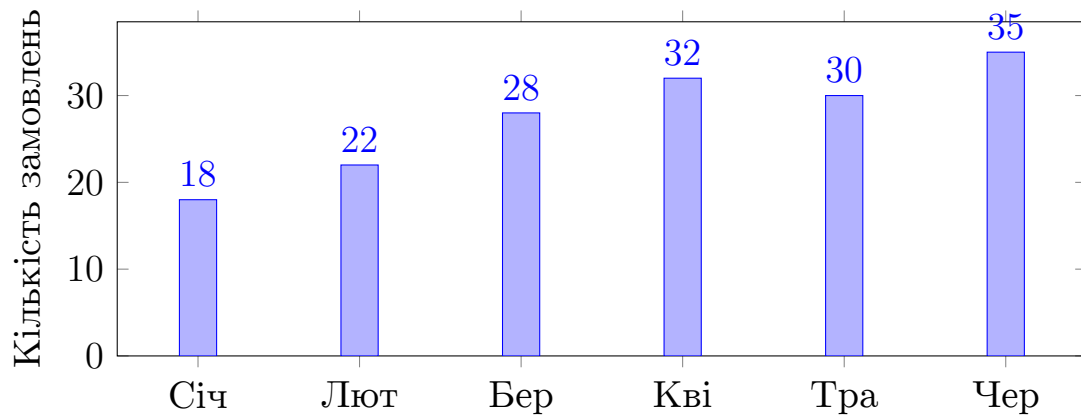


Рис. 11: Динаміка замовлень за півріччя

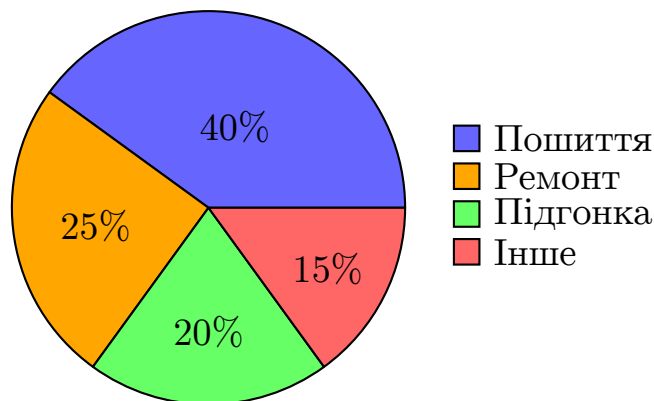


Рис. 12: Розподіл типів послуг

11. Висновки

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