

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
[НАЗВА УНІВЕРСИТЕТУ]

Факультет [назва факультету]
Кафедра [назва кафедри]

КУРСОВА РОБОТА

з дисципліни «[Назва дисципліни]»

на тему:

«ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ АТЕЛЬЄ»

Виконав:
студент групи [номер групи]
Б студента

Перевірив:
Б викладача
сада, науковий ступінь

Зміст

1	ВСТУП	3
1.1	Актуальність теми	3
1.2	Мета та завдання курсової роботи	3
1.3	Об'єкт та предмет дослідження	4
2	АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	5
2.1	Опис предметної області	5
2.2	Основні сутності предметної області	5
2.3	Функціональні вимоги до системи	6
3	ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	7
3.1	Діаграма прецедентів	7
3.2	Діаграма класів	8
3.3	Модель бази даних	9
3.3.1	Опис таблиць бази даних	9
3.4	Діаграма діяльності	13
3.5	Архітектура системи	14
3.5.1	Діаграма компонентів	14
3.6	Діаграма послідовності	15
3.7	Діаграма станів	16
4	РЕАЛІЗАЦІЯ СИСТЕМИ	17
4.1	Технології та інструменти	17
4.2	Структура проекту	18
4.3	SQL-скрипти створення таблиць	19
4.4	Приклади коду	23
5	ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ	25
5.1	План тестування	25
5.2	Приклади тестових сценаріїв	25
5.3	Впровадження системи	26
6	ВИСНОВКИ	28
7	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	30

А ДОДАТОК А. Діаграма розгортання	31
Б ДОДАТОК Б. Словник термінів	31

1 ВСТУП

1.1 Актуальність теми

Сучасний розвиток інформаційних технологій створює нові можливості для автоматизації бізнес-процесів у різних галузях економіки. Ательє як підприємство, що надає послуги з пошиття та ремонту одягу, потребує ефективної системи управління для підвищення якості обслуговування клієнтів та оптимізації виробничих процесів.

Актуальність розробки інформаційної системи для ательє обумовлена наступними факторами:

- необхідність автоматизації обліку замовлень та клієнтів;
- потреба у контролі виконання замовлень та завантаженості працівників;
- важливість ведення складського обліку матеріалів;
- необхідність формування звітності для аналізу діяльності підприємства;
- підвищення конкурентоспроможності за рахунок покращення якості обслуговування.

1.2 Мета та завдання курсової роботи

Мета роботи: розробка інформаційної системи управління ательє для автоматизації основних бізнес-процесів підприємства.

Завдання роботи:

1. Проаналізувати предметну область та виявити основні бізнес-процеси ательє;
2. Визначити функціональні вимоги до інформаційної системи;
3. Розробити діаграми прецедентів для визначення взаємодії користувачів з системою;
4. Створити діаграму класів для опису структури системи;
5. Розробити модель бази даних;

6. Описати архітектуру системи;
7. Розробити діаграми діяльності для ключових бізнес-процесів.

1.3 Об'єкт та предмет дослідження

Об'єкт дослідження: бізнес-процеси ательє з пошиття та ремонту одягу.

Предмет дослідження: методи та засоби автоматизації управління діяльністю ательє.

2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Опис предметної області

Ательє — це підприємство, що надає послуги з пошиття одягу на замовлення, ремонту та підгонки одягу за розміром клієнта. Основними процесами ательє є:

- **Прийом замовлень** — консультація клієнта, зняття мірок, вибір тканини та фасону, визначення вартості та термінів виконання;
- **Виробництво** — розкрій тканини, пошиття виробу, примірка та коригування;
- **Видача замовлень** — перевірка якості, розрахунок з клієнтом, видача готового виробу;
- **Складський облік** — облік тканин, фурнітури та інших матеріалів;
- **Управління персоналом** — розподіл замовлень між кравцями, облік робочого часу.

2.2 Основні сутності предметної області

1. **Клієнт** — особа, яка замовляє послуги ательє;
2. **Замовлення** — конкретне замовлення клієнта на виготовлення або ремонт виробу;
3. **Працівник** — співробітник ательє (кравець, закрійник, адміністратор);
4. **Послуга** — вид роботи (пошиття сукні, ремонт брюк, підгонка піджака тощо);
5. **Матеріал** — тканина, фурнітура та інші матеріали;
6. **Мірки** — індивідуальні параметри клієнта.

2.3 Функціональні вимоги до системи

Інформаційна система управління ательє повинна забезпечувати:

1. Управління клієнтами:

- реєстрація нових клієнтів;
- зберігання контактної інформації;
- зберігання мірок клієнтів;
- історія замовлень клієнта.

2. Управління замовленнями:

- створення нових замовлень;
- відстеження статусу виконання;
- призначення виконавців;
- розрахунок вартості;
- контроль термінів виконання.

3. Управління матеріалами:

- облік наявності матеріалів на складі;
- облік витрачання матеріалів на замовлення;
- контроль залишків та формування замовлень постачальникам.

4. Управління персоналом:

- облік працівників;
- розподіл замовлень між виконавцями;
- контроль завантаженості.

5. Звітність:

- звіт по виконаних замовленнях;
- фінансовий звіт;
- звіт по завантаженості працівників;
- звіт по залишках матеріалів.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Діаграма прецедентів

Діаграма прецедентів (use case diagram) відображає взаємодію користувачів з системою та основні функції, які вона надає.

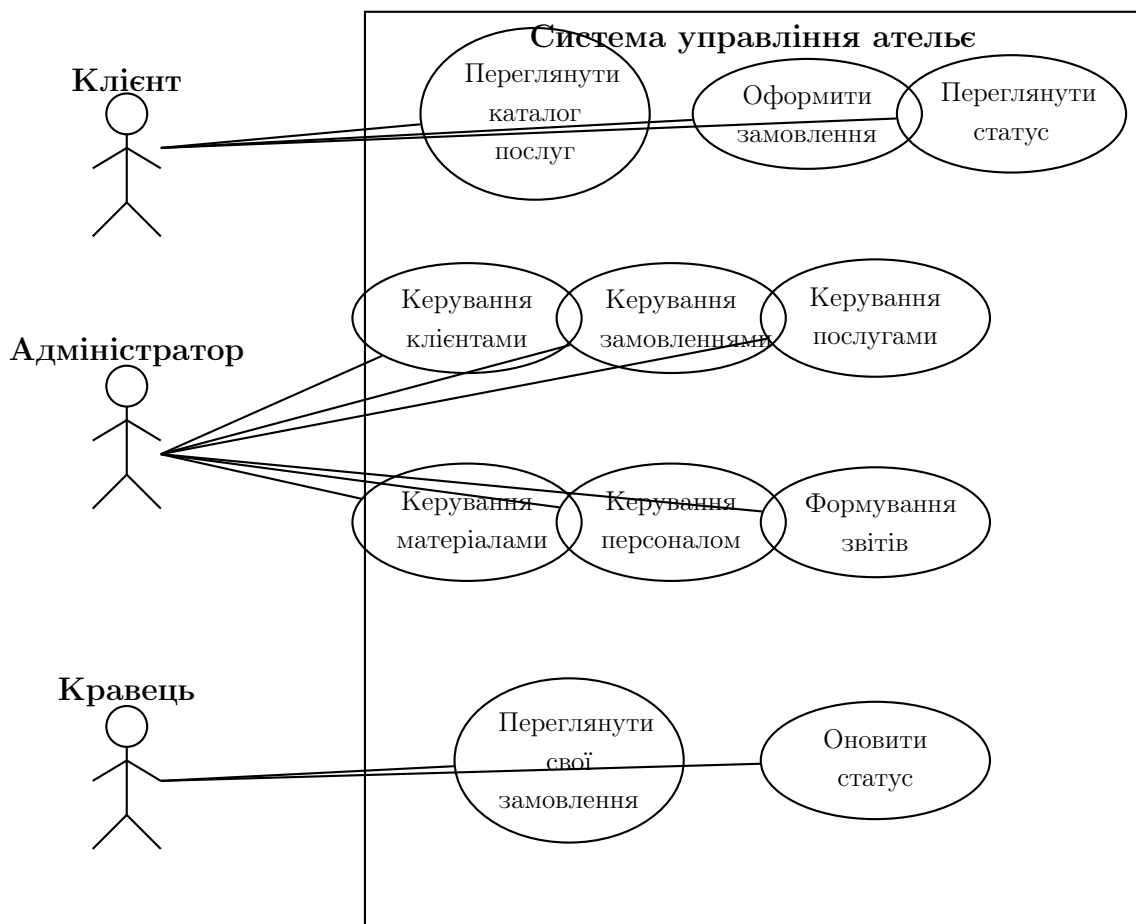


Рис. 1: Діаграма прецедентів інформаційної системи ательє

Опис основних прецедентів:

- **Оформити замовлення** — клієнт або адміністратор створює нове замовлення, вказуючи необхідні послуги, матеріали та терміни виконання;
- **Керування замовленнями** — адміністратор може створювати, редагувати та видаляти замовлення, призначати виконавців;
- **Оновити статус** — кравець змінює статус замовлення по мірі виконання роботи;
- **Формування звітів** — адміністратор формує різні види звітів для аналізу діяльності.

3.2 Діаграма класів

Діаграма класів відображає структуру системи, основні класи та зв'язки між ними.

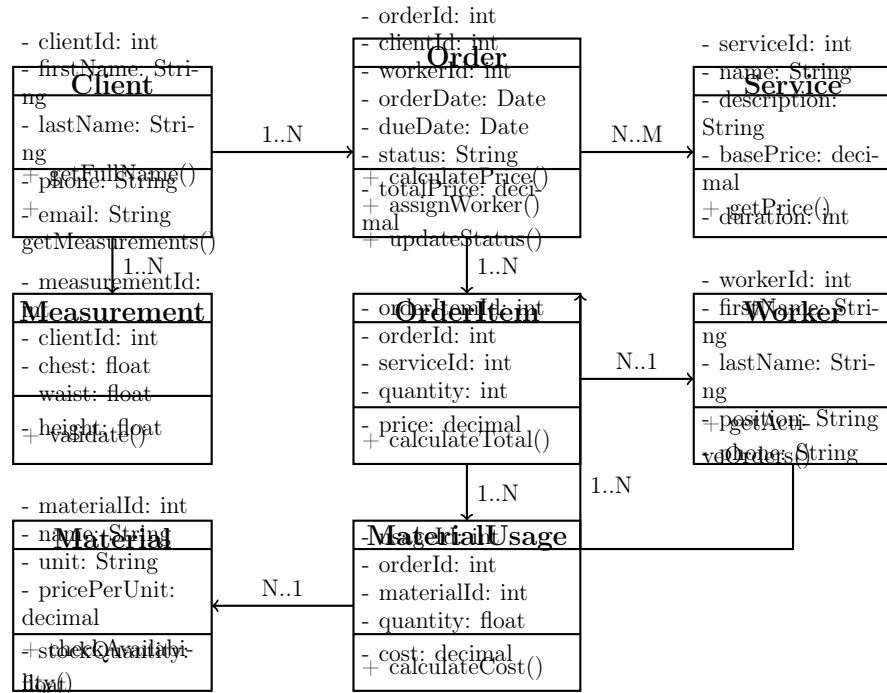


Рис. 2: Діаграма класів інформаційної системи

3.3 Модель бази даних

Концептуальна модель бази даних представлена у вигляді ER-діаграми, яка відображає основні сутності та зв'язки між ними.

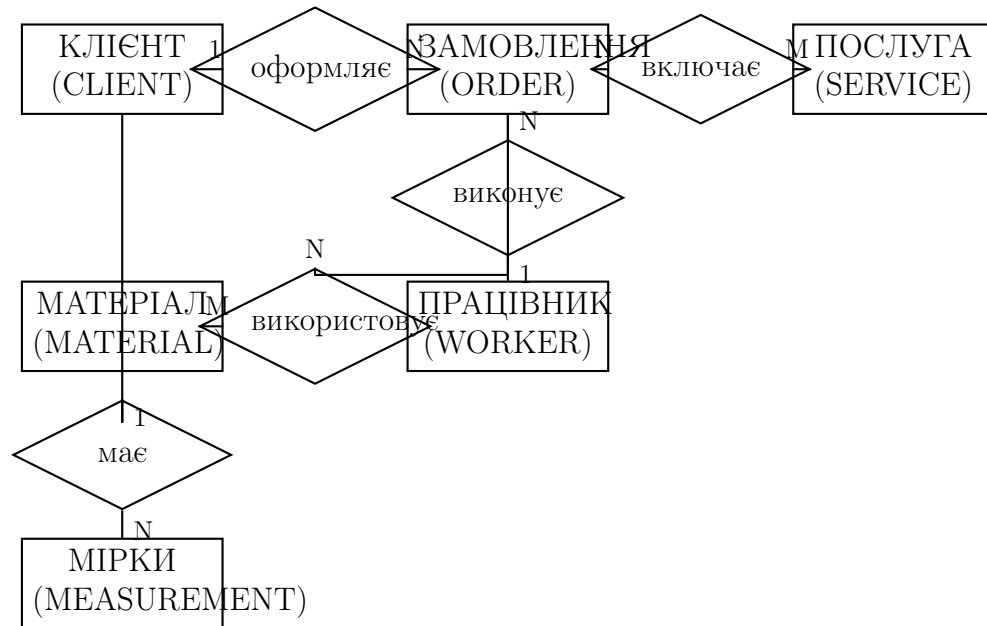


Рис. 3: ER-діаграма бази даних

3.3.1 Опис таблиць бази даних

Таблиця CLIENT (Клієнти):

- client_id (PK) — унікальний ідентифікатор клієнта
- first_name — ім'я
- last_name — прізвище
- phone — номер телефону
- email — електронна пошта
- registration_date — дата реєстрації
- notes — додаткові примітки

Таблиця MEASUREMENT (Мірки):

- measurement_id (PK) — унікальний ідентифікатор

- client_id (FK) — посилання на клієнта
- chest — обхват грудей
- waist — обхват талії
- hips — обхват стегон
- height — зріст
- measurement_date — дата зняття мірок

Таблиця WORKER (Працівники):

- worker_id (PK) — унікальний ідентифікатор
- first_name — ім'я
- last_name — прізвище
- position — посада
- phone — номер телефону
- hire_date — дата прийняття на роботу
- salary — заробітна плата

Таблиця SERVICE (Послуги):

- service_id (PK) — унікальний ідентифікатор
- name — назва послуги
- description — опис
- base_price — базова ціна
- duration_hours — тривалість виконання
- category — категорія послуги

Таблиця ORDER (Замовлення):

- order_id (PK) — унікальний ідентифікатор

- client_id (FK) — посилання на клієнта
- worker_id (FK) — посилання на виконавця
- order_date — дата оформлення
- due_date — термін виконання
- completion_date — дата завершення
- status — статус (новий, в роботі, готовий, виданий)
- total_price — загальна вартість
- notes — примітки

Таблиця ORDER_ITEM (Позиції замовлення):

- order_item_id (PK) — унікальний ідентифікатор
- order_id (FK) — посилання на замовлення
- service_id (FK) — посилання на послугу
- quantity — кількість
- price — ціна
- description — опис

Таблиця MATERIAL (Матеріали):

- material_id (PK) — унікальний ідентифікатор
- name — назва матеріалу
- unit — одиниця виміру
- price_per_unit — ціна за одиницю
- stock_quantity — кількість на складі
- min_stock_level — мінімальний рівень запасів
- supplier — постачальник

Таблиця MATERIAL_USAGE (Використання матеріалів):

- usage_id (PK) — унікальний ідентифікатор
- order_id (FK) — посилання на замовлення
- material_id (FK) — посилання на матеріал
- quantity — кількість використаного матеріалу
- cost — вартість
- usage_date — дата використання

3.4 Діаграма діяльності

Діаграма діяльності для процесу оформлення та виконання замовлення.

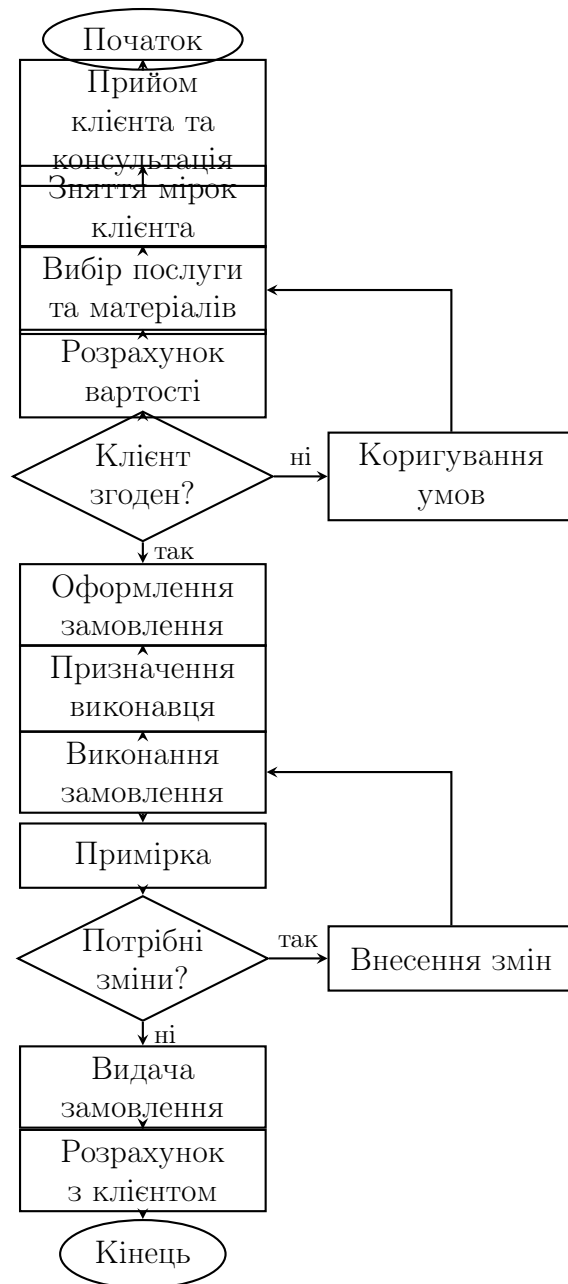


Рис. 4: Діаграма діяльності: процес оформлення та виконання замовлення

3.5 Архітектура системи

Інформаційна система управління ательє побудована за трирівневою архітектурою:

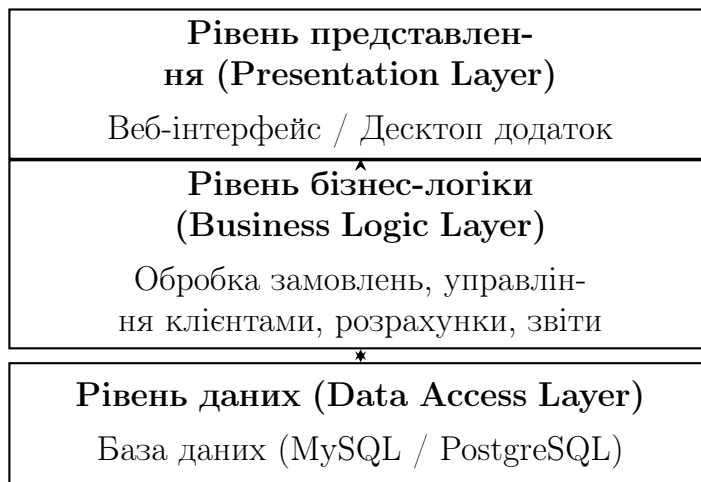


Рис. 5: Трирівнева архітектура системи

Опис рівнів:

1. **Рівень представлення:** забезпечує взаємодію користувача з системою через графічний інтерфейс. Може бути реалізований як веб-додаток або десктоп-програма.
2. **Рівень бізнес-логіки:** містить основну логіку роботи системи, обробляє запити користувачів, виконує валідацію даних, здійснює розрахунки та формує звіти.
3. **Рівень даних:** відповідає за зберігання та отримання даних з бази даних, забезпечує цілісність та безпеку інформації.

3.5.1 Діаграма компонентів

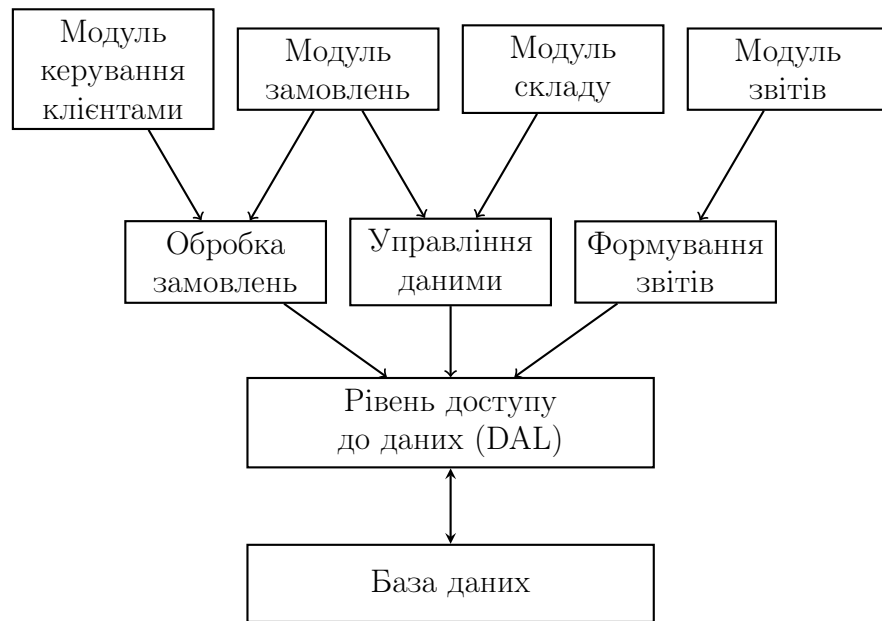


Рис. 6: Діаграма компонентів системи

3.6 Діаграма послідовності

Діаграма послідовності для процесу створення нового замовлення.

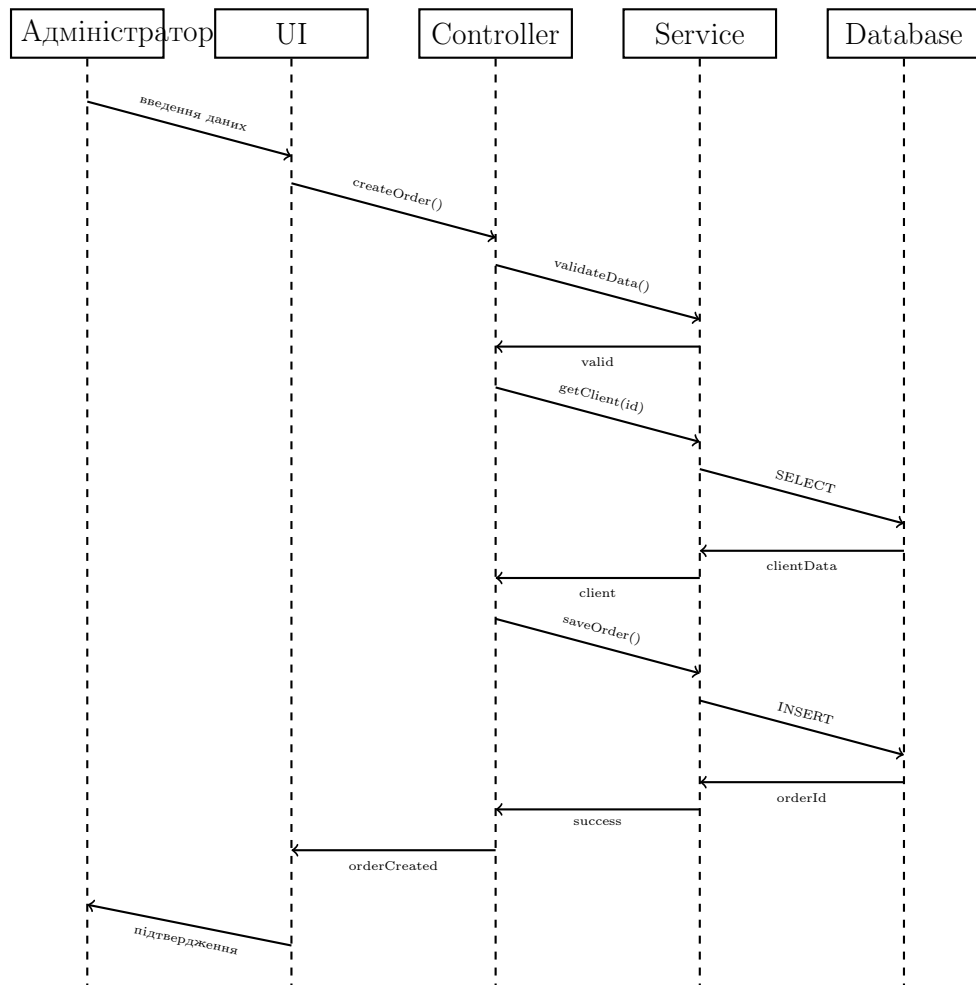


Рис. 7: Діаграма послідовності: створення замовлення

3.7 Діаграма станів

Діаграма станів для сутності "Замовлення" відображає можливі стани замовлення та переходи між ними.

Опис станів:

- **Новий** — замовлення створено, але ще не прийнято в роботу;
- **Прийнято в роботу** — замовлення призначено виконавцю;
- **В процесі виконання** — виконавець працює над замовленням;
- **Примірка** — клієнт проводить примірку виробу;
- **Готовий** — виріб готовий до видачі;
- **Виданий** — замовлення видано клієнту;
- **Скасовано** — замовлення скасовано клієнтом або адміністратором.

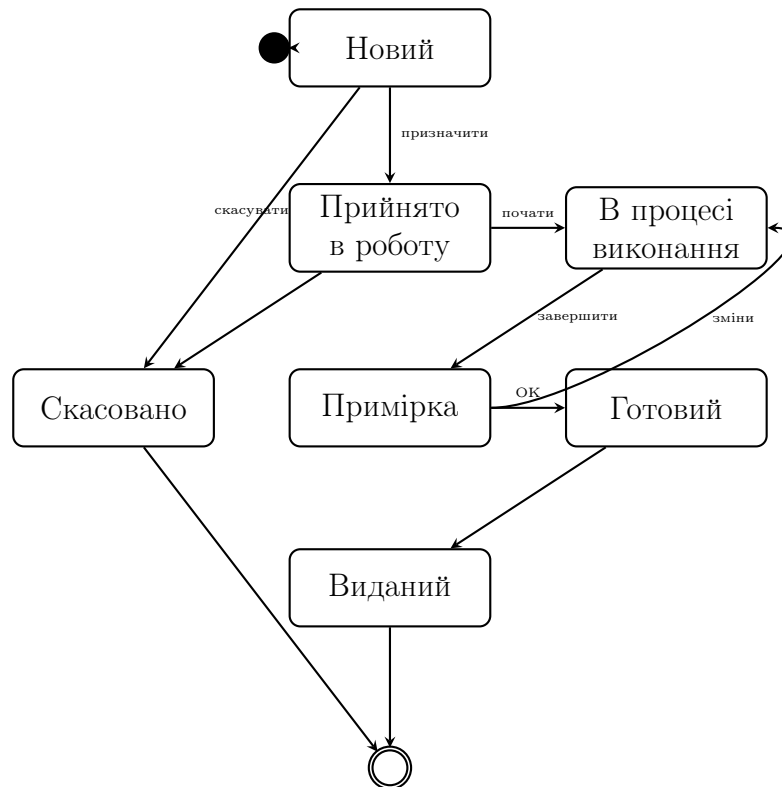


Рис. 8: Діаграма станів замовлення

4 РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 Технології та інструменти

Для реалізації інформаційної системи управління ательє пропонується використовувати наступний стек технологій:

Backend (серверна частина):

- Мова програмування: Java / Python / C#
- Фреймворк: Spring Boot / Django / ASP.NET Core
- ORM: Hibernate / Django ORM / Entity Framework
- База даних: PostgreSQL / MySQL

Frontend (клієнтська частина):

- HTML5, CSS3, JavaScript
- Фреймворк: React / Angular / Vue.js
- UI-бібліотека: Material-UI / Bootstrap

Інструменти розробки:

- Система контролю версій: Git
- IDE: IntelliJ IDEA / Visual Studio Code / PyCharm
- Інструменти для тестування: JUnit / pytest / NUnit
- Документування API: Swagger / OpenAPI

4.2 Структура проекту

Приклад структури проекту для веб-додатку:

```
atelier-system/  
  backend/  
    src/  
      controllers/      # Контролери  
      models/           # Моделі даних  
      services/         # Бізнес-логіка  
      repositories/     # Робота з БД  
      config/           # Конфігурація  
      tests/            # Тести  
  frontend/  
    src/  
      components/      # React компоненти  
      pages/           # Сторінки  
      services/        # API сервіси  
      utils/           # Утиліти  
    public/            # Статичні файли  
  database/  
    migrations/         # Міграції БД  
    seeds/              # Початкові дані
```

4.3 SQL-скрипти створення таблиць

Лістинг 1: Створення таблиць бази даних

```
-- Tablycyu klientiv
CREATE TABLE client (
    client_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    phone VARCHAR(20),
    email VARCHAR(100),
    registration_date DATE DEFAULT CURRENT_DATE,
    notes TEXT
);

-- Tablycyu mirok
CREATE TABLE measurement (
    measurement_id SERIAL PRIMARY KEY,
    client_id INTEGER REFERENCES client(client_id),
    chest DECIMAL(5,2),
    waist DECIMAL(5,2),
    hips DECIMAL(5,2),
    height DECIMAL(5,2),
    measurement_date DATE DEFAULT CURRENT_DATE
);

-- Tablycyu pratsivnykiv
CREATE TABLE worker (
    worker_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    position VARCHAR(50),
    phone VARCHAR(20),
    hire_date DATE,
    salary DECIMAL(10,2)
);

-- Tablycyu posluh
CREATE TABLE service (
    service_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
```

```
base_price DECIMAL(10,2),  
duration_hours INTEGER,  
category VARCHAR(50)  
);
```

Лістинг 2: Продовження створення таблиць

```
-- Tablycyia zamovlen
CREATE TABLE "order" (
    order_id SERIAL PRIMARY KEY,
    client_id INTEGER REFERENCES client(client_id),
    worker_id INTEGER REFERENCES worker(worker_id),
    order_date DATE DEFAULT CURRENT_DATE,
    due_date DATE,
    completion_date DATE,
    status VARCHAR(20) DEFAULT 'new',
    total_price DECIMAL(10,2),
    notes TEXT
);

-- Tablycyia pozytsiy zamovlennya
CREATE TABLE order_item (
    order_item_id SERIAL PRIMARY KEY,
    order_id INTEGER REFERENCES "order"(order_id),
    service_id INTEGER REFERENCES service(service_id),
    quantity INTEGER DEFAULT 1,
    price DECIMAL(10,2),
    description TEXT
);

-- Tablycyia materialiv
CREATE TABLE material (
    material_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    unit VARCHAR(20),
    price_per_unit DECIMAL(10,2),
    stock_quantity DECIMAL(10,2),
    min_stock_level DECIMAL(10,2),
    supplier VARCHAR(100)
);

-- Tablycyia vykorystannya materialiv
CREATE TABLE material_usage (
    usage_id SERIAL PRIMARY KEY,
    order_id INTEGER REFERENCES "order"(order_id),
    material_id INTEGER REFERENCES material(material_id),
    quantity DECIMAL(10,2),
```

```
cost DECIMAL(10,2),  
usage_date DATE DEFAULT CURRENT_DATE  
);
```

4.4 Приклади коду

Клас Order (Java):

ЛІСТИНГ 3: Клас Order

```
public class Order {
    private int orderId;
    private int clientId;
    private int workerId;
    private LocalDate orderDate;
    private LocalDate dueDate;
    private LocalDate completionDate;
    private OrderStatus status;
    private BigDecimal totalPrice;
    private String notes;

    public enum OrderStatus {
        NEW, ACCEPTED, IN_PROGRESS, FITTING,
        READY, COMPLETED, CANCELLED
    }

    public BigDecimal calculateTotalPrice(
        List<OrderItem> items,
        List<MaterialUsage> materials) {

        BigDecimal total = BigDecimal.ZERO;

        for (OrderItem item : items) {
            total = total.add(
                item.getPrice().multiply(
                    new BigDecimal(item.getQuantity())
                )
            );
        }

        for (MaterialUsage material : materials) {
            total = total.add(material.getCost());
        }

        return total;
    }
}
```



```
public void updateStatus(OrderStatus newStatus) {
    if (this.status == OrderStatus.COMPLETED ||
        this.status == OrderStatus.CANCELLED) {
        throw new IllegalStateException(
            "Cannot change completed order status"
        );
    }
    this.status = newStatus;

    if (newStatus == OrderStatus.COMPLETED) {
        this.completionDate = LocalDate.now();
    }
}
}
```

5 ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ

5.1 План тестування

Тестування інформаційної системи включає наступні етапи:

1. Модульне тестування (Unit Testing):

- Тестування окремих методів та функцій
- Перевірка коректності обчислень
- Валідація вхідних даних

2. Інтеграційне тестування:

- Тестування взаємодії між модулями
- Перевірка роботи з базою даних
- Тестування API endpoints

3. Системне тестування:

- Тестування повного функціоналу системи
- Перевірка бізнес-процесів
- Тестування продуктивності

4. Приймальне тестування:

- Тестування користувачами
- Перевірка відповідності вимогам
- Виявлення зауважень

5.2 Приклади тестових сценаріїв

Тестовий сценарій 1: Створення нового замовлення

1. Адміністратор входить в систему
2. Вибирає клієнта зі списку або створює нового
3. Додає послуги до замовлення

4. Вказує необхідні матеріали
5. Система розраховує загальну вартість
6. Адміністратор вказує термін виконання
7. Система зберігає замовлення
8. Очікуваний результат: замовлення створено, статус "Новий"

Тестовий сценарій 2: Відстеження виконання замовлення

1. Кравець входить в систему
2. Переглядає список своїх замовлень
3. Вибирає замовлення в роботі
4. Оновлює статус на "В процесі виконання"
5. Після завершення змінює статус на "Готовий"
6. Очікуваний результат: статус оновлено, клієнт може бути повідомлений

5.3 Впровадження системи

Етапи впровадження:

1. Підготовчий етап:

- Налаштування серверного обладнання
- Встановлення програмного забезпечення
- Створення бази даних

2. Міграція даних:

- Перенесення існуючих даних про клієнтів
- Імпорт каталогу послуг
- Завантаження довідників

3. Навчання персоналу:

- Проведення тренінгів для адміністраторів
- Навчання кравців роботі з системою
- Підготовка інструкцій користувача

4. Пілотний запуск:

- Робота в тестовому режимі
- Виявлення та усунення помилок
- Збір зворотного зв'язку

5. Промислова експлуатація:

- Повний перехід на нову систему
- Технічна підтримка
- Моніторинг роботи системи

6 ВИСНОВКИ

В результаті виконання курсової роботи була розроблена інформаційна система управління ательє, яка дозволяє автоматизувати основні бізнес-процеси підприємства.

Основні результати роботи:

1. Проведено аналіз предметної області та виявлено основні бізнес-процеси ательє, що потребують автоматизації.
2. Визначено функціональні вимоги до інформаційної системи, які включають управління клієнтами, замовленнями, матеріалами, персоналом та формування звітності.
3. Розроблено діаграму прецедентів, що відображає взаємодію трьох типів користувачів з системою: клієнта, адміністратора та кравця.
4. Створено діаграму класів, яка описує структуру системи та зв'язки між основними сутностями.
5. Спроектовано модель бази даних, що складається з 8 таблиць та забезпечує зберігання всієї необхідної інформації.
6. Розроблено діаграму діяльності для процесу оформлення та виконання замовлення, що демонструє послідовність дій.
7. Описано трирівневу архітектуру системи, яка забезпечує розділення відповідальності та масштабованість.
8. Розроблено діаграму станів для сутності "Замовлення що визначає життєвий цикл замовлення.
9. Підготовлено SQL-скрипти для створення таблиць бази даних та приклади коду основних класів.

Переваги розробленої системи:

- Автоматизація обліку клієнтів та їх замовлень
- Спрощення процесу оформлення та відстеження замовлень

- Контроль витрачання матеріалів та складських залишків
- Ефективний розподіл навантаження між працівниками
- Можливість формування аналітичних звітів
- Покращення якості обслуговування клієнтів
- Підвищення продуктивності роботи ательє

Перспективи подальшого розвитку:

- Додавання мобільного додатку для клієнтів
- Інтеграція з онлайн-оплатою
- Впровадження системи лояльності для постійних клієнтів
- Додавання модуля для роботи з постачальниками
- Інтеграція з бухгалтерськими системами
- Впровадження аналітики на основі машинного навчання для прогнозування попиту

Розроблена інформаційна система є повноцінним інструментом для управління ательє та може бути впроваджена на підприємствах різного масштабу.

7 СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб.: Питер, 2020. — 368 с.
2. Фаулер М. UML. Основы. — 3-е изд. — СПб.: Символ-Плюс, 2019. — 192 с.
3. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. — 3-е изд. — М.: Вильямс, 2018. — 1440 с.
4. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. — 2-е изд. — М.: ДМК Пресс, 2019. — 496 с.
5. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. — М.: Финансы и статистика, 2018. — 544 с.
6. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. — К.: Держстандарт України, 1995.
7. Офіційна документація PostgreSQL [Електронний ресурс]. — Режим доступу: <https://www.postgresql.org/docs/>
8. Офіційна документація Spring Framework [Електронний ресурс]. — Режим доступу: <https://spring.io/projects/spring-framework>
9. Sommerville I. Software Engineering. — 10th ed. — Pearson, 2019. — 816 p.
10. Pressman R.S., Maxim B.R. Software Engineering: A Practitioner's Approach. — 9th ed. — McGraw-Hill Education, 2020. — 976 p.

А ДОДАТОК А. Діаграма розгортання

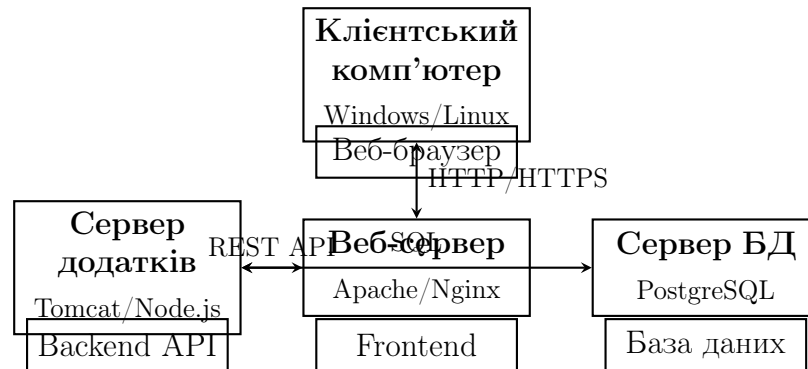


Рис. 9: Діаграма розгортання системи

Б ДОДАТОК Б. Словник термінів

- **Ательє** — підприємство, що надає послуги з пошиття та ремонту одягу
- **API** (Application Programming Interface) — інтерфейс програмування додатків
- **Backend** — серверна частина додатку
- **Frontend** — клієнтська частина додатку
- **CRUD** — Create, Read, Update, Delete (базові операції з даними)
- **ER-діаграма** — діаграма "сутність-зв'язок"
- **ORM** (Object-Relational Mapping) — об'єктно-реляційне відображення
- **REST** (Representational State Transfer) — архітектурний стиль для веб-сервісів
- **SQL** (Structured Query Language) — мова структурованих запитів
- **UML** (Unified Modeling Language) — уніфікована мова моделювання