

Лабораторна робота 5

«Розробка UML-діаграм розроблюваної системи: діаграма послідовності»

Мета: навчитись будувати діаграми послідовності розроблюваної системи. використання системи.

Завдання **1. Розробити послідовності**

Форма звіту

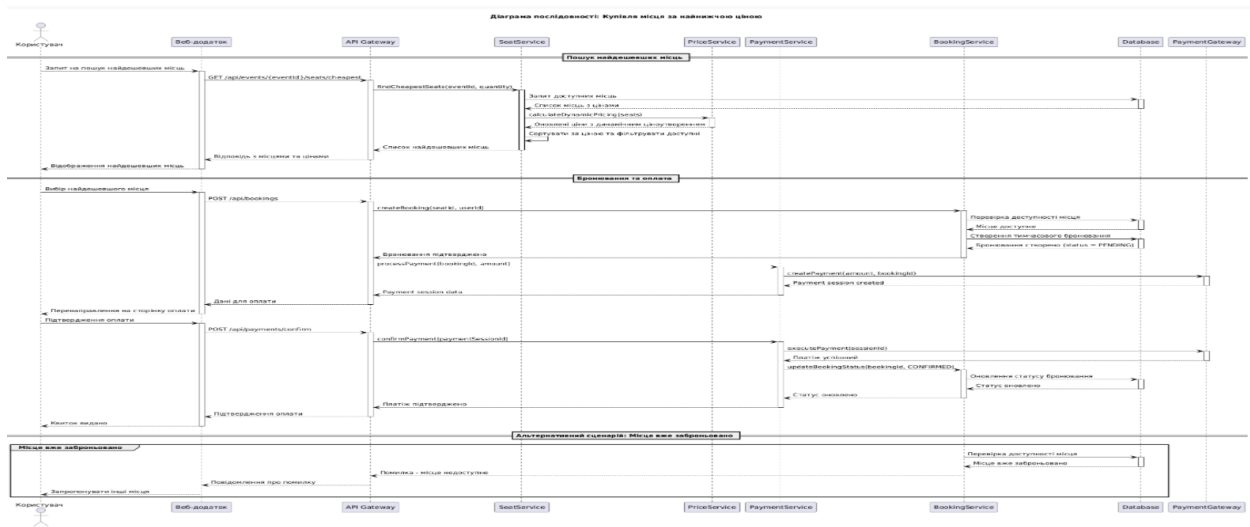
1. Тема та мета.
2. Опис завдання згідно розроблюваної (на вибір) студентом системи.
3. UML-код для діаграми
4. (optional) Відповіді на запитання по теорії та за результатами виконаної роботи.

ПРИКЛАД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ З ПОКРОКОВИМ ОПИСОМ ЗАВДАНЬ

Розробка UML-діаграм послідовності розроблюваної системи

Завдання 1 Розробити ДІАГРАМУ ПОСЛІДОВНОСТІ.

ДІАГРАМА АКТИВНОСТІ ДЛЯ ФУНКЦІЇ «КУПІВЛЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»
ЗОБРАЖЕНА НА РИСУНКУ 4



Код:

@startuml Купівля місця за найнижчою ціною

title Діаграма послідовності: Купівля місця за найнижчою ціною

actor Користувач as User

participant "Веб-додаток" as WebApp

participant "API Gateway" as Gateway

participant "SeatService" as SeatService

participant "PriceService" as PriceService

participant "PaymentService" as PaymentService

participant "BookingService" as BookingService

participant "Database" as DB

participant "PaymentGateway" as PaymentGateway

== Пошук найдешевших місць ==

User -> WebApp: Запит на пошук найдешевших місць

activate WebApp

WebApp -> Gateway: GET /api/events/{eventId}/seats/cheapest

activate Gateway

Gateway -> SeatService: findCheapestSeats(eventId, quantity)

activate SeatService

SeatService -> DB: Запит доступних місць

activate DB

DB --> SeatService: Список місць з цінами

deactivate DB

SeatService -> PriceService: calculateDynamicPricing(seats)

activate PriceService

PriceService --> SeatService: Оновлені ціни з динамічним ціноутворенням

deactivate PriceService

SeatService -> SeatService: Сортувати за ціною та фільтрувати доступні

SeatService --> Gateway: Список найдешевших місць

deactivate SeatService

Gateway --> WebApp: Відповідь з місцями та цінами

deactivate Gateway

WebApp --> User: Відображення найдешевших місць

deactivate WebApp

== Бронювання та оплата ==

User -> WebApp: Вибір найдешевшого місця

activate WebApp

WebApp -> Gateway: POST /api/bookings

activate Gateway

Gateway -> BookingService: createBooking(seatId, userId)

activate BookingService

BookingService -> DB: Перевірка доступності місця

activate DB

DB --> BookingService: Місце доступне

deactivate DB

BookingService -> DB: Створення тимчасового бронювання

activate DB

DB --> BookingService: Бронювання створено (status = PENDING)

deactivate DB

BookingService --> Gateway: Бронювання підтверджено

deactivate BookingService

Gateway -> PaymentService: processPayment(bookingId, amount)

activate PaymentService

PaymentService -> PaymentGateway: createPayment(amount, bookingId)

activate PaymentGateway

PaymentGateway --> PaymentService: Payment session created

deactivate PaymentGateway

PaymentService --> Gateway: Payment session data

deactivate PaymentService

Gateway --> WebApp: Дані для оплати

deactivate Gateway

WebApp --> User: Перенаправлення на сторінку оплати

deactivate WebApp

User -> WebApp: Підтвердження оплати

activate WebApp

WebApp -> Gateway: POST /api/payments/confirm

activate Gateway

Gateway -> PaymentService: confirmPayment(paymentSessionId)

activate PaymentService

PaymentService -> PaymentGateway: executePayment(sessionId)

activate PaymentGateway

PaymentGateway --> PaymentService: Платіж успішний

deactivate PaymentGateway

PaymentService -> BookingService: updateBookingStatus(bookingId, CONFIRMED)

activate BookingService

BookingService -> DB: Оновлення статусу бронювання

activate DB

DB --> BookingService: Статус оновлено

deactivate DB

BookingService --> PaymentService: Статус оновлено

deactivate BookingService

PaymentService --> Gateway: Платіж підтверджено

deactivate PaymentService

Gateway --> WebApp: Підтвердження оплати

deactivate Gateway

WebApp --> User: Квиток видано

deactivate WebApp

== Альтернативний сценарій: Місце вже заброньовано ==

Group Місце вже заброньовано

BookingService -> DB: Перевірка доступності місця

activate DB

DB --> BookingService: Місце вже заброньовано

deactivate DB

BookingService --> Gateway: Помилка - місце недоступне

Gateway --> WebApp: Повідомлення про помилку

WebApp --> User: Запропонувати інші місця

end

@enduml