Università degli Studi di Salerno
Corso di Ingegneria del Software

AniTour
Object Design Document (ODD)
Versione 1.0

Data: 19/12/2025

Progetto: AniTour	Versione: 1.0
Documento: ODD	Data: 19/12/2025

Indice

1. Introduction

1.1 Object design trade-offs

1.2 Interface documentation guidelines

1.3 Definitions, acronyms, and abbreviations

1.4 References

2. Packages

3. Class interfaces

3.1 BookingControl

Overview
Classe di controllo (stereotype <<Control>>) responsabile del coordinamento del processo di acquisto. Gestisce la transazione logica, l'interazione con il carrello e l'invocazione dei servizi di pagamento.

Dependencies

- IPaymentGateway (per processare i pagamenti)
- IBookingRepository (per la persistenza)
- CartManager (per recuperare i dati di sessione)

Operations & Contracts:

3.1.1 processCheckout

```
public Booking processCheckout(String sessionToken, PaymentDTO paymentData) throws Exception
```

- **Descrizione:** Finalizza l'acquisto convertendo il carrello corrente in un ordine confermato.
- **OCL Contract:**

```
context BookingControl::processCheckout(sessionToken, paymentData)
pre:
    -- Il token di sessione non deve essere nullo e il carrello deve contenere elementi
    sessionToken <> null and self.cartManger.getCart(sessionToken).isEmpty() = false
pre:
    -- I dati di pagamento devono essere validi sintatticamente
```

```
    paymentData <> null and paymentData.isValid() =
true

post:
    -- Viene creato un nuovo ordine nel sistema (il
conteggio ordini aumenta di 1)
    self.bookingRepository.count() =
self.bookingRepository.count()@pre + 1
post:
    -- L'ordine restituito è in stato 'CONFIRMED'
    result.status = BookingStatus.CONFIRMED
post:
    -- Il carrello viene svuotato dopo l'acquisto

self.cartManager.getCart(sessionToken).isEmpty() =
true
```

3.1.2 getUserHistory

```
public List<Booking> getUserHistory(int userId)
```

- **Descrizione:** Restituisce la lista delle prenotazioni effettuate da un utente.
- **OCL Contract:**

```
context BookingControl::getUserHistory(userId)
pre:
    -- L'ID utente deve essere positivo ed esistere
nel sistema
    userId > 0 and
self.userRepository.exists(userId) = true
post:
    -- La lista restituita non è mai nulla (al
massimo è vuota)
    result <> null
```

3.2 IPaymentGateway

Overview: Interfaccia (stereotype <<Interface>>) che astrae i servizi di pagamento esterni (Pattern Adapter). Permette al sistema di essere indipendente dal provider specifico (es. Stripe, PayPal).

Operations & Contracts:

3.2.1 processPayment

```
boolean processPayment(double amount, PaymentDTO
cardDetails)
```

- **Descrizione:** Tenta di addebitare l'importo sulla carta fornita.
- **OCL Contract:**

```
context IPaymentGateway::processPayment(amount,
cardDetails)
pre:
    -- L'importo deve essere positivo
    amount > 0.00
post:
    -- Il metodo restituisce true solo se la banca
autorizza, altrimenti false
    result = true implies
(BankSystem.authorize(amount, cardDetails) = true)
```

3.3 IBookingRepository

Overview: Interfaccia (stereotype <<Interface>>) per l'accesso ai dati persistenti (Pattern DAO/Repository). Isola la logica di business dai dettagli del database MySQL.

Operations & Contracts:

3.3.1 Save

```
int save(Booking booking)
```

- **Descrizione:** Persiste una nuova prenotazione nel database.
- **OCL Contract:**

```
context IBookingRepository::save(booking)
pre:
    -- L'oggetto booking non deve essere nullo
    booking <> null
post:
    -- Restituisce un ID positivo che rappresenta la
chiave primaria generata
    result > 0
```

3.3.2 checkAvailability

```
boolean checkAvailability(int tourId, int requiredSeats)
```

- **Descrizione:** Verifica se un tour ha abbastanza posti liberi.
- **OCL Contract:**

```
context
IBookingRepository::checkAvailability(tourId,
requiredSeats)
pre:
```

```
    requiredSeats > 0
post:
    -- Restituisce true se i posti disponibili nel
DB sono sufficienti
    result = (self.tours.select(t | t.id =
tourId).seatsAvailable >= requiredSeats)
```

4. Glossary

Revision History

Data	Versione	Descrizione	Autore
19/12/2025	1.0	Creazione documento ODD. Specifica delle interfacce dei moduli del sottosistema da implementare	Salvatore Merola

Partecipanti:

Nome	Matricola
Vincenzo Chiocca	**0512119182**
Salvatore Merola	**0512120979**