

AniTour
Requirements Analysis Document (RAD)
Versione 1.2



Data: 10/11/2025

Progetto: AniTour	Versione: 1.2
Documento: RAD	Data: 10/11/2025

Indice

1. Introduction
 - 1.1 Purpose of the system
 - 1.2 Scope of the system
 - 1.3 Objectives and success criteria of the project
 - 1.4 Definitions, acronyms, and abbreviations
 - 1.5 References
 - 1.6 Overview
2. Current system
3. Proposed system
 - 3.1 Overview
 - 3.2 Functional requirements
 - 3.3 Nonfunctional requirements
 - 3.3.1 Usability
 - 3.3.2 Reliability
 - 3.3.3 Performance
 - 3.3.4 Supportability
 - 3.3.5 Implementation
 - 3.3.6 Interface
 - 3.3.7 Packaging
 - 3.3.8 Legal
 - 3.4 System models
 - 3.4.1 Scenarios
 - 3.4.2 Use case model
 - 3.4.3 Object model
 - 3.4.4 Dynamic model
 - 3.4.5 User interface—navigational paths and screen mock-ups
4. Glossary

1. Introduction

1.1 Purpose of the system

Lo scopo del software AniTour è offrire una piattaforma di e-commerce specializzata nella vendita di tour tematici dedicati alla cultura nerd, con particolare attenzione al mondo degli anime, dei videogiochi e della cultura pop giapponese. Il sistema consente agli utenti di scoprire, prenotare e acquistare viaggi organizzati in base ai propri interessi specifici, offrendo esperienze personalizzate che combinano turismo e passioni culturali. AniTour mira a semplificare l'intero processo di scelta e prenotazione dei tour, garantendo un'interfaccia intuitiva, un catalogo aggiornato e un'esperienza di acquisto fluida e coinvolgente.

1.2 Scope of the system

AniTour si rivolge a un pubblico sia di appassionati/viaggiatori (clienti) sia di organizzatori di tour (fornitori), offrendo funzionalità di ricerca, selezione, personalizzazione, acquisto e gestione di pacchetti viaggio, incluse le funzionalità di pubblicazione, modifica e gestione tour da parte degli organizzatori.

1.3 Objectives and success criteria of the project

- Offerta di un'esperienza di prenotazione integrata per tour tematici ispirati ad anime, videogiochi e cultura nerd/pop giapponese.
- Unificazione di servizi turistici eterogenei in un unico portale fruibile tramite browser web.
- Personalizzazione dinamica dei pacchetti e suggerimenti automatizzati basati sugli interessi utente.
- Interfaccia responsive e fluida.
- Performance: tempi risposta sempre <2s, alta affidabilità per almeno 500 utenti simultanei.
- Sicurezza: protezione dati personali e transazioni certificate.

1.4 Definitions, acronyms, and abbreviations

- UI: User Interface (Interfaccia utente)
- UX: User Experience (Esperienza utente)
- DBMS: Database Management System
- CRUD: Create Read Update Delete
- Guest: utente non autenticato
- Cliente: utente autenticato che può acquistare tour
- Organizzatore: utente che può creare e gestire i tour
- Operatore: personale di AniTour abilitato alla revisione/approvazione dei tour

1.5 References

- Problem Statement AniTour v1.1
- Linee guida di progetto e documentazione del corso

1.6 Overview

AniTour è una piattaforma digitale per la ricerca, prenotazione, pubblicazione e gestione di tour tematici.

2. Current system

Attualmente non esiste un sistema integrato con tali specializzazioni; l'offerta di tour nerd è frammentaria e affidata a singole agenzie o forum, senza management centralizzato, strumenti di prenotazione unificati o verifica qualità.

3. Proposed system

3.1. Overview

Il sistema proposto fornisce catalogo e gestione pacchetti tour, autenticazione e gestione ruoli (guest, cliente, organizzatore, operatore), workflow di acquisto integrato, strumenti per l'organizzazione e la pubblicazione, backend di amministrazione per operatori, notifiche e pagamenti sicuri.

3.2. Functional requirements

AniTour supporta quattro tipi di utenti:

- L'*operatore* deve essere in grado di gestire i tour e gli utenti presenti sul sito.
- L'*organizzatore* deve essere in grado di creare e gestire i propri tour.
- Il *cliente* deve essere in grado di navigare il sito, cercare tour e finalizzarne l'acquisto.
- Il *guest* deve essere in grado di navigare liberamente sul sito e visualizzare tour, ma non ha la possibilità di procedere all'acquisto (occorre la creazione del proprio account).

3.3. Nonfunctional requirements

- L'architettura deve permettere di gestire almeno 500 utenti simultanei.
- Il sito deve essere pienamente operativo su browser moderni (Chrome, Firefox, Edge, ecc...).
- Il sito deve disporre di un layout fluido e responsive, per permetterne la visualizzazione su dispositivi mobili con schermi di dimensioni variabili.
- Il codice sorgente deve essere strutturato in moduli indipendenti, in modo che nuove funzionalità (es. nuovi tipi di tour o categorie) possano essere aggiunte senza modificare i componenti esistenti.

3.3.1. Usability

Interfaccia utente esplicita, layout responsive e chiaro su diversi dispositivi

3.3.2. Reliability

Uptime annuo $\geq 99.5\%$, failover su servizi critici

3.3.3. Performance

Almeno 500 utenti in contemporanea senza degrado

3.3.4. Supportability

Codice in moduli disaccoppiati, facile manutenzione ed evoluzione

3.3.5. Implementation

Web app accessibile da browser moderni, senza installazione

3.3.6. Interface

API RESTful per integrazione esterna

3.3.7. Packaging

Deploy automatico per ambienti di test e produzione

3.3.8. Legal

GDPR compliance e privacy by design

3.4. System models

3.4.1. Scenarios

SC 1:

1. L'utente apre il browser e naviga all'URL "www.anitour.it"
2. Nella homepage, l'utente utilizza la barra di ricerca inserendo il testo "Persona"
3. Il sistema visualizza 3 risultati. L'utente seleziona il tour "Persona 5 Tour"
4. Il sistema mostra la pagina di dettaglio del tour con le seguenti informazioni:
 1. Titolo: "Persona 5 Tour"
 2. Organizzatore: "Japan Games Adventures S.r.l."
 3. Date: 15/03/2026 – 22/03/2026
 4. Prezzo: €5.000
 5. Tappe: Shibuya Station, Yongen-Jaya, Shinjuku, Akihabara
5. L'utente seleziona dal menù a tendina "Numero partecipanti" il valore "2" e clicca sul pulsante "Aggiungi al carrello".
6. Il sistema mostra il carrello con:
 1. 2x "Persona 5 Tour"
 2. Totale: €10.000
7. L'utente clicca sul pulsante "Procedi all'acquisto".
8. Il sistema rileva che l'utente non è autenticato e reindirizza alla pagina di login.
9. L'utente clicca sul link "Registrati".
10. L'utente compila il form di registrazione con i seguenti campi:
 - **Username** (campo testuale): aldopersonafan04
 - **Nome** (campo testuale): Aldo
 - **Cognome** (campo testuale): Esposito
 - **Email** (campo testuale email): aldoesp@email.it
 - **Password** (campo password): ILovePersona5!
 - **Conferma password** (campo password): ILovePersona5!
 - **Telefono** (campo testuale): +39333234567
 - **Indirizzo** (campo testuale): Via Roma 1
 - **CAP** (campo numerico): 8000

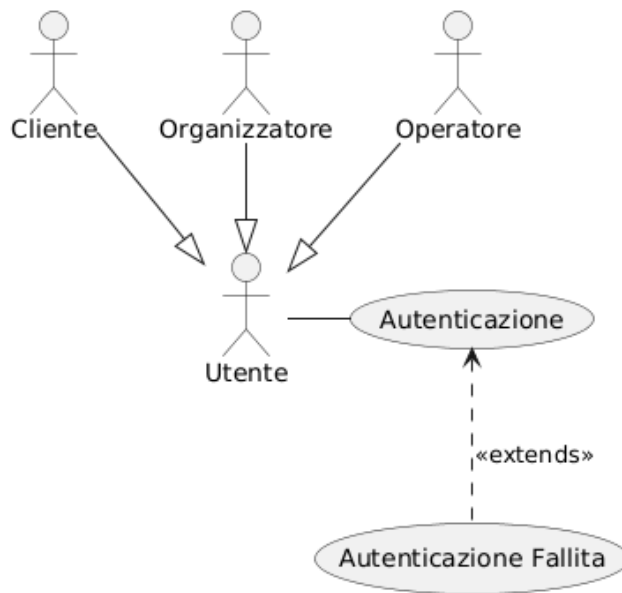
- **Città** (campo testuale): Napoli
 - **Paese** (menù a tendina): Italia
 - **Sesso** (menù a tendina): seleziona "Uomo"
11. L'utente clicca sul pulsante "Registrati"
 12. Il sistema convalida i dati, crea l'account e reindirizza automaticamente al carrello
 13. L'utente clicca sul pulsante "Procedi al pagamento"
 14. Il sistema mostra il form di pagamento. L'utente inserisce:
 - Numero carta (campo numerico): 4532 1234 5678 9012
 - Intestatario (campo testuale): Aldo Esposito
 - Scadenza (due menù a tendina mese/anno): 05 / 2029
 - CVV (campo numerico): 123
 15. L'utente clicca sul pulsante "Conferma pagamento"
 16. Il sistema elabora il pagamento e mostra la pagina di conferma ordine con:
 1. Numero ordine: #ORD-2025-00001
 2. Data: 19/10/2025
 3. Importo totale: €10.000
 4. Ricevuta inviata a: aldoesp@email.it
 17. L'utente riceve via email i voucher del tour

SC 2

1. Un organizzatore di viaggi che può essere un'azienda o un privato in possesso di Partita IVA desidera sponsorizzare un tour a tema anime da lui ideato.
2. Poiché è difficile ottenere visibilità nel vasto mercato turistico, decide di promuovere il proprio tour su *AniTour*, hub centrale dedicato ai viaggi tematici di questo tipo.
3. L'organizzatore accede al sito e clicca su "Registrazione", selezionando la modalità registrazione per organizzatori, dove inserisce i propri dati personali (stessa procedura di registrazione per utenti normali) e la Partita IVA.
4. Una volta completata la registrazione, gli operatori di *AniTour* verificano l'autenticità delle informazioni fornite e la validità della Partita IVA. In caso di esito positivo, l'organizzatore riceve un'e-mail di conferma che dovrà accettare per attivare il profilo.
5. L'organizzatore può quindi proporre un nuovo tour, compilando i campi richiesti nella pagina dedicata e curando la descrizione per renderla chiara e coinvolgente. È tenuto inoltre a indicare le aziende di trasporto che verranno utilizzate per gli spostamenti.
6. Una volta completata la proposta, l'organizzatore invia una richiesta di approvazione agli operatori di *AniTour*, che ne verificano la validità e l'affidabilità. In caso di approvazione, riceve un'e-mail di conferma e il tour viene pubblicato sul sito; in caso contrario, la proposta viene respinta e non resa visibile al pubblico.

3.4.2. Use case model

UC 1 Autenticazione:



Attore: Utente

Entry condition: L'utente si trova nella schermata di autenticazione di AniTour.

Flusso di eventi:

1. L'utente inserisce username e password
2. L'utente invia i dati al sistema
3. Il sistema controlla le credenziali e il controllo ha esito positivo
4. Il sistema reindirizza l'utente alla sua home page

Exit condition: L'utente è autenticato e si trova sulla sua home page.

Flussi alternativi / Eccezioni:

Se al punto 3 il sistema rileva credenziali non corrette, il sistema mostrerà il messaggio di errore "username o password non corrette" e ripresenterà la schermata di autenticazione (UC 1.1 Autenticazione Fallita).

Alternativa

Se al punto 3 il sistema rileva credenziali non corrette, verrà eseguito il caso d'uso UC 1.1 Autenticazione Fallita

UC 1.1 Autenticazione fallita:

Attore: Utente

Entry Condition: L'utente ha provato ad autenticarsi e le credenziali non erano corrette

Flusso di eventi:

1. Il sistema mostra il messaggio di errore "Nome utente o password errati."
2. Il sistema ripresenta all'utente la schermata di autenticazione

Exit Condition: Il sistema visualizza nuovamente il form di autenticazione con il messaggio di errore "Nome utente o password errati".

UC 2 Registrazione:



Attore: Guest

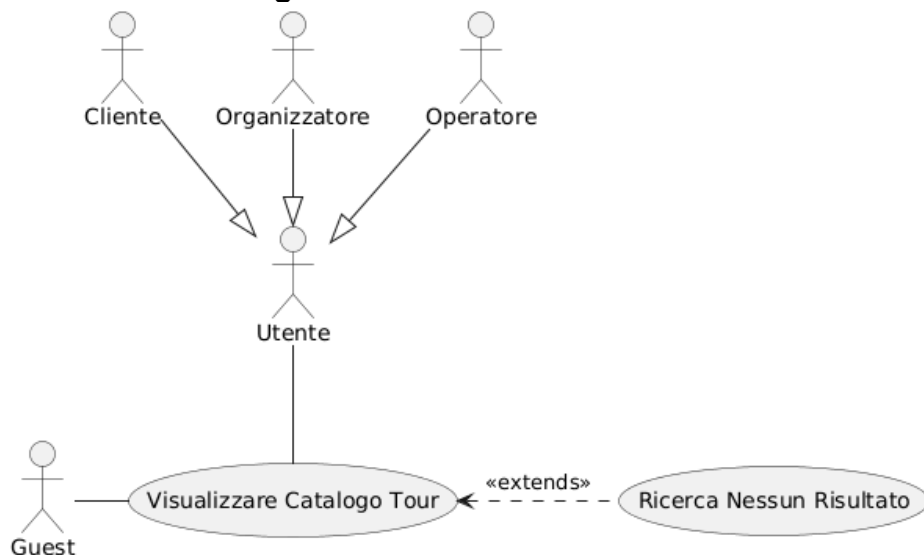
Entry Condition: Il guest si trova sulla pagina di registrazione di AniTour e non possiede un account.

Flusso di eventi:

1. Il guest accede alla pagina di registrazione
2. Il guest compila il form con nome, cognome, e-mail, password e conferma password
3. Il guest inserisce una password con requisiti validi (min 8 caratteri)
4. Il guest clicca il pulsante "Registrati"
5. Il sistema valida i dati inseriti
6. Il sistema verifica che l'e-mail non sia già registrata
7. Il sistema crea un nuovo account utente
8. Il sistema reindirizza l'utente (prima guest) alla home page

Exit Condition: Il nuovo account è creato e l'utente è reindirizzato alla home page.

UC 3 Visualizzare catalogo tour:



Attori: Utente, Guest

Entry Condition: L'utente si trova sulla home page del sito e clicca "Esplora tutti i tour" oppure usa la ricerca.

Flusso di eventi (Percorso A - Esplora tutti i tour):

1. L'utente si trova sulla home page
2. L'utente clicca il pulsante "Esplora tutti i tour"
3. Il sistema reindirizza alla pagina dei tour disponibili (/tours)
4. Il sistema visualizza griglia completa di tutti i tour disponibili
5. L'utente visualizza card con immagine, titolo, descrizione, date e prezzo

Flusso di eventi (Percorso B - Ricerca filtrata tour):

1. L'utente si trova sulla home page

2. L'utente decide di utilizzare la search bar per filtrare il tour per tema (nome tour)
3. L'utente digita nel campo di ricerca "Persona 5"
4. L'utente clicca "Cerca"
5. Il sistema reindirizza alla pagina dei tour disponibili (/tours) filtrata
6. Il sistema visualizza solo i tour che corrispondono ai criteri "nome tour="Persona 5")
7. L'utente visualizza card con immagine, titolo, descrizione, date e prezzo

Exit Condition: L'utente visualizza il catalogo e può selezionare un tour dai risultati.

Flussi alternativi / Eccezioni:

Se al punto 6 Percorso B il sistema non trova nessun tour che corrisponda ai criteri dati dall'utente, il sistema mostrerà la pagina dei tour disponibili (/tours) con il messaggio "Nessun tour disponibile per questa ricerca."

Alternativa

Se al punto 6 Percorso B il sistema non trova nessun tour che corrisponda ai criteri dati all'utente, verrà eseguito il caso d'uso UC 3.1 Ricerca nessun risultato.

UC 3.1 Ricerca nessun risultato:

Attori: Utente, Guest

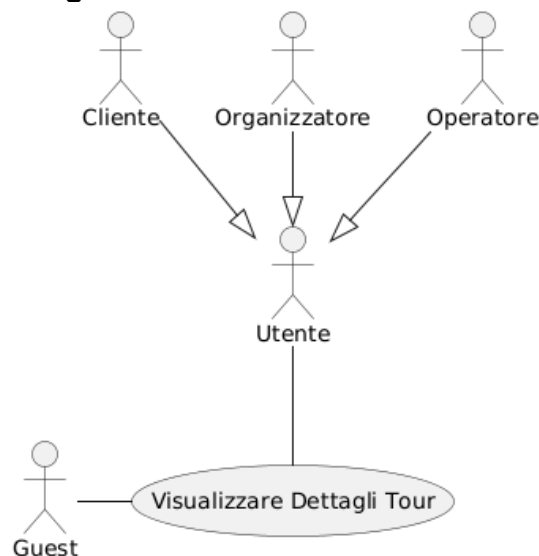
Entry Condition: I filtri applicati non producono risultati

Flusso di eventi:

1. Il sistema verifica che nessun tour corrisponda ai criteri
2. Il sistema mostra il messaggio di errore "Nessun tour disponibile per questa ricerca."

Exit Condition: L'utente rimane sulla pagina dei tour con il messaggio di errore e i filtri ancora applicati. Può cliccare il pulsante "Torna alla ricerca".

UC 4 Visualizzare dettagli tour:



Attori: Utente, Guest

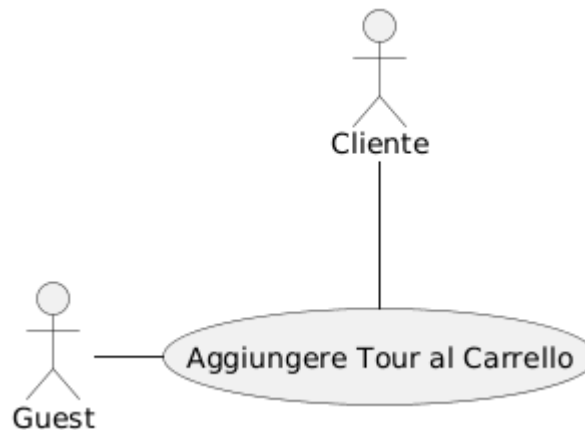
Entry Condition: L'utente ha cliccato su un tour dal catalogo (UC 3)

Flusso di eventi:

1. Il sistema carica la pagina dettagli del tour selezionato
2. Il sistema mostra immagine copertina, titolo, descrizione, organizzatore, date, prezzo
3. Il sistema mostra elenco tappe numerate con titolo e descrizione
4. Il sistema mostra un campo numerico modificabile "Quantità"
5. L'utente può cliccare sul pulsante "Aggiungi al carrello"

Exit Condition: L'utente visualizza la pagina dei dettagli del tour selezionato.

UC 5 Aggiungere tour al carrello:



Attori: Cliente, Guest

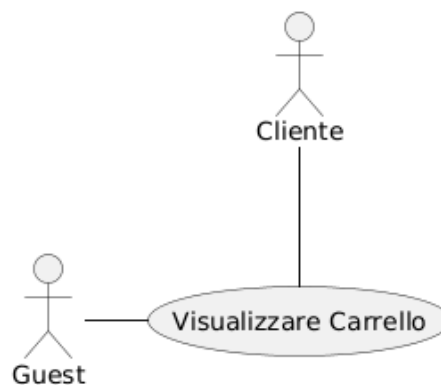
Entry Condition: L'utente (Guest o Cliente autenticato) visualizza i dettagli di un tour (UC 4) e il tour è disponibile con stock sufficiente.

Flusso di eventi:

1. L'utente clicca il pulsante "Aggiungi al carrello"
2. Il sistema valida la quantità selezionata
3. Il sistema aggiunge il tour al carrello con i parametri selezionati
4. Il sistema aggiorna il badge del carrello (numero tour aumenta)
5. Il sistema reindirizza alla pagina del carrello (UC 6)

Exit Condition: Tour aggiunto al carrello e utente reindirizzato sulla pagina del carrello (UC 6).

UC 6 Visualizza carrello:



Attori: Cliente, Guest

Entry Condition: L'utente clicca l'icona carrello oppure è reindirizzato dopo aver aggiunto un tour.

Flusso di eventi:

1. Il sistema visualizza la pagina "Il tuo carrello"
2. Il sistema mostra tabella con colonne: immagine tour, titolo, quantità, prezzo unitario, subtotale, azioni
3. Il sistema mostra il prezzo totale
4. Il sistema mostra due pulsanti principali:

Percorso A – Se utente è guest:

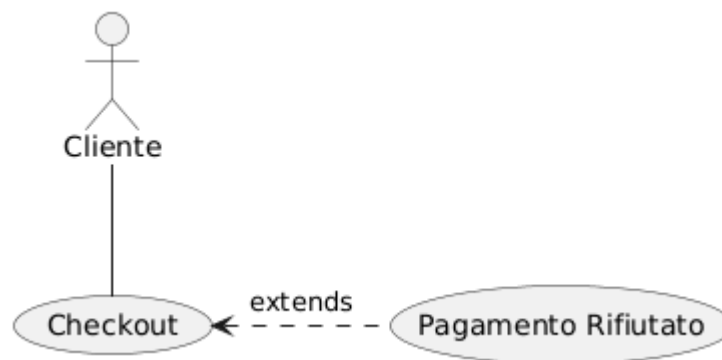
- Pulsante "Accedi per procedere all'acquisto" che reindirizza ad Autenticazione (UC 1) / Registrazione (UC 2)
- Dopo il login con successo. Il sistema reindirizza direttamente al Checkout (UC 7) con il carrello che si è salvato e non viene svuotato

Percorso B – Se utente è cliente:

- Pulsante "Procedi al checkout" che reindirizza direttamente al Checkout (UC 7)

Exit Condition: L'utente visualizza la pagina del carrello dove può procedere al checkout (solo se autenticato).

UC 7 Checkout:



Attori: Cliente

Entry Condition: Il cliente è autenticato e ha cliccato "Procedi al checkout" dal carrello (UC 6) oppure è stato reindirizzato qui dopo login dal carrello. Il carrello contiene almeno un tour.

Flusso di eventi:

1. Il sistema visualizza la pagina di checkout per effettuare il pagamento
2. Il sistema presenta campi da compilare:

Informazioni di spedizione:

- Campo "Nome completo" (input testuale)
- Campo "Email" (input e-mail)
- Campo "Telefono" (input testuale, formato internazionale)
- Campo "Indirizzo" (input testuale)
- Campo "Città" (input testuale)
- Campo "CAP" (input numerico, 4-6 cifre)
- Campo "Paese" (dropdown lista paesi)

Dati pagamento:

- Campo "Numero carta di credito" (input numerico che accetta formati internazionali carte di credito)
 - Campo "Data di scadenza (MM/AA)" (due input: MM e AA)
 - Campo "CVV" (input numeri, 3 cifre)
 - Campo "Intestatario carta" (input testuale)
3. Il sistema mostra il riepilogo ordine
 4. Il cliente compila tutti i campi
 5. Il cliente clicca sul pulsante "Effettua ordine"
 6. Il sistema crea un nuovo ordine nel database
 7. Il sistema manda il voucher al cliente tramite e-mail
 8. Il sistema svuota il carrello
 9. L'utente viene reindirizzato sulla pagina di successo dell'ordine

Exit Condition: Pagamento confermato, ordine creato nel database e voucher inviato all'e-mail dell'utente. Utente reindirizzato a pagina di successo ordine

Flussi alternativi / Eccezioni:

Se al punto 5 il sistema non accetta il pagamento dall'utente, verrà eseguito il caso d'uso UC 7.1 Pagamento rifiutato.

Alternativa

Se al punto 5 il sistema rifiuta la transazione (fondi insufficienti, carta scaduta, ecc), viene visualizzato un messaggio di errore e consente il retry.

UC 7.1 Pagamento rifiutato:

Attore: Cliente

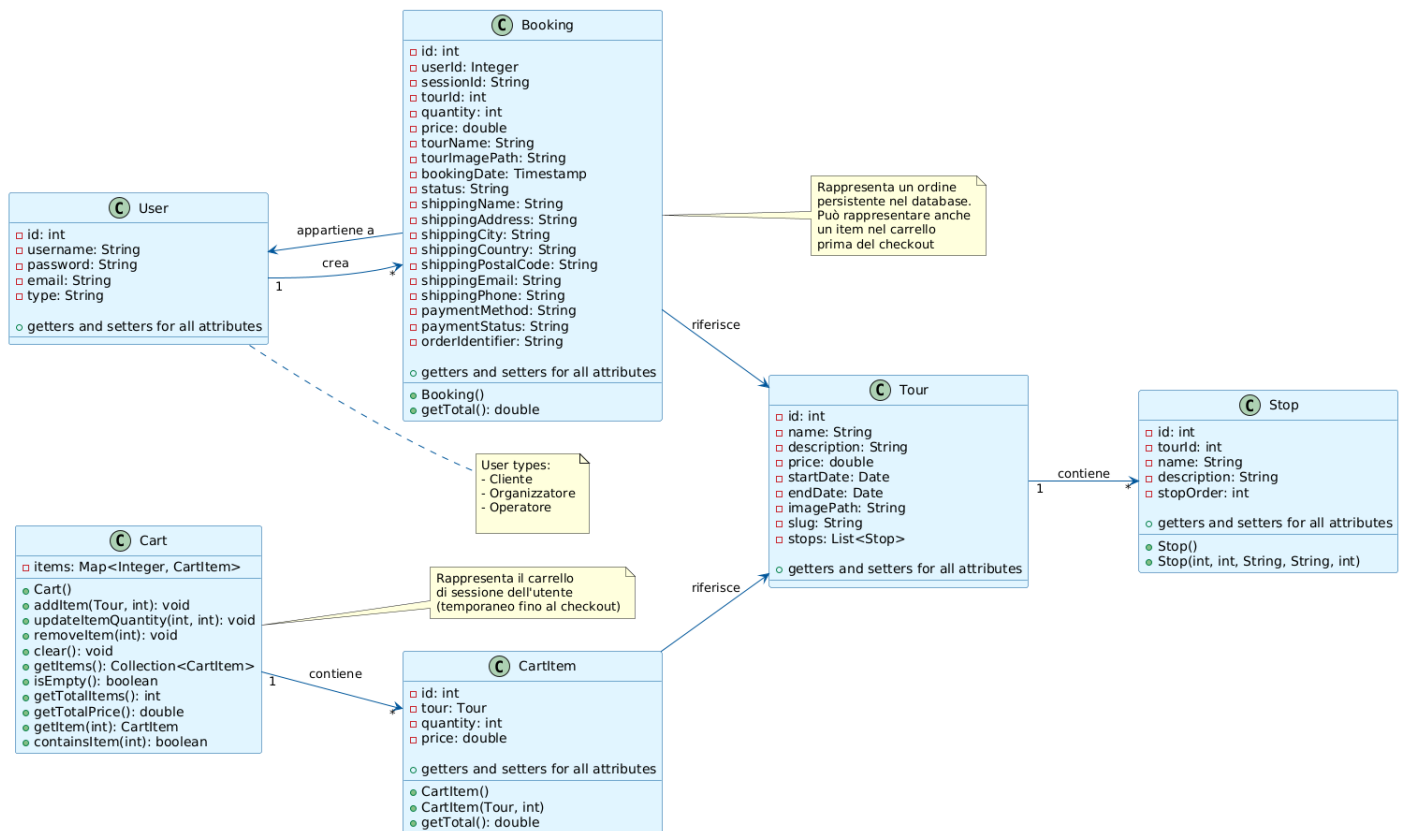
Entry Condition: Il sistema ha rifiutato il pagamento

Flusso di eventi:

1. Il sistema rifiuta il pagamento
2. Il sistema mostra messaggio di errore "Pagamento rifiutato"

Exit Condition: L'utente rimane sulla pagina di checkout con il messaggio di errore visualizzato, i dati di spedizione mantenuti e quelli di pagamento svuotati.

3.4.3. Object model



Data dictionary

Entità: User

Descrizione: Rappresenta un utente del sistema AniTour. Può assumere i ruoli di Cliente, Organizzatore, Operatore o rimanere come utente non autenticato (Guest).

Attributo	Tipo	Vincoli	Descrizione
id	int	PK, AUTO INCREMENT	Identificatore univoco dell'utente
username	String(50)	NOT NULL, UNIQUE	Nome utente per autenticazione (UC 1)
password	String(255)	NOT NULL, HASHED	Password crittografata
email	String(100)	NOT NULL, UNIQUE	Indirizzo email univoco
type	String(20)	NOT NULL	Ruolo: 'cliente', 'organizzatore', 'operatore'

Riferimenti Use Case: UC 1 (Autenticazione), UC 2 (Registrazione)

Note Implementative: La password deve essere memorizzata crittografata utilizzando algoritmi sicuri (es. bcrypt). L'email serve anche come metodo di recupero password.

Entità: Tour

Descrizione: Rappresenta un pacchetto viaggio organizzato all'interno della piattaforma AniTour. Contiene informazioni complete sul tour, date, prezzo e le tappe che lo compongono.

Attributo	Tipo	Vincoli	Descrizione
id	int	PK, AUTO INCREMENT	Identificatore univoco del tour
name	String(200)	NOT NULL	Titolo del tour
description	String(2000)	NOT NULL	Descrizione completa e dettagliata del tour
price	double	NOT NULL, > 0	Prezzo per persona del tour
startDate	Date	NOT NULL	Data di partenza del tour
endDate	Date	NOT NULL, >= startDate	Data di ritorno del tour
imagePath	String(255)	NOT NULL	Percorso relativo dell'immagine copertina
slug	String(255)	NOT NULL, UNIQUE	URL-friendly identifier generato da name
stops	List	FK	Collection di Stop ordinati appartenenti al tour

Riferimenti Use Case: UC 3 (Visualizzare catalogo), UC 4 (Dettagli tour), UC 5 (Aggiungere al carrello)

Note Implementative:

- Lo slug viene generato automaticamente dal campo name (es. "Tour Persona 5" → "tour-persona-5") ed è utilizzato nelle URL amichevoli
- La relazione 1..* con Stop è mantenuta in memoria tramite una List
- Gli attributi startDate e endDate determinano la disponibilità del tour nel catalogo
- Il campo price rappresenta il prezzo per una singola persona

Entità: Stop

Descrizione: Rappresenta una tappa singola di un tour. Ogni tappa ha un ordine sequenziale e contiene informazioni sul luogo visitato.

Attributo	Tipo	Vincoli	Descrizione
id	int	PK, AUTO INCREMENT	Identificatore univoco della tappa
tourId	int	FK(Tour.id), NOT NULL	Riferimento al tour di appartenenza
name	String(150)	NOT NULL	Nome della tappa (città, luogo, attrazione)
description	String(1000)	NOT NULL	Descrizione dettagliata della tappa
stopOrder	int	NOT NULL, >= 1	Ordine sequenziale della tappa nel tour

Riferimenti Use Case: UC 4 (Visualizzare dettagli tour)

Note Implementative:

- Le tappe vengono visualizzate in ordine numerico sequenziale nella pagina di dettagli del tour (UC 4)
- L'attributo stopOrder deve essere unique per coppia (tourId, stopOrder) per evitare duplicati
- Modifiche alle tappe devono essere riflesse nei tour in stato "Draft" ma non in tour "Published"

Entità: Cart

Descrizione: Rappresenta il carrello di sessione dell'utente. Contiene gli item (tour) aggiunti dall'utente prima del checkout. È un oggetto **non persistente** nel database, mantenuto in memoria di sessione HTTP.

Attributo	Tipo	Vincoli	Descrizione
items	Map	NOT NULL	Mappa di CartItem indicizzati per tourId

Riferimenti Use Case: UC 5 (Aggiungere tour al carrello), UC 6 (Visualizzare carrello)

Metodi Pubblici:

- addItem(Tour, int): Aggiunge un tour al carrello o incrementa la quantità se già presente
- removeItem(int tourId): Rimuove un item dal carrello
- updateItemQuantity(int tourId, int quantity): Modifica la quantità di un item
- getTotalItems(): Ritorna il numero totale di item nel carrello
- getTotalPrice(): Calcola il prezzo totale di tutti gli item
- clear(): Svuota completamente il carrello (es. dopo checkout completato)
- isEmpty(): Verifica se il carrello è vuoto
- getItems(): Ritorna la collezione di CartItem

Note Implementative:

- La Map evita duplicati per lo stesso tour: aggiungere lo stesso tour due volte incrementa la quantità anziché creare un nuovo CartItem
- Non è persistente nel database; viene cancellato al termine della sessione HTTP
- Viene svuotato automaticamente dopo il checkout completato (UC 7)
- Per utenti autenticati, il carrello può essere ripristinato dalla sessione precedente

Entità: CartItem

Descrizione: Rappresenta un singolo item all'interno del carrello. Associa un tour a una quantità specifica di posti e mantiene uno snapshot del prezzo al momento dell'aggiunta.

Attributo	Tipo	Vincoli	Descrizione
id	int	PK, AUTO INCREMENT	Identificatore univoco del CartItem
tour	Tour	NOT NULL, FK	Riferimento all'oggetto Tour inserito nel carrello
quantity	int	NOT NULL, >= 1	Quantità di posti prenotati per questo tour
price	double	NOT NULL, > 0	Snapshot del prezzo al momento dell'aggiunta al carrello

Riferimenti Use Case: UC 5 (Aggiungere al carrello), UC 6 (Visualizzare carrello)

Metodi Pubblici:

- getTotal(): Calcola il subtotale = price × quantity

Note Implementative:

- Il campo price è uno **snapshot del prezzo al momento dell'aggiunta** per evitare che modifiche al prezzo del tour impattino sugli item già nel carrello
- La quantity rappresenta il numero di posti (persone) da prenotare per quel tour
- Se il prezzo del tour viene modificato dopo l'aggiunta al carrello, il CartItem mantiene il prezzo originale fino al checkout

Entità: Booking

Descrizione: Rappresenta una prenotazione di tour. Ha un doppio utilizzo: (1) traccia gli item nel carrello durante la sessione di acquisto; (2) memorizza gli ordini completati. La persistenza nel database consente la cronologia completa delle transazioni.

Attributo	Tipo	Vincoli	Descrizione
id	int	PK, AUTO INCREMENT	Identificatore univoco della prenotazione
userId	Integer	FK(User.id), NULLABLE	ID dell'utente autenticato (NULL per guest)
sessionId	String(100)	NOT NULL	ID di sessione HTTP per tracciare guest e carrelli
tourId	int	FK(Tour.id), NOT NULL	Riferimento al tour prenotato
quantity	int	NOT NULL, >= 1	Numero di posti/personone prenotate
price	double	NOT NULL, > 0	Prezzo totale della prenotazione (price × quantity)
tourName	String(200)	NOT NULL	Snapshot del nome del tour (per cronologia)
tourImagePath	String(255)	NOT NULL	Snapshot del percorso immagine del tour
bookingDate	Timestamp	NOT NULL, DEFAULT NOW()	Data e ora della creazione della prenotazione
status	String(20)	NOT NULL	Stato: 'cart', 'completed', 'cancelled'
shippingName	String(100)	NULLABLE	Nome completo per spedizione voucher
shippingAddress	String(255)	NULLABLE	Indirizzo completo di spedizione
shippingCity	String(100)	NULLABLE	Città di spedizione
shippingCountry	String(100)	NULLABLE	Paese di spedizione
shippingPostalCode	String(20)	NULLABLE	Codice postale di spedizione
shippingEmail	String(100)	NULLABLE	Email di contatto per spedizione
shippingPhone	String(20)	NULLABLE	Numero telefono di contatto
paymentMethod	String(50)	NULLABLE	Metodo di pagamento: 'credit_card', 'paypal', etc.
paymentStatus	String(20)	NULLABLE	Stato pagamento: 'pending', 'authorized', 'failed'
orderIdIdentifier	String(50)	NULLABLE, UNIQUE	ID ordine univoco per tracking (es. ORDER-20251111-00123)

Riferimenti Use Case: UC 5 (Aggiungere al carrello), UC 6 (Visualizzare carrello), UC 7 (Checkout)

Metodi Pubblici:

- getTotal(): Ritorna il prezzo totale della prenotazione

Ciclo di Vita Booking:

1. **Status = 'cart'** (UC 5, UC 6)

- Booking viene creato quando un CartItem diventa persistente
- L'utente può aggiungere/rimuovere/modificare quantità
- I campi di shipping e pagamento sono NULL

1. **Status = 'completed'** (UC 7 - Success)

- Pagamento autorizzato
- I dati di shipping e pagamento vengono valorizzati
- orderIdentifier viene generato e reso univoco
- Voucher PDF viene generato e inviato via email
- Il carrello dell'utente viene svuotato

1. **Status = 'cancelled'**

- Ordine annullato dall'utente o dal sistema
- Rimane nel database per cronologia

Note Implementative:

- La classe Booking rappresenta **due concetti** per ridurre la complessità del modello:
- **Item nel carrello** (status='cart'): persistenza temporanea prima del checkout
- **Ordine completato** (status='completed'): prenotazione confermata e pagata
- I campi tourName, tourImagePath, price sono **snapshot** per mantenere la cronologia storica indipendente da modifiche future ai tour
- Il sessionId è essenziale per tracciare sia guest (userId=NULL) che utenti autenticati (userId NOT NULL)
- L'orderIdentifier deve essere generato solo quando status='completed' ed è utilizzato per il tracking e il supporto al cliente
- I campi di shipping sono valorizzati solo durante il checkout (UC 7)

Relazioni tra Entità

Relazione	Molteplicità	Descrizione
User → Booking	1..*	Un utente crea multiple prenotazioni/ordini nel tempo
Tour → Stop	1..*	Un tour contiene una o più tappe ordinate sequenzialmente
Tour → CartItem	1..*	Un tour può essere aggiunto al carrello di più utenti
CartItem → Cart	*..1	Più CartItem appartengono a un singolo Cart
CartItem → Tour	*..1	CartItem riferisce a esattamente un Tour
Booking → Tour	*..1	Più Booking possono riferire allo stesso Tour
Booking → User	*..1	Più Booking appartengono a uno User (opzionale)

Vincoli e Regole di Business

Regola	Entità	Descrizione
Primary Key	Tutte	Ogni entità ha un id come primary key auto-incrementale
Username Univoco	User	Due utenti non possono avere lo stesso username
Email Unica	User	Due utenti non possono avere la stessa email
User Type	User	type deve essere uno di: 'cliente', 'organizzatore', 'operatore'
Tour Price	Tour	price deve essere > 0
Date Validity	Tour	endDate >= startDate
Slug Univoco	Tour	slug generato da name deve essere univoco
Stop Order	Stop	stopOrder >= 1 e deve essere univoco per coppia (tourId, stopOrder)
Cart Quantity	CartItem	quantity >= 1
Booking Price	Booking	price deve essere > 0
Booking Status	Booking	status deve essere uno di: 'cart', 'completed', 'cancelled'
Session Tracking	Booking	sessionId deve essere sempre valorizzato
Order ID Univoco	Booking	orderIdIdentifier must be unique quando status='completed'
Guest Tracking	Booking	Per guest: userId=NULL, sessionId=NOT NULL
Snapshot Fields	Booking	tourName, tourImagePath, price non devono essere modificati dopo creazione

Note di Implementazione Generale

Persistenza e Sessione:

- Cart è **non persistente** (memoria di sessione HTTP)
- Booking è **persistente** (database) e traccia sia il carrello (status='cart') che ordini completati
- SessionId è essenziale per recuperare il carrello di utenti non autenticati

Snapshot Fields:

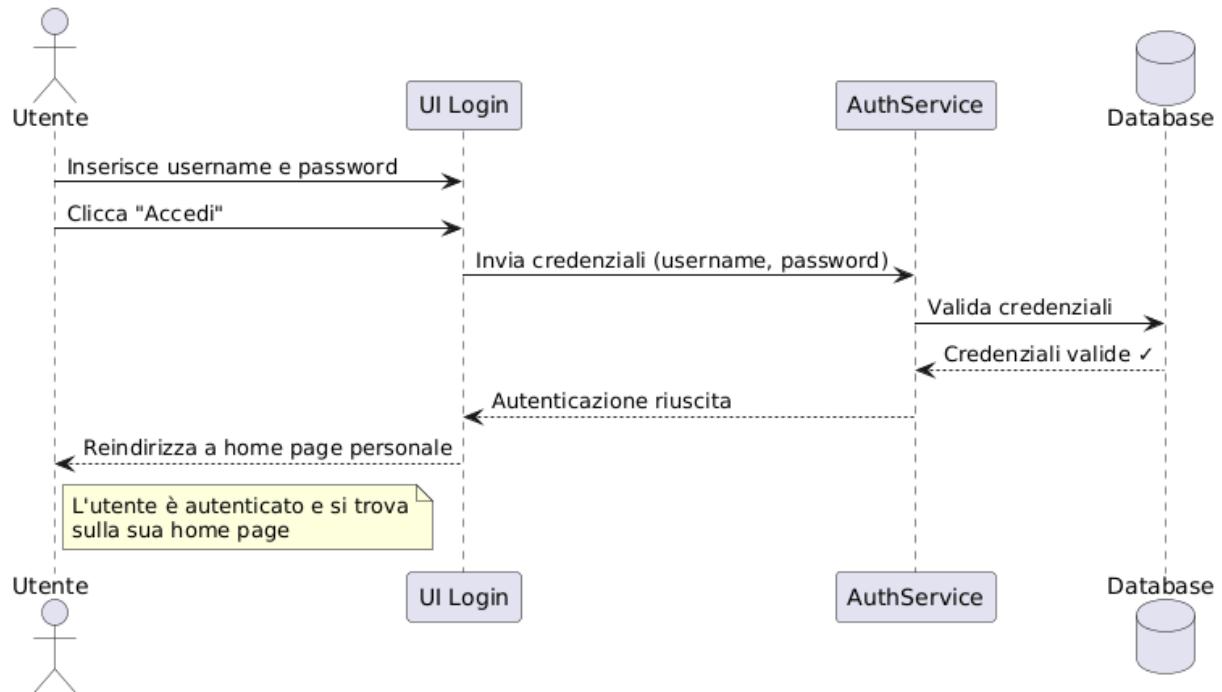
- tourName, tourImagePath in Booking mantengono uno snapshot del tour al momento dell'ordine
- Modifiche successive al tour non impattano sulla cronologia degli ordini
- Essenziale per conformità normativa e cronologia storica accurate

Relazioni In-Memoria vs Persistenti:

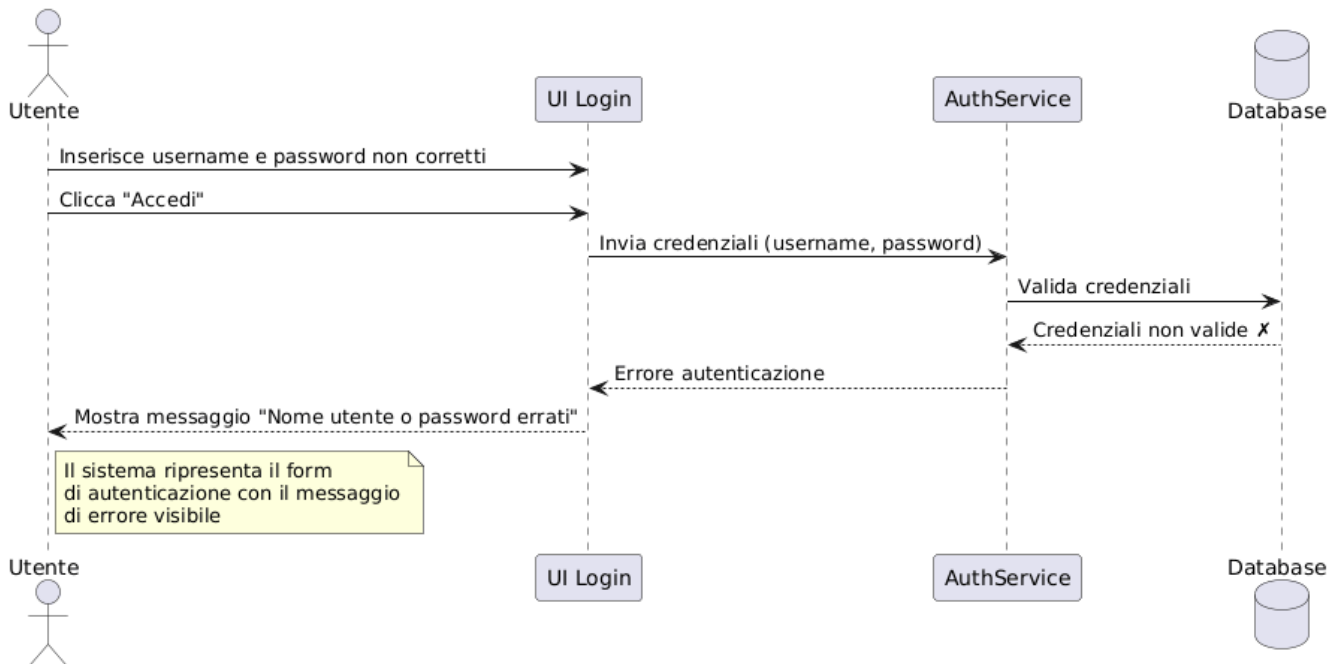
- Cart.items: relazione in-memoria (Map)
- Tour.stops: relazione in-memoria (List) ma i dati sono persistenti nel database
- Booking: relazione persistente, completa e tracciabile

3.4.4. Dynamic model

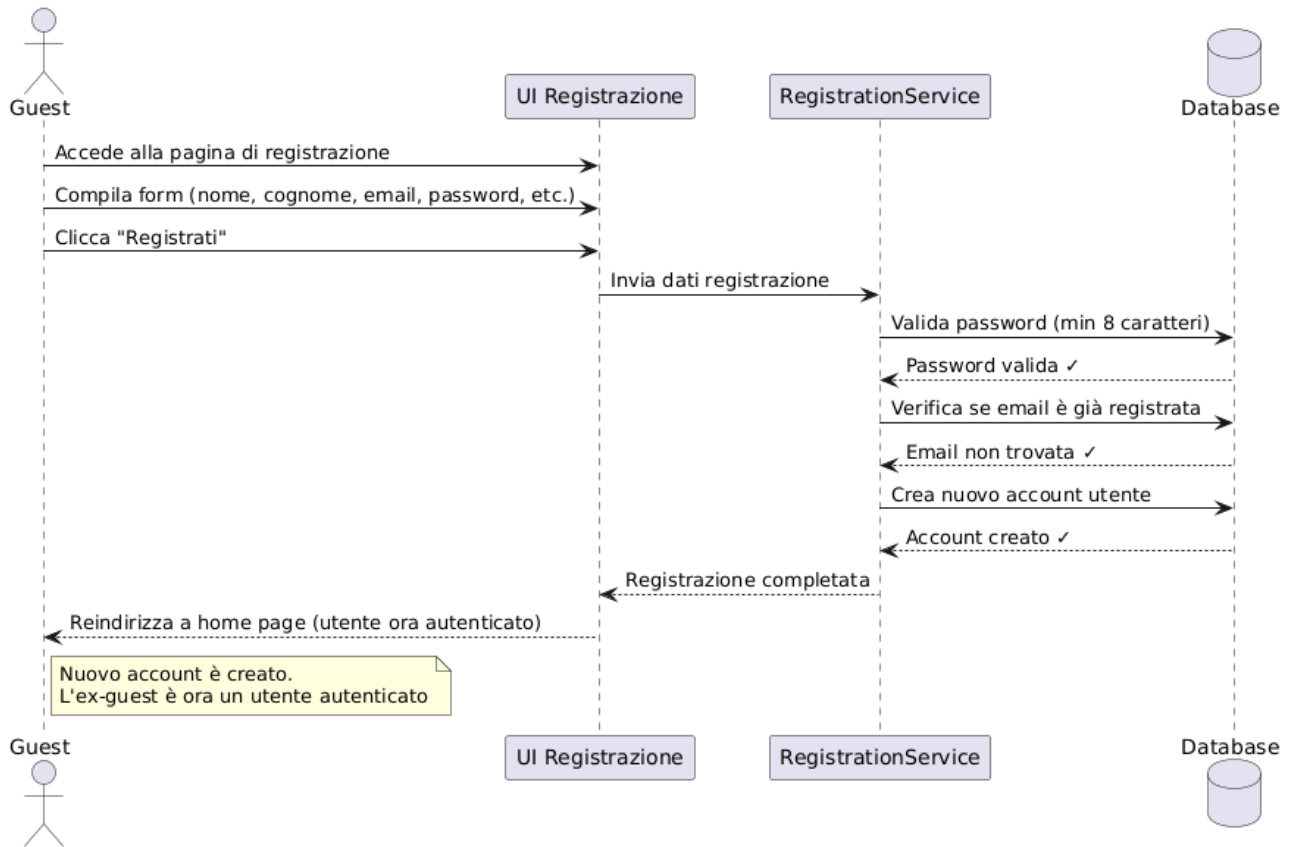
UC 1 - Autenticazione (Success Path)



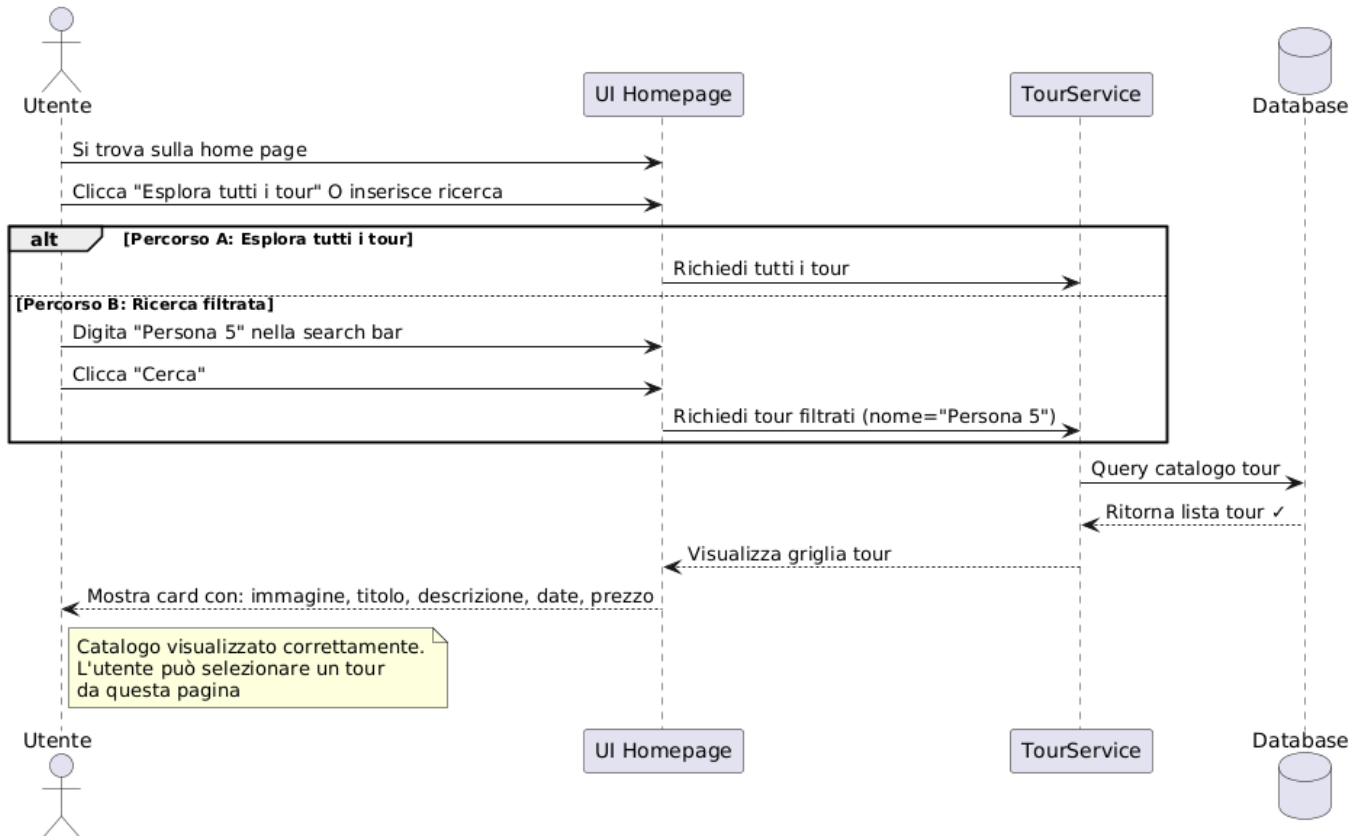
UC 1.1 - Autenticazione Fallita (Exception Path)



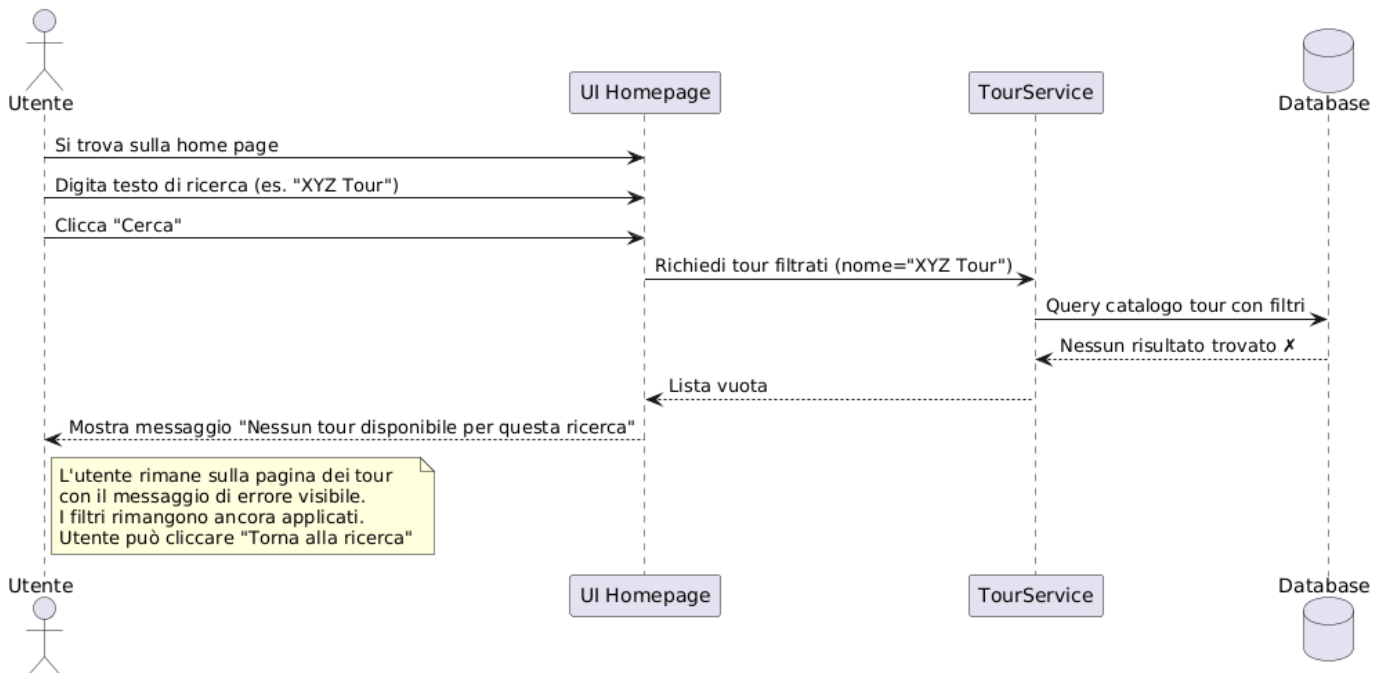
UC 2 - Registrazione



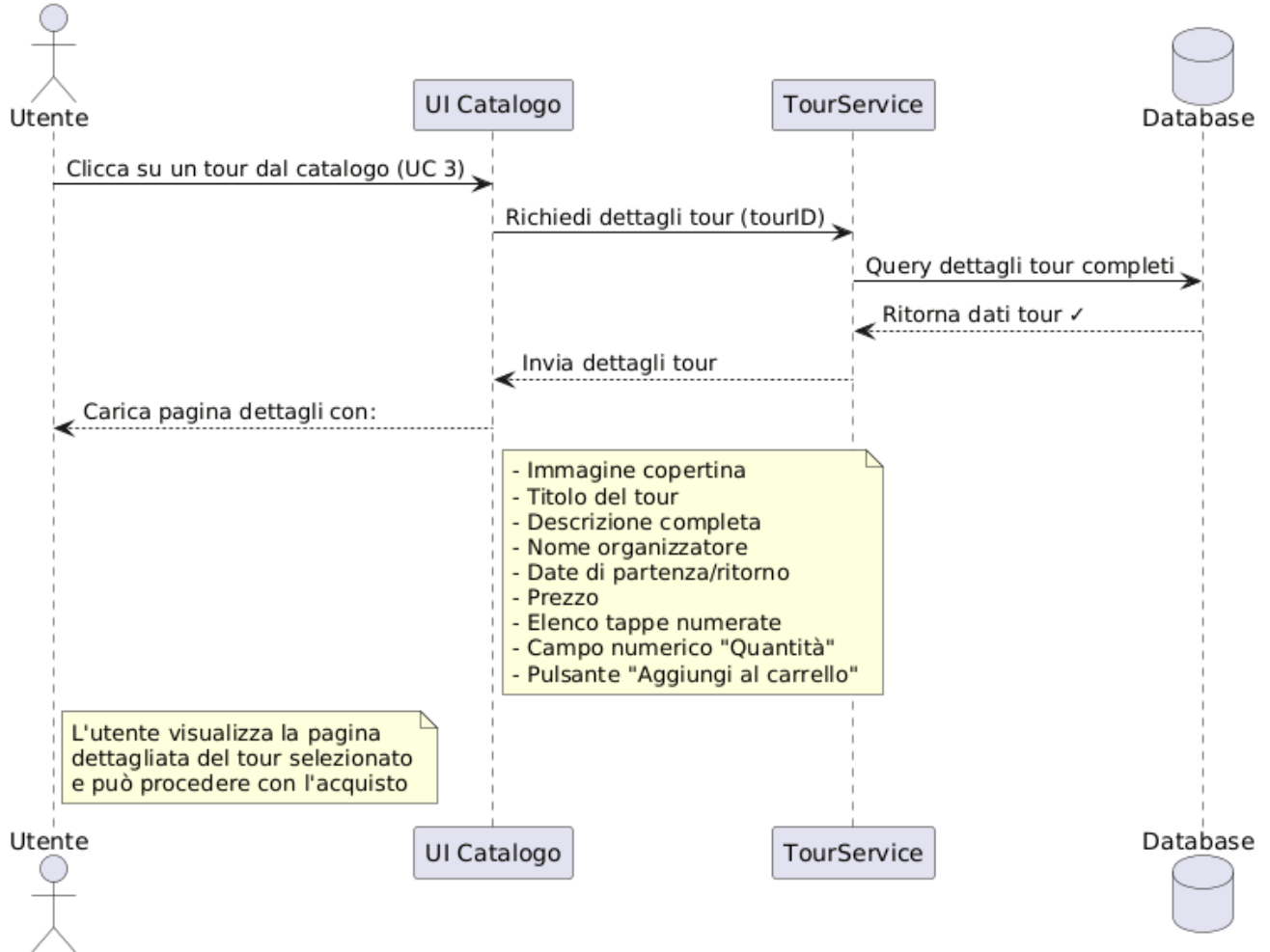
UC 3 - Visualizzare Catalogo Tour (Success Path)



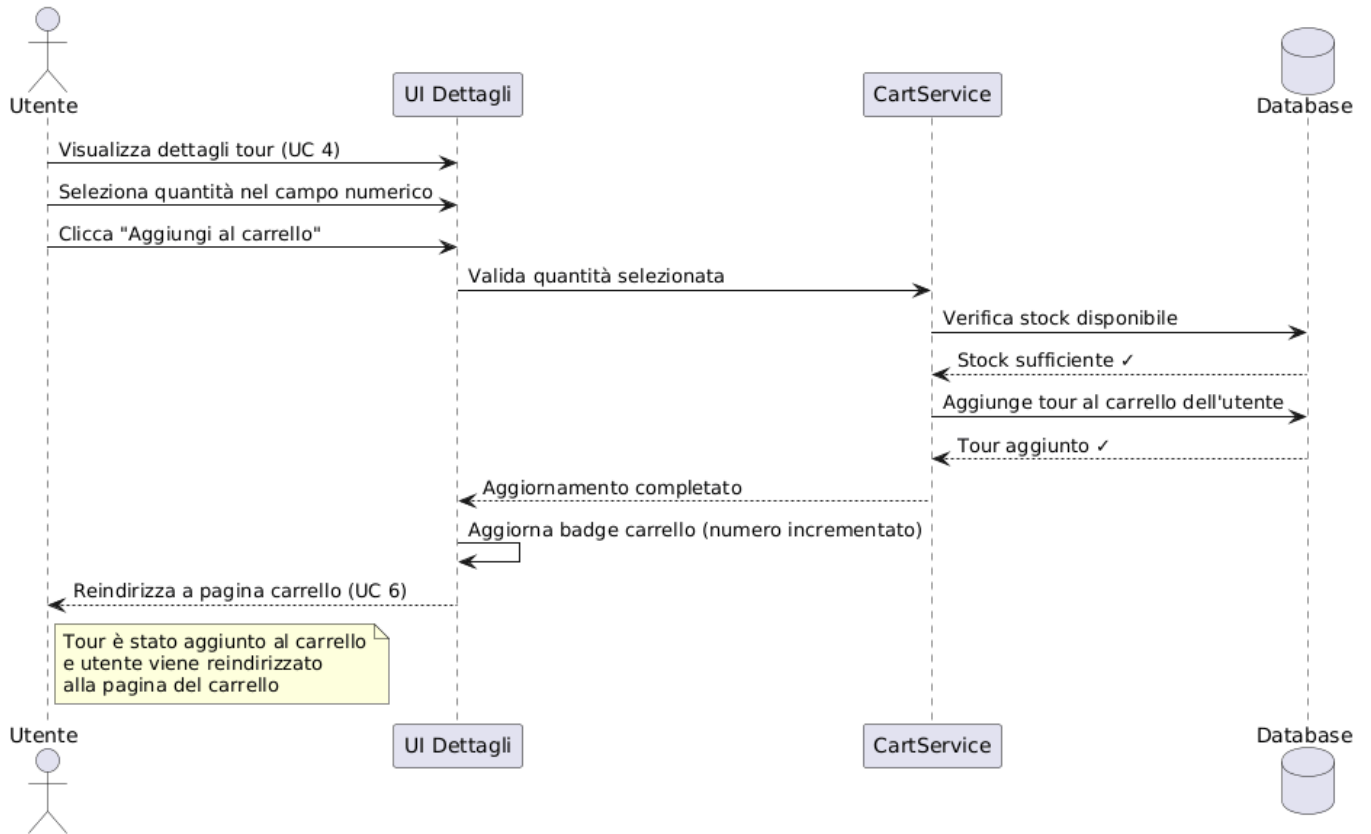
UC 3.1 - Ricerca Nessun Risultato (Exception Path)



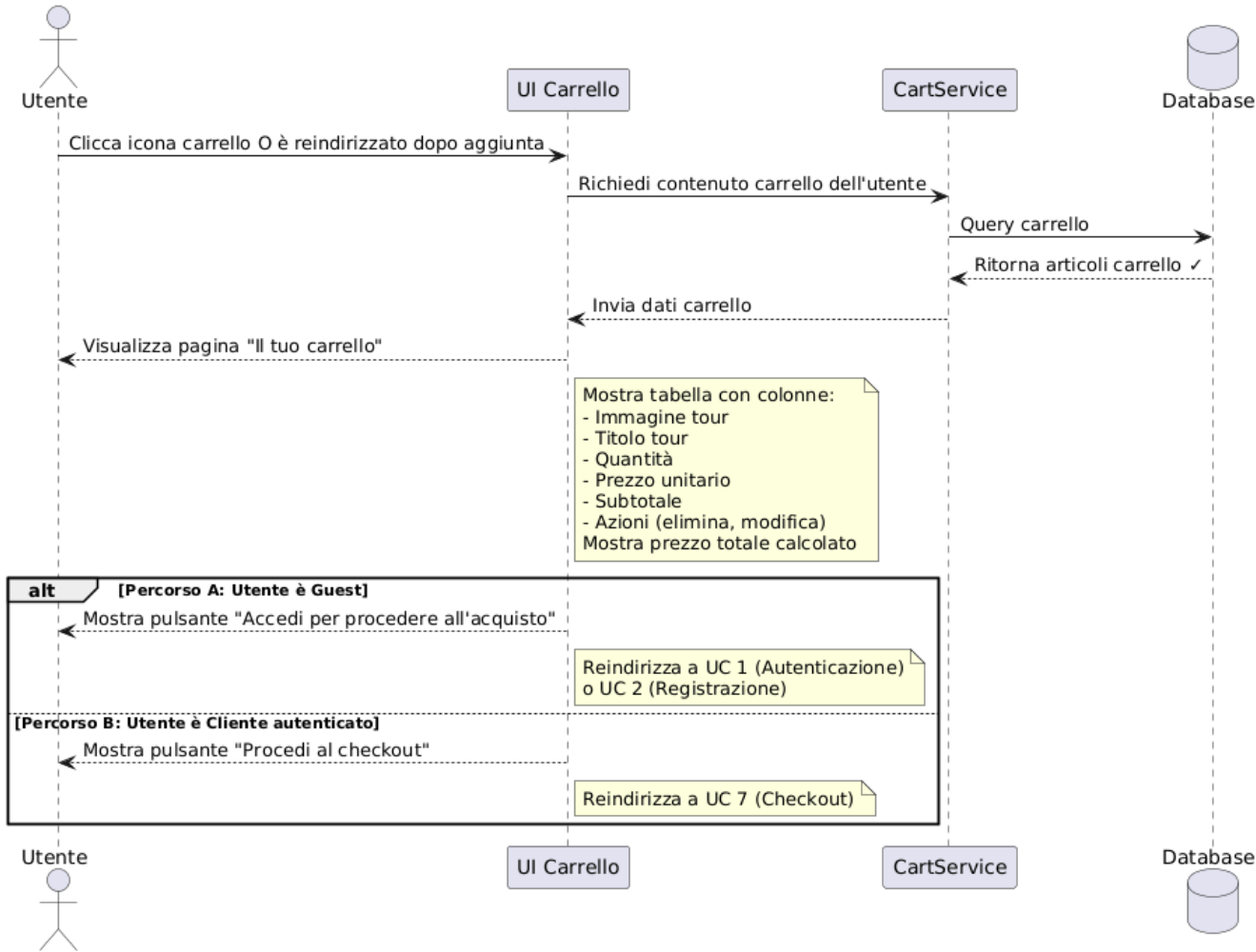
UC 4 - Visualizzare Dettagli Tour



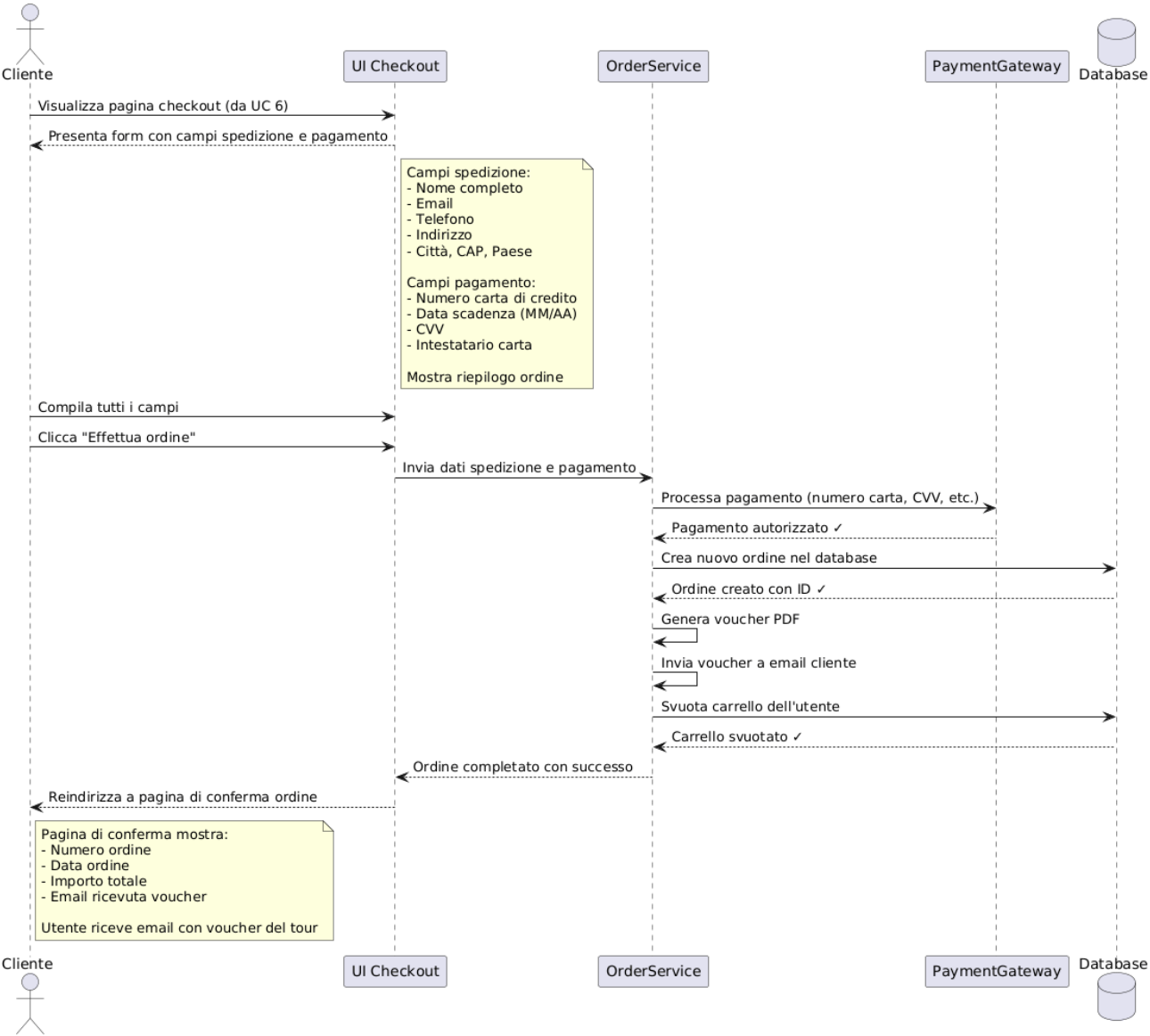
UC 5 - Aggiungere Tour al Carrello



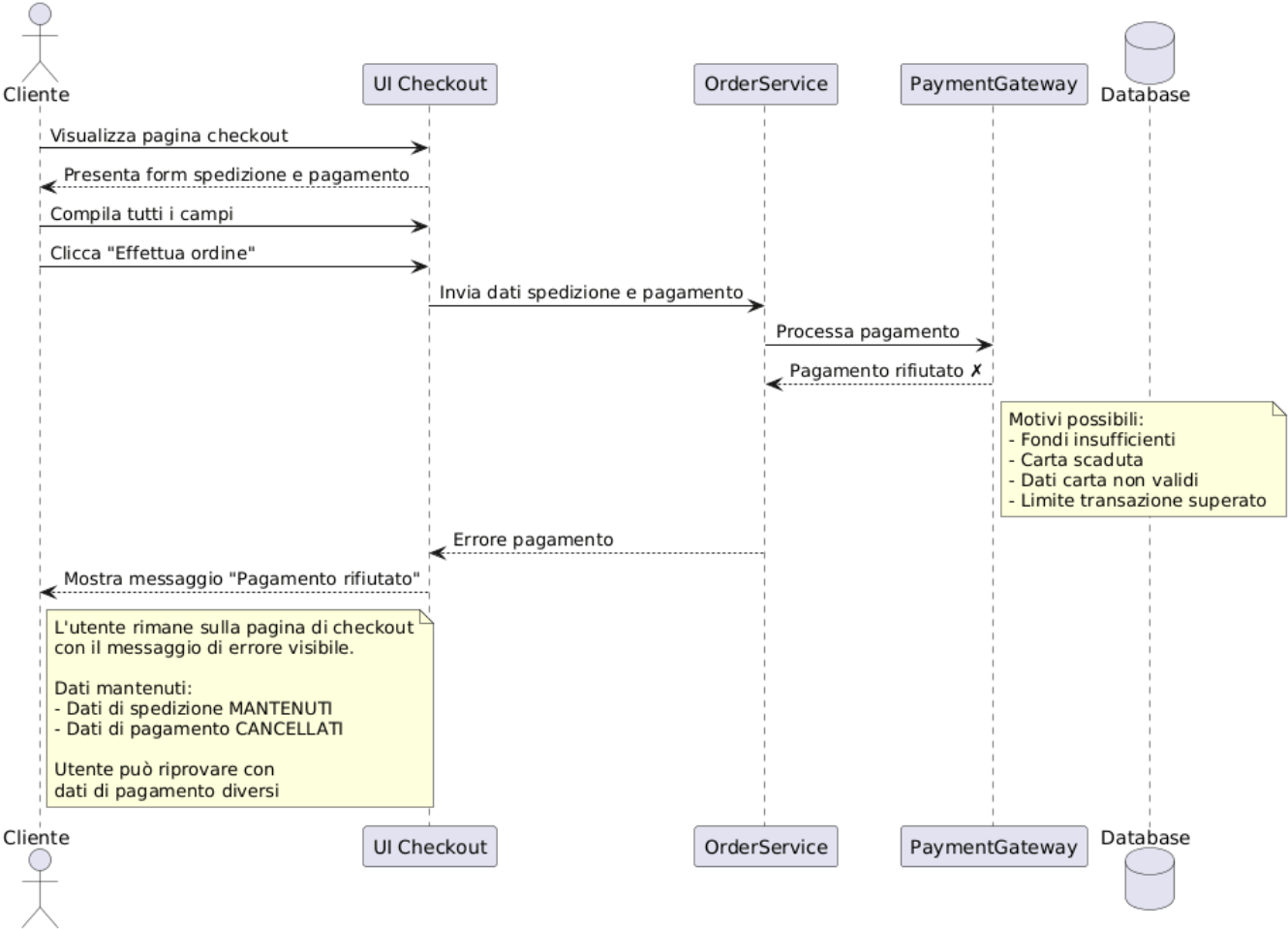
UC 6 - Visualizza Carrello



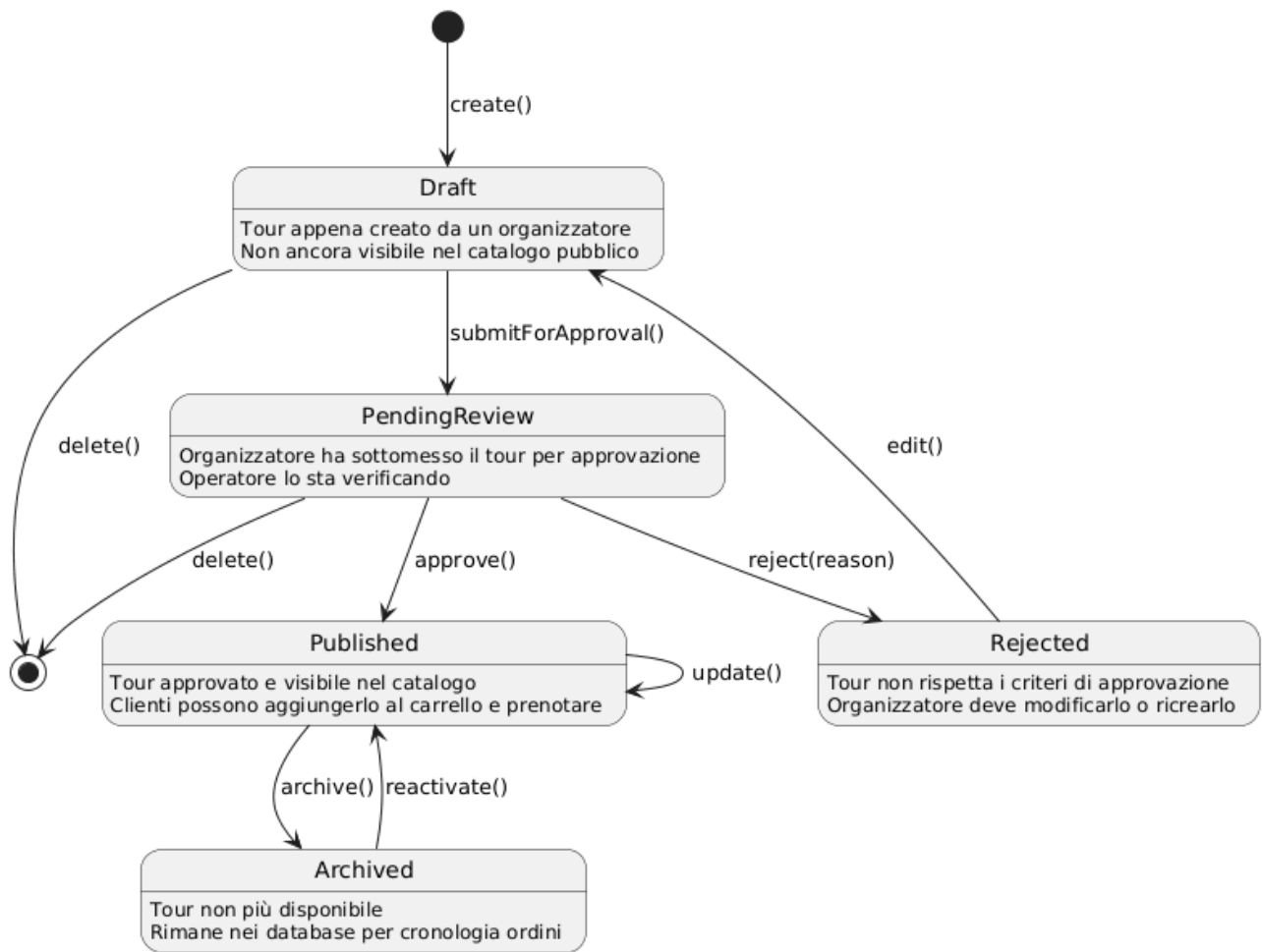
UC 7 - Checkout (Success Path)



UC 7.1 - Pagamento Rifiutato (Exception Path)

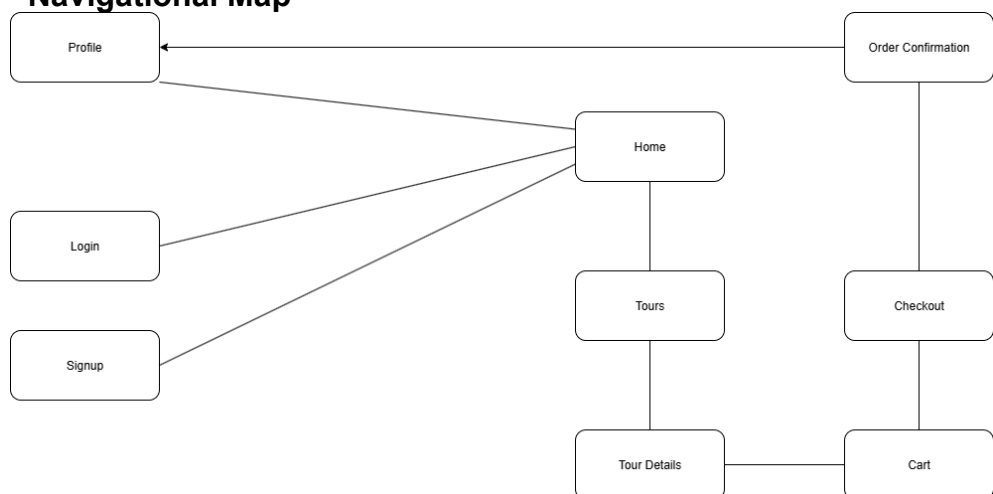


Tour - (State Chart)

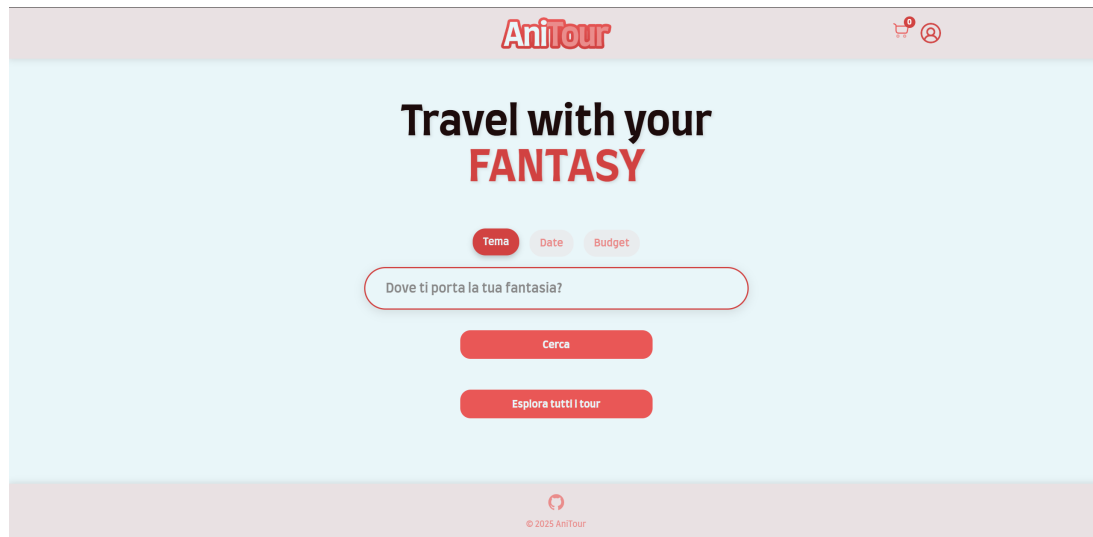


3.4.5. User interface-navigational paths and screen mock-ups

3.4.5.1. Navigational Map



3.4.5.2. Screen mockups

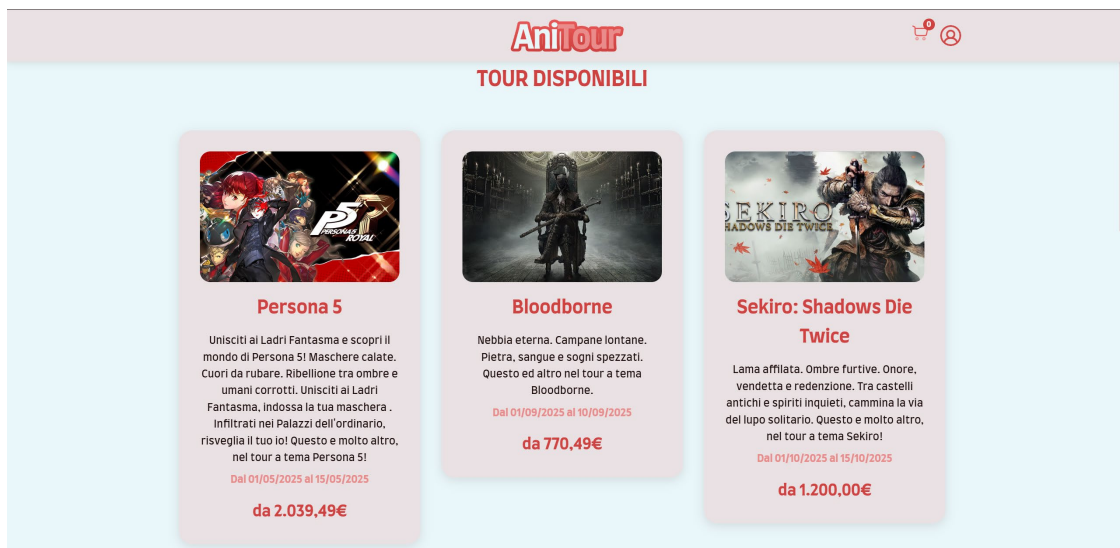


M-01: Homepage

Attori coinvolti: Guest, Utente

Use Cases correlato: UC 3 (Visualizzare catalogo tour), UC 1 (Autenticazione), UC 6 (Visualizza carrello)

Descrizione: Pagina di accesso principale che presenta pulsante per mostrare il catalogo tour con funzionalità di ricerca e filtro. Accesso a login/registrazione per utenti non autenticati.

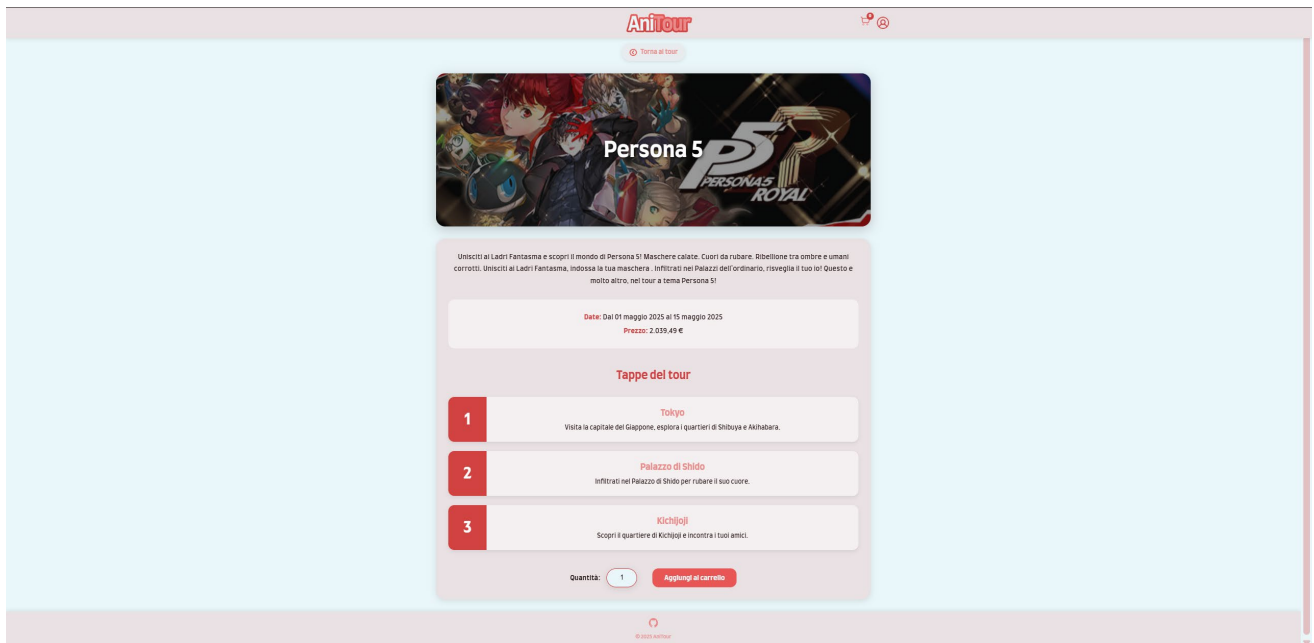


M-02: Pagina tour disponibili

Attori coinvolti: Guest, Utente

Use Cases correlati: UC 3 (Visualizzare catalogo tour), UC 4 (Visualizza dettagli tour), UC 1 (Autenticazione), UC 6 (Visualizza carrello)

Descrizione: Pagina di visualizzazione di tour dopo la ricerca o dopo aver cliccato sul pulsante per mostrarli tutti.

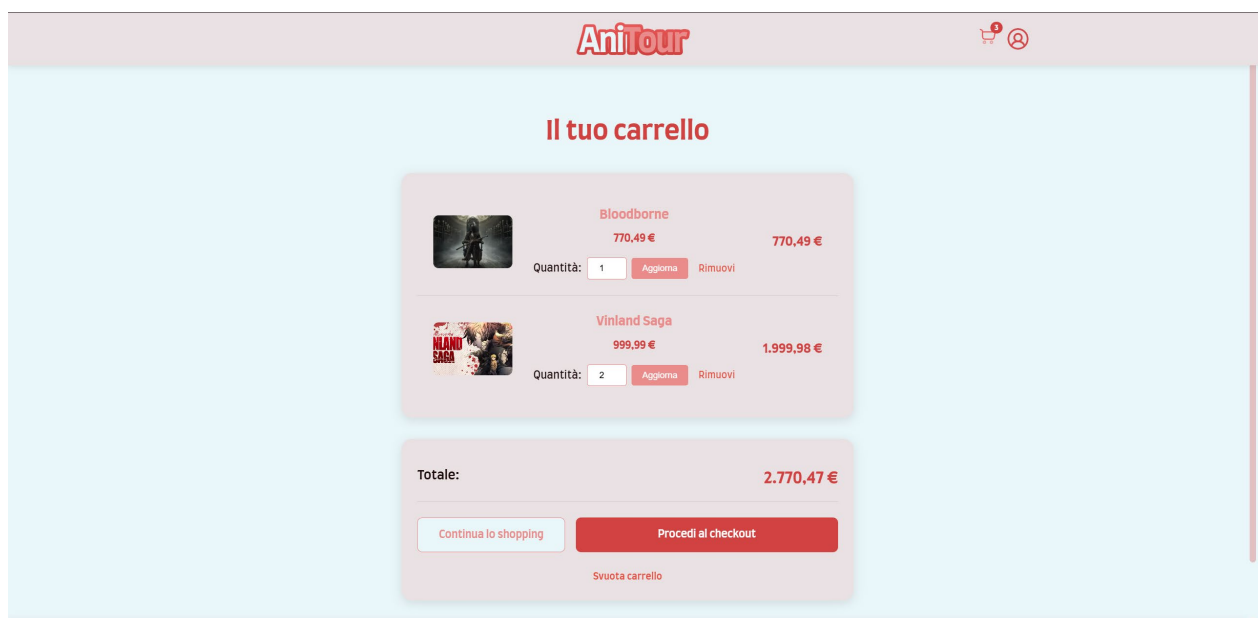


M-03: Pagina dettagli tour

Attori coinvolti: Guest, Utente

Use Cases correlati: UC 4 (Visualizzare dettagli tour) UC 5 (Aggiungere tour al carrello), UC 1 (Autenticazione), UC 6 (Visualizza carrello)

Descrizione: Pagina dettagliata di un tour specifico con descrizione completa, tappe, prezzi e pulsante di acquisto.



M-04: Pagina carrello



Attori coinvolti: Guest, Cliente

Use Cases correlati: UC 6 (Visualizzare carrello) UC 7 (Checkout), UC 1 (Autenticazione)

Descrizione: Pagina dettagliata del carrello con riepilogo elementi aggiunti

e prezzo totale.

AniTour



Checkout

Informazioni di spedizione

Nome completo

Email

Telefono

Indirizzo

Città

CAP

Paese

Dati di pagamento

Numero carta di credito

1234 5678 9012 3456

Data di scadenza (MM/AA)

CVV

MM/AA

123

Intestatario carta

Torna al carrello

Effettua ordine

Riepilogo ordine

Bloodborne

770.49 €

Quantità: 1


Vinland Saga

1.999.98 €

Quantità: 2

Totale:

2.770,47 €



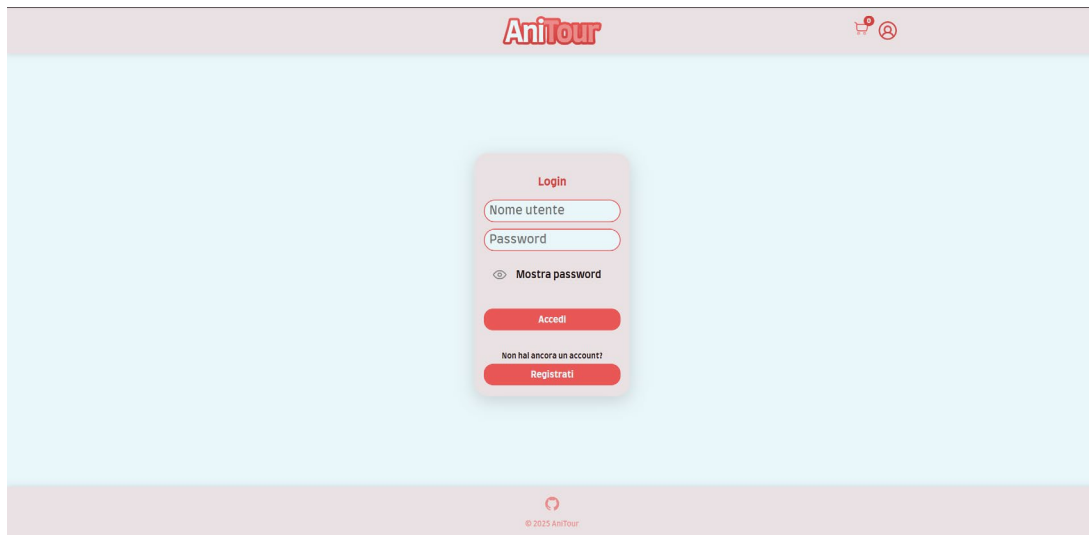
© 2025 AniTour

M-05: Pagina di checkout

Attori coinvolti: Cliente

Use Cases correlati: UC 7 (Checkout), UC 1 (Autenticazione), UC 6 (Visualizzare carrello)

Descrizione: Pagina di checkout per finalizzare l'ordine da parte del cliente.



M-06: Pagina di autenticazione

Attori coinvolti: Guest, Utente

Use Cases correlati: UC 1 (Autenticazione), UC 2 (Registrazione), UC 6 (Visualizzare carrello)

Descrizione: Pagina che permette l'autenticazione di un utente o la registrazione di un guest.

4. Glossary

- Tour: Pacchetto viaggio tematico su base nerd/anime/videogiochi/giapponese
- Guest: Non autenticato
- Cliente: Registrato/autenticato che acquista
- Organizzatore: Crea/gestisce tour
- Operatore: Gestore del sistema
- Voucher: Documento PDF ricevuto dopo acquisto tour

Revision History

Data	Versione	Descrizione	Autore
28/10/2025	1.0	Creazione documento RAD	Vincenzo Chiocca
01/11/25	1.1	Aggiunti Use Case con i rispettivi diagrammi, aggiunti mockups. Effettuate altre correzioni minori al documento.	Vincenzo Chiocca
10/11/25	1.2	Sviluppati punti inerenti ad Object model e Dynamic model	Vincenzo Chiocca

Partecipanti:

Nome	Matricola
Vincenzo Chiocca	0512119182
Salvatore Merola	0512120979