

1 Diagrammi dei casi d'uso

Seguono ora i diagrammi dei casi d'uso e per ciascuno di essi una dettagliata descrizione dei casi d'uso presenti. Per rendere il più possibile comoda la lettura da parte del lettore, gli autori hanno scelto di ordinare la descrizione dei casi d'uso secondo il loro identificativo numerico crescente. Tale identificativo è stato riportato anche sulle rappresentazioni grafiche dei diagrammi.

Si tenga inoltre presente che, al fine di non appesantire le rappresentazioni grafiche, sono state omesse le relazioni di associazione tra attore e caso d'uso qualora esse fossero state già esplicitate in altri diagrammi e ripeterle nell'attuale non avrebbe contribuito in modo indispensabile alla comprensione della logica del sistema. Sempre nel perseguimento di tale scopo si è scelto di colorare in grigio i casi d'uso presenti in un diagramma, ma originari di un'altro diagramma.

Il diagramma dei casi d'uso è composto da 5 *packages* oltre al *package* degli attori; nel dettaglio essi sono:

- Gestione Cliente (CU 01) - (CU 10)
- Gestione Segreteria (CU 11) - (CU 15)
- Gestione Amministrazione (CU 16) - (CU 23)
- Gestione Sistema (CU 24) - (CU 27)
- Gestione Statistiche (CU 28) - (CU 35)

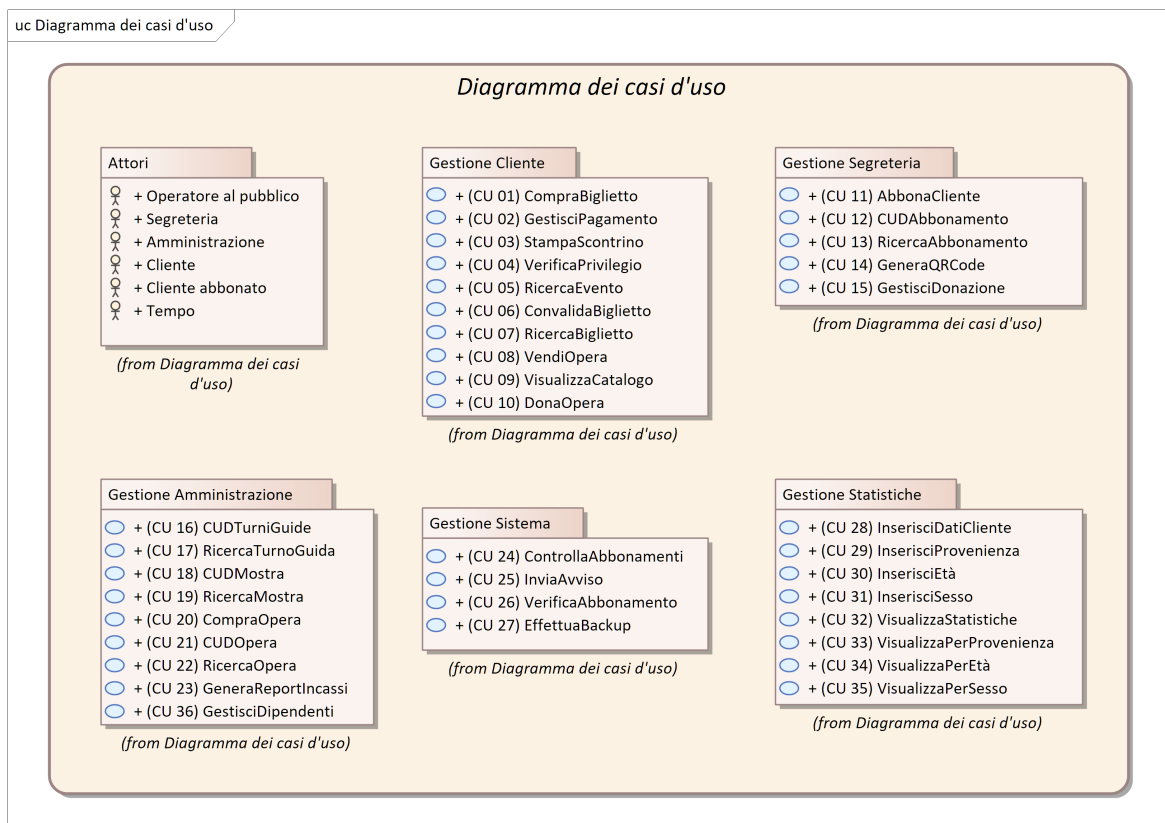


Figura 1: diagramma dei casi d'uso: **Diagramma dei casi d'uso**

2 Gestione Cliente

Il package degli attori contiene i 'soggetti' del dominio del problema in esame, ovvero coloro che interagiranno con il presente sistema al fine di svolgere il loro lavoro. Nel dettaglio i 6 attori sono:

- Cliente
- Cliente abbonato
- Operatore al pubblico
- Segreteria
- Amministrazione
- Tempo

Si noti la relazione di generalizzazione che prende posto tra l'attore *Cliente* e l'attore *Cliente abbonato*. Mentre del secondo il sistema possiede le informazioni anagrafiche lasciate al momento della sua registrazione, non possiede che dati in forma anonima del *Cliente*, ovvero di colui che usufruisce dei servizi offerti dal museo senza essere in possesso di un abbonamento. In altre parole, poiché il presente sistema non obbliga il *Cliente* a registrarsi per accedere alla struttura e usufruire dei suoi servizi, gli unici dati che il sistema memorizza su di lui sono quelli (facoltativi) rilasciati in forma anonima per le statistiche.

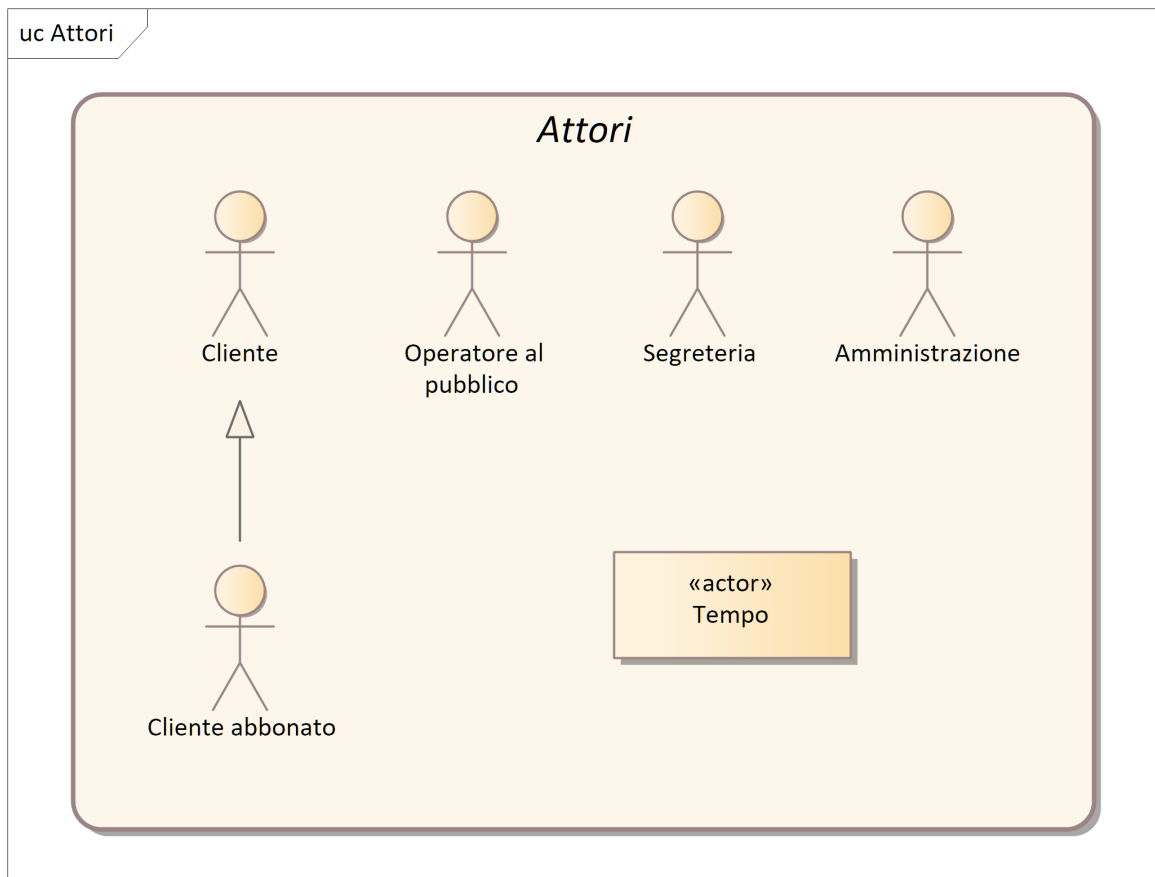


Figura 2: Package: **Attori**

3 Gestione Cliente

Il diagramma dei casi d'uso **Gestione Cliente** racchiude i casi d'uso volti a fornire servizi al cliente. Nel dettaglio, i casi d'uso di pertinenza sono:

- CompraBiglietto (CU 01)
- GestisciPagamento (CU 02)
- StampaScontrino (CU 03)
- VerificaPrivilegio (CU 04)
- RicercaEvento (CU 05)
- ConvalidaBiglietto (CU 06)
- RicercaBiglietto (CU 07)
- VendiOpera (CU 08)
- VisualizzaCatalogo (CU 09)
- DonaOpera (CU 10)

Gli attori coinvolti in questi casi d'uso sono il *Cliente abbonato*, generalizzato nell'attore *Cliente*, e l'attore *Operatore al pubblico*.

Si noti che, poiché questo sistema informatico non prevede terminali sui quali il cliente può interfacciarsi direttamente, i casi d'uso istanziabili dal cliente hanno come attore secondario l'operatore al pubblico, il quale, una volta compresa la necessità del cliente, avvia la procedura interfacciandosi con il sistema.

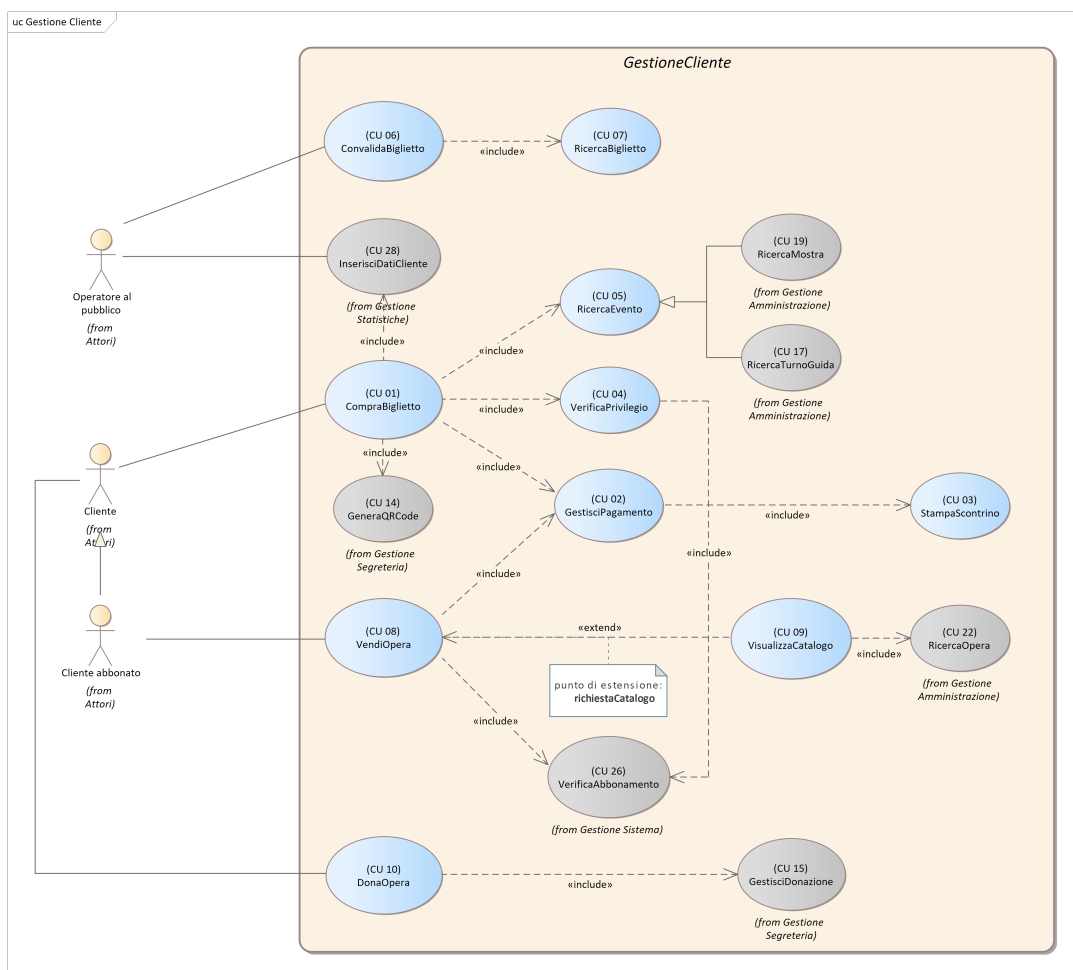


Figura 3: diagramma dei casi d'uso: **Gestione Cliente**

CompraBiglietto (CU 01)

Questo caso d'uso consente l'acquisto di un biglietto per l'accesso al museo o ad una mostra.

Attori primari: Cliente.

Attori secondari: Operatore al pubblico.

Precondizioni: Nessuna.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario esprime la volontà di acquistare un biglietto.
2. L'attore secondario prende coscienza della volontà dell'attore primario e lo assiste nell'operazione.
3. L'attore secondario effettua l'accesso con le sue credenziali se richiesto.
4. **if** l'attore primario vuole accedere alla mostra:
 - 4.1. L'attore secondario seleziona la mostra attuale.
5. L'attore secondario inserisce nel sistema i dati relativi alla tipologia del biglietto.
6. **if** l'attore primario richiede anche il servizio guida:
 - 6.1. `include(RicercaTurnoGuida)`.
 - 6.2. L'attore secondario comunica all'attore primario gli orari dei turni delle guide disponibili.
 - 6.3. **if** c'è almeno un turno disponibile **and** l'attore primario ne sceglie uno:
 - 6.3.1. L'attore secondario inserisce la scelta dell'attore primario nel sistema.
7. **if** l'attore primario comunica all'attore secondario di voler usufruire della tariffa ridotta:
 - 7.1. `include(VerificaPrivilegio)`.
8. Il sistema memorizza le informazioni e calcola il costo totale del biglietto, mostrando in un opportuno il resoconto dell'operazione.
9. `include(InserisciDatiCliente)`.
10. `include(GestisciPagamento)`.
11. Il sistema genera un identificativo univoco per il biglietto.
12. `include(GeneraQRCode)`.
13. Il sistema applica il codice QR sul fronte del biglietto e lo stampa.

Sequenza eventi alternativa: *PrivilegioNonVerificato*

1. La sequenza di eventi alternativa inizia dopo il passo 6.1. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore secondario la non validità del privilegio necessario per poter usufruire della tariffa ridotta.

Sequenza eventi alternativa: *AbbonamentoScaduto*

1. La sequenza di eventi alternativa inizia dopo il passo 6.1. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore secondario la non validità dell'abbonamento inserito.

Sequenza eventi alternativa: *PagamentoRifutato*

1. La sequenza di eventi alternativa inizia dopo il passo 8. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore primario l'errore avvenuto durante il pagamento ed esortandolo a riprovare.

GestisciPagamento (CU 02)

Questo caso d'uso permette di gestire il pagamento di un servizio offerto dal museo.

Attori primari: Segreteria, Operatore al pubblico.

Attori secondari: Cliente.

Precondizioni:

1. Il sistema sta elaborando la richiesta da parte del cliente di acquisto di un biglietto, di un abbonamento **or** di un'opera.
2. L'attore primario ha comunicato al sistema il bene che il cliente desidera acquistare.

Postcondizioni:

1. Il sistema ha registrato il pagamento.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario deve gestire il pagamento di un acquisto effettuato da parte del cliente.
2. Il sistema comunica all'attore primario il costo totale dell'acquisto.
3. L'attore primario comunica all'attore secondario il totale da pagare e gli chiede la modalità di pagamento che desidera seguire.
4. **if** l'attore secondario comunica di voler pagare in contanti:
 - 4.1. L'attore primario gestisce manualmente il pagamento usufruendo della cassa per erogare l'eventuale resto e riporre il denaro.
 - 4.2. L'attore primario comunica l'avvenuto pagamento al sistema.
5. **else if** l'attore secondario comunica di voler pagare utilizzando il *Point Of Sale*:
 - 5.1. L'attore primario consegna all'attore secondario il *Point Of Sale*.
 - 5.2. L'attore secondario inserisce la sua carta di credito, di debito **or** prepagata, e segue i passaggi richiesti per autenticarla.
 - 5.3. Il sistema riceve l'esito della transazione dal *Point Of Sale*.
6. Il sistema registra l'avvenuto pagamento.
7. `include(StampaScontrino)`

Sequenza eventi alternativa: *TransazioneNonRiuscita*

1. La sequenza di eventi alternativa inizia dopo il passo 5.3. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore primario l'errore avvenuto durante il pagamento ed esortandolo a ripetere l'operazione.

StampaScontrino (CU 03)

Questo caso d'uso permette di stampare lo scontrino con l'importo pagato.

Attori primari: Segreteria, Operatore al pubblico.

Attori secondari: Cliente.

Precondizioni:

1. Il processo di pagamento terminato ed è andato a buon fine.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario deve stampare lo scontrino per l'avvenuto pagamento.
2. Il sistema comunica al registratore di cassa telematico le informazioni del pagamento.
3. L'attore primario attende l'avvenuta stampa e consegna all'attore secondario.

Sequenza eventi alternativa: *ConnessioneAssente*

1. La sequenza di eventi alternativa inizia dopo il passo 2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore primario l'impossibilità di connessione da parte del registratore di cassa telematico, e lo esorta a riprovare **or** a riportare a mano lo scontrino su carta.

VerificaPrivilegio (CU 04)

Questo caso d'uso permette di verificare il privilegio di un cliente ad aderire al costo ridotto o di essere esente dal pagamento per l'acquisto di un biglietto.

Attori primari: Operatore al pubblico.

Attori secondari: Cliente.

Precondizioni:

1. Si è verificata la volontà da parte del cliente di acquistare un biglietto.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario desidera verificare se il cliente può usufruire dei privilegi di costo ridotto o nullo.
2. L'attore primario chiede l'attore secondario quale privilegio desidera sfruttare.
3. **if** l'attore secondario chiede di utilizzare il privilegio sull'età :
 - 3.1. L'attore primario chiede all'attore secondario di mostrare un documento d'identità.
 - 3.2. **if** L'età dell'attore secondario è minore o uguale a 5 anni:
 - 3.2.1. L'attore primario conferma l'esenza dal pagamento.
 - 3.3. **else if** L'età dell'attore secondario è minore di 18 anni **or** è maggiore di 59 anni:
 - 3.3.1. L'attore primario conferma l'aderenza alla tariffa ridotta.
4. **else if** l'attore secondario chiede di utilizzare il privilegio sull'occupazione :
 - 4.1. L'attore primario chiede all'attore secondario di mostrare il badge di studente **or** il badge di professore.
 - 4.2. **if** l'attore secondario ha consegnato il badge di studente **and** l'attore secondario ritiene valido il badge consegnatogli:
 - 4.2.1. L'attore primario conferma l'aderenza alla tariffa ridotta.
 - 4.3. **else if** l'attore secondario ha consegnato il badge di professore **and** l'attore secondario ritiene valido il badge consegnatogli:
 - 4.3.1. L'attore primario conferma l'esenza dal pagamento.
5. **else if** l'attore secondario chiede di utilizzare il privilegio sul possesso di certificato di invalidità:
 - 5.1. L'attore primario chiede all'attore secondario di mostrare il certificato di invalidità.
 - 5.2. L'attore primario inserisce nel sistema il codice del certificato di invalidità.
 - 5.3. **if** il sistema conferma la validità del certificato:
 - 5.3.1. L'attore primario conferma l'esenza dal pagamento.
6. **else if** l'attore secondario chiede di utilizzare il privilegio sul possesso di abbonamento:
 - 6.1. L'attore primario chiede all'attore secondario di mostrare il suo abbonamento.
 - 6.2. `include(VerificaAbbonamento)`
 - 6.3. **if** l'abbonamento non è scaduto:
 - 6.3.1. L'attore primario conferma l'esenza dal pagamento.
7. **else** l'attore primario non applica alcuna riduzione al costo del biglietto.

Sequenza eventi alternativa: Nessuna.

RicercaEvento (CU 05)

Questo caso d'uso consente di ricercare un evento.

Attori primari: Operatore al pubblico.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole effettuare la ricerca di un evento **or** il sistema deve effettuare la ricerca di un evento.
2. **if** il caso d'uso è stato avviato dall'attore:
 - 2.1. Il sistema chiede i criteri di ricerca.
 - 2.2. L'attore inserisce i criteri di ricerca richiesti.
3. Il sistema registra i parametri di ricerca.
4. **for each** evento nel database:
 - 4.1. Il sistema controlla se l'evento soddisfa i criteri di ricerca.
 - 4.2. **if** l'evento soddisfa il criterio:
 - 4.2.1. Il sistema aggiunge l'evento alla lista dei risultati.
5. **if** la lista dei risultati contiene almeno un evento:
 - 5.1. Il sistema mostra la lista dei risultati con tutte le loro informazioni.
6. **else:**
 - 6.1. Il sistema comunica all'attore che nessun evento soddisfa il criterio specificato.

Sequenza eventi alternativa: Nessuna.

ConvalidaBiglietto (CU 06)

Questo caso d'uso consente di verificare la validità di un biglietto mediante il suo identificativo stampato sul fronte sotto forma di codice QR.

Attori primari: Operatore al pubblico.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.
2. L'attore primario dispone di un biglietto con codice QR.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole eseguire la convalida di un biglietto.
2. Il sistema chiede il codice identificativo del biglietto.
3. **while** il codice identificativo non corrisponde ad alcun biglietto memorizzato nel sistema:
 - 3.1. L'attore primario inserisce il codice identificativo eseguendo una scannerizzazione del codice QR stampato sul fronte del biglietto.
 - 3.2. `include(RicercaBiglietto)`
4. **if** la data attuale coincide con il giorno di validità del biglietto:
 - 4.1. Il sistema mostra un opportuno messaggio di successo dell'operazione.
5. **else:**
 - 5.1. Il sistema mostra un opportuno messaggio di errore.

Sequenza eventi alternativa: *IdentificativoNonTrovato*

1. La sequenza di eventi alternativa inizia dopo il passo 3.2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore l'inesistenza dell'identificativo all'interno del database ed esortandolo a ripetere il processo di scannerizzazione.
3. La sequenza di eventi principale prosegue dal punto 3.

RicercaBiglietto (CU 07)

Questo caso d'uso consente di ricercare un biglietto nel database del sistema mediante il suo identificativo.

Attori primari: Operatore al pubblico.

Attori secondari: Nessuno.

Precondizioni:

1. L'attore primario ha inserito il codice identificativo del biglietto nel sistema.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole effettuare la ricerca di un biglietto **or** il sistema deve effettuare la ricerca di un biglietto.
2. Il sistema ottiene il codice identificativo del biglietto.
3. **for each** biglietto nel database dei biglietti:
 - 3.1. **if** l'identificativo del biglietto coincide con l'identificativo fornito al sistema:
 - 3.1.1. Il sistema raccoglie tutte le informazioni del biglietto.
4. **if** la ricerca non ha prodotto alcun risultato:
 - 4.1. Il sistema mostra un opportuno messaggio di errore.
5. **else if** il caso d'uso è stato avviato dall'attore primario:
 - 5.1. Il sistema mostra un opportuno messaggio di successo dell'operazione.
 - 5.2. Il sistema mostra, in una schermata opportuna, tutte le informazioni del biglietto, tra le quali la sua tipologia e l'eventuale presenza del servizio di guida.

Sequenza eventi alternativa: Nessuna.

VendiOpera (CU 08)

Questo caso d'uso consente la vendita di un'opera da parte del museo.

Attori primari: Cliente abbonato.

Attori secondari: Operatore al pubblico.

Precondizioni:

1. L'attore primario deve disporre di un abbonamento valido.

Postcondizioni:

1. Nel caso in cui l'operazione sia andata a buon fine, il sistema ha segnato come vendute le opere oggetto di transazione.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario esprime la volontà di acquistare un'opera di una mostra.
2. L'attore secondario riceve la richiesta e chiede all'attore primario di mostrare il suo abbonamento.
3. `include(VerificaAbbonamento)`
4. `extention point: richiestaCatalogo`
5. **for each** opera scelta dall'attore primario:
 - 5.1. L'attore secondario aggiunge l'opera al carrello.
6. L'attore secondario comunica al sistema la conferma dell'operazione.
7. `include(GestisciPagamento)`
8. **for each** opera venduta:
 - 8.1. Il sistema segna come venduta l'opera.

Sequenza eventi alternativa: *OperazioneAnnullata*

1. La sequenza di eventi alternativa inizia in qualunque momento.
2. L'attore primario annulla l'acquisto delle opere.

Sequenza eventi alternativa: *AbbonamentoScaduto*

1. La sequenza di eventi alternativa inizia dopo il passo 3. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore secondario la non validità dell'abbonamento inserito.

Sequenza eventi alternativa: *PagamentoRifutato*

1. La sequenza di eventi alternativa inizia dopo il passo 7. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore primario l'errore avvenuto durante il pagamento ed esortandolo a riprovare.

VisualizzaCatalogo (CU 09)

Questo caso d'uso permette di generare ed eventualmente stampare una lista delle opere appartenenti alle mostre del museo ed in quanto tali vendibili al cliente richiedente.

Attori primari: Operatore al pubblico.

Attori secondari: Nessuno.

Precondizioni del segmento 1:

1. L'attore secondario ha richiesto esplicitamente la volontà di visualizzare il catalogo delle opere vendibili.

Postcondizioni del segmento 1: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario riceve la richiesta di visualizzare il catalogo delle opere vendibili.
2. **for each** mostra nel database delle mostre:
 - 2.1. **include**(RicercaMostra)
 - 2.2. Il sistema genera una presentazione della mostra.
 - 2.3. **for each** opera nella mostra:
 - 2.3.1. **include**(RicercaOpera)
 - 2.3.2. Il sistema genera una presentazione dell'opera e la aggiunge alla presentazione della mostra.
 - 2.4. Il sistema aggiunge la presentazione della mostra al catalogo.
3. Il sistema mostra a schermo il catalogo generato, offrendo all'attore primario la possibilità di mandarlo in stampa.

Sequenza eventi alternativa del segmento 1: *NessunaOperaVendibile*

1. La sequenza di eventi alternativa inizia dopo il passo 3. della sequenza di eventi principale.
2. Il sistema comunica all'attore primario l'assenza di opere vendibili e lo esorta a portare a termine l'operazione di vendita.

DonaOpera (CU 10)

Questo caso d'uso consente la donazione di un'opera da parte di un cliente.

Attori primari: Cliente.

Attori secondari: Operatore al pubblico, Segreteria.

Precondizioni: Nessuna.

Postcondizioni:

1. Il sistema ha memorizzato la richiesta insieme a tutti i dati dell'opera in fase di donazione.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario esprime la volontà di donare un'opera al museo.
2. L'attore secondario *Operatore al pubblico* riceve la richiesta ed inserisce nel sistema il nome, l'autore, le dimensioni, il tipo e l'immagine dell'opera.
3. L'attore secondario *Operatore al pubblico* chiede all'attore primario anche un indirizzo di posta elettronica attraverso il quale verrà comunicata la scelta della segreteria.
4. L'attore secondario *Operatore al pubblico* ripone l'opera in un magazzino appropriato in attesa dell'esito della richiesta.
5. Il sistema registra le informazioni e notifica l'attore *Segreteria* della richiesta.
6. `include(GestisciDonazione)`

Sequenza eventi alternativa: Nessuna.

4 Gestione Segreteria

Il diagramma dei casi d'uso **Gestione Segreteria** racchiude i casi d'uso necessari all'attore *Segreteria* per svolgere le sue operazioni orientate a migliorare il lato servizio-cliente. Nel dettaglio, i casi d'uso di pertinenza sono:

- AbbonaCliente (CU 11)
- CUDAbbonamento (CU 12)
- RicercaAbbonamento (CU 13)
- GeneraQRCode (CU 14)
- GestisciDonazione (CU 15)

L'attore coinvolto in questi casi d'uso è la *Segreteria*.

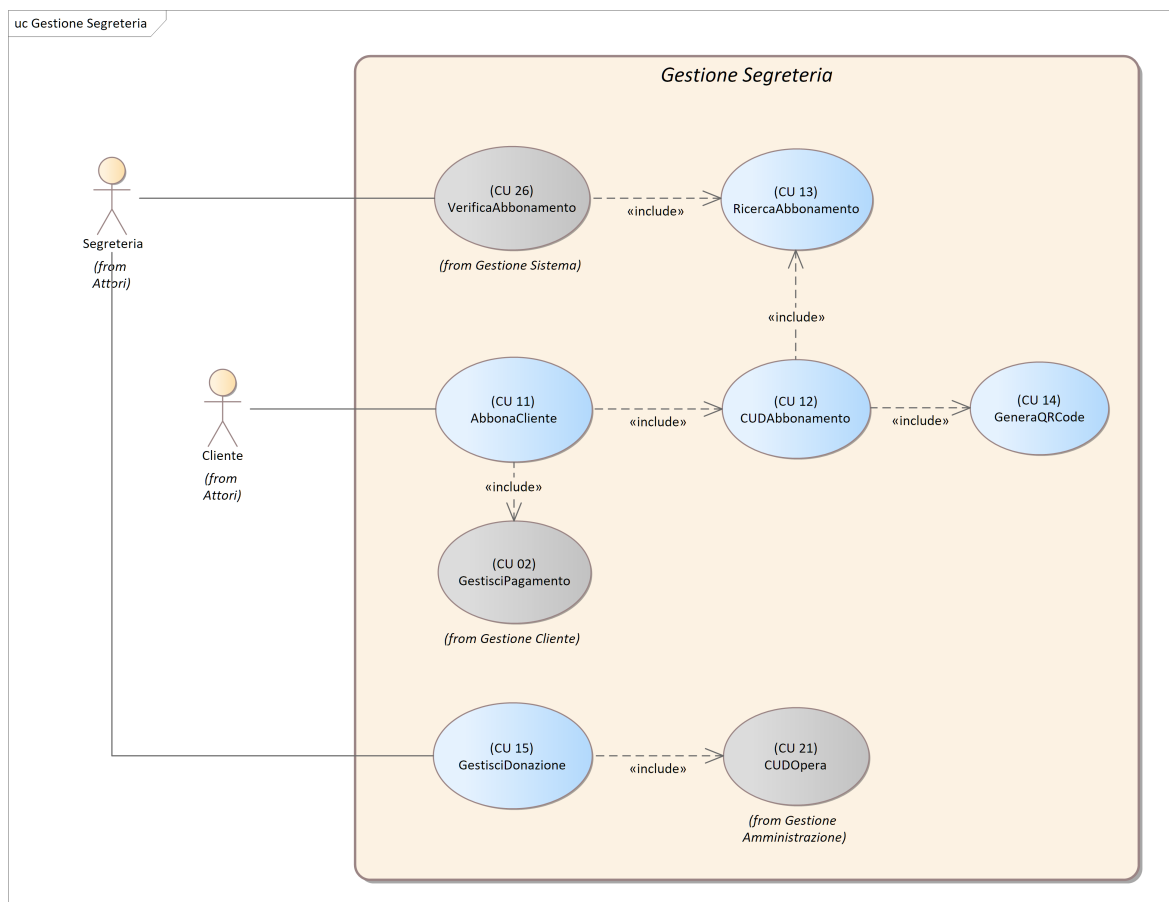


Figura 4: diagramma dei casi d'uso: **Gestione Segreteria**

AbbonaCliente (CU 11)

Questo caso d'uso consente la creazione di un abbonamento per un cliente non abbonato e il rinnovo di un abbonamento per un cliente abbonato.

Attori primari: Cliente non abbonato.

Attori secondari: Segreteria.

Precondizioni: Nessuno.

Postcondizioni:

1. Il sistema ha memorizzato il nuovo abbonamento nel database degli abbonamenti.
2. Il sistema ha memorizzato il pagamento ricevuto.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario esprime la volontà di abbonarsi o di rinnovare il suo abbonamento.
2. L'attore secondario prende coscienza della volontà dell'attore primario e lo assiste nell'operazione.
3. **if** l'attore primario vuole abbonarsi:
 - 3.1. L'attore secondario richiede all'attore primario i dati anagrafici e li registra nel sistema.
4. **else if** l'attore primario vuole rinnovare il suo abbonamento:
 - 4.1. L'attore secondario chiede all'attore primario di mostrare il suo abbonamento.
 - 4.2. L'attore secondario inserisce il codice identificativo eseguendo una scannerizzazione del codice QR stampato sul fronte dell'abbonamento.
 - 4.3. **include**(RicercaAbbonamento)
5. L'attore secondario informa l'attore primario sulle tipologie di abbonamento disponibili in base alla loro durata e al loro costo, quindi chiede all'attore secondario i dati necessari per l'abbonamento.
6. L'attore primario sceglie una tipologia di abbonamento e comunica all'attore secondario i dati richiesti.
7. L'attore secondario inserisce i dati nel sistema.
8. **include**(CUDAbbonamento)
9. L'attore secondario comunica all'attore primario l'importo da pagare.
10. **include**(GestisciPagamento)
11. L'attore secondario comunica all'attore primario l'esito positivo dell'operazione.
12. Il sistema applica il codice QR sul fronte dell'abbonamento e lo stampa.

Sequenza eventi alternativa: *OperazioneAnnullata.*

1. La sequenza di eventi alternativa inizia in qualunque momento.
2. L'attore primario annulla la richiesta di abbonarsi.

Sequenza eventi alternativa: *PagamentoRifutato.*

1. La sequenza di eventi alternativa inizia dopo il passo 9. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore primario l'errore avvenuto ed esortandolo a riprovare.

CUDAbbonamento (CU 12)

Questo caso d'uso consente di eseguire la creazione, la modifica o l'eliminazione di un abbonamento.

Attori primari: Segreteria.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni:

1. Il sistema ha memorizzato le eventuali modifiche nel database degli abbonamenti.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole eseguire un'operazione CUD relativa all'abbonamento.
2. **if** l'attore primario vuole creare un nuovo abbonamento:
 - 2.1. L'attore primario inserisce i dati nel sistema.
 - 2.2. Il sistema provvede a memorizzare i dati.
 - 2.3. Il sistema genera un identificativo univoco per l'abbonamento.
 - 2.4. **include**(GeneraQRCode)
 - 2.5. Il sistema inserisce il codice QR nell'abbonamento.
 - 2.6. Il sistema mostra a schermo l'abbonamento creato e un messaggio contenente l'esito dell'operazione.
3. **else if** l'attore vuole modificare l'abbonamento esistente:
 - 3.1. **include**(RicercaAbbonamento)
 - 3.2. **if** l'abbonamento viene trovato:
 - 3.2.1. l'attore primario aggiorna i dati dell'abbonamento trovato.
 - 3.2.2. Il sistema provvede a salvare le modifiche.
4. **else if** l'attore primario **or** il sistema vuole eliminare l'abbonamento:
 - 4.1. **if** il caso d'uso è stato attivato dall'attore primario
 - 4.1.1. L'attore secondario chiede al cliente di mostrare il suo abbonamento.
 - 4.1.2. L'attore secondario inserisce il codice identificativo eseguendo una scannerizzazione del codice QR stampato sul fronte dell'abbonamento.
 - 4.1.3. **include**(RicercaAbbonamento)
 - 4.1.4. Il sistema mostra un messaggio di conferma.
 - 4.1.5. **if** l'attore primario conferma l'operazione:
 - 4.1.5.1. Il sistema provvede ad eliminare l'abbonamento.
 - 4.2. **else if** il caso d'uso è stato attivato dal sistema:
 - 4.2.1. **include**(RicercaAbbonamento)
 - 4.2.2. Il sistema provvede ad eliminare l'abbonamento.
 - 4.3. Il sistema notifica a schermo l'esito positivo dell'operazione.

Sequenza eventi alternativa: *AbbonamentoNonTrovato.*

1. La sequenza di eventi alternativa inizia dopo i passi 3.1., 4.1.3. **or** 4.2.1. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore primario di non aver trovato un abbonamento corrispondente a quello ricercato.

RicercaAbbonamento (CU 13)

Questo caso d'uso consente di eseguire la ricerca di un abbonamento all'interno del database.

Attori primari: Segreteria.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario desidera effettuare la ricerca di un abbonamento **or** il sistema deve effettuare la ricerca.
2. **if** il caso d'uso è stato attivato dall'attore primario
 - 2.1. L'attore primario inserisce il codice identificativo dell'abbonamento da ricercare.
3. **for each** abbonamento nel database degli abbonamenti:
 - 3.1. **if** il codice identificativo dell'abbonamento corrisponde a quello inserito nel sistema:
 - 3.1.1. Il sistema mostra in un'opportuna finestra l'abbonamento trovato con tutte le relative informazioni.
4. **if** la ricerca non ha prodotto un risultato:
 - 4.1. il sistema comunica di non aver trovato l'abbonamento.

Sequenza eventi alternativa: Nessuna.

GeneraQRCode (CU 14)

Questo caso d'uso consente di generare un codiceQR a partire da un identificativo fornito.

Attori primari: Segreteria.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha generato un identificativo.
2. Tale identificativo è univoco all'interno del database.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando il sistema deve generare un codice QR.
2. Il sistema utilizza l'identificativo fornito per generare il codice QR.

Sequenza eventi alternativa: Nessuna.

GestisciDonazione (CU 15)

Questo caso d'uso consente di gestire una richiesta di donazione di un'opera da parte di un cliente.

Attori primari: Segreteria.

Attori secondari: Nessuno.

Precondizioni: Nel sistema è presente almeno una richiesta di donazione di un'opera.

Postcondizioni:

1. Il sistema ha autenticato l'attore primario.
2. Nel caso in cui la richiesta sia stata accettata, il sistema ha provveduto a memorizzare la nuova opera nel database delle opere.
3. Il sistema ha mandato un avviso comunicante l'esito della scelta al cliente donante mediante la posta elettronica.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario prende in carico la donazione.
2. Il sistema mostra in un'opportuna schermata tutti i dettagli della richiesta.
3. L'attore primario riceve le informazioni mostrate e decide se accettare la donazione.
4. L'attore primario inserisce la scelta nel sistema.
5. **if** il sistema ha ricevuto una conferma:
 - 5.1. `include(CUDOpera)`
6. Il sistema genera un messaggio di notifica in base alla scelta effettuata e la invia all'indirizzo di posta elettronica allegato alla richiesta.

5 Gestione Amministrazione

Il diagramma dei casi d'uso **Gestione Amministrazione** racchiude i casi d'uso necessari all'attore *Amministrazione* per svolgere le sue operazioni che riguardano la gestione interna e organizzativa del museo. Nel dettaglio, i casi d'uso di pertinenza sono:

- CUDTurniGuide (CU 16)
- RicercaTurnoGuida (CU 17)
- CUDMostra (CU 18)
- RicercaMostra (CU 19)
- CompraOpera (CU 20)
- CUDOpera (CU 21)
- RicercaOpera (CU 22)
- GeneraReportIncassi (CU 23)
- GestisciDipendenti (CU 36)

Gli attori coinvolti in questi casi d'uso sono l' *Amministrazione* e il *Cliente abbonato*.

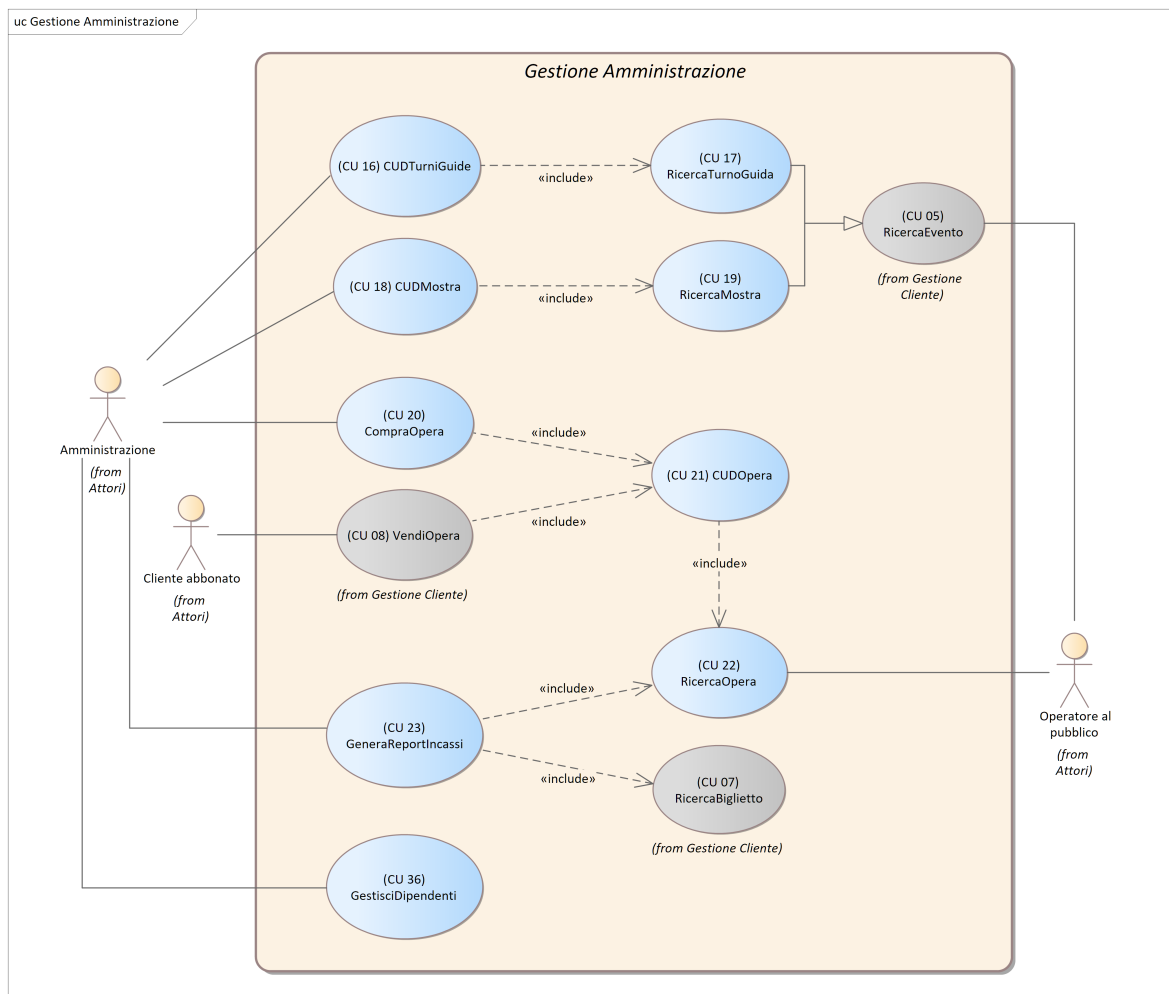


Figura 5: diagramma dei casi d'uso: **Gestione Amministrazione**

CUDTurniGuide (CU 16)

Questo caso d'uso consente l'inserimento, la modifica e la rimozione dei turni relativi alle guide che effettueranno i tour della mostra.

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni:

1. Il sistema ha memorizzato tutte le modifiche nel database dei turni guide.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CUD relativa ai turni delle guide.
2. **if** l'attore primario vuole inserire un nuovo turno:
 - 2.1. L'attore inserisce l'ora del nuovo turno e il nome dell' operatore al pubblico.
 - 2.2. `include(RicercaTurnoGuida)`
 - 2.3. Il sistema provvede ad aggiungere il nuovo turno.
3. **else if** l'attore primario vuole modificare un turno già esistente:
 - 3.1. L'attore primario inserisce l'ora del turno da modificare e il nome dell' operatore al pubblico.
 - 3.2. `include(RicercaTurnoGuida)`
 - 3.3. L'attore primario inserisce l'ora dove spostare il turno.
 - 3.4. `include(RicercaTurnoGuida)`
 - 3.5. Il sistema provvede a modificare l'orario del turno.
4. **else if** l'attore primario vuole eliminare un turno:
 - 4.1. L'attore primario inserisce l'ora del turno da eliminare e il nome dell' operatore al pubblico.
 - 4.2. `include(RicercaTurnoGuida)`
 - 4.3. Il sistema mostra un messaggio di conferma.
 - 4.4. **if** l'attore primario conferma l'operazione:
 - 4.4.1. Il sistema provvede ad eliminare il turno.

Sequenza eventi alternativa: *TurnoGiàEsistente*

1. La sequenza di eventi alternativa inizia dopo il passo 2.2. **or** 3.4. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando che per l'operatore al pubblico indicato è già programmato un turno nell'orario scelto.

Sequenza eventi alternativa: *TurnoNonTrovato*

1. La sequenza di eventi alternativa inizia dopo il passo 3.2. **or** 4.2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando che non è stato trovato un turno relativo ai parametri di ricerca.

RicercaTurnoGuida (CU 17)

Questo caso d'uso consente di ricercare un turno di una guida.

ID padre: (CU 05).

Attori primari: Amministrazione, Operatore al pubblico.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. (o 1.) Il caso d'uso inizia quando l'attore primario vuole effettuare la ricerca di un turno di una guida **or** il sistema deve effettuare la ricerca di un turno di una guida.
2. (2.) **if** il caso d'uso è stato avviato dall'attore:
 - 2.1. (2.1.) Il sistema chiede i criteri di ricerca.
 - 2.2. (o 2.2.) **if** l'attore non inserisce la data **nor** il nome della guida:
 - 2.2.1. Il sistema usa la data attuale come parametro di ricerca.
3. (3.) Il sistema registra i parametri di ricerca.
4. (o 4.) **for each** turno nel database dei turni delle guide:
 - 4.1. (o 4.1.) Il sistema controlla se il turno soddisfa i criteri di ricerca.
 - 4.2. (o 4.2.) **if** il turno soddisfa il criterio:
 - 4.2.1. (o 4.2.1.) Il sistema aggiunge il turno alla lista dei risultati.
5. (o 5.) **if** la lista dei risultati contiene almeno un turno:
 - 5.1. (5.1.) Il sistema mostra la lista dei risultati con tutte le loro informazioni.
6. (6.) **else:**
 - 6.1. (o 6.1.) Il sistema comunica all'attore che nessun turno di alcuna guida soddisfa il criterio specificato.

Sequenza eventi alternativa: Nessuna.

CUDMostra (CU 18)

Questo caso d'uso consente l'inserimento, la modifica e la rimozione di una mostra.

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni:

1. Il sistema ha memorizzato tutte le modifiche nel database delle mostre.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CUD relativa alla mostra.
2. **if** l'attore primario vuole inserire una nuova mostra:
 - 2.1. Il sistema chiede all'attore primario la data nella quale egli desidera organizzare una mostra.
 - 2.2. `include(RicercaMostra)`
 - 2.3. **for each** opera nel database:
 - 2.3.1. Il sistema aggiunge l'opera alla schermata.
 - 2.4. Il sistema chiede all'attore di selezionare le opere da aggiungere alla mostra.
 - 2.5. Il sistema provvede a creare la nuova mostra.
3. **else if** l'attore vuole modificare una mostra esistente:
 - 3.1. Il sistema chiede all'attore primario la data nella quale egli desidera modificare la mostra.
 - 3.2. `include(RicercaMostra)`
 - 3.3. **for each** opera nella mostra **and for each** opera nel database, ma non nella mostra:
 - 3.3.1. Il sistema aggiunge l'opera alla schermata.
 - 3.4. L'attore rimuove e aggiunge le opere alla mostra.
 - 3.5. Il sistema provvede a salvare le modifiche.
4. **else if** l'attore primario vuole eliminare una mostra esistente:
 - 4.1. Il sistema chiede all'attore primario la data nella quale egli desidera rimuovere la mostra.
 - 4.2. `include(RicercaMostra)`
 - 4.3. Il sistema mostra un messaggio di conferma.
 - 4.4. **if** l'attore primario conferma l'operazione:
 - 4.4.1. Il sistema provvede ad eliminare la mostra.

Sequenza eventi alternativa: *MostraGiàEsistente*

1. La sequenza di eventi alternativa inizia dopo il passo 2.2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando che è già programmata una mostra nella data specificata.

Sequenza eventi alternativa: *MostraNonTrovata*

1. La sequenza di eventi alternativa inizia dopo il passo 3.2. **or** 4.2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando che non è stata trovata alcuna mostra nella data specificata.

RicercaMostra (CU 19)

Questo caso d'uso consente di ricercare una mostra.

ID padre: (CU 05).

Attori primari: Amministrazione, Operatore al pubblico.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. (o 1.) Il caso d'uso inizia quando l'attore primario vuole effettuare la ricerca di una mostra **or** il sistema deve effettuare la ricerca di una mostra.
2. (2.) **if** il caso d'uso è stato avviato dall'attore:
 - 2.1. (2.1.) Il sistema chiede i criteri di ricerca.
 - 2.2. (o 2.2.) **if** l'attore non inserisce la data **nor** il nome della mostra:
 - 2.2.1. Il sistema usa la data attuale come parametro di ricerca.
3. (3.) Il sistema registra i parametri di ricerca.
4. (o 4.) **for each** mostra nel database delle mostre:
 - 4.1. (o 4.1.) Il sistema controlla se la mostra soddisfa i criteri di ricerca.
 - 4.2. (o 4.2.) **if** la mostra soddisfa il criterio:
 - 4.2.1. (o 4.2.1.) Il sistema aggiunge la mostra alla lista dei risultati.
5. (o 5.) **if** la lista dei risultati contiene almeno una mostra:
 - 5.1. (5.1.) Il sistema mostra la lista dei risultati con tutte le loro informazioni.
6. (6.) **else**:
 - 6.1. (o 6.1.) Il sistema comunica all'attore che nessuna mostra soddisfa il criterio specificato.

Sequenza eventi alternativa: Nessuna.

CompraOpera (CU 20)

Questo caso d'uso permette di comprare opere da enti esterni, di gestire le relative transazioni di denaro, e di aggiungerle al database delle opere.

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni:

1. Il sistema ha memorizzato la nuova opera nel database insieme a tutte le sue informazioni.
2. Il sistema ha memorizzato la transazione effettuata per l'acquisto.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario desidera comprare una nuova opera.
2. **if** l'attore primario ha già concluso un affare con un ente esterno privatamente:
 - 2.1. Il sistema chiede all'attore primario di inserire tutti i dettagli dell'opera acquistata.
 - 2.2. Il sistema chiede anche le informazioni sul costo della transazione.
3. **else**
 - 3.1. Il sistema si collega al sito web per la compravendita di opere tra enti privati.
 - 3.2. **for each** opera disponibile per l'acquisto:
 - 3.2.1. Il sistema preleva tutte le sue informazioni e aggiunge l'opera alla schermata.
 - 3.3. l'attore primario seleziona un'opera alle quale è interessato.
4. Il sistema mostra un messaggio di conferma.
5. **if** l'attore primario conferma l'operazione:
 - 5.1. Il sistema provvede a completare l'acquisto memorizzando la transazione.
 - 5.2. `include(CUDOpera)`

Sequenza eventi alternativa: *OperazioneAnnullata.*

1. La sequenza di eventi alternativa inizia in qualunque momento.
2. L'attore primario annulla l'operazione di acquisto.

Sequenza eventi alternativa: *ConnessioneNonRiuscita.*

1. La sequenza di eventi alternativa inizia dopo il passo 3.1. della sequenza di eventi principale.
2. Il sistema comunica all'attore primario, in un opportuno messaggio, l'indisponibilità del sito web per la compravendita di opere e lo esorta ad attendere prima di ripetere l'operazione.

CUDOpera (CU 21)

Questo caso d'uso consente l'inserimento, la modifica e la rimozione di un'opera.

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni:

1. Il sistema ha memorizzato tutte le modifiche nel database delle opere.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CUD relativa ad un'opera.
2. **if** l'attore primario vuole inserire una nuova opera:
 - 2.1. L'attore primario inserisce i dati relativi all'opera.
 - 2.2. `include(RicercaOpera)`
 - 2.3. Il sistema provvede ad aggiungere la nuova opera.
3. **else if** l'attore vuole modificare un'opera già esistente:
 - 3.1. L'attore primario inserisce i dati dell'opera da ricercare.
 - 3.2. `include(RicercaOpera)`
 - 3.3. L'attore aggiorna i dati dell'opera.
 - 3.4. Il sistema provvede a salvare le modifiche.
4. **else if** l'attore primario vuole eliminare un'opera:
 - 4.1. L'attore primario inserisce i dati dell'opera da eliminare.
 - 4.2. `include(RicercaOpera)`
 - 4.3. Il sistema mostra un messaggio di conferma.
 - 4.4. **if** l'attore primario conferma l'operazione:
 - 4.4.1. Il sistema provvede ad eliminare l'opera.

Sequenza eventi alternativa: *OperaGiàEsistente*

1. La sequenza di eventi alternativa inizia dopo il passo 2.2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando che l'opera è già presente nel database.

Sequenza eventi alternativa: *OperaNonTrovata*

1. La sequenza di eventi alternativa inizia dopo il passo 3.2. **or** 4.2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando l'attore primario che non è stata trovata alcuna opera conforme ai parametri di ricerca inseriti.

RicercaOpera (CU 22)

Questo caso d'uso permette di ricercare opere all'interno del database delle opere.

Attori primari: Amministrazione, Operatore al pubblico.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando uno degli attori primari vuole effettuare la ricerca di un'opera **or** il sistema deve ricercare un'opera.
2. **if** il caso d'uso è stato avviato da uno degli attori primari:
 - 2.1. Il sistema chiede i criteri di ricerca.
 - 2.2. L'attore inserisce i criteri di ricerca richiesti.
3. Il sistema preleva i parametri di ricerca.
4. **for each** opera nel database:
 - 4.1. Il sistema confronta i parametri di ricerca con i parametri dell'opera.
 - 4.2. **if** l'opera rispetta i parametri richiesti:
 - 4.2.1. Il sistema aggiunge l'opera alla lista del risultato di ricerca.
5. **if** il sistema ha trovato almeno un'opera corrispondente ai parametri di ricerca:
 - 5.1. Il sistema fornisce tutti i dati relativi alle opere trovate.
6. **else**
 - 6.1. Il sistema informa di non aver trovato nessuna opera corrispondente ai parametri ricercati.

Sequenza eventi alternativa: Nessuna.

GeneraReportIncassi (CU 23)

Questo caso d'uso consente di generare il report degli incassi su tutte le transazioni effettuate da inizio mese fino al momento della richiesta oppure in mesi precedenti.

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario desidera generare il report degli incassi.
2. Il sistema chiede all'attore primario il mese nel quale desidera generare il report degli incassi.
3. **if** l'attore primario inserisce un mese antecedente il mese corrente:
 - 3.1. Il sistema utilizza il mese inserito come criterio di ricerca delle transazioni.
4. **else if** l'attore primario non inserisce alcuna data:
 - 4.1. Il sistema utilizza il mese corrente come criterio di ricerca delle transazioni.
5. **for each** biglietto nel database dei biglietti:
 - 5.1. `include(RicercaBiglietto)`
 - 5.2. **if** il biglietto è stato venduto nel mese selezionato:
 - 5.2.1. Il sistema aggiunge la transazione alla lista delle transizioni.
6. **for each** abbonamento nel database degli abbonamenti:
 - 6.1. `include(RicercaAbbonamento)`
 - 6.2. **if** l'abbonamento è stato venduto **or** è stato rinnovato nel mese selezionato :
 - 6.2.1. Il sistema aggiunge la transazione alla lista delle transizioni.
7. **for each** opera nel database delle opere:
 - 7.1. `include(RicercaOpera)`
 - 7.2. **if** l'opera è stata venduta **or** l'opera è stata acquistata nel mese selezionato :
 - 7.2.1. Il sistema aggiunge la transazione alla lista delle transizioni.
8. Il sistema emette il report con tutte le transazioni recuperate e calcola il totale degli incassi come differenza tra le entrate e le uscite.

Sequenza eventi alternativa: *Report Vuoto*

1. La sequenza di eventi alternativa inizia dopo il passo 8. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando l'attore primario dell'assenza di transazione nel mese selezionato.

GestisciDipendenti (CU 36)

Questo caso d'uso consente di gestire i dipendenti che lavorano all'interno del museo.

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni:

1. Il sistema ha modificato tutte le modifiche effettuate.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario desidera gestire i dipendenti del museo.
2. Il sistema chiede all'attore primario il reparto da gestire.
3. **if** l'attore primario vuole gestire la reception:
 - 3.1. Il sistema mostra all'attore primario una lista degli operatori al pubblico dipendenti del museo.
 - 3.2. L'attore apporta le modifiche necessarie al personale degli operatori al pubblico.
4. **else if** l'attore primario vuole gestire la segreteria:
 - 4.1. Il sistema mostra all'attore primario una lista dei segretari dipendenti del museo.
 - 4.2. L'attore apporta le modifiche necessarie al personale dei segretari.
5. **else if** l'attore primario vuole gestire l'amministrazione:
 - 5.1. Il sistema mostra all'attore primario una lista degli amministratori dipendenti del museo.
 - 5.2. L'attore apporta le modifiche necessarie al personale degli amministratori.
6. Il sistema memorizza tutte le modifiche effettuate al personale.

6 Gestione Sistema

Il diagramma dei casi d'uso **Gestione Sistema** racchiude i casi d'uso istanziabili direttamente solo dal sistema stesso, che permettono di eseguire azioni di **scheduling**, quali effettuare backup locale e su cloud e avvisare i possessori di abbonamenti prossimi alla scadenza. Nel dettaglio, i casi d'uso di pertinenza sono:

- ControllaAbbonamenti (CU 24)
- InviaAvviso (CU 25)
- VerificaAbbonamento (CU 26)
- EffettuaBackup (CU 27)

Gli attori coinvolti in questi casi d'uso sono l'attore *Tempo*, capace di eseguire lo **scheduling** delle azioni e l'attore *Amministrazione* che ha la facoltà di eseguire un backup anche fuori dall'orario prestabilito dallo scheduler.

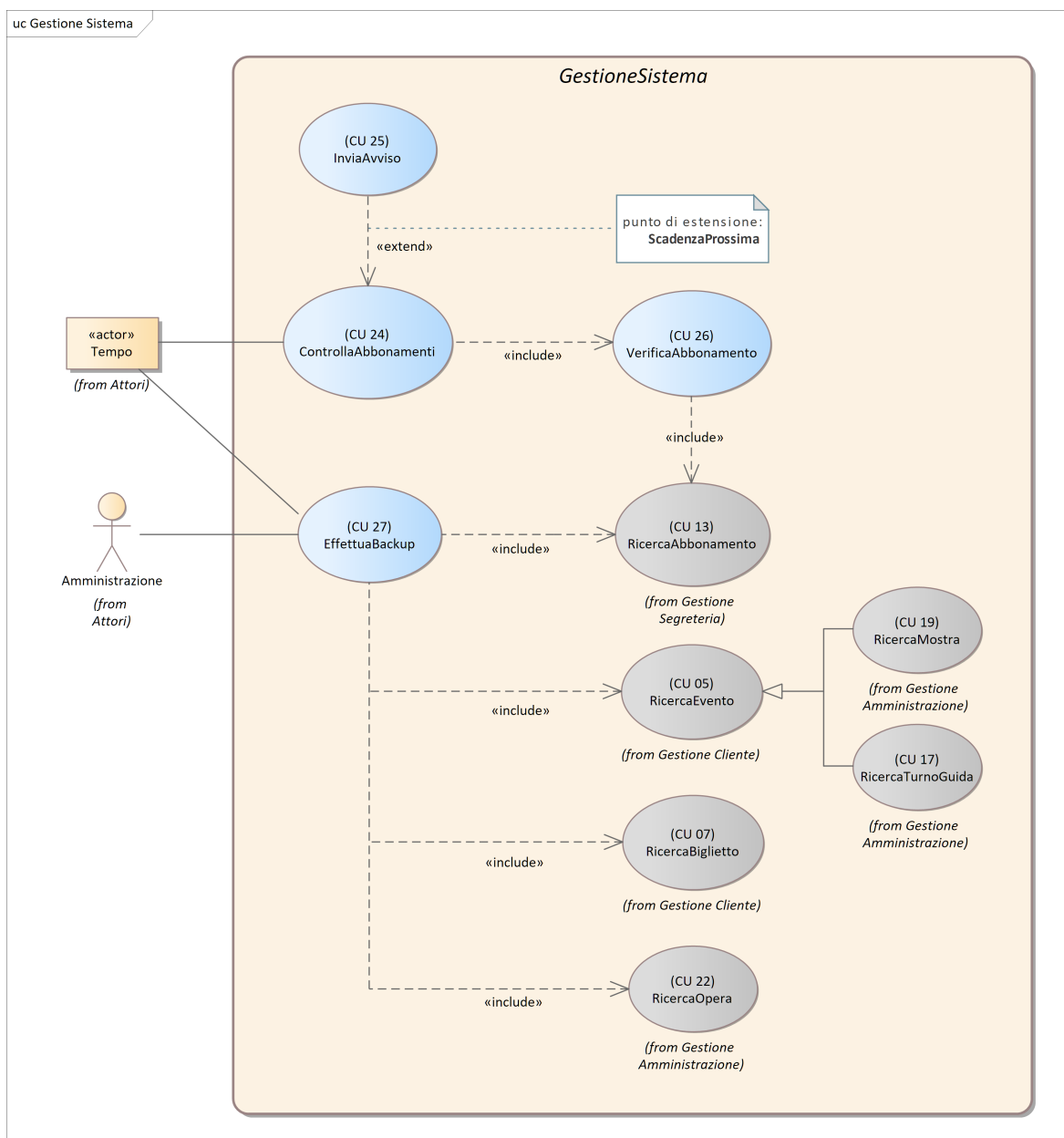


Figura 6: diagramma dei casi d'uso: **Gestione Sistema**

ControllaAbbonamenti (CU 24)

Questo caso d'uso consente di controllare gli abbonamenti attivi e segnalare con un avviso i possessori di abbonamenti con scadenza imminente.

Attori primari: Tempo.

Attori secondari: Nessuno.

Precondizioni: Nessuna.

Postcondizioni:

1. Gli abbonamenti sono stati controllati.

Sequenza eventi principale:

1. Il caso d'uso inizia ogni giorno alle ore 23:59 secondo il fuso orario locale.
2. **for each** abbonamento nel database degli abbonamenti:
 - 2.1. `include(VerificaAbbonamento)`
 - 2.2. `extention point: scadenzaProssima`
 - 2.3. Il sistema aggiorna la data di ultimo controllo dell'abbonamento con la data attuale.

Sequenza eventi alternativa: Nessuna.

InviaAvviso (CU 25)

Segmento 1: Questo caso d'uso permette al sistema di inviare un avviso ai clienti possessori di un abbonamento in fase di scadenza.

Attori primari: Tempo.

Attori secondari: Nessuno.

Precondizioni del segmento 1:

1. La data di scadenza dell'abbonamento antecede la data corrente di meno di 5 giorni.

Postcondizioni del segmento 1:

1. Il sistema ha inviato una comunicazione via posta elettronica al possessore dell'abbonamento.

Sequenza eventi principale del segmento 1:

1. Il caso d'uso inizia quando il sistema deve mandare un messaggio di avviso per rinnovare un abbonamento.
2. Il sistema preleva l'indirizzo di posta elettronica dall'abbonamento.
3. Il sistema genera un messaggio di avviso di rinnovo dell'abbonamento personalizzato e lo invia all'indirizzo di posta elettronica prelevato.

Sequenza eventi alternativa del segmento 1: Nessuna.

VerificaAbbonamento (CU 26)

Questo caso d'uso permette di controllare se un abbonamento è valido, scaduto o in fase di scadenza.

Attori primari: Segreteria, Operatore al Pubblico, Tempo.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha ricevuto l'identificativo dell'abbonamento da controllare.

Postcondizioni:

1. Il sistema ha verificato lo stato dell'abbonamento.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole verificare lo stato di un abbonamento **or** il sistema deve verificare lo stato di un abbonamento.
2. **include(RicercaAbbonamento)**
3. Il sistema legge la data di scadenza dell'abbonamento e la confronta con la data corrente.
4. **if** la data di scadenza dell'abbonamento precede la data corrente:
 - 4.1. Il sistema informa l'attore primario che l'abbonamento è scaduto.
5. **else if** la data di scadenza dell'abbonamento antecede la data corrente di meno di 5 giorni:
 - 5.1. Il sistema informa l'attore primario che l'abbonamento è in fase di scadenza.

Sequenza eventi alternativa: *AbbonamentoNonTrovato.*

1. La sequenza di eventi alternativa inizia dopo il passo 2. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore comunicando all'attore primario di non aver trovato alcun abbonamento avente lo stesso codice identificativo inserito, e lo esorta a ripetere il processo di scannerizzazione del codice QR sul fronte dell'abbonamento.

Sequenza eventi alternativa: *OperazioneAnnullata.*

1. La sequenza di eventi alternativa inizia in qualunque momento.
2. L'attore primario annulla l'operazione verifica dell'abbonamento.

EffettuaBackup (CU 27)

Questo caso d'uso consente di effettuare il backup dei dati relativi al sistema informativo di riferimento.

Attori primari: Amministrazione, Tempo.

Attori secondari: Amministrazione.

Precondizioni:

1. Se il caso d'uso è stato avviato dall'attore primario *Amministrazione*, Il sistema lo ha autenticato.

Postcondizioni: I dati sono stati copiati.

Sequenza eventi principale:

1. Il caso d'uso inizia ogni sera alle 23:59 secondo il fuso orario locale.
2. **for each** biglietto nel database dei biglietti:
 - 2.1. `include(RicercaBiglietto)`.
 - 2.2. Il sistema preleva tutte le informazioni riguardati quel biglietto.
 - 2.3. Il sistema copia sul disco i dati del biglietto.
3. **for each** abbonamento nel database degli abbonamenti:
 - 3.1. `include(RicercaAbbonamento)`.
 - 3.2. Il sistema preleva tutte le informazioni riguardati quell'abbonamento.
 - 3.3. Il sistema copia sul disco i dati dell'abbonamento.
4. **for each** opera nel database delle opere:
 - 4.1. `include(RicercaOpera)`.
 - 4.2. Il sistema preleva tutte le informazioni riguardati quell'opera.
 - 4.3. Il sistema copia sul disco i dati dell'opera.
5. **for each** evento nel database degli eventi:
 - 5.1. `include(RicercaEvento)`.
 - 5.2. Il sistema preleva tutte le informazioni riguardati quell'evento.
 - 5.3. Il sistema copia sul disco i dati dell'evento.
6. **for each** dato sul visitatore nel database dei dati sui visitatori:
 - 6.1. Il sistema preleva tutte le informazioni riguardati quel visitatore.
 - 6.2. Il sistema copia sul disco i dati del visitatore.
7. Il sistema organizza in una cartella adeguata i dati copiati.
8. Il sistema effettua l'upload sul cloud utilizzato dal museo del backup appena effettuato.

Sequenza eventi alternativa: *DiscoPieno*.

1. La sequenza di eventi alternativa inizia dopo il passo 6. della sequenza di eventi principale.
2. Il sistema invia un avviso all'attore secondario comunicante l'impossibilità di effettuare il backup sul disco locale per via della mancanza di spazio sufficiente, e lo esorta a risolvere il problema.

Sequenza eventi alternativa: *UploadNonRiuscito*.

1. La sequenza di eventi alternativa inizia dopo il passo 7. della sequenza di eventi principale.
2. Il sistema notifica all'attore secondario l'impossibilità di effettuare l'upload del backup sul cloud e visualizza l'errore.

7 Gestione Statistiche

Il diagramma dei casi d'uso **Gestione Statistiche** racchiude i casi d'uso necessari per la raccolta dei dati sui clienti in forma anonima e la generazione delle statistiche in base all provenienza, all'età e al sesso del visitatore. Nel dettaglio, i casi d'uso di pertinenza sono:

- InserisciDatiCliente (CU 28)
- InserisciProvenienza (CU 29)
- InserisciEtà (CU 30)
- InserisciSesso (CU 31)
- VisualizzaStatistiche (CU 32)
- VisualizzaPerProvenienza (CU 33)
- VisualizzaPerEtà (CU 34)
- VisualizzaPerSesso (CU 35)

Gli attori coinvolti in questi casi d'uso sono l' *Operatore al pubblico* per la popolazione dei dati sui quali inferire le statistiche e l' *Amministrazione* per la visualizzazione delle stesse.

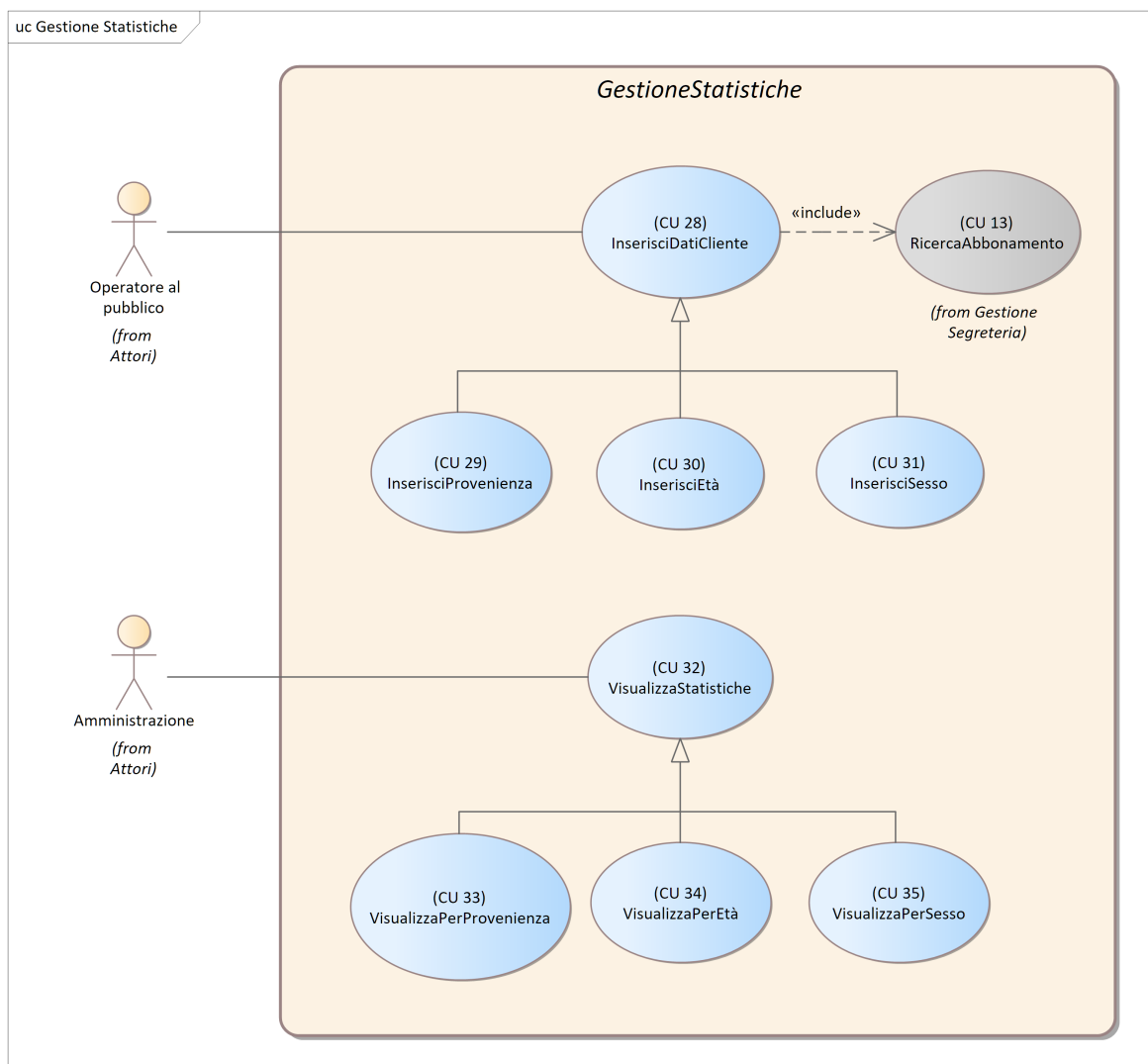


Figura 7: diagramma dei casi d'uso: **Gestione Statistiche**

InserisciDatiCliente (CU 28)

Questo caso d'uso consente di inserire i dati del cliente al momento dell'acquisto del biglietto.

Attori primari: Operatore al pubblico.

Attori secondari: Cliente.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.
2. Nel caso in cui il cliente disponga di un abbonamento, l'attore primario ha inserito nel sistema l'identificativo dell'abbonamento eseguendo una scannerizzazione del codice QR.

Postcondizioni:

1. Il sistema ha memorizzato i dati inseriti in forma anonima.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole memorizzare i dati di un cliente.
2. **if** il cliente dispone di un abbonamento:
 - 2.1. Il sistema recupera i dati necessari dal suo abbonamento.
 - 2.2. `include(RicercaAbbonamento)`
3. **else:**
 - 3.1. L'attore primario chiede al cliente i dati necessari.
 - 3.2. L'attore primario inserisce i dati nel sistema.
4. Il sistema memorizza i dati.
5. Il sistema produce un messaggio con l'esito dell'operazione.

Sequenza eventi alternativa: *DatiNonFornitiDalCliente*

1. La sequenza di eventi alternativa inizia dopo il passo 3.1. della sequenza di eventi principale.
2. L'attore secondario si rifiuta di fornire le informazioni.
3. L'attore primario annulla l'operazione.
4. Il sistema termina l'operazione senza memorizzare alcun dato riguardante il cliente.

InserisciProvenienza (CU 29)

Questo caso d'uso consente di inserire lo Stato di nascita del cliente.

Id padre: (CU 28)

Attori primari: Operatore al pubblico.

Attori secondari: Cliente.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.
2. Nel caso in cui il cliente disponga di un abbonamento, l'attore primario ha inserito nel sistema l'identificativo dell'abbonamento eseguendo una scannerizzazione del codice QR.

Postcondizioni:

1. Il sistema ha memorizzato i dati inseriti in forma anonima.

Sequenza eventi principale:

1. (o1) Il caso d'uso inizia quando l'attore primario vuole memorizzare il Paese d'origine del cliente.
2. (2.) **if** il cliente dispone di un abbonamento:
 - 2.1. (2.1.) Il sistema recupera i dati necessari dal suo abbonamento.
 - 2.2. (2.2.) `include(RicercaAbbonamento)`
3. (3.) **else**:
 - 3.1. (o3.1) L'attore primario chiede al cliente il suo Paese d'origine.
 - 3.2. (o3.2.) L'attore primario inserisce il Paese d'origine del cliente nel sistema.
4. (4.) Il sistema memorizza i dati.
5. (5.) Il sistema produce un messaggio con l'esito dell'operazione.

Sequenza eventi alternativa: *DatiNonFornitiDalCliente*

1. (1.) La sequenza di eventi alternativa inizia dopo il passo 3.1. della sequenza di eventi principale.
2. (o2.) L'attore secondario si rifiuta di fornire l'informazione del suo Paese di origine.
3. (3.) L'attore primario annulla l'operazione.
4. (4.) Il sistema termina l'operazione senza memorizzare alcun dato riguardante il cliente.

InserisciEtà (CU 30)

Questo caso d'uso consente di inserire l'età del cliente.

Id padre: (CU 28)

Attori primari: Operatore al pubblico.

Attori secondari: Cliente.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.
2. Nel caso in cui il cliente disponga di un abbonamento, l'attore primario ha inserito nel sistema l'identificativo dell'abbonamento eseguendo una scannerizzazione del codice QR.

Postcondizioni:

1. Il sistema ha memorizzato i dati inseriti in forma anonima.

Sequenza eventi principale:

1. (o1) Il caso d'uso inizia quando l'attore primario vuole memorizzare la data di nascita del cliente.
2. (2.) **if** il cliente dispone di un abbonamento:
 - 2.1. (2.1.) Il sistema recupera i dati necessari dal suo abbonamento.
 - 2.2. (2.2.) `include(RicercaAbbonamento)`
3. (3.) **else**:
 - 3.1. (o3.1) L'attore primario chiede al cliente la sua data di nascita.
 - 3.2. (o3.2.) L'attore primario inserisce la data di nascita del cliente nel sistema.
4. (4.) Il sistema memorizza i dati.
5. (5.) Il sistema produce un messaggio con l'esito dell'operazione.

Sequenza eventi alternativa: *DatiNonFornitiDalCliente*

1. (1.) La sequenza di eventi alternativa inizia dopo il passo 3.1. della sequenza di eventi principale.
2. (o2.) L'attore secondario si rifiuta di fornire l'informazione della sua data di nascita.
3. (3.) L'attore primario annulla l'operazione.
4. (4.) Il sistema termina l'operazione senza memorizzare alcun dato riguardante il cliente.

InserisciSesso (CU 31)

Questo caso d'uso consente di inserire il genere del cliente.

Id padre: (CU 28)

Attori primari: Operatore al pubblico.

Attori secondari: Cliente.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.
2. Nel caso in cui il cliente disponga di un abbonamento, l'attore primario ha inserito nel sistema l'identificativo dell'abbonamento eseguendo una scannerizzazione del codice QR.

Postcondizioni:

1. Il sistema ha memorizzato i dati inseriti in forma anonima.

Sequenza eventi principale:

1. (o1) Il caso d'uso inizia quando l'attore primario vuole memorizzare il genere del cliente.
2. (2.) **if** il cliente dispone di un abbonamento:
 - 2.1. (2.1.) Il sistema recupera i dati necessari dal suo abbonamento.
 - 2.2. (2.2.) `include(RicercaAbbonamento)`
3. (3.) **else**:
 - 3.1. (o3.1) L'attore primario chiede al cliente il suo genere.
 - 3.2. (o3.2.) L'attore primario inserisce il genere del cliente nel sistema.
4. (4.) Il sistema memorizza i dati.
5. (5.) Il sistema produce un messaggio con l'esito dell'operazione.

Sequenza eventi alternativa: *DatiNonFornitiDalCliente*

1. (1.) La sequenza di eventi alternativa inizia dopo il passo 3.1. della sequenza di eventi principale.
2. (o2.) L'attore secondario si rifiuta di fornire l'informazione del suo genere.
3. (3.) L'attore primario annulla l'operazione.
4. (4.) Il sistema termina l'operazione senza memorizzare alcun dato riguardante il cliente.

VisualizzaStatistiche (CU 32)

Questo caso d'uso consente di generare statistiche in un particolare mese sulla base dei dati sui clienti raccolti in forma anonima dal sistema.

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole visualizzare le statistiche.
2. Il sistema chiede all'attore primario il mese nel quale desidera generare le statistiche.
3. **if** l'attore primario inserisce un mese antecedente il mese corrente:
 - 3.1. Il sistema utilizza il mese inserito come criterio di ricerca.
4. **else if** l'attore primario non inserisce alcuna data:
 - 4.1. Il sistema utilizza il mese corrente come criterio di ricerca.
5. **for each** dato sul cliente nel database dei dati sui clienti:
 - 5.1. **if** il dato sul cliente è stato registrato nel sistema nel mese selezionato:
 - 5.1.1. Il sistema aggiunge il dato sul cliente alla lista dei dati raccolti.
6. Il sistema organizza i dati sulla base dei quali genera un grafico e lo mostra a schermo.
7. Il sistema calcola deviazione standard campionaria sulla base dei dati raccolti e mostra a schermo i risultati.
8. Il sistema mostra a schermo anche il numero di visite avvenute dall'inizio del mese, fino al momento della richiesta.

Sequenza eventi alternativa: *NessunDato*

1. La sequenza di eventi alternativa inizia dopo il passo 6. della sequenza di eventi principale.
2. Il sistema mostra un opportuno messaggio di errore informando l'attore primario dell'assenza di dati sui clienti nel mese selezionato.

VisualizzaPerProvenienza (CU 33)

Questo caso d'uso consente di eseguire la ricerca di un abbonamento all'interno del database.

Id padre: (CU 32)

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. (1.) Il caso d'uso inizia quando l'attore primario vuole visualizzare le statistiche.
2. (2.) Il sistema chiede all'attore primario il mese nel quale desidera generare le statistiche.
3. (3.) **if** l'attore primario inserisce un mese antecedente il mese corrente:
 - 3.1. (3.1.) Il sistema utilizza il mese inserito come criterio di ricerca.
4. (4.) **else if** l'attore primario non inserisce alcuna data:
 - 4.1. (4.1.) Il sistema utilizza il mese corrente come criterio di ricerca.
5. (5.) **for each** dato sul cliente nel database dei dati sui clienti:
 - 5.1. (5.1.) **if** il dato sul cliente è stato registrato nel sistema nel mese selezionato:
 - 5.1.1. (5.1.1.) Il sistema aggiunge il Paese di origine del cliente alla lista dei dati raccolti.
6. (6.) Il sistema organizza i dati sulla base dei quali genera un grafico a torta e lo mostra a schermo.
7. (7.) Il sistema calcola la deviazione standard campionaria sulla base dei dati raccolti e mostra a schermo i risultati.
8. (8.) Il sistema mostra a schermo anche il numero di visite avvenute dall'inizio del mese, fino al momento della richiesta.

Sequenza eventi alternativa: *NessunDato*

1. (1.) La sequenza di eventi alternativa inizia dopo il passo 6. della sequenza di eventi principale.
2. (2.) Il sistema mostra un opportuno messaggio di errore informando l'attore primario dell'assenza di dati sui clienti nel mese selezionato.

VisualizzaPerEtà (CU 34)

Questo caso d'uso consente di eseguire la ricerca di un abbonamento all'interno del database.

Id padre: (CU 32)

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. (1.) Il caso d'uso inizia quando l'attore primario vuole visualizzare le statistiche.
2. (2.) Il sistema chiede all'attore primario il mese nel quale desidera generare le statistiche.
3. (3.) **if** l'attore primario inserisce un mese antecedente il mese corrente:
 - 3.1. (3.1.) Il sistema utilizza il mese inserito come criterio di ricerca.
4. (4.) **else if** l'attore primario non inserisce alcuna data:
 - 4.1. (4.1.) Il sistema utilizza il mese corrente come criterio di ricerca.
5. (5.) **for each** dato sul cliente nel database dei dati sui clienti:
 - 5.1. (5.1.) **if** il dato sul cliente è stato registrato nel sistema nel mese selezionato:
 - 5.1.1. (5.1.1.) Il sistema raccoglie la data di nascita del cliente e la utilizza per calcolare l'età.
 - 5.1.2. (5.1.2.) Il sistema aggiunge l'età del cliente alla lista dei dati raccolti.
6. (6.) Il sistema organizza i dati sulla base dei quali genera un grafico istogramma e lo mostra a schermo.
7. (7.) Il sistema calcola la media campionaria e la deviazione standard campionaria sulla base dei dati raccolti e mostra a schermo i risultati.
8. (8.) Il sistema mostra a schermo anche il numero di visite avvenute dall'inizio del mese, fino al momento della richiesta.

Sequenza eventi alternativa: *NessunDato*

1. (1.) La sequenza di eventi alternativa inizia dopo il passo 6. della sequenza di eventi principale.
2. (2.) Il sistema mostra un opportuno messaggio di errore informando l'attore primario dell'assenza di dati sui clienti nel mese selezionato.

VisualizzaPerSesso (CU 35)

Questo caso d'uso consente di eseguire la ricerca di un abbonamento all'interno del database.

Id padre: (CU 32)

Attori primari: Amministrazione.

Attori secondari: Nessuno.

Precondizioni:

1. Il sistema ha autenticato l'attore primario.

Postcondizioni: Nessuna.

Sequenza eventi principale:

1. (1.) Il caso d'uso inizia quando l'attore primario vuole visualizzare le statistiche.
2. (2.) Il sistema chiede all'attore primario il mese nel quale desidera generare le statistiche.
3. (3.) **if** l'attore primario inserisce un mese antecedente il mese corrente:
 - 3.1. (3.1.) Il sistema utilizza il mese inserito come criterio di ricerca.
4. (4.) **else if** l'attore primario non inserisce alcuna data:
 - 4.1. (4.1.) Il sistema utilizza il mese corrente come criterio di ricerca.
5. (5.) **for each** dato sul cliente nel database dei dati sui clienti:
 - 5.1. (5.1.) **if** il dato sul cliente è stato registrato nel sistema nel mese selezionato:
 - 5.1.1. (5.1.1.) Il sistema aggiunge il genere del cliente alla lista dei dati raccolti.
6. (6.) Il sistema organizza i dati sulla base dei quali genera un grafico a torta e lo mostra a schermo.
7. (7.) Il sistema calcola la deviazione standard campionaria sulla base dei dati raccolti e mostra a schermo i risultati.
8. (8.) Il sistema mostra a schermo anche il numero di visite avvenute dall'inizio del mese, fino al momento della richiesta.

Sequenza eventi alternativa: *NessunDato*

1. (1.) La sequenza di eventi alternativa inizia dopo il passo 6. della sequenza di eventi principale.
2. (2.) Il sistema mostra un opportuno messaggio di errore informando l'attore primario dell'assenza di dati sui clienti nel mese selezionato.