

Università degli Studi di Napoli Federico II

**Corso di Laurea in Ingegneria Informatica**

**Corso di Ingegneria del Software**

**Prof. R. Pietrantuono- A.A. 2024 – 25**

***Progetto***

Società di Trasporto Regionale

Studente:

Gaetano Emanuele Izzo N46007081 [gaetan.izzo@studenti.unina.it](mailto:gaetan.izzo@studenti.unina.it)

Eugenio Iandoli N46007067

eu.iandoli@studenti.unina.it

Aniello Gaudino N46007219

aniel.gaudino@studenti.unina.it

**Indice**

**1. Specifiche informali 1**

**2. Analisi e specifica dei requisiti 2**

2.1 Revisione dei requisiti 2

2.2 Glossario dei termini 2

2.3 Classificazione dei requisiti 3

2.3.1 Requisiti funzionali 3

2.3.2 Vincoli / Altri requisiti 3

2.4 Modellazione dei casi d’uso 4

2.4.1 Attori e casi d’uso 4

2.4.2 Diagramma dei casi d’uso 4

2.4.3 Scenari 4

2.5 Modellazione dei dati 2

2.6 Diagramma delle classi 2

2.7 Diagrammi di sequenza 2

**3. Stima dei costi 3**

**4. Piano di test funzionale 4**

**5. Progettazione 6**

5.1 Diagramma delle classi 6

5.2 Diagrammi di sequenza 6

**6. Implementazione 7**

**7. Testing 8**

7.1 Test funzionale 8

# Specifiche informali

*Si vuole realizzare un’applicazione per una società di trasporto su autobus. La compagnia fornisce servizi di trasporto passeggeri operati da autobus su tratte regionali. La compagnia opera un insieme di corse giornaliere tra N località. Ogni autobus ha una capienza in termini di numero di posti disponibili per i passeggeri e svolge servizio giornalmente sempre sulla stessa tratta. Le biglietterie, site nelle località servite dalla compagnia, possono emettere titoli di viaggio, validi per una specifica corsa. Ogni corsa è identificata da un codice e ha orario e città di partenza e di arrivo, e un prezzo del biglietto prefissato. Prima di emettere un biglietto, il sistema deve valutare l’effettiva disponibilità del posto. All’atto dell’emissione dovrà essere aggiornata la disponibilità residua. L’applicazione supporta l’emissione di biglietti da parte degli impiegati nelle biglietterie. Ogni impiegato deve accedere al sistema solo previa autenticazione. Per ogni biglietto il sistema deve stampare la data e l’ora di emissione. Per ciascun biglietto il sistema deve registrare internamente, al momento dell’emissione, data e ora di emissione e codice dell’impiegato che lo ha emesso. È altresì previsto l’uso dell’applicazione da parte di clienti, che possono acquistare via web un biglietto passeggero, o anche modificare un biglietto emesso: la modifica riguarda il cambio corsa (giorno e/o ora), ma non delle località di partenza e arrivo. Quando acquista un biglietto, il cliente riceve una mail con in allegato il biglietto elettronico (in formato pdf), contenente un codice QR che il conducente dell’autobus deve leggere, con il dispositivo mobile di cui viene dotato, quando il passeggero sale sull’autobus. Il cliente può anche richiedere l’invio del codice QR sul proprio cellulare. Settimanalmente (di norma, il lunedì alle 8), il sistema produce automaticamente ed invia per email al direttore della compagnia un report dei biglietti venduti per ogni giorno sulle varie tratte*.

# Analisi e specifica dei requisiti

**Analisi nomi-verbi**

Si vuole realizzare un’applicazione per una società di trasporto su autobus. La compagnia fornisce servizi di trasporto passeggeri operati da autobus su tratte regionali. La compagnia opera un insieme di corse giornaliere tra N località. Ogni autobus ha una capienza in termini di numero di posti disponibili per i passeggeri e svolge servizio giornalmente sempre sulla stessa tratta. Le biglietterie, site nelle località servite dalla compagnia, possono emettere titoli di viaggio, validi per una specifica corsa. Ogni corsa è identificata da un codice e ha orario e città di partenza e di arrivo, e un prezzo del biglietto prefissato. Prima di emettere un biglietto, il sistema deve valutare l’effettiva disponibilità del posto. All’atto dell’emissione dovrà essere aggiornata la disponibilità residua. L’applicazione supporta l’emissione di biglietti da parte degli impiegati nelle biglietterie. Ogni impiegato deve accedere al sistema solo previa autenticazione. Per ogni biglietto il sistema deve stampare la data e l’ora di emissione. Per ciascun biglietto il sistema deve registrare internamente, al momento dell’emissione, data e ora di emissione e codice dell’impiegato che lo ha emesso. È altresì previsto l’uso dell’applicazione da parte di clienti, che possono acquistare via web un biglietto passeggero, o anche modificare un biglietto emesso: la modifica riguarda il cambio corsa (giorno e/o ora), ma non delle località di partenza e arrivo. Quando acquista un biglietto, il cliente riceve una mail con in allegato il biglietto elettronico (in formato pdf), contenente un codice QR che il conducente dell’autobus deve leggere, con il dispositivo mobile di cui viene dotato, quando il passeggero sale sull’autobus. Il cliente può anche richiedere l’invio del codice QR sul proprio cellulare. Settimanalmente (di norma, il lunedì alle 8), il sistema produce automaticamente ed invia per email al direttore della compagnia un report dei biglietti venduti per ogni giorno sulle varie tratte.

*LEGENDA:  
Classe  
Attributo  
Funzionalità  
Attore*

*Classe-Attore*

## Revisione dei requisiti

1. *Il sevizio di trasporto ha a dispozione N autobus.*
2. *Ogni autobus ha una propria capienza e una singola tratta associata.*
3. *La compagnia opera con un insieme di corse su N località.*
4. *Ogni corsa è identificata da: codice, orario di partenza,orario di arrivo,città di partenza,città di arrivo e prezzo del biglietto.*
5. *Ogni biglietto ha una data, un orario di emissione e il codice identificativo dell'impiegato lo ha emesso.*
6. *Il sistema prima di consentire l'emissione di un biglietto deve valutare la disponibilità del posto.*
7. *Il sistema deve, all'atto dell'emissione di un biglietto, aggiornare la disponibilità dei posti residui.*
8. *Il sistema deve consentire l’accesso agli impiegati solo previa autenticazione.*
9. *Di ogni impiegato si vuole memorizzare il codice impiegato associato.*
10. *Il sistema deve permettere agli impiegati, previa autenticazione, di emettere biglietti per una corsa.*
11. *Il sistema per ciascun biglietto deve stampare data e orario di emissione.*
12. *Il sistema deve registrare internamente, per ogni biglietto, la data, l'orario d'emissione e il codice dell'impiegato che lo ha emesso.*
13. *Il sistema deve permettere al cliente di acquistare un biglietto via web.*
14. *Del cliente si vogliono memorizzare indirizzo email ed opzionalmente il cellulare.*
15. *Il sistema deve consentire ai clienti di modificare il giorno e/o l'orario della corsa di un biglietto già acquistato, mantenendo inalterate le località di partenza e arrivo.*
16. *Il sistema in seguito all'acquisto di un biglietto elettronico, deve inviare al cliente una mail con in allegato il biglietto ed un codice QR.*
17. *Il sistema deve consentire ai clienti di richiedere l’invio del codice QR direttamente sul proprio cellulare.*
18. *Il sistema deve generare automaticamente, ogni lunedì alle 8:00, un report dei biglietti venduti per ciascun giorno e per ogni tratta.*
19. *Il sistema deve inviare via e-mail il report settimanale al direttore della compagnia.*

## Glossario dei termini

| Termine | Descrizione | Sinonimi |
| --- | --- | --- |
| Capienza | Il numero di posti disponibili in un autobus per i passeggeri |  |
| Tratta | Percorso che un autobus svolge giornalmente fra due località specifiche con fermate intermedie. | Percorso |
| Corsa | Viaggio specifico effettuato da un autobus in una determinata giornata su una determinata tratta. |  |
| Biglietto | Documento che autorizza un passeggero a viaggiare su una determinata corsa, emesso previa verifica. | Titolo di viaggio |
| Biglietteria | Punto fisico o online dove i biglietti vengono emessi o acquistati dai passeggeri. |  |
| Autenticazione | Procedura di verifica dell'identità di un impiegato per l'accesso al sistema. |  |
| QRCode | Codice a barre bidimensionale che viene utilizzato per il controllo e la validazione del biglietto. | Codice QR |
| Dispositivo Mobile | Dispositivo che consente di leggere il QR code per validare l'accesso del passeggero sull'autobus | Scanner QR |
| Report Settimanale | Documento inviato automaticamente al direttore della compagnia, contenente informazioni sulle vendite dei biglietti. |  |
| Posto | Unità di capacità in un autobus, che rappresenta un posto a sedere per un passeggero. |  |
| Biglietto Elettronico | Biglietto inviato via email con formato PDF contenente un QR code, che può essere stampato o letto tramite un dispositivo mobile. Biglietto fisico rilasciato al cliente, generalmente acquistato presso la biglietteria. |  |
| Email | Messaggio elettronico inviato per comunicare informazioni, come la conferma d’acquisto o i biglietti. |  |

## Classificazione dei requisiti

## Requisiti funzionali

| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| --- | --- | --- |
| RF01 | Il sistema prima di consentire l'emissione di un biglietto deve valutare la disponibilità del posto. | 6 |
| RF02 | Il sistema deve, all'atto dell'emissione di un biglietto, aggiornare la disponibilità dei posti residui. | 7 |
| RF03 | Il sistema deve consentire l’accesso agli impiegati solo previa autenticazione. | 8 |
| RF04 | Il sistema deve permettere agli impiegati, previa autenticazione, di emettere biglietti per una corsa. | 10 |
| RF05 | Il sistema per ciascun biglietto deve stampare data e orario di emissione | 11 |
| RF07 | Il sistema deve registrare internamente, per ogni biglietto emesso, la data, l'ora di emissione e il codice identificativo dell'impiegato che lo ha emesso. | 12 |
| RF08 | Il sistema deve consentire ai clienti di acquistare un biglietto online. | 13 |
| RF09 | Il sistema deve consentire ai clienti di modificare il giorno e/o l'orario della corsa di un biglietto già acquistato, mantenendo inalterate le località di partenza e arrivo. | 15 |
| RF10 | Il sistema, a seguito dell'acquisto di un biglietto elettronico, deve inviare automaticamente una e-mail al cliente contenente il biglietto in formato PDF e il codice QR. | 16 |
| RF11 | Il sistema deve consentire ai clienti di richiedere l’invio del codice QR direttamente sul proprio cellulare. | 17 |
| RF12 | Il sistema deve generare automaticamente, ogni lunedì alle 8:00, un report dei biglietti venduti per ciascun giorno e per ogni tratta. | 18 |
| RF13 | Il sistema deve inviare via e-mail il report settimanale al direttore della compagnia. | 19 |

## Vincoli / Altri requisiti

| ID | Requisito |
| --- | --- |
| V/RF01 | Ogni autobus opera su tratte regionali. |
| V/RF02 | La compagnia opera su un insieme di corse giornaliere tra N località. |
| V/RF03 | Ogni autobus ha una capienza massima (numero di posti disponibili). |
| V/RF04 | Ogni autobus svolge servizio giornaliero sempre sulla stessa tratta. |
| V/RF05 | Ogni corsa è identificata da un codice e ha orario e città di partenza e di arrivo, e un prezzo del biglietto prefissato. |
| V/RF06 | Non può essere erogato un biglietto qualora sia stata raggiunta la massima capienza. |
| V/RF07 | Ogni biglietto ha una data, un orario di emissione e il codice identificativo dell'impiegato lo ha emesso. |
| V/RF08 | Gli impiegati solo previa autenticazione possono emettere biglietti per una corsa. |
| V/RF09 | Di ogni biglietto il cliente può modificare il giorno e/o l'orario della corsa di un biglietto già acquistato, mantenendo inalterate le località di partenza e arrivo. |
| V/RF10 | Il cliente al momento dell’acquisto via web, deve ricevere il biglietto in formato pdf. |
| V/RF11 | Ogni lunedì alle 8:00 il direttore deve ricevere via mail un report (dei biglietti venduti per ogni giorno sulle varie tratte) prodotto in automatico dal sistema. |

## Modellazione dei casi d’uso

## Attori e casi d’uso

***Attori Primari:***

* Cliente
* Impiegato
* Tempo
* Conducente

***Attori Secondari:***

* Stampante
* Servizio email
* Dispositivo mobile

***Casi d’uso:***

* Acquista biglietto
* Vendi biglietto
* Lettura biglietto
* Invia biglietto
* Autenticazione
* Genera report

***Casi d’ uso di inclusione:***

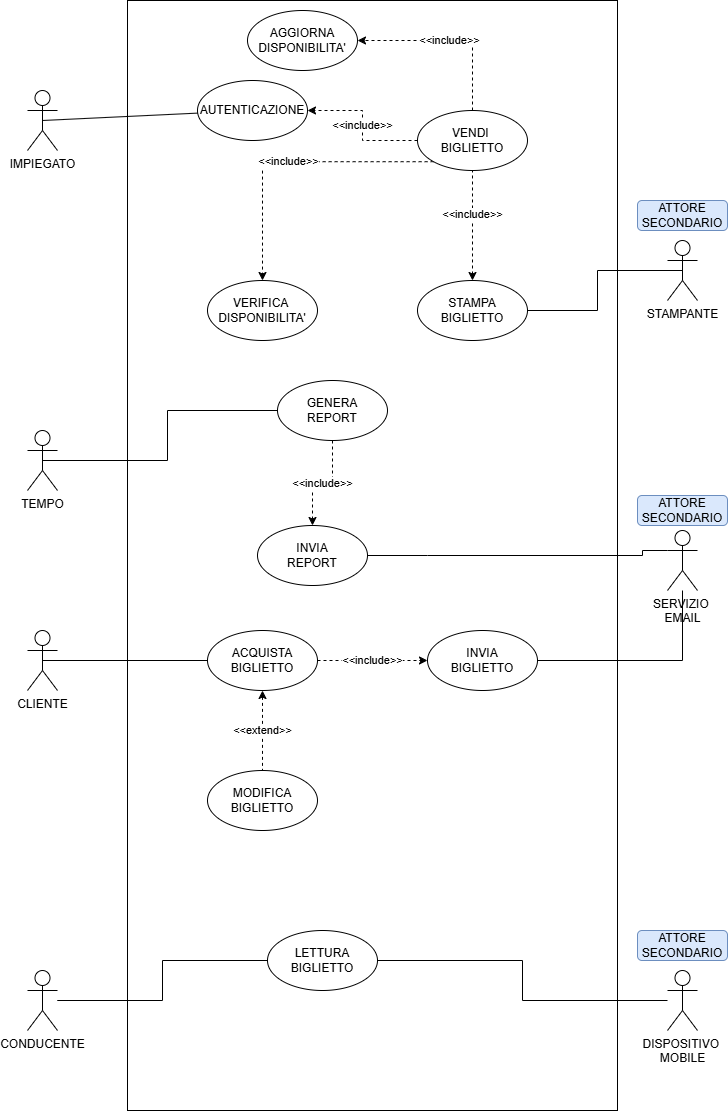
* Invia biglietto
* Stampa biglietto
* Aggiorna disponibilità
* Verifica disponibilità
* Invia report

***Casi d’ uso di estensione:***

* Modifica biglietto

| **Caso d’uso** | **Attori Primari** | **Attori Secondari** | **Incl. / Ext.** | **Requisiti corrispondenti** |
| --- | --- | --- | --- | --- |
| UC1: Acquista biglietto | Cliente | … | Modifica biglietto  Invia biglietto | RF08 |
| UC2: Modifica biglietto | Cliente | … | Estende  Acquista biglietto | RF09 |
| UC 3: Invia biglietto | … | Servizio email | Incluso in  Acquista biglietto | RF10 |
| UC4: Autenticazione | Impiegato | ... |  | RF03 |
| UC5: Vendi biglietto | Impiegato | … | Autenticazione  Stampa biglietto  Verifica disponibilità  Aggiorna disponibilità | RF04 |
| UC6: Stampa biglietto | Impiegato | Stampante | Incluso in  Vendi biglietto | RF05 |
| UC7: Verifica disponibilità | … | … | Incluso in  Vendi biglietto | RF01 |
| UC8: Aggiorna disponibilità | … | … | Incluso in  Vendi biglietto | RF02 |
| UC9: Genera report | Tempo | Servizio email | Include  Invia report | RF12 |
| UC10: Invia report | … | Servizio email | Incluso in  Genera report | RF13 |
| UC11: Lettura biglietto | Conducente | Dispositivo mobile |  |  |

## Diagramma dei casi d’uso



## Scenari

| **Caso d’uso:** | **Acquista biglietto** |
| --- | --- |
| **Attore primario** | Cliente |
| **Attore secondario** | Servizio email |
| **Descrizione** | Il cliente acquista un biglietto elettronico tramite il sito web per una specifica corsa. |
| **Pre-Condizioni** | Ci sono posti disponibili per la corsa richiesta dall'utente. |
| **Sequenza di eventi**  **principale** | * Il caso d’uso inizia quando il Cliente richiede l’acquisto di un biglietto. * Il Cliente seleziona località di partenza e arrivo, giorno, ora della corsa desiderata. * Il sistema controlla che ci siano posti disponibili per quella corsa. * Il sistema mostra i dettagli della corsa e il prezzo del biglietto. * Il Cliente conferma l’acquisto e inserisce i dati richiesti per il pagamento. * Il sistema registra data e ora dell’acquisto, e genera un biglietto contenente il codice QR. * Il sistema aggiorna il numero di posti residui per la corsa. * Il sistema invia il biglietto in formato PDF via email e, se richiesto dal cliente, anche via SMS con il codice QR. |
| **Post-Condizioni** | Il cliente ha acquistato un biglietto per una corsa specifica, che riceve in formato elettronico sulla mail specificata.  Il sistema ha aggiornato la disponibilità dei posti per quella corsa. |
| **Casi d’uso correlati** | Verifica disponibilità, Aggiorna disponibilità, Invia biglietto, Modifica biglietto. |
| **Sequenza di eventi**  **alternativi** | Se la corsa inserita non ha posti disponibili :   * il sistema mostra al Cliente un messaggio di errore. |

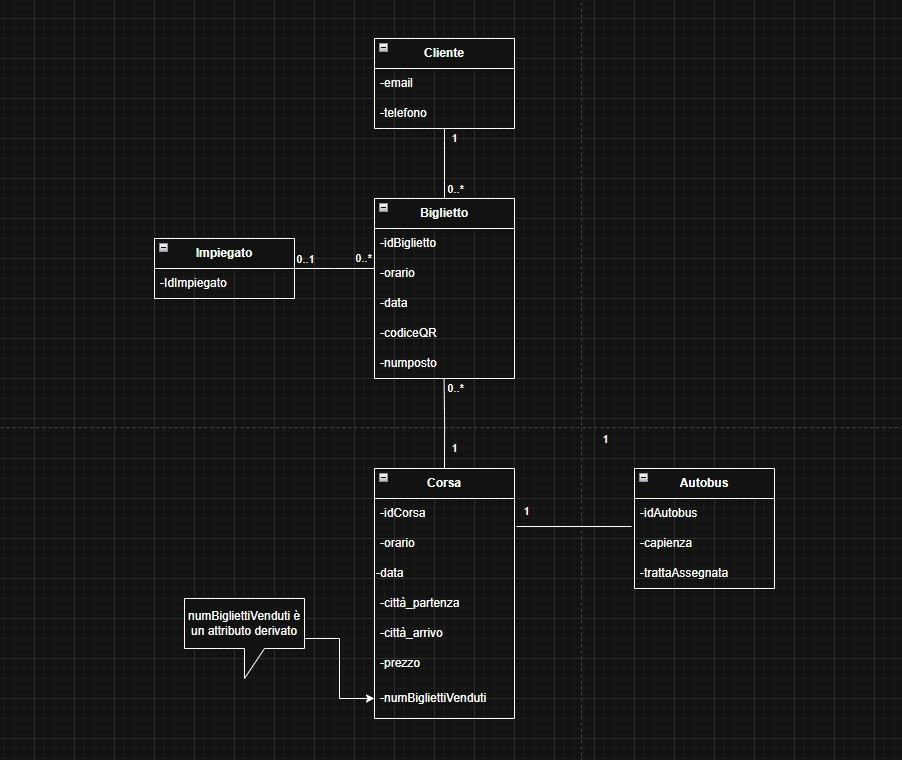
| **Caso d’uso:** | **Vendi biglietto** |
| --- | --- |
| **Attore primario** | Impiegato |
| **Attore secondario** | Stampante |
| **Descrizione** | L’impiegato vende un biglietto tramite biglietteria. |
| **Pre-Condizioni** | L'impiegato ha effettuato l'autenticazione al sistema.  Ci sono posti disponibili per la corsa richiesta dall'utente. |
| **Sequenza di eventi**  **principale** | * Il caso d'uso inizia quando l’impiegato effettua l'autenticazione al sistema * L'impiegato seleziona località di partenza e arrivo, giorno, ora della corsa desiderata. * Il sistema controlla che ci siano posti disponibili per quella corsa. * L’impiegato conferma la vendita. * Il sistema registra per la corsa : data, ora e codice impiegato. * Il sistema aggiorna il numero di posti residui per la corsa. * Il sistema stampa il biglietto cartaceo tramite la stampante. |
| **Post-Condizioni** | Il biglietto viene stampato.  Il sistema ha aggiornato la disponibilità dei posti per quella corsa.  Il sistema traccia la vendita internamente. |
| **Casi d’uso correlati** | Verifica disponibilità, Aggiorna disponibilità, Stampa biglietto, Autenticazione. |
| **Sequenza di eventi**  **alternativi** | Se la corsa inserita non ha posti disponibili :   * il sistema mostra all'impiegato un messaggio di errore.   Se l'impiegato non ha effettuato l'autenticazione correttamente :   * il sistema nega l'accesso alle funzionalità e richiede all'impiegato di effettuare l'autenticazione. |

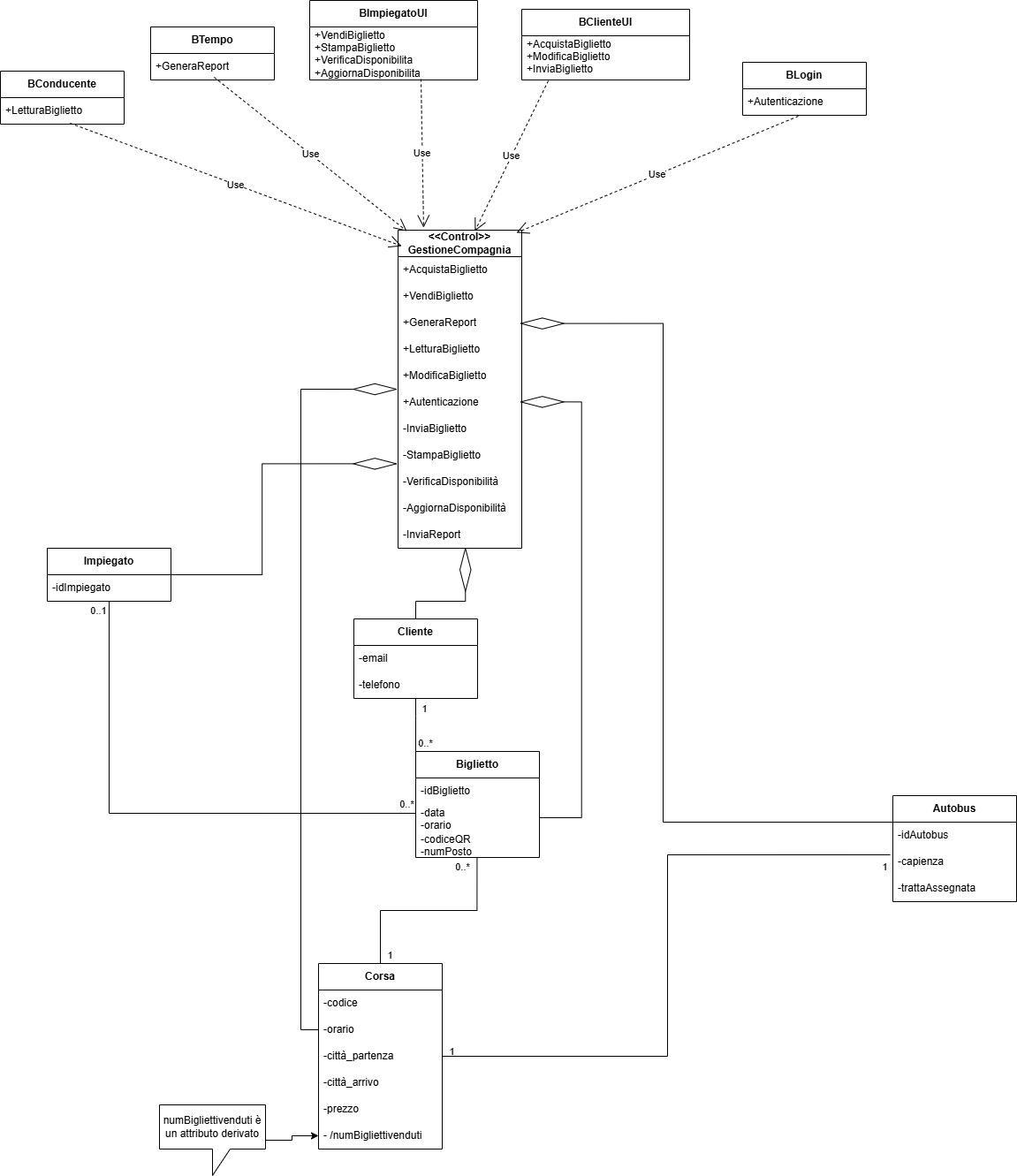
| **Caso d’uso:** | **Genera report** |
| --- | --- |
| **Attore primario** | Tempo |
| **Attore secondario** | Servizio mail |
| **Descrizione** | Ogni lunedì il sistema invia al Direttore un report delle vendite. |
| **Pre-Condizioni** | Il sistema ha registrato dati di vendita |
| **Sequenza di eventi**  **principale** | * Il caso d'uso inizia quando il sistema, ogni lunedì della settimana alle 8:00, attiva il processo di generazione report dei biglietti venduti. * Il sistema aggrega i dati di vendita giornalieri per ogni tratta. * Il sistema compone il file del report. * Il sistema invia il file del report all'email del direttore. |
| **Post-Condizioni** | Il Direttore riceve il report. |
| **Casi d’uso correlati** | Invia report |
| **Sequenza di eventi**  **alternativi** |  |

| **Caso d’uso:** | **Lettura biglietto** |
| --- | --- |
| **Attore primario** | Conducente |
| **Attore secondario** | Dispositivo mobile |
| **Descrizione** | Il conducente verifica il biglietto tramite lettura del codice QR prima che il passeggero salga sull'autobus. |
| **Pre-Condizioni** | Il passeggero possiede un biglietto valido.  Il conduce dispone di un dispositivo mobile in grado di verificare la validità dei biglietti. |
| **Sequenza di eventi**  **principale** | * Il caso d'uso inizia quando il passeggero mostra il codice QR. * Il conducente scansiona il codice QR con il dispositivo mobile. * Il dispositivo mobile verifica la validità del codice. * Se il biglietto è valido, il dispositivo mostra esito positivo. * Il passeggero può salire sull'autobus. |
| **Post-Condizioni** | Accesso consentito al passeggero. |
| **Casi d’uso correlati** | *nessuno* |
| **Sequenza di eventi**  **alternativi** | Se il codice QR del cliente non è valido :   * Il conducente nega al passeggero di salire sull'autobus. |

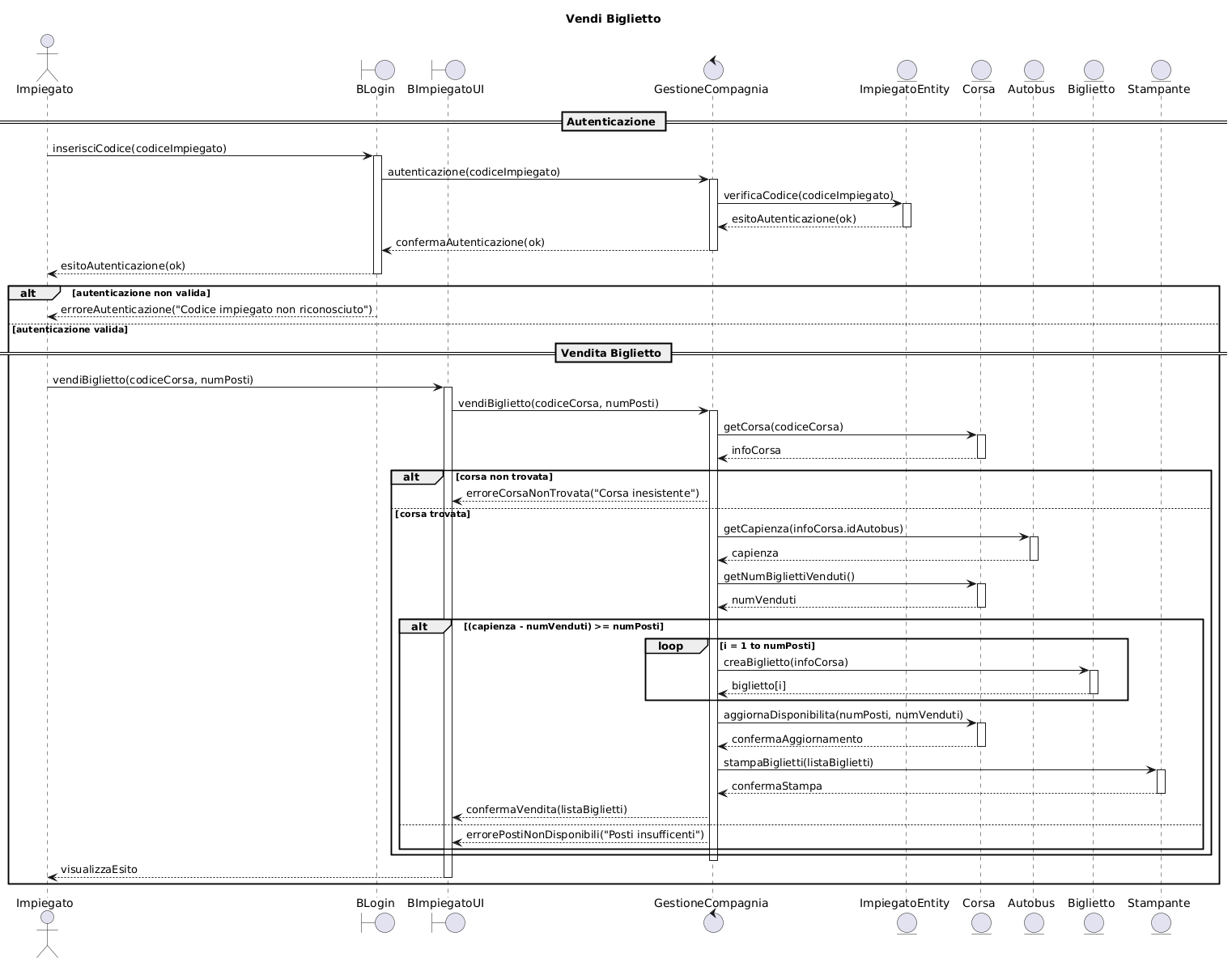
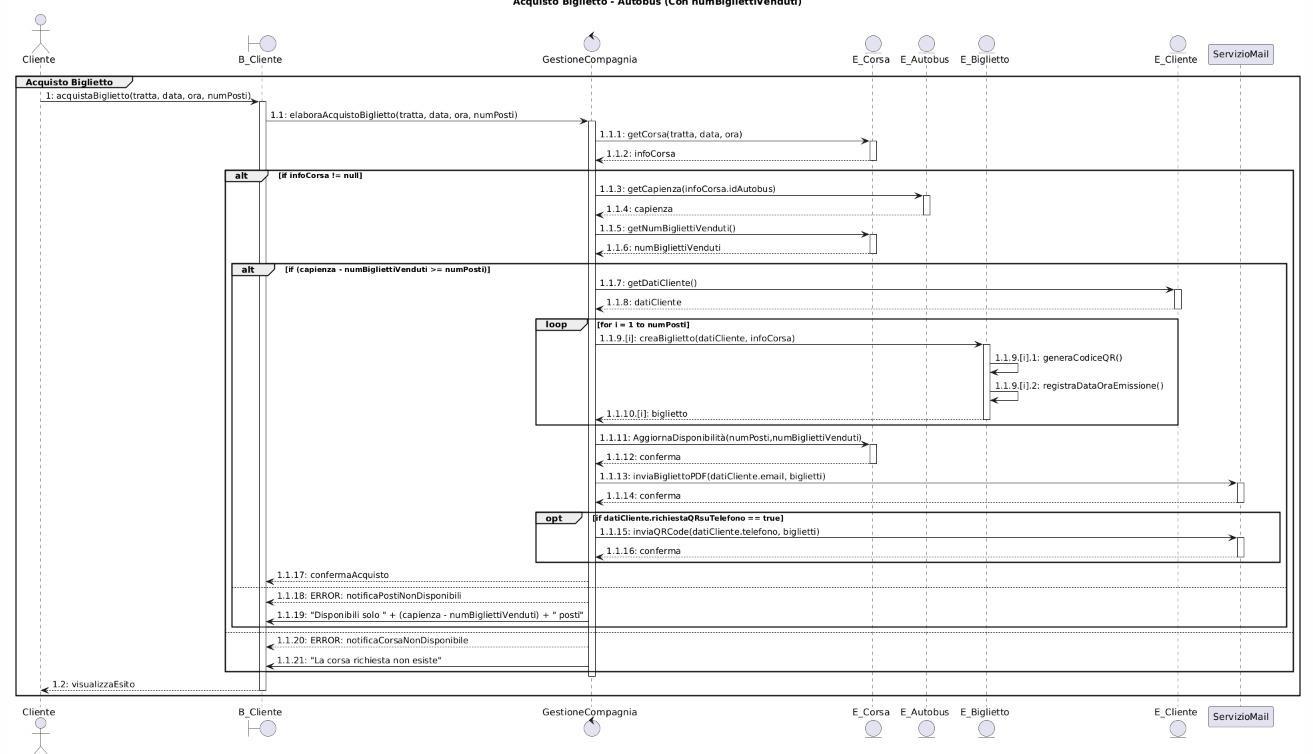
## Modellazione dei dati

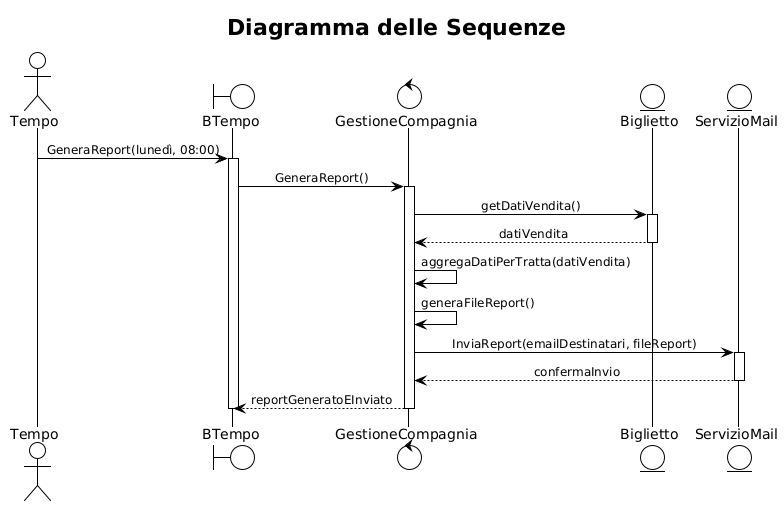
## Diagramma delle classi





## Diagrammi di sequenza





# Stima dei costi

| FATTORI CORRETTIVI | | | |
| --- | --- | --- | --- |
| COMUNICAZIONE DATI | | | 3 |
| DISTRIBUZIONE ELABORAZIONE | | | 1 |
| PRESTAZIONI | | | 3 |
| UTILIZZO INTENSIVO CONFIGURAZIONE | | | 3 |
| FREQUENZA DELLE TRANSAZIONI | | | 4 |
| INSERIMENTO DATI INTERATTIVO | | | 4 |
| EFFICIENZA PER L’UTENTE FINALE | | | 4 |
| AGGIORNAMENTO INTERATTIVO | | | 4 |
| COMPLESSITA’ ELABORATIVA | | | 3 |
| RIUSABILITA’ | | | 3 |
| FACILITA’ INSTALLAZIONE | | | 2 |
| FACILITA’ GESTIONE OPERATIVA |  |  | 2 |
| MOLTEPLICITA’ DI SITI | | | 1 |
| FACILITA’ DI MODIFICA | | | 3 |

**40**

1. Analisi delle Funzionalità e Conteggio UFP (Unadjusted Function Points)

Internal Logical Files (ILF):

* EntityAutobus (gestione dati autobus)
* EntityBiglietto (gestione dati biglietti)
* EntityCliente (gestione dati clienti)
* EntityCorsa (gestione dati corse)
* EntityImpiegato (gestione dati impiegati)

Totale ILF = 5

External Inputs (EI):

* Creazione Autobus ( AutobusDAO.createAutobus )
* Creazione Biglietto ( BigliettoDAO.createBiglietto , orchestrato da acquistaBiglietto / vendiBiglietto )
* Creazione Cliente ( ClienteDAO.createCliente , orchestrato da acquistaBiglietto )
* Aggiornamento Autobus ( AutobusDAO.updateAutobus )
* Aggiornamento Biglietto ( BigliettoDAO.updateBiglietto , orchestrato da modificaBiglietto )
* Aggiornamento Corsa ( CorsaDAO.updateCorsa , orchestrato da acquistaBiglietto / vendiBiglietto / modificaBiglietto )
* Vendita Biglietto (funzionalità complessa GestioneCompagnia.vendiBiglietto )
* Acquisto Biglietto (funzionalità complessa GestioneCompagnia.acquistaBiglietto )
* Modifica Biglietto (funzionalità complessa GestioneCompagnia.modificaBiglietto )
* Autenticazione Impiegato ( GestioneCompagnia.autenticazione )
* Eliminazione Autobus ( AutobusDAO.deleteAutobus )

Totale EI = 11

External Outputs (EO):

* Generazione PDF Biglietto ( PDFGenerator.generaBigliettoPDF )
* Invio Email (tramite EmailSender.inviaBiglietto e EmailSender.inviaEmailConferma )
* Invio SMS ( EmailSender.inviaSMS )

Totale EO = 3

External Inquiries (EQ):

* Lettura Autobus ( AutobusDAO.readAutobus )
* Lettura Biglietto per QR ( BigliettoDAO.getBigliettoByCodiceQR )
* Lettura Cliente ( ClienteDAO.readCliente )
* Lettura Corsa ( CorsaDAO.readCorsa )
* Lettura Impiegato ( ImpiegatoDAO.readImpiegato )
* Ottieni tutti gli Autobus ( AutobusDAO.getAllAutobus )
* Ottieni Biglietti per Corsa ( BigliettoDAO.getBigliettiByCorsa )
* Ottieni Posti Occupati ( BigliettoDAO.getPostiOccupati )
* Ottieni Corse Disponibili ( GestioneCompagnia.getCorseDisponibili )
* Ottieni Corse Alternative ( GestioneCompagnia.getCorseAlternative )

Totale EQ = 10

2. Calcolo degli UFP (Unadjusted Function Points)



3. Calcolo dei Punti Funzione Aggiustati (FP)

Utilizzando la formula FP = UFP \* (0.65 + 0.01 \* Σ Fi) :

FP = 149 \* (0.65 + 0.01 \* 40)

FP = 149 \* (0.65 + 0.40)

FP = 149 \* 1.05

FP = 156.45

Arrotondando, 156 FP.

4. Stima delle Linee di Codice (LOC)

LOC = FP \* LLOC/FP

LOC = 156 \* 53

**LOC = 8268**

Conclusioni della Stima:

La stima dei costi per il progetto, secondo il metodo dei Punti Funzione, risulta in 156 Punti Funzione Aggiustati; questo corrisponde a una stima di circa 8268 Linee di Codice (LOC) per il linguaggio Java.

# Piano di test funzionale

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Acquista Biglietto*”.**

| **Email Cliente** | **Numero di telefono Cliente** | **Numero posti** | **Numero carta** | **Data di scadenza** | **CVV** |
| --- | --- | --- | --- | --- | --- |
| * Stringa con “@” * Stringa senza “@”**[ERROR]** | * numero di cifre = 10 * numero di cifre != 10 **[ERROR]** * Contiene lettere **[ERROR]** | * numero di posti desiderati = numero di posti selezionati * numero di posti desiderati != numero di posti selezionati**[ERROR]** | * numero di cifre = 16 * numero di cifre != 16 **[ERROR]** | * Formato data MM/AA * Formato diverso **[ERROR]** | * Numero di cifre = 3 * Numero di cifre != 3 **[ERROR]** |

### 

### 

### 

### 

### Numero di Test Totali Possibili (senza vincoli):

2 × 3 × 2 × 2 × 2 × 2 **= 96**

### Numero di Test con Vincoli Applicati (1 valido + 7 [ERROR]):

1 + 7 [ERROR] **= 8**

**TEST SUITE**

| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| --- | --- | --- | --- | --- | --- | --- |
| TC-AB-01 | Tutti i dati validi | Email con “@”, Telefono = 10 cifre, Posti desiderati = selezionati, Carta = 16 cifre, Data = MM/AA, CVV = 3 cifre | L’utente ha selezionato una corsa dal menu a tendina | Email: mario@gmail.com  Telefono: 3331234567  Posti desiderati: 2  Posti selezionati: 2  Carta: 1234567812345678  Scadenza: 06/25  CVV: 123 | Biglietto acquistato con successo, riceverai una mail di conferma | Email di conferma inviata (con QR e PDF) |
| TC-AB-02 | Email senza “@” | Email non valida **[ERROR]** (non contiene @) | L’utente ha selezionato una corsa dal menu a tendina | Email: mariogmail.com | l’Indirizzo email deve contenere il carattere “@” | Acquisto non completato |
| TC-AB-03 | Numero di telefono con meno o più di 10 cifre | Telefono ≠ 10 cifre **[ERROR]** | L’utente ha selezionato una corsa dal menu a tendina | Telefono: 3331234566559 | Messaggio di errore | Acquisto non completato |
| TC-AB-04 | Numero di telefono con lettere | Telefono contiene lettere **[ERROR]** | L’utente ha selezionato una corsa dal menu a tendina | Telefono: 33312ab456 | Messaggio di errore | Acquisto non completato |
| TC-AB-05 | Posti desiderati ≠ posti selezionati | Posti desiderati ≠ selezionati **[ERROR]** | L’utente ha selezionato una corsa dal menu a tendina | Posti desiderati: 3  Posti selezionati: 2 | Messaggio di errore | Acquisto non completato |
| TC-AB-06 | Numero carta con meno o più di 16 cifre | Carta ≠ 16 cifre **[ERROR]** | Il cliente ha inserito correttamente i dati della sezione precedente | Carta: 12345678123456  (< 16 cifre) | Messaggio di errore | Acquisto non completato |
| TC-AB-07 | Formato scadenza non MM/AA | Scadenza ≠ formato MM/AA **[ERROR]** | Il cliente ha inserito correttamente i dati della sezione precedente | Scadenza: 2025-06 | Messaggio di errore | Acquisto non completato |
| TC-AB-08 | CVV con 2 cifre | CVV ≠ 3 cifre **[ERROR]** | Il cliente ha inserito correttamente i dati della sezione precedente | CVV: 12 | Messaggio di errore | Acquisto non completato |

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Genera Report*”.**

| **Data/Ora di attivazione** | **Presenza dati vendita** | **Invio email** |
| --- | --- | --- |
| * Lunedì ore 08:00 (attivazione prevista) | * Dati presenti * Nessun dato presente (inviato comunque) | * Successo |

### Numero di Test Totali Possibili (senza vincoli):

1 × 2 × 1 **= 2**

### Numero di Test con Vincoli Applicati (1 valido + 1 caso limite):

1 (dati presenti) + 1 (nessun dato presente) **= 2**

**TEST SUITE**

| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| --- | --- | --- | --- | --- | --- | --- |
| GR-01 | Generazione del report con dati di vendita | Lunedì ore 08:00, Dati presenti, Invio email: Successo | Il sistema ha vendite registrate e il timer è impostato per attivarsi alle 08:00 | No input(Attivazione automatica) | Report generato e inviato via email | Il direttore riceve il report settimanale via email |
| GR-02 | Generazione del report senza dati di vendita | Lunedì ore 08:00, Nessun dato presente, Invio email: Successo | Nessuna vendita registrata; il timer è impostato per attivarsi alle 08:00 | No input(Attivazione automatica) | Report generato (vuoto) e inviato via email | Il direttore riceve il report settimanale via email |

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Vendi Biglietto*”.**

| **ID Impiegato** | **Numero posti** | **Scelta stampa** |
| --- | --- | --- |
| * Numero intero registrato nel database * Numero intero non registrato nel database **[ERROR]** * Contiene dei caratteri alfabetici **[ERROR]** | * numero di posti desiderati = numero di posti selezionati * numero di posti desiderati != numero di posti selezionati **[ERROR]** | * Si * No |

### Numero di Test Totali Possibili (senza vincoli):

3 × 2 × 2 = **12**

### Numero di Test con Vincoli Applicati (1 valido + 3 [ERROR]):

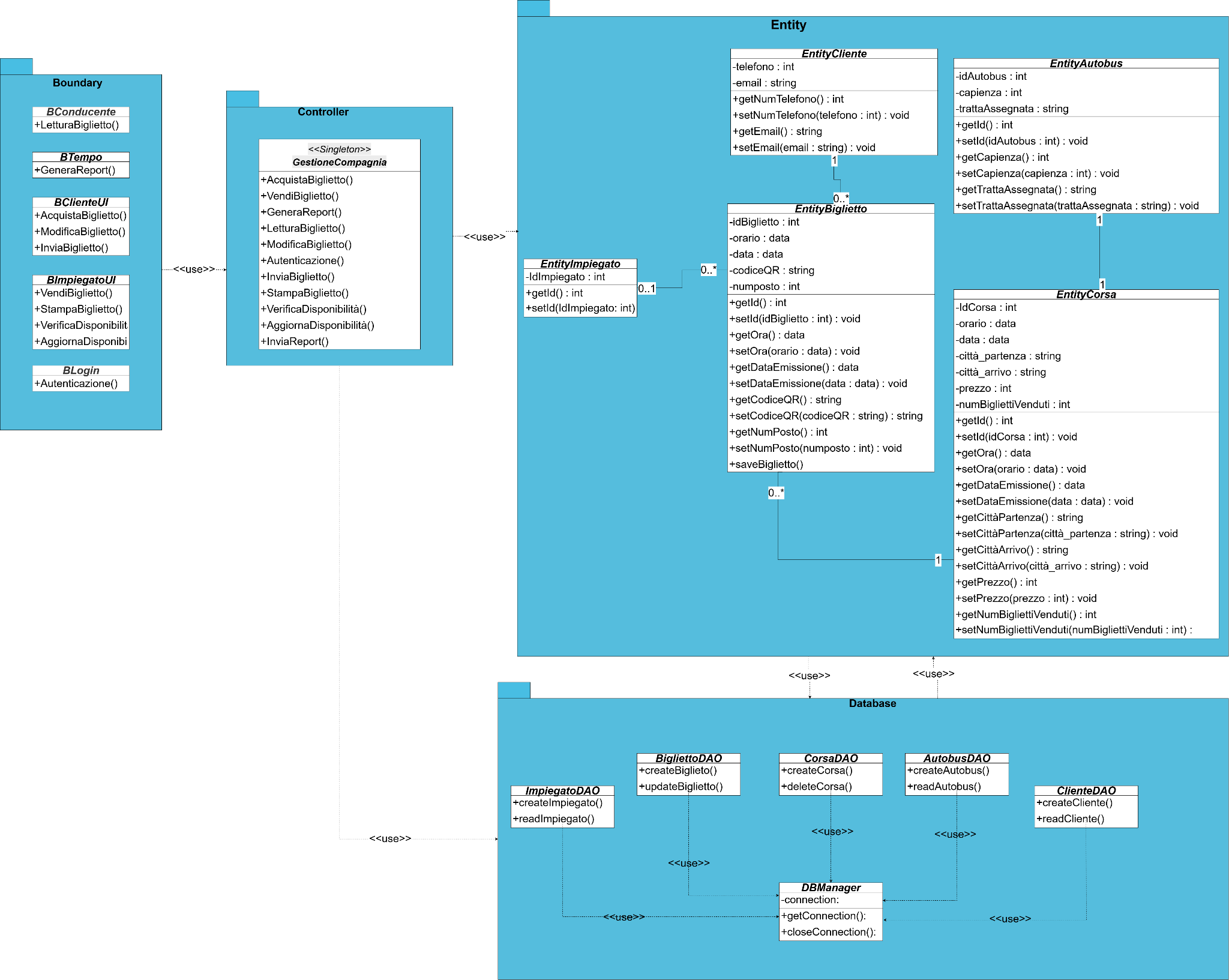
2 validi + 3 [ERROR] = **5**

**TEST SUITE**

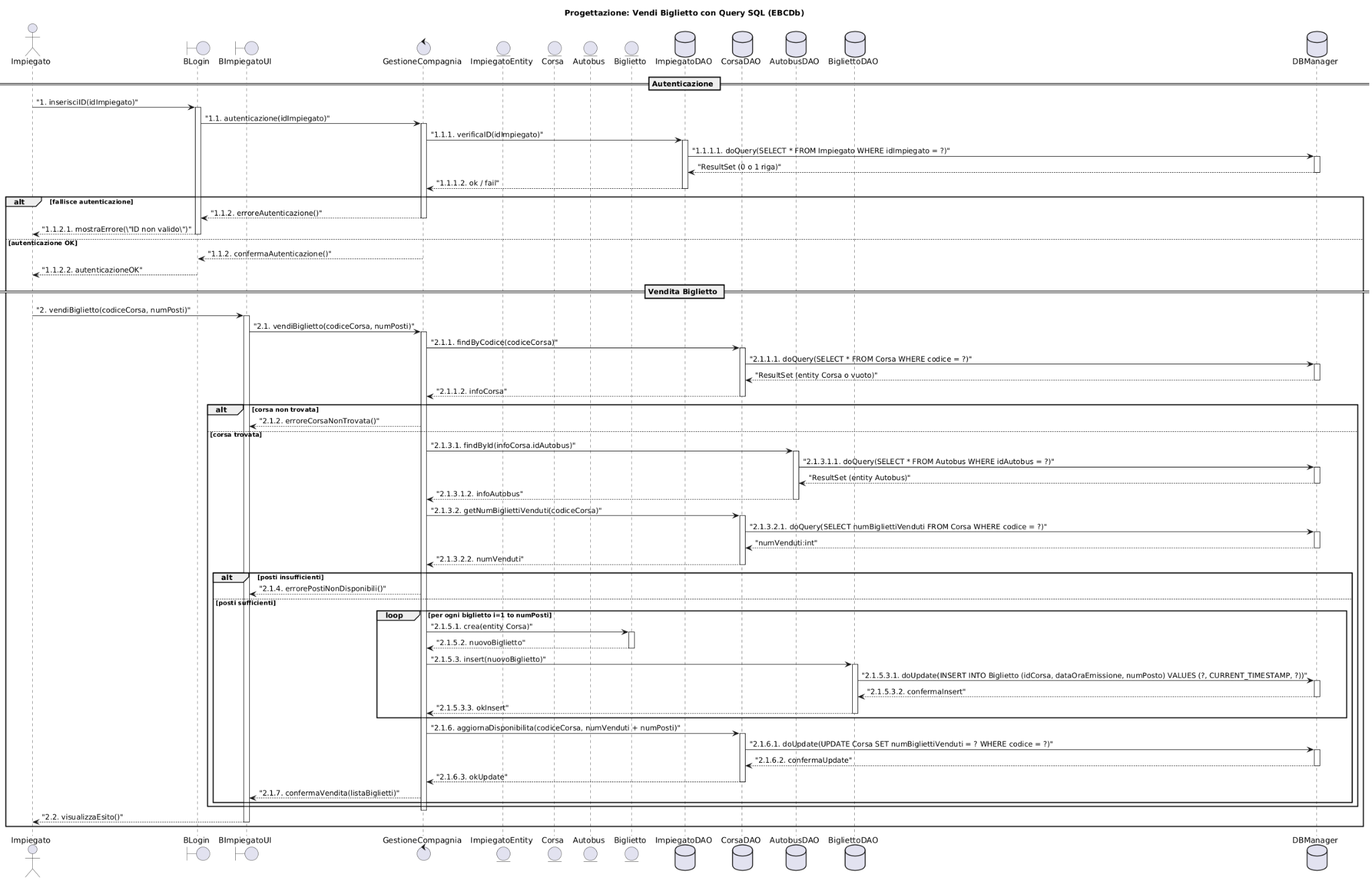
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| --- | --- | --- | --- | --- | --- | --- |
| TC\_VB\_01 | Vendita riuscita con stampa | ID Impiegato valido, posti selezionati correttamente, stampa: Sì | Impiegato autenticato, corsa selezionata | ID: 2, Num. posti desiderati: 2, Posti selezionati: 2, **Stampa: Sì** | Biglietto venduto con successo, richiesta di stampa (accettata) | Biglietto registrato e stampato |
| TC\_VB\_02 | Vendita riuscita senza stampa | ID Impiegato valido, posti selezionati correttamente, stampa: No | Impiegato autenticato, corsa selezionata | ID: 5, Num. posti desiderati: 1, Posti selezionati: 1, **Stampa:** **No** | Biglietto venduto con successo, richiesta di stampa (rifiutata) | Biglietto registrato (niente stampa) |
| TC\_VB\_03 | Errore ID impiegato non registrato | ID non registrato**[ERROR]** | nessuna | ID: 999999 | Messaggio di errore, richiesta reinserimento ID | Nessuna vendita effettuata |
| TC\_VB\_04 | Errore ID con caratteri alfabetici | ID con caratteri **[ERROR]** | nessuna | ID: 2AB | Messaggio di errore, richiesta reinserimento ID | Nessuna vendita effettuata |
| TC\_VB\_05 | Errore mismatch numero posti | Numero posti desiderati ≠ selezionati **[ERROR]** | Impiegato autenticato, corsa selezionata | Num. posti desiderati: 3, Posti selezionati: 2 | Messaggio di errore | Nessuna vendita effettuata |

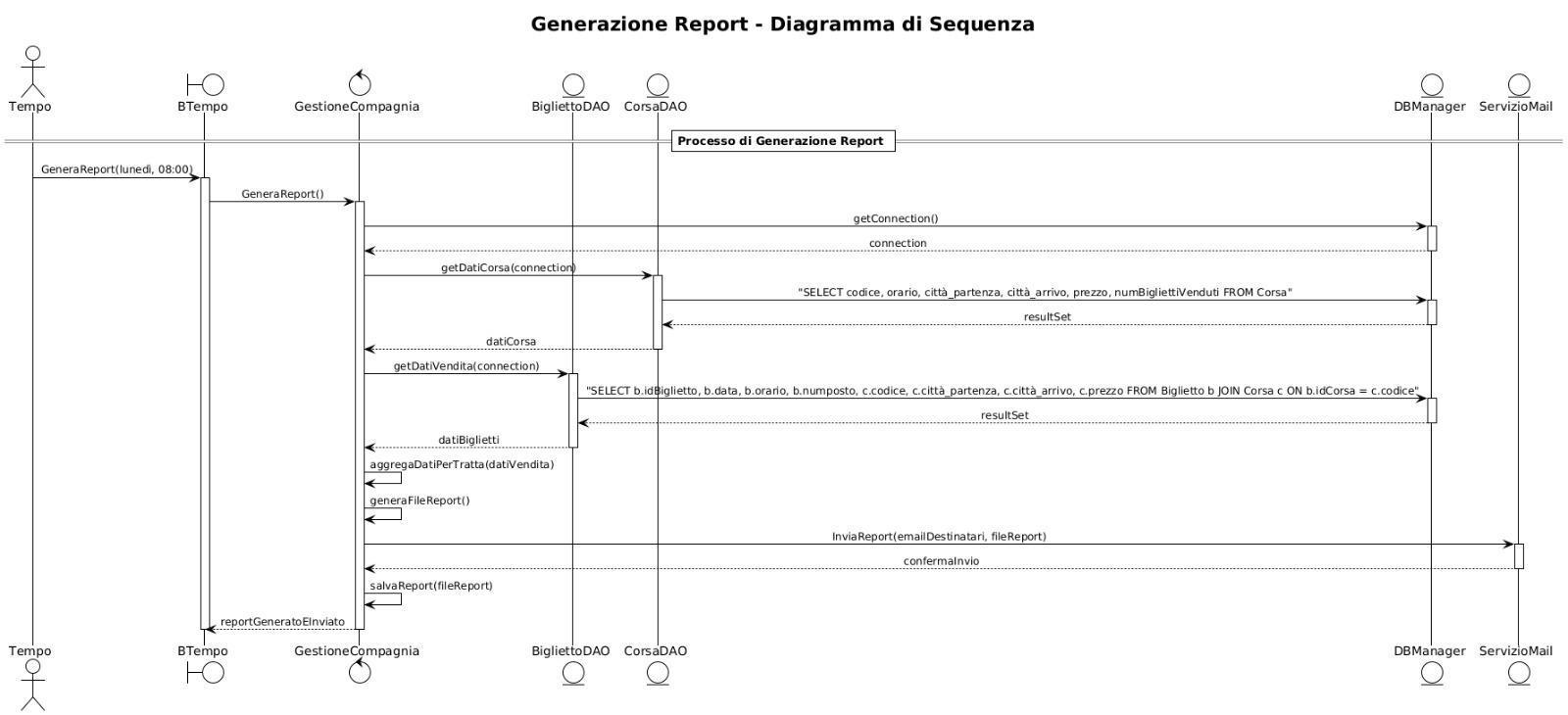
# Progettazione

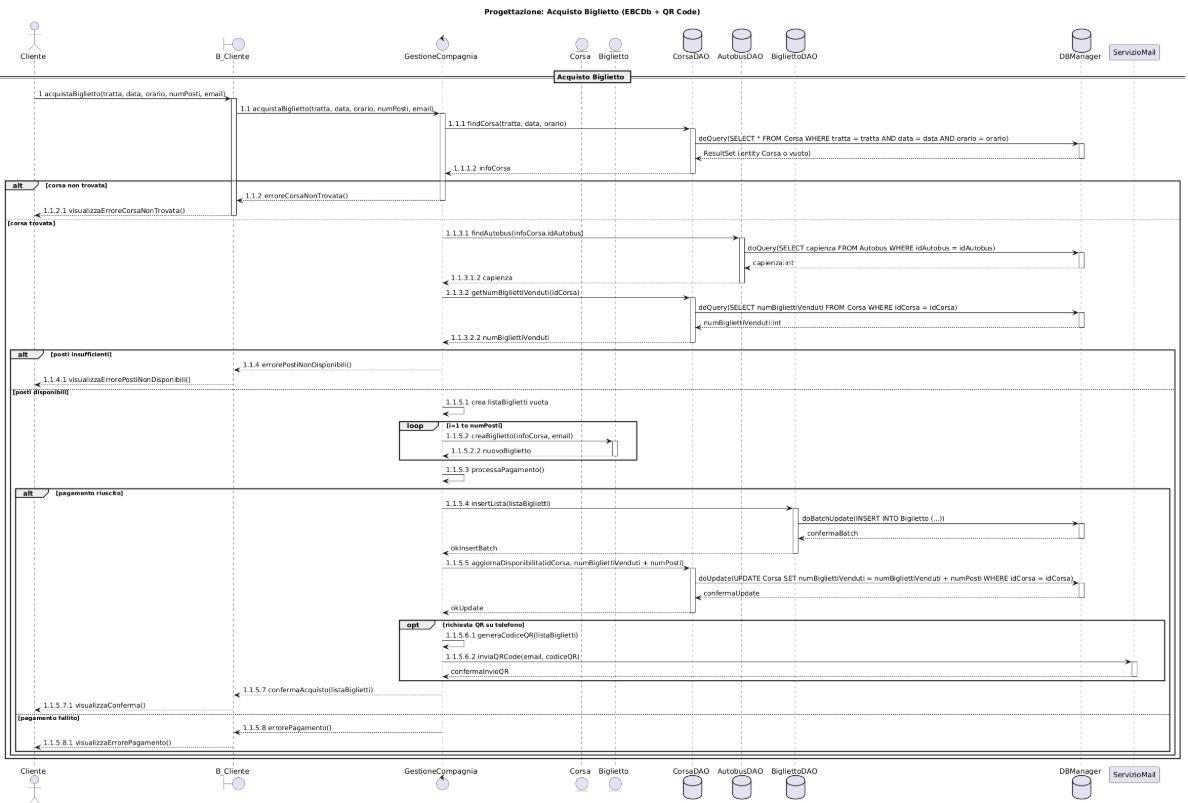
## Diagramma delle classi



## Diagrammi di sequenza







# Implementazione

Descrizione dell'Implementazione del Sistema di Trasporto Regionale degli Autobus:

1. Struttura del Progetto

Il sistema di trasporto regionale degli autobus è stato implementato seguendo l'architettura BCED (Boundary-Control-Entity-Database), che separa chiaramente le responsabilità tra i diversi componenti del sistema.

1.1 Package e Classi

Package boundary

Contiene le classi che gestiscono l'interfaccia utente:

BClienteUI: Interfaccia utente per i clienti che desiderano acquistare biglietti

BImpiegatoUI: Interfaccia utente per gli impiegati della compagnia

BLogin: Gestisce l'autenticazione degli utenti

BMainUI: Interfaccia principale dell'applicazione

BTempo: Gestisce la programmazione e l'invio automatico dei report settimanali

Package control

Contiene le classi che implementano la logica di business:

GestioneCompagnia: Controller principale che gestisce tutte le operazioni di business

Package database

Contiene le classi per l'accesso ai dati:

AutobusDAO: Gestisce le operazioni CRUD per gli autobus

BigliettoDAO: Gestisce le operazioni CRUD per i biglietti

ClienteDAO: Gestisce le operazioni CRUD per i clienti

CorsaDAO: Gestisce le operazioni CRUD per le corse

DBManager: Gestisce la connessione al database

DatabaseConnection: Gestisce la connessione al database (versione aggiornata)

DatabaseInitializer: Inizializza il database con dati di esempio

ImpiegatoDAO: Gestisce le operazioni CRUD per gli impiegati

Package entity

Contiene le classi che rappresentano le entità del dominio:

* EntityAutobus: Rappresenta un autobus
* EntityBiglietto: Rappresenta un biglietto
* EntityCliente: Rappresenta un cliente
* EntityCorsa: Rappresenta una corsa
* EntityImpiegato: Rappresenta un impiegato
* MetodoPagamento: Enumerazione dei metodi di pagamento disponibili

Package exception

Contiene le classi che gestiscono le eccezioni personalizzate:

* AutenticazioneException: Eccezione lanciata in caso di errori di autenticazione
* CorsaNonTrovataException: Eccezione lanciata quando una corsa non viene trovata
* PrinterException: Eccezione lanciata in caso di errori di stampa
* VenditaBigliettiException: Eccezione lanciata in caso di errori nella vendita dei biglietti

Package main

Contiene le classi principali per l'avvio dell'applicazione:

Main: Punto di ingresso dell'applicazione

Package util

Contiene classi di utilità:

* EmailSender: Gestisce l'invio di email
* PDFGenerator: Genera documenti PDF
* PaymentProcessor: Elabora i pagamenti
* PrinterManager: Gestisce la stampa dei biglietti
* QRCodeUtil: Genera e gestisce i codici QR
* ThemeManager: Gestisce i temi dell'interfaccia utente

1.2 Tipi di Eccezione Definiti

Il sistema definisce le seguenti eccezioni personalizzate:

AutenticazioneException: Lanciata quando si verificano problemi durante l'autenticazione di un impiegato, ad esempio quando l'ID impiegato non è registrato nel sistema.

CorsaNonTrovataException: Lanciata quando una corsa richiesta non viene trovata nel database, ad esempio durante la ricerca di una corsa per ID o per criteri specifici.

PrinterException: Lanciata quando si verificano errori durante la stampa dei biglietti, come problemi di connessione con la stampante o errori di formattazione.

VenditaBigliettiException: Lanciata quando si verificano problemi durante il processo di vendita dei biglietti, come la mancata disponibilità di posti o errori di pagamento.

2. Artefatti Necessari per l'Installazione ed Esecuzione

2.1 Requisiti di Sistema

Java Runtime Environment (JRE): Versione 11 o superiore

Sistema Operativo: Windows, macOS o Linux

2.2 Database

H2 Database: Versione 1.4.200

File del database: trasporto\_regionale\_db.mv.db

File di traccia: trasporto\_regionale\_db.trace.db

2.3 Librerie Esterne

Il sistema utilizza le seguenti librerie esterne:

Librerie Core

h2-1.4.200.jar: Database H2

javax.mail-1.6.2.jar: API per l'invio di email

activation-1.1.1.jar: Framework JavaBeans Activation

itextpdf-5.5.13.3.jar: Generazione di documenti PDF

pdfbox-2.0.29.jar: Manipolazione di documenti PDF

core-3.5.2.jar: Libreria core per ZXing (generazione QR code)

javase-3.5.2.jar: Estensione Java SE per ZXing

okhttp-4.9.3.jar: Client HTTP

okio-2.10.0.jar: Libreria I/O per OkHttp

twilio-9.14.1.jar: API per l'invio di SMS

Librerie UI

flatlaf-3.4.jar: Look and feel moderno per Swing

flatlaf-extras-3.4.jar: Componenti aggiuntivi per FlatLaf

flatlaf-intellij-themes-3.4.jar: Temi IntelliJ per FlatLaf

2.4 File Compilati

File .class compilati per tutte le classi Java (da modificare se si fa utilizzo di una versione Java differente e si utilizzano percorsi differenti delle librerie rispetto a quelle dichiarate).

Risorse come icone e immagini nella directory resources/icons.

File temporanei per i codici QR nella directory temp/qrcodes.

2.5 Procedura di Installazione

Assicurarsi che sia installato Java 11 o superiore

*Se non si vuole installare ogni libreria necessaria per l’avvio, si può scaricare e far utilizzo di Maven, dichiarare le librerie in un file pom.xml, in modo che lo strumento di build automation le scarichi in automatico.*

*Una volta scaricato, per farlo funzionare bisognerà porre il percorso del /bin di maven nel path delle variabili d’ambiente, e nel terminale digitare cd “percorso-progetto” e dopo esserci porsi nel path possiamo eseguire il comando mvn clean install, che permetterà di installare le librerie dichiarate nel file pom.*

Estrarre l'archivio CompagniaAutobusModificatoV2.rar in una directory a scelta

Verificare che tutti i file JAR delle librerie siano presenti nella directory lib

Verificare che i file del database H2 siano presenti nella directory principale

2.6 Procedura di Esecuzione

Per eseguire l'applicazione:

* Su Eclipse:

1. Caricare il progetto “nome\_cartella\_progetto”;
2. Aprire il package “main”;
3. Tasto destro del mouse sul file “Main.java” poi “Run as” —> “1. Java Application”.

3. Documentazione delle Funzionalità Implementate

3.1 Funzionalità Principali

3.1.1 Gestione Compagnia (Control)

La classe GestioneCompagnia è il controller principale dell'applicazione e implementa la logica di business per la compagnia di trasporto regionale. Utilizza il pattern Singleton per garantire un'unica istanza in tutto il sistema.

Funzionalità principali:

* Autenticazione degli impiegati
* Vendita dei biglietti
* Acquisto dei biglietti da parte dei clienti
* Modifica dei biglietti da parte dei clienti
* Generazione di report settimanali
* Verifica della disponibilità dei posti
* Gestione delle corse e degli autobus

3.1.2 Entità (Entity)

Le classi Entity rappresentano le entità del dominio:

EntityBiglietto: Rappresenta un biglietto con attributi come ID, orario, data, codice QR, numero posto, ecc.

EntityCorsa: Rappresenta una corsa con attributi come ID, orario, data, città di partenza e arrivo, prezzo, ecc.

EntityAutobus: Rappresenta un autobus con attributi come ID, capienza, tratta assegnata, ecc.

EntityCliente: Rappresenta un cliente con attributi come telefono, email, ecc.

EntityImpiegato: Rappresenta un impiegato con attributo ID.

3.1.3 Accesso ai Dati (DAO)

Le classi DAO implementano l'accesso ai dati per le diverse entità:

BigliettoDAO: Gestisce le operazioni CRUD per i biglietti

CorsaDAO: Gestisce le operazioni CRUD per le corse

AutobusDAO: Gestisce le operazioni CRUD per gli autobus

ClienteDAO: Gestisce le operazioni CRUD per i clienti

ImpiegatoDAO: Gestisce le operazioni CRUD per gli impiegati

3.1.4 Interfaccia Utente (Boundary)

Le classi Boundary implementano l'interfaccia utente:

BClienteUI: Interfaccia per i clienti che permette di cercare corse e acquistare biglietti

BImpiegatoUI: Interfaccia per gli impiegati che permette di vendere biglietti

BLogin: Gestisce l'autenticazione degli impiegati

BMainUI: Interfaccia principale che permette di scegliere tra area cliente e area impiegato

BTempo: Gestisce la programmazione e l'invio automatico dei report settimanali

3.1.5 Utilità (Util)

Le classi Util forniscono funzionalità di supporto:

EmailSender: Gestisce l'invio di email con i biglietti o i report

PDFGenerator: Genera documenti PDF per biglietti

QRCodeUtil: Genera codici QR per i biglietti

PaymentProcessor: Elabora i pagamenti per l'acquisto dei biglietti

PrinterManager: Gestisce la stampa dei biglietti

ThemeManager: Gestisce i temi dell'interfaccia utente

3.2 Casi d'Uso Implementati

3.2.1 Vendita Biglietti

Il caso d'uso "Vendita Biglietti" permette agli impiegati di vendere biglietti ai clienti. Il processo include:

* Autenticazione dell'impiegato
* Selezione della corsa
* Verifica della disponibilità dei posti
* Inserimento dei dati del cliente
* Selezione del metodo di pagamento
* Generazione e stampa del biglietto

3.2.2 Acquisto Biglietti

Il caso d'uso "Acquisto Biglietti" permette ai clienti di acquistare biglietti autonomamente. Il processo include:

* Ricerca delle corse disponibili
* Selezione della corsa
* Inserimento dei dati personali
* Selezione del metodo di pagamento
* Ricezione del biglietto via email o SMS

3.2.3 Generazione Report

Il caso d'uso "Generazione Report" permette agli impiegati di generare report sui biglietti venduti. Il sistema può anche inviare automaticamente report settimanali al direttore ogni lunedì alle 8:00. Il processo include:

* Generazione del report in formato PDF
* Visualizzazione o invio del report via email

4. Conclusioni

Il sistema di trasporto regionale degli autobus è stato implementato seguendo l'architettura BCED, che separa chiaramente le responsabilità tra i diversi componenti del sistema. Il sistema è in grado di gestire la vendita e l'acquisto di biglietti, la generazione di report e altre funzionalità richieste.

Il sistema è stato progettato per essere facilmente estensibile e manutenibile, con una chiara separazione tra l'interfaccia utente, la logica di business e l'accesso ai dati. Inoltre, il sistema include test unitari per verificare il corretto funzionamento delle funzionalità principali.

# Testing

## Test funzionale

| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Output Ottenuti** | **Post-condizioni Ottenute** | **Esito**  **(*FAIL*, *PASS*)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TC-AB-01 | Tutti i dati validi | Email con “@”, Telefono = 10 cifre, Posti desiderati = selezionati, Carta = 16 cifre, Data = MM/AA, CVV = 3 cifre | L’utente ha selezionato una corsa dal menu a tendina | Email: nellogaudino0@gmail.com  Telefono: 3792631098  Posti desiderati: 2  Posti selezionati: 2  Carta: 5342321157366589  Scadenza: 02/29  CVV: 884 | Biglietto acquistato con successo, riceverai una mail di conferma | Email di conferma inviata (con QR e PDF) | Biglietto acquistato con successo, riceverai una mail di conferma | Email di conferma inviata (con QR e PDF) | PASS |
| TC-AB-02 | Email senza “@” | Email non valida **[ERROR]** (non contiene @) | L’utente ha selezionato una corsa dal menu a tendina | Email: mariogmail.com | l’Indirizzo email deve contenere il carattere “@” | Acquisto non completato | l’Indirizzo email deve contenere il carattere “@” | Acquisto non completato | PASS |
| TC-AB-03 | Numero di telefono con meno o più di 10 cifre | Telefono ≠ 10 cifre **[ERROR]** | L’utente ha selezionato una corsa dal menu a tendina | Telefono: 3331234566559 | Il numero di telefono non é valido, deve contenere esattamente 10 cifre | Acquisto non completato | Il numero di telefono non é valido, deve contenere esattamente 10 cifre | Acquisto non completato | PASS |
| TC-AB-04 | Numero di telefono con lettere | Telefono contiene lettere **[ERROR]** | L’utente ha selezionato una corsa dal menu a tendina | Telefono: 33312ab456 | Il numero di telefono non é valido, deve contenere esattamente 10 cifre | Acquisto non completato | Il numero di telefono non é valido, deve contenere esattamente 10 cifre | Acquisto non completato | PASS |
| TC-AB-05 | Posti desiderati ≠ posti selezionati | Posti desiderati ≠ selezionati **[ERROR]** | L’utente ha selezionato una corsa dal menu a tendina | Posti desiderati: 3  Posti selezionati: 2 | Devi selezionare esattamente X posti. Hai selezionato Y posti | Acquisto non completato | Devi selezionare esattamente X posti. Hai selezionato Y posti | Acquisto non completato | PASS |
| TC-AB-06 | Numero carta con meno o più di 16 cifre | Carta ≠ 16 cifre **[ERROR]** | Il cliente ha inserito correttamente i dati della sezione precedente | Carta: 12345678123456  (< 16 cifre) | Numero carta non valido | Acquisto non completato | Numero carta non valido | Acquisto non completato | PASS |
| TC-AB-07 | Formato scadenza non MM/AA | Scadenza ≠ formato MM/AA **[ERROR]** | Il cliente ha inserito correttamente i dati della sezione precedente | Scadenza: 2025-06 | Data di scadenza della carta non Valida | Acquisto non completato | Data di scadenza della carta non Valida | Acquisto non completato | PASS |
| TC-AB-08 | CVV con 2 cifre | CVV ≠ 3 cifre **[ERROR]** | Il cliente ha inserito correttamente i dati della sezione precedente | CVV: 12 | CVV deve essere di 3 cifre | Acquisto non completato | Acquisto non completato | CVV deve essere di 3 cifre | PASS |
| TC\_VB\_01 | Vendita riuscita con stampa | ID Impiegato valido, posti selezionati correttamente, stampa: Sì | Impiegato autenticato, corsa selezionata | ID: 2, Num. posti desiderati: 2, Posti selezionati: 2, **Stampa: Sì** | Biglietti venduti con successo: - posto X: codice qr (formato cifre + caratteri) | Biglietto registrato e stampato | Biglietti venduti con successo: - posto X: codice qr (formato cifre + caratteri) | Biglietto registrato e stampato | PASS |
| TC\_VB\_02 | Vendita riuscita senza stampa | ID Impiegato valido, posti selezionati correttamente, stampa: No | Impiegato autenticato, corsa selezionata | ID: 5, Num. posti desiderati: 1, Posti selezionati: 1, **Stampa:** **No** | Biglietto venduto con successo, richiesta di stampa (rifiutata) | Biglietto registrato (niente stampa) | Biglietti venduti con successo: - posto X: codice qr (formato cifre + caratteri) | Biglietto registrato (niente stampa) | PASS |
| TC\_VB\_03 | Errore ID impiegato non registrato | ID non registrato**[ERROR]** | nessuna | ID: 999999 | ID impiegato non valido o non presente nel sistema | Nessuna vendita effettuata | ID impiegato non valido o non presente nel sistema | Nessuna vendita effettuata | PASS |
| TC\_VB\_04 | Errore ID con caratteri alfabetici | ID con caratteri **[ERROR]** | nessuna | ID: 2AB | L’ID deve essere un numero | Nessuna vendita effettuata | L’ID deve essere un numero | Nessuna vendita effettuata | PASS |
| TC\_VB\_05 | Errore mismatch numero posti | Numero posti desiderati ≠ selezionati **[ERROR]** | Impiegato autenticato, corsa selezionata | Num. posti desiderati: 3, Posti selezionati: 2 | Devi selezionare esattamente X posti. Hai selezionato Y posti | Nessuna vendita effettuata | Devi selezionare esattamente X posti. Hai selezionato Y posti | Nessuna vendita effettuata | PASS |