



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**FEDERICO II**

**CORSO DI LAUREA IN INGEGNERIA INFORMATICA**

**CORSO DI INGEGNERIA DEL SOFTWARE**

**PROF. R. PIETRANTUONO- A.A. 2021 – 22**



**POSEIDON**

**STUDENTI:**

LEONARDO	CATELLO	N46004862
DAIANA	CIPOLLARO	N46004941
FRANCESCO	DI SERIO	N46004817

<a href="mailto:l.catello@studenti.unina.it">l.catello@studenti.unina.it</a>
<a href="mailto:da.cipollaro@studenti.unina.it">da.cipollaro@studenti.unina.it</a>
<a href="mailto:f.diserio@studenti.unina.it">f.diserio@studenti.unina.it</a>

# Indice

## 1. SPECIFICHE INFORMALI

## 2. ANALISI E SPECIFICA DEI REQUISITI

- 2.1 ANALISI NOMI-VERBI
- 2.2 REVISIONE DEI REQUISITI
- 2.3 GLOSSARIO DEI TERMINI
- 2.4 CLASSIFICAZIONE DEI REQUISITI
  - 2.4.1 REQUISITI FUNZIONALI
  - 2.4.2 REQUISITI SUI DATI
  - 2.4.3 VINCOLI / ALTRI REQUISITI
- 2.5 MODELLAZIONE DEI CASI D'USO
  - 2.5.1 ATTORI E CASI D'USO
  - 2.5.2 DIAGRAMMA DEI CASI D'USO
  - 2.5.3 SCENARI
- 2.6 DIAGRAMMA DELLE CLASSI
- 2.7 DIAGRAMMI DI SEQUENZA

## 3. STIMA DEI COSTI

## 4. PIANO DI TEST FUNZIONALE

## 5. PROGETTAZIONE

- 5.1 DIAGRAMMA DELLE CLASSI
- 5.2 DIAGRAMMI DI SEQUENZA

## 6. IMPLEMENTAZIONE

## 7. TESTING

- 7.1 TEST STRUTTURALE
  - 7.1.1 COMPLESSITÀ CICLOMATICA
- 7.2 TEST FUNZIONALE

# 1. Specifiche informali

## ID 5: BIGLIETTERIA COMPAGNIA DI NAVIGAZIONE

Si vuole realizzare un'applicazione per compagnie di navigazione per l'emissione dei biglietti per viaggi marittimi.

Una compagnia di navigazione (caratterizzata dal proprio nome: *Poseidon*) fornisce servizi di trasporto passeggeri (operati con aliscafi e traghetti) e di trasporto passeggeri e autoveicoli (operati con traghetti). Ogni nave ha un nome e una capienza in termini di numero di posti disponibili per i passeggeri, per i traghetti, per gli autoveicoli.

La compagnia di navigazione opera con le proprie navi un insieme di corse giornaliere tra N porti. Esistono due tipi di biglietti, per passeggero e per autoveicolo. Ogni corsa è identificata da un codice e ha un orario e porto di partenza e di arrivo, e un prezzo dei biglietti prefissato. Una corsa è effettuata sempre dalla stessa nave.

La biglietteria di una compagnia può emettere i biglietti viaggio solo per una specifica corsa in un determinato giorno. Prima di emettere un biglietto, il sistema deve valutare le effettive disponibilità del posto. All'atto dell'emissione dovrà essere aggiornata alla disponibilità residua.

L'applicazione supporta l'emissione di biglietti da parte degli impiegati di una compagnia appunto 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 ora di emissione e codice dell'impiegato che lo ha emesso. Per i biglietti per i veicoli occorre registrare e stampare la targa del veicolo.

È altresì previsto l'uso dell'applicazione da parte di utenti clienti di una compagnia, che possono prenotare via web un biglietto passeggero e/o per autoveicolo. L'emissione del biglietto prenotato avverrà tuttavia a cura di un impiegato della compagnia, cui il cliente dovrà esibire una stampa della prenotazione che riporti gli estremi della corsa e, eventualmente, la targa del veicolo.

## 2. Analisi e specifica dei requisiti

### 2.1 Revisione dei requisiti

1. La compagnia di navigazione da gestire è identificata dal proprio nome.
2. La compagnia di navigazione fornisce servizi di trasporto passeggeri e autoveicoli.
3. Il trasporto può avvenire sia con aliscafi che con traghetti.
4. Gli autoveicoli possono essere trasportati solo su traghetti.
5. I passeggeri possono essere trasportati sia su traghetti che aliscafi.
6. La compagnia vende due tipi di biglietti:
  - a. Passeggero
  - b. Autoveicolo
7. Ogni nave che realizza un trasporto viene caratterizzata da un *nome, capienza in termini di numero di posti disponibili*.
8. Di ogni traghetto il sistema deve memorizzare la capienza disponibile sia per passeggeri che per autoveicoli.
9. Di ogni aliscafo il sistema deve memorizzare il numero di posti disponibili per passeggeri.
10. La compagnia opera con le proprie navi un certo numero di corse giornaliere tra N porti.
11. Ogni corsa è identificata da un *codice, orario di partenza, porto di partenza, orario di arrivo, porto di arrivo* e un *prezzo dei biglietti prefissato*.
12. Una corsa è effettuata sempre dalla stessa nave.
13. Il sistema può emettere i biglietti viaggio solo per una specifica corsa in un determinato giorno.
14. Prima di emettere un biglietto, il sistema deve valutare le effettive disponibilità del posto.
15. All'atto dell'emissione, il sistema deve aggiornare la disponibilità residua dei posti.
16. L'applicazione supporta l'emissione di biglietti da parte degli impiegati di una compagnia.
17. Ogni impiegato deve accedere al sistema solo previa autenticazione.
18. Al momento dell'autenticazione, il sistema deve richiedere *nome, cognome, password* dell'impiegato.
19. Per ogni biglietto il sistema deve stampare la *data* e l'*ora di emissione*.
20. Per ciascun biglietto il sistema deve registrare internamente, al momento dell'emissione, *data di emissione, ora di emissione* e *codice dell'impiegato che lo ha emesso*.
21. Per i biglietti per i veicoli il sistema deve registrare e stampare la targa del veicolo.
22. È altresì previsto l'uso dell'applicazione da parte di utenti clienti di una compagnia.
23. Il sistema deve permettere ai clienti di prenotare via web un biglietto passeggero e/o per autoveicolo.
24. Al momento della prenotazione, il sistema deve richiedere al cliente di autenticarsi fornendo *nome, cognome e password*.
25. L'emissione del biglietto prenotato avverrà a cura di un impiegato della compagnia.
26. Al momento dell'emissione, il cliente dovrà esibire una stampa della prenotazione che riporti *gli estremi della corsa e, eventualmente, la targa del veicolo*.
27. Al momento dell'acquisto, il sistema deve memorizzare *il codice del cliente, gli estremi della corsa, la targa del veicolo (se presente) e la ricevuta di acquisto* nella cronologia acquisti del cliente.
28. Il sistema deve permettere al cliente di visualizzare la propria cronologia acquisti.

## 2.2 Glossario dei termini

Termine	Descrizione	Sinonimi
<b>Nave</b>	Mezzo di trasporto marittimo che possa esso essere traghetto o aliscafo	Traghetto Aliscafo
<b>Capienza</b>	Numero di posti totali della nave	Disponibilità
<b>Utente</b>	Persona fisica che interagisce con il sistema	Dipendente Cliente
<b>Dipendente</b>	Utente che lavora presso la compagnia di navigazione, munito di password per l'autenticazione	Impiegato, Addetto
<b>Cliente</b>	Utente che usufruisce del servizio di emissione dei biglietti fornito dall'applicazione, munito di password per l'autenticazione	Acquirente
<b>Sistema</b>	Software che permette di gestire tutti i servizi offerti dalla compagnia di navigazione	Applicazione
<b>Biglietto</b>	Documento cartaceo o digitale rilasciato da un impiegato della compagnia in seguito all'acquisto di un posto a bordo di una nave	Attestato di acquisto
<b>Cronologia Acquisti</b>	Documento che certifica la prenotazione di un posto a bordo di una nave, riportante le informazioni relative al codice del cliente, alla corsa selezionata, alla ricevuta erogata e, in seguito all'emissione, al biglietto acquistato.	Attestato di prenotazione

## 2.3 Classificazione dei requisiti

### 2.4 Requisiti funzionali

ID	Requisito	Origine (numero del requisito revisionato)
RF01	Di ogni traghetto il sistema deve memorizzare la capienza disponibile sia per passeggeri che per autoveicoli.	8
RF02	Di ogni aliscafo il sistema deve memorizzare il numero di posti disponibili per passeggeri.	9
RF03	Il sistema può emettere i biglietti viaggio solo per una specifica corsa in un determinato giorno.	13
RF04	Prima di emettere un biglietto, il sistema deve valutare le effettive disponibilità del posto.	14
RF05	All'atto dell'emissione, il sistema deve aggiornare la disponibilità residua dei posti.	15
RF06	L'applicazione supporta l'emissione di biglietti da parte degli impiegati.	16
RF07	Ogni impiegato deve accedere al sistema solo previa autenticazione.	17
RF08	Al momento dell'autenticazione, il sistema deve richiedere nome, cognome, password dell'impiegato.	18
RF09	Per ogni biglietto il sistema deve stampare la data e l'ora di emissione.	19
RF10	Per ciascun biglietto il sistema deve registrare internamente, al momento dell'emissione, data ora di emissione e codice dell'impiegato che lo ha emesso.	20
RF11	Per i biglietti per i veicoli il sistema deve registrare e stampare la targa del veicolo.	21
RF12	Il sistema deve permettere ai clienti di prenotare via web un biglietto passeggero e/o per autoveicolo.	23
RF13	Al momento della prenotazione, il sistema deve richiedere al cliente di autenticarsi fornendo nome, cognome e password.	24
RF14	Al momento dell'acquisto, il sistema deve memorizzare il codice del cliente, gli estremi della corsa, la targa del veicolo (se presente) e la ricevuta di acquisto nella cronologia acquisti del cliente.	27
RF15	Il sistema deve permettere al cliente di visualizzare la propria cronologia acquisti.	28

## 2.5 Requisiti sui dati

ID	Requisito	Origine (numero del requisito revisionato)
RD01	Il trasporto può avvenire sia con aliscafi che con traghetti.	3
RD02	Gli autoveicoli possono essere trasportati solo su traghetti.	4
RD03	I passeggeri possono essere trasportati sia su traghetti che aliscafi.	5
RD04	La compagnia vende due tipi di biglietti: Passeggero Autoveicolo	6
RD05	Ogni nave che realizza un trasporto viene caratterizzata da un nome, capienza in termini di numero di posti disponibili.	7
RD06	La compagnia fa un certo numero di corse giornaliere tra N porti.	10
RD07	Ogni corsa è identificata da un codice, orario, porto di partenza, porto di arrivo e un prezzo dei biglietti prefissato.	11
RD08	Una corsa è effettuata sempre dalla stessa nave.	12

## 2.6 Vincoli / Altri requisiti

RNF01: Il sistema da sviluppare deve essere una applicazione web, cui i clienti e i dipendenti accedono da uno tra i seguenti browser: Firefox, Chrome.

RNF02: Il sistema è conforme alle normative sulla privacy rispetto ai dati personali dei clienti registrati all'applicazione e agli acquisti da loro effettuati.

RNF03: Il sistema deve gestire accessi contemporanei.

RNF04: Il sistema deve prevedere almeno 500 operazioni al giorno.

RNF05: Multiutenza: le informazioni trattate dal sistema software devono poter essere gestite da diverse postazioni (terminali). Ciascun impiegato deve disporre delle funzionalità dell'applicazione.

RNF06: L'applicazione deve essere progettata in maniera tale che il codice sia fortemente riusabile.

RNF07: Il sistema software richiede l'uso di un DBMS (Data Base Management System).

## 2.7 Modellazione dei casi d'uso

### 2.8 Attori e casi d'uso

#### Attori Primari:

- Utente
- Cliente
- Dipendente

#### Attori Secondari:

- Stampante

#### Casi d'uso:

- Ricerca corsa
- Autenticazione dipendente
- Autenticazione cliente
- Inserimento corsa
- Modifica corsa
- Cancellazione corsa
- Acquisto biglietto
- Emissione biglietto
- Visualizzazione cronologia acquisti

RICERCA : Consente al cliente e al dipendente la ricerca di una corsa.

- La ricerca è effettuata per tratta da percorrere, data, ora, codice e porto di partenza e di arrivo.
- In seguito ad un'operazione di ricerca, l'applicazione deve mostrare al cliente i seguenti dati:
  - Codice della corsa
  - Categoria: aliscafo, traghetto
  - Prezzo (fisso in base alla corsa)
  - Nome Nave (fissa in base alla corsa)
  - Numero disponibilità di posti passeggeri
  - Numero disponibilità di posti autoveicoli
  - Porto di partenza
  - Porto di arrivo

#### AUTENTICAZIONE DIPENDENTE

- Avviene contestualmente ad una delle operazioni di inserimento, cancellazione, modifica di una corsa, emissione di un biglietto.
- Prima che un addetto effettui una di queste operazioni, il sistema deve autenticarlo. Pertanto, l'utente deve fornire la propria identità al sistema (nome, cognome, password).
- Successivamente all'inserimento delle informazioni di identificazione, l'addetto visualizzerà le funzionalità per la gestione delle operazioni critiche (inserimento, modifica, ...).



- Non è previsto un messaggio esplicito relativo all'inserimento dell'identità dell'operatore, ma la corretta esecuzione di tale operazione viene notificata all'addetto direttamente attraverso la visualizzazione delle funzionalità messe a disposizione.

#### AUTENTICAZIONE CLIENTE

- Avviene contestualmente ad una delle operazioni di acquisto di un biglietto e visualizzazione della propria cronologia acquisti.
- Prima che un cliente effettui una di queste operazioni, il sistema deve autenticarlo. Pertanto, l'utente deve fornire la propria identità al sistema (nome, cognome, password).
- Non è previsto un messaggio esplicito relativo all'inserimento dell'identità del cliente, ma la corretta esecuzione di tale operazione viene notificata all'acquirente direttamente attraverso la visualizzazione delle funzionalità messe a disposizione.

#### INSERIMENTO CORSA

- L'inserimento consente all'addetto di registrare una nuova tratta.
- Preliminarmente, l'addetto deve autenticarsi. In seguito, devono essere inserite all'interno della relativa pagina web dell'applicazione, le informazioni che riguardano la tratta e la nave che la compie. L'addetto deve inserire quindi il codice, data e ora, prezzo, porto di partenza e di arrivo.
- L'applicazione informa l'utente che l'inserimento è avvenuto con successo.

#### MODIFICA CORSA

- Consente all'impiegato di modificare le informazioni relative ad una corsa già presente nell'applicazione. Tale operazione permette quindi all'operatore di correggere eventuali errori di battitura, occorsi durante la fase di inserimento.
- Preliminarmente l'addetto deve effettuare la ricerca e la selezione della corsa di cui si vuole effettuare la modifica, in seguito deve autenticarsi. Successivamente devono essere inserite, all'interno della relativa pagina web dell'applicazione, le informazioni da modificare che riguardano la tratta.
- L'applicazione informa l'utente che la modifica è avvenuta con successo.

#### CANCELLAZIONE CORSA

- Consente all'addetto di cancellare una corsa dal sistema.
- L'addetto effettua preliminarmente la ricerca e la selezione della tratta da eliminare; quindi, effettua l'operazione di autenticazione infine di cancellazione. Inoltre, deve confermare le operazioni di eliminazione.
- L'applicazione informa l'utente che l'eliminazione è avvenuta con successo.

#### ACQUISTO BIGLIETTO

- Consente al cliente di acquistare un biglietto per una corsa. L'applicazione deve aggiornare la quantità di posti disponibile per i passeggeri e per gli autoveicoli sulla nave che diminuisce all'aumentare del numero di vendite.
- Il cliente effettua la ricerca e la selezione della corsa da acquistare. Quindi utilizza la funzionalità di acquisto, fornendo al sistema la sua identità. In seguito, il sistema inserisce

all'interno della relativa pagina web le informazioni che riguardano la vendita del biglietto: ricevuta, codice del cliente, informazioni sulla corsa. Per i biglietti per i veicoli occorre registrare anche la targa del veicolo.

- L'applicazione informa l'utente che l'acquisto è avvenuto con successo.

#### EMISSIONE BIGLIETTO

- Consente al dipendente di emettere un biglietto per una corsa.
- Preliminarmente l'impiegato effettua l'operazione di autenticazione. In seguito, inserisce il codice della corsa e il tipo di biglietto da emettere.
- Il sistema verifica l'effettiva disponibilità di posti per la corsa selezionata. Successivamente, registra e stampa la data e l'ora di emissione del biglietto; inoltre, registra anche il codice dell'impiegato che lo ha emesso.
- Per i biglietti per i veicoli, il sistema registra e stampa la targa del veicolo.
- L'applicazione rende possibile la visualizzazione dell'attestato nella cronologia acquisti del cliente e aggiorna la disponibilità residua.

#### VISUALIZZAZIONE CRONOLOGIA ACQUISTI

- Consente al cliente di visualizzare *data e ora di emissione, il codice della corsa, il codice del biglietto, il codice dell'impiegato, orario e porto di partenza e di arrivo, il prezzo, l'Id del cliente e la ricevuta* dei biglietti che ha acquistato presso la compagnia di navigazione, conservati nella propria cronologia acquisti.

#### **Casi d'uso di estensione:**

- a. Emissione biglietto
- b. Visualizza cronologia acquisti

#### **Casi d'uso d'inclusione:**

- a. Ricerca corsa

Caso d'uso	Attori Primari	Attori Secondari	Estende
Ricerca corsa	Utente	-	-
Autenticazione dipendente	Dipendente	-	-
Autenticazione cliente	Cliente	-	-
Inserimento corsa	Dipendente	-	-
Modifica corsa	Dipendente	-	Ricerca corsa
Cancellazione corsa	Dipendente	-	Ricerca corsa
Acquisto biglietto	Cliente	Sistema di Pagamento	Ricerca corsa
Emissione biglietto	Dipendente	Stampante	Acquisto biglietto
Visualizzazione cronologia acquisti	Cliente	-	Acquisto biglietto

## 2.9 Diagramma dei casi d'uso

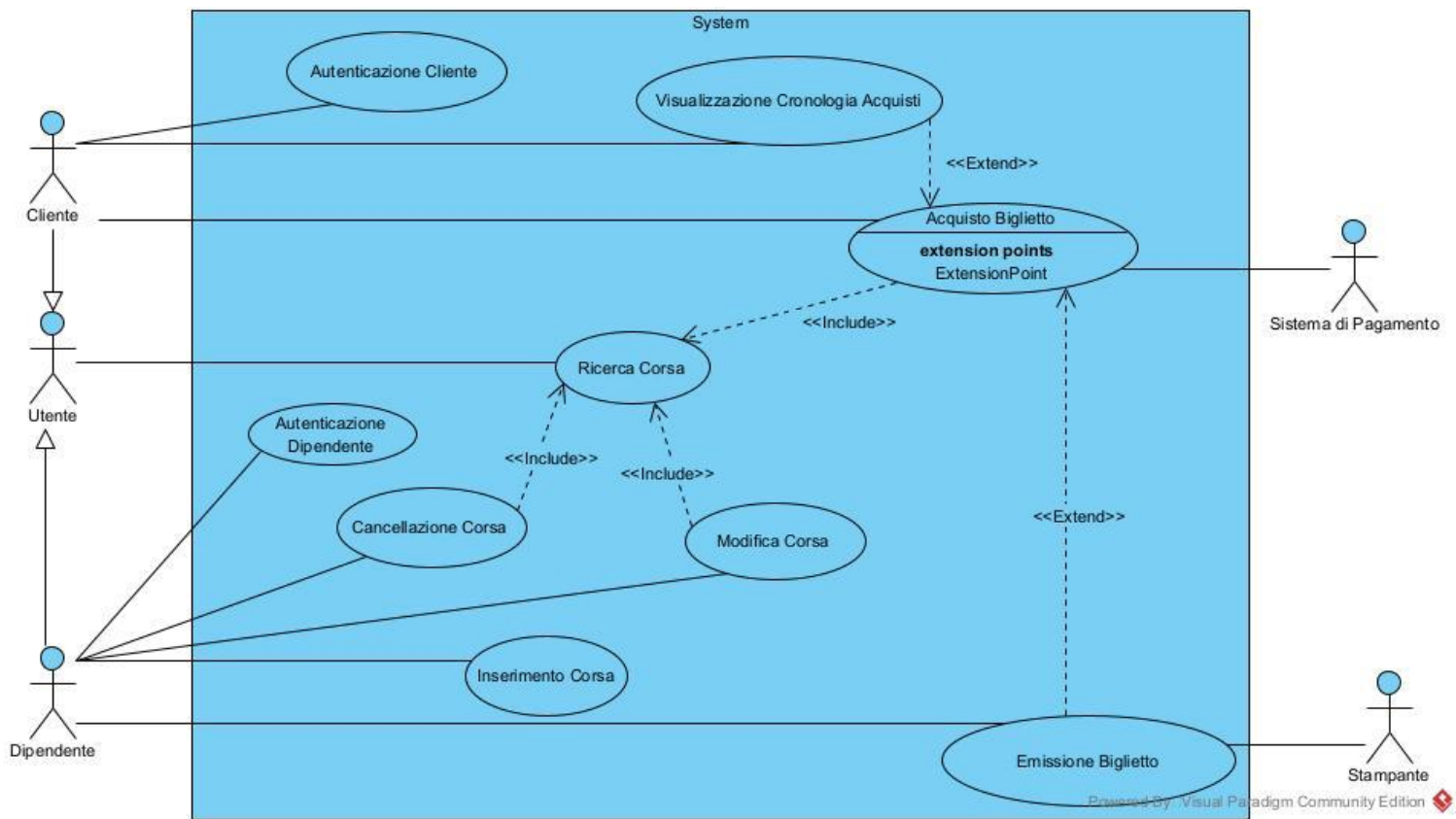


Figura 1 Use Case Diagram del Sistema POSEIDON

## 2.10 Scenari

Si riporta lo scenario principale del caso d'uso *EMISSIONE BIGLIETTO*.

<b>Caso d'uso:</b>	<b>EMISSIONE BIGLIETTO</b>
<b>Attore primario</b>	Dipendente
<b>Attori secondari</b>	Stampante
<b>Descrizione</b>	Un dipendente della compagnia emette un biglietto viaggio, valido per una specifica corsa in un determinato giorno, su richiesta di un cliente.
<b>Pre-Condizioni</b>	Il dipendente si è autenticato per l'accesso al sistema.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"><li>1. Il caso d'uso inizia quando il dipendente seleziona "emetti biglietto".</li><li>2. Il dipendente inserisce il tipo di biglietto da emettere (per passeggero o per autoveicolo).</li><li>3. Il sistema valuta l'effettiva disponibilità del posto in base ai biglietti emessi.</li><li>4. Il sistema inserisce nel database il nuovo biglietto con data e ora di emissione, codice della corsa a cui afferisce e codice dell'impiegato che lo ha emesso.</li><li>5. Se il biglietto è per un autoveicolo:<ol style="list-style-type: none"><li>a. Il dipendente inserisce la targa del veicolo.</li><li>b. Il sistema aggiorna il biglietto inserendo la targa del veicolo.</li></ol></li><li>6. Il sistema stampa il biglietto con data e ora di emissione e, se il biglietto è per un autoveicolo, targa del veicolo.</li><li>7. Il sistema rende disponibile il biglietto nella cronologia acquisti del cliente.</li><li>8. Il sistema aggiorna la disponibilità residua dei posti.</li></ol>
<b>Post-Condizioni</b>	L'attestato di acquisto è stato rilasciato, il biglietto è stato memorizzato nel database e la disponibilità dei posti per la corsa afferita è stata aggiornata.
<b>Casi d'uso correlati</b>	ACQUISTO BIGLIETTO
<b>Sequenza di eventi alternativi</b>	<ol style="list-style-type: none"><li>1. Se nessun posto risulta disponibile:<ol style="list-style-type: none"><li>a. Il sistema notifica l'utente dell'assenza di posti disponibili per la tipologia di biglietto prescelta.</li></ol></li><li>2. Se la stampante non funziona:<ol style="list-style-type: none"><li>a. Il sistema notifica al dipendente un warning con la mancata emissione e riprova dopo dieci minuti.</li></ol></li></ol>

Si riporta lo scenario principale del caso d'uso *INSERIMENTO CORSA*.

<b>Caso d'uso:</b>	<b>INSERIMENTO CORSA</b>
<b>Attore primario</b>	Dipendente
<b>Attori secondari</b>	<i>nessuno</i>
<b>Descrizione</b>	Inserimento di una nuova corsa
<b>Pre-Condizioni</b>	L'utente accede al sistema autenticandosi come dipendente
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"><li>1. Il caso d'uso inizia quando il dipendente seleziona "inserisci corsa".</li><li>2. Il dipendente controlla se il porto da usare per effettuare la corsa sia già memorizzato, altrimenti fa una richiesta di crearlo.</li><li>3. Il sistema effettua un controllo aggiuntivo prima di memorizzare il nuovo porto, altrimenti va avanti.</li><li>4. Il dipendente inserisce nel database la nuova corsa con: orario di partenza, orario di arrivo, prezzo, porto di partenza, porto di arrivo, nome della nave che effettuerà la corsa.</li></ol>

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

	<ol style="list-style-type: none"> <li>5. Il sistema effettua un op. di ricerca per controllare se il nome della nave è già nel database.</li> <li>6. Se la nave è già presente nel DB l'operazione si interrompe (dato che ad ogni nave corrisponde già una specifica corsa).</li> <li>7. Se la nave non è registrata nel DB: <ol style="list-style-type: none"> <li>a. la registra inserendo: nome, categoria.</li> </ol> </li> <li>8. Il sistema fa un controllo sulla categoria della nave: <ol style="list-style-type: none"> <li>a. Se la nave usata è un traghetto: <ol style="list-style-type: none"> <li>i. Richiede la capienza massima dei passeggeri e la capienza massima degli autoveicoli</li> </ol> </li> <li>b. Se la nave è un aliscafo: <ol style="list-style-type: none"> <li>i. Richiede la capienza massima dei passeggeri</li> </ol> </li> </ol> </li> <li>9. Il sistema aggiorna il database con la nuova corsa.</li> <li>10. Il sistema restituisce un messaggio di conferma per informare l'impiegato che l'inserimento è avvenuto con successo.</li> </ol>
<b>Post-Condizioni</b>	La corsa è stata memorizzata nel database ed è resa disponibile ai clienti.
<b>Casi d'uso correlati</b>	<i>nessuno</i>
<b>Sequenza di eventi alternativi</b>	<ol style="list-style-type: none"> <li>1. Se l'inserimento non termina correttamente: <ol style="list-style-type: none"> <li>a. Il sistema notifica al dipendente un warning con la mancata memorizzazione della corsa e richiede di ripetere l'operazione.</li> </ol> </li> </ol>

Si riporta lo scenario principale del caso d'uso *RICERCA CORSA*.

<b>Caso d'uso:</b>	<b>RICERCA CORSA</b>
<b>Attore primario</b>	Utente
<b>Attori secondari</b>	<i>nessuno</i>
<b>Descrizione</b>	Un utente effettua una ricerca di una corsa presente in un determinato giorno e da un determinato porto in una fascia di prezzo personalizzata.
<b>Pre-Condizioni</b>	La corsa deve essere stata memorizzata e resa disponibile agli utenti
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando il dipendente/cliente seleziona "ricerca corsa".</li> <li>2. L'utente inserisce i filtri relativi alla corsa che si sta cercando: codice della nave, porto e orario di partenza e/o di arrivo e il prezzo del biglietto.</li> <li>3. Il sistema presenta all'utente uno o più risultati. <ol style="list-style-type: none"> <li>a. Per ogni corsa trovata <ol style="list-style-type: none"> <li>i. Il sistema mostra il prezzo base della corsa;</li> <li>ii. Il sistema mostra le informazioni relative ai porti;</li> <li>iii. Il sistema mostra le informazioni relative agli orari.</li> </ol> </li> </ol> </li> <li>4. Il sistema fornirà a schermo la lista filtrata.</li> </ol>
<b>Post-Condizioni</b>	La corsa è stata trovata ed è disponibile all'acquisto da parte del cliente o alla revisione da parte del dipendente.
<b>Casi d'uso correlati</b>	ACQUISTO BIGLIETTO, MODIFICA CORSA, CANCELLAZIONE CORSA
<b>Sequenza di eventi alternativi</b>	<ol style="list-style-type: none"> <li>1. Se nessuna corsa risulta disponibile: <ol style="list-style-type: none"> <li>a. Il sistema notifica l'utente dell'assenza di corse disponibili che rientrano nei filtri utilizzati.</li> </ol> </li> </ol>

Si riporta lo scenario principale del caso d'uso *ACQUISTA BIGLIETTO*.

<b>Caso d'uso:</b>	<b>ACQUISTA BIGLIETTO</b>
--------------------	---------------------------

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

<b>Attore primario</b>	Cliente Autenticato
<b>Attori secondari</b>	Sistema di Pagamento
<b>Descrizione</b>	Un utente procede con l'acquisto di un biglietto viaggio valido per una specifica corsa in un determinato giorno.
<b>Pre-Condizioni</b>	Il cliente/dipendente si è autenticato per l'accesso al sistema.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Include (RICERCA CORSA).</li> <li>2. Il caso d'uso inizia quando l'utente seleziona "acquista biglietto".</li> <li>3. L'utente seleziona la tipologia di biglietto che si vuole acquistare (per passeggero o per autoveicolo).</li> <li>4. Il sistema valuta l'effettiva disponibilità del posto in base ai biglietti emessi.</li> <li>5. L'utente fornisce le informazioni anagrafiche.</li> <li>6. Il cliente procede attraverso un sistema di pagamento esterno, fornendo la metodologia di pagamento.</li> <li>7. Il sistema procede con l'elaborazione dell'acquisto ritornando una ricevuta: <ul style="list-style-type: none"> <li>Punto di estensione: (VISUALIZZA CRONOLOGIA ACQUISTI)</li> <li>Punto di estensione: (EMISSIONE BIGLIETTO)</li> </ul> </li> </ol>
<b>Post-Condizioni</b>	L'acquisto è stato effettuato e inserito in cronologia e il biglietto è pronto per l'emissione.
<b>Casi d'uso correlati</b>	RICERCA CORSA, EMISSIONE BIGLIETTO, VISUALIZZA CRONOLOGIA ACQUISTI
<b>Sequenza di eventi alternativi</b>	<ol style="list-style-type: none"> <li>1. Se nessun posto risulta disponibile: <ol style="list-style-type: none"> <li>a. Il sistema notifica l'utente dell'assenza di posti disponibili per la tipologia di biglietto prescelta.</li> </ol> </li> <li>2. Se l'elaborazione dell'acquisto fallisce: <ol style="list-style-type: none"> <li>a. Il sistema notifica l'utente del fallimento di acquisto.</li> </ol> </li> </ol>

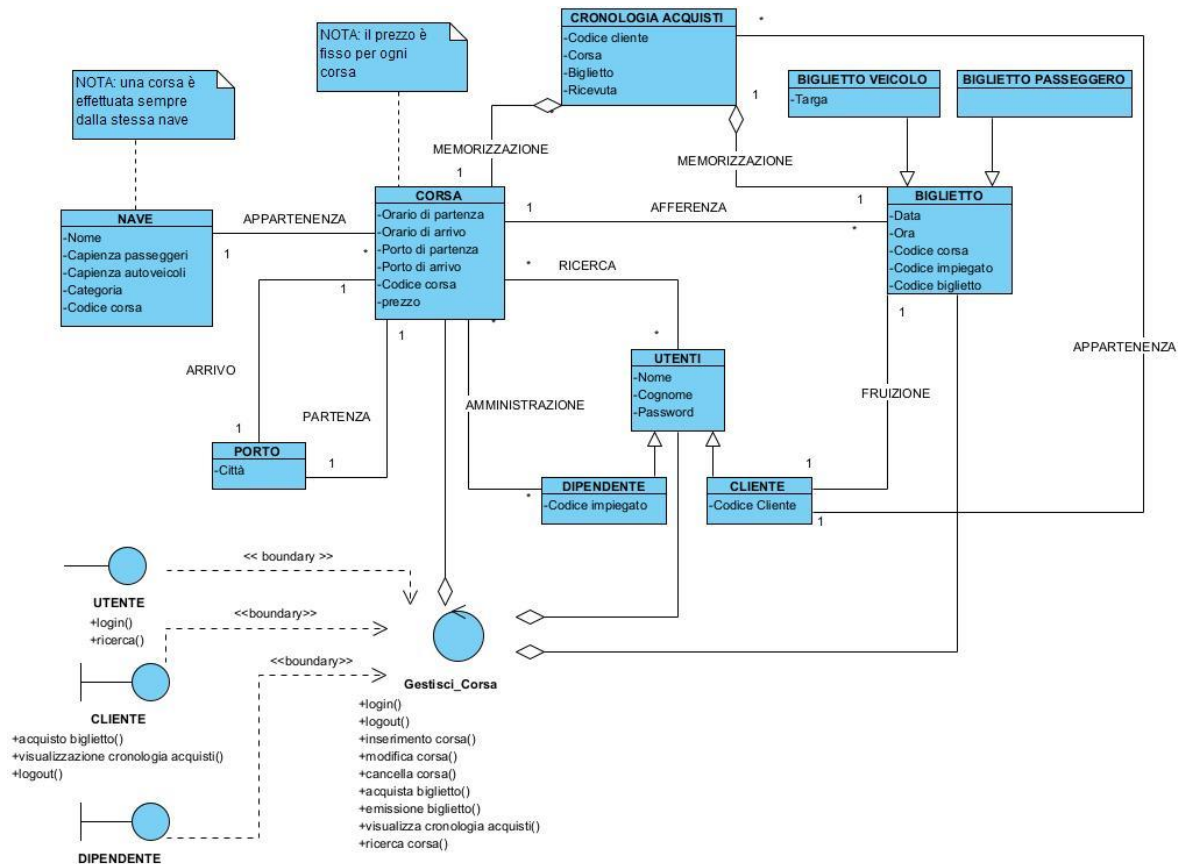
## 2.11 Diagramma delle classi

Dopo aver individuato i casi d'uso del sistema, si passa ad un'analisi singolare e iterativa di ogni singolo caso d'uso con l'obiettivo di produrre:

- Le **classi di analisi**, esse modellano i concetti chiave di entità e relazioni tra esse;
- La **realizzazione dei casi d'uso**, essa mostra le interazioni tra le classi di analisi con la finalità di mostrare il comportamento del sistema preso in considerazione.

Questo insieme di documenti costituisce il **modello di analisi del sistema**.

Nella fase di analisi, il **diagramma delle classi** è usato per fornire una rappresentazione delle entità del mondo reale presenti nel dominio di interesse e cattura le relazioni tra di esse.



*Figura 2 Class Diagram di Prima Analisi*

2.12 Diagrammi di sequenza di prima analisi

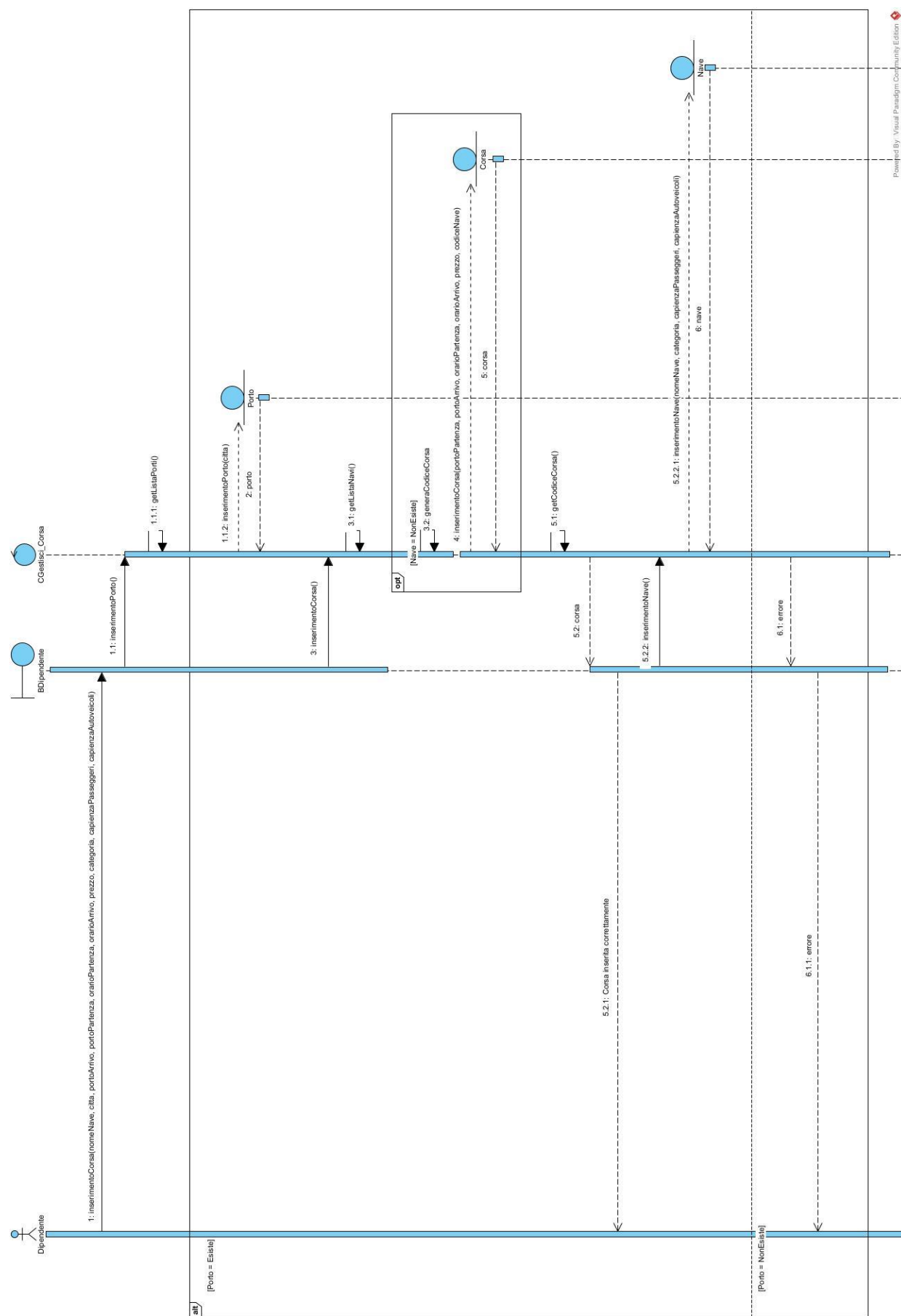


Figura 3 Sequence Diagram di Analisi del caso d'uso Inserimento Corsa



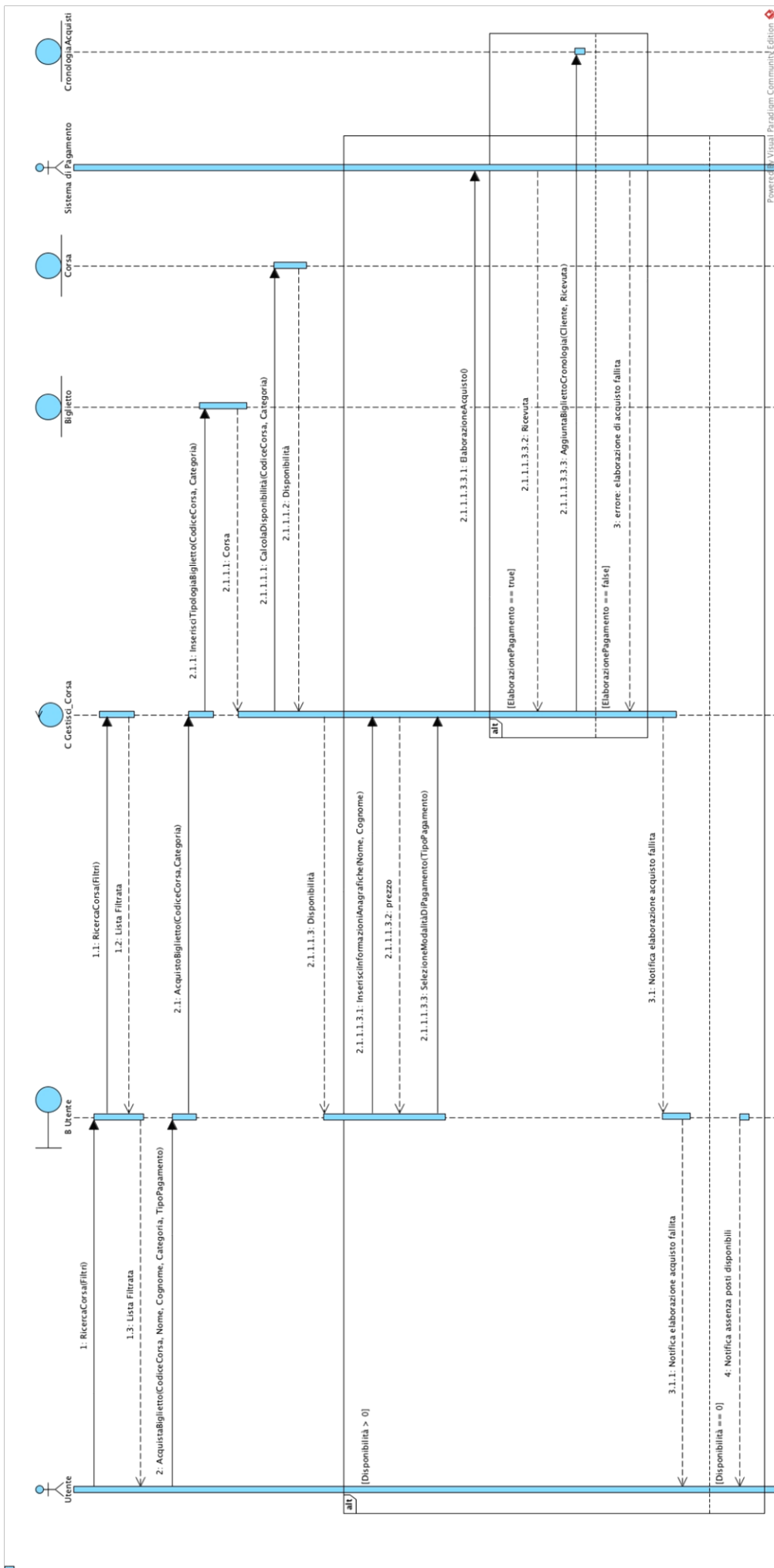


Figura 4 Sequence Diagram di Analisi del caso d'uso Acquista Biglietto

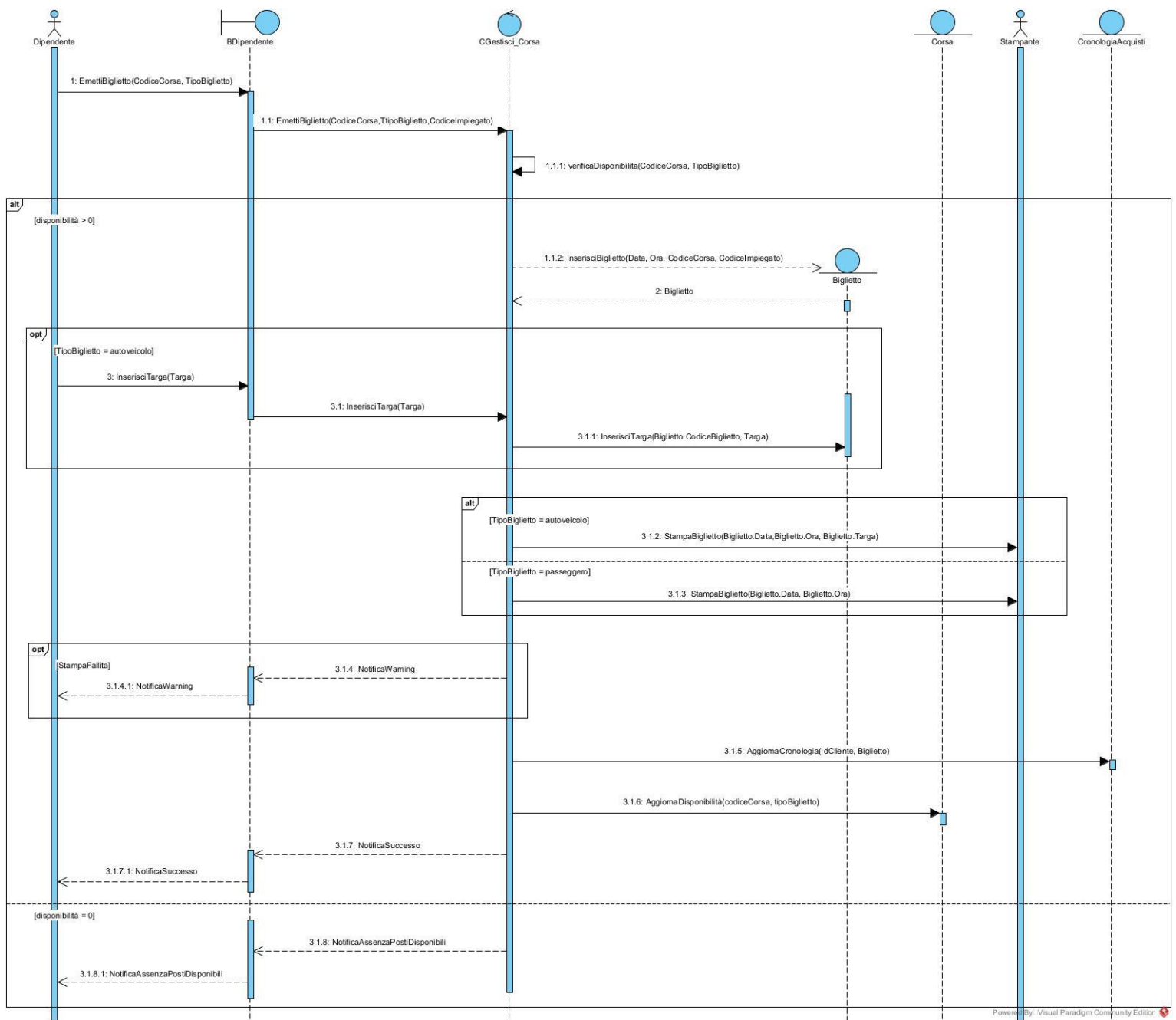


Figura 5 Sequence Diagram di Analisi del caso d'uso Emissione Biglietto

### 3. Stima dei costi

#### VALUTAZIONE DEGLI UFP

- Determinazione del tipo di conteggio:  
Conteggio per sviluppo di progetto
- Identificazione dell'ambito di conteggio e dei confini applicativi + modello ER:

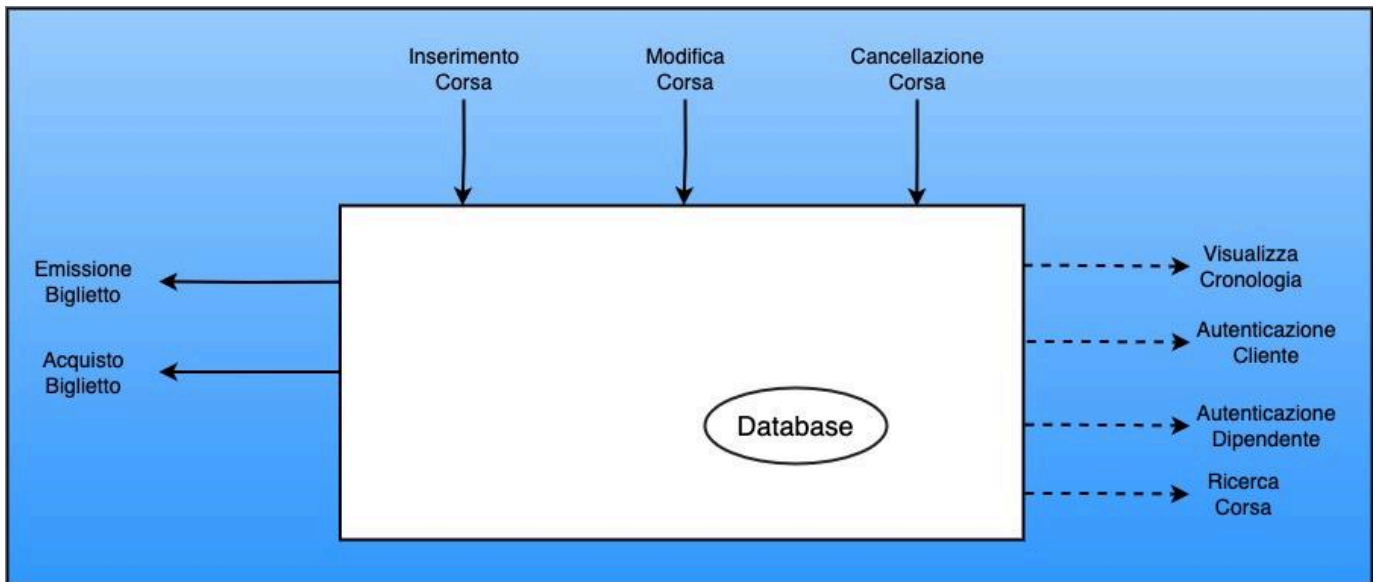


Figura 6 Diagramma dei confini applicativi

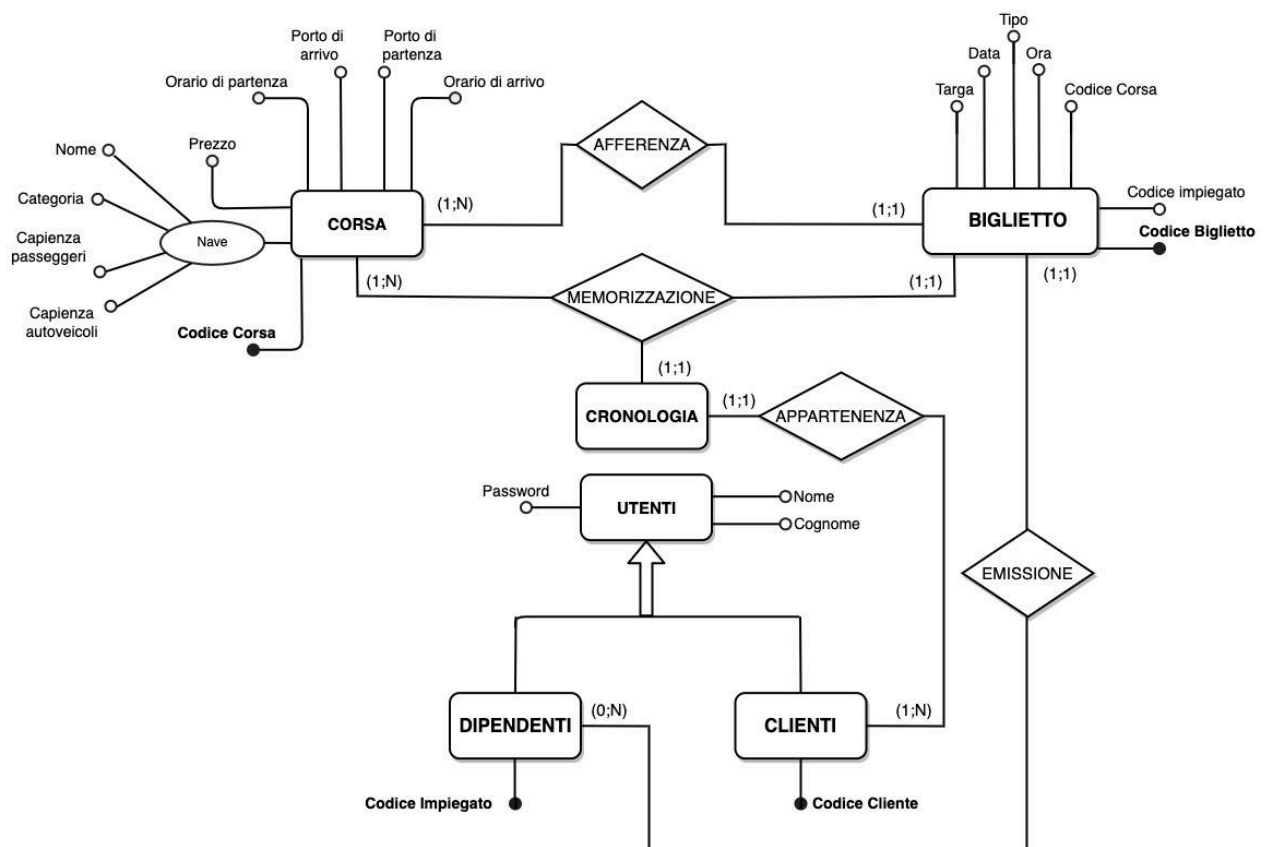


Figura 7 Modello E-R

c. Identificazione delle funzioni dati:

DATI

- **ILF: 1**

Database (5 RET: corsa, dipendenti, clienti, biglietto, cronologia / 39 DET: 10 attributi di corsa + 4 attributi di dipendenti + 4 attributi di clienti + 7 attributi di biglietto + 14 attributi di cronologia)

d. Identificazione delle funzioni transazionali:

TRANSAZIONI

- **EI: 3**

Inserimento Corsa (1 FTR: corsa / 10 DET: i dieci attributi di corsa)

Modifica Corsa (1 FTR: corsa / 10 DET: i dieci attributi di corsa)

Cancellazione Corsa (1 FTR: corsa / 1 DET: messaggio di conferma)

- **EO: 2**

Emissione Biglietto (5 FTR: corsa, dipendenti, clienti, biglietto, cronologia) / 14 DET: i sette attributi di biglietto + sette attributi di cronologia escludendo le ripetizioni)

Acquisto Biglietto (3 FTR: corsa, clienti, cronologia) / 14 DET: i quattordici attributi di cronologia)

- **EQ: 4**

Ricerca Corsa (1 FTR: corsa / 10 DET: i dieci attributi di corsa)

Visualizza Cronologia Acquisti (1 FTR: cronologia / 14 DET: i quattordici attributi di cronologia)

Autenticazione dipendente (1 FTR: dipendenti / 4 DET: i quattro attributi di dipendenti)

Autenticazione cliente (1 FTR: clienti / 4 DET: i quattro attributi di clienti)

e. Conteggio UFP:

DESCRIZIONE	TIPO	DET	RET/FTR	COMPLESSITÀ	VALORE
<b>Database</b>	ILF	39	5	Media	10
<b>Inserimento Corsa</b>	EI	10	1	Bassa	3
<b>Modifica Corsa</b>	EI	10	1	Bassa	3
<b>Cancellazione Corsa</b>	EI	1	1	Bassa	3
<b>Emissione Biglietto</b>	EO	14	5	Alta	7
<b>Acquisto Biglietto</b>	EO	14	3	Media	5
<b>Ricerca Corsa</b>	EQ	10	1	Bassa	3
<b>Visualizza Cronologia Acquisti</b>	EQ	14	1	Bassa	3
<b>Autenticazione dipendente</b>	EQ	4	1	Bassa	3
<b>Autenticazione cliente</b>	EQ	4	1	Bassa	3

VALUTAZIONE DEI FATTORI DI AGGIUSTAMENTO

- Fattori correttivi: adegua il valore ottenuto dalla valutazione degli UFP

## FATTORI CORRETTIVI

CARATTERISTICHE GENERALI	VALORE
Comunicazione dati	3
Distribuzione elaborazione	3
Prestazioni	1
Utilizzo intensivo configurazione	2
Frequenza delle transazioni	1
Inserimento dati interattivo	2
Efficienza per l'utente finale	4
Aggiornamento interattivo	4
Complessità elaborativa	3
Riusabilità	4
Facilità di installazione	0
Facilità di gestione operativa	1
Molteplicità dei siti	3
Facilità di modifica	1

- Si ottiene il conteggio finale:

$$FP = UFP \times (0.65 + 0.01 \times \sum_{i=1}^{14} F_i) = 41.71 \approx 42.$$

Il numero di linee di codice stimato se l'applicazione è sviluppata in C:  $42 \cdot 128 = 5376$

Il numero di linee di codice stimato se l'applicazione è sviluppata in Java:  $42 \cdot 6 = 252$

## 4. Piano di test funzionale

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

### PARAMETRI E CATEGORIE

- **Biglietto**
  - Codice Impiegato
  - Codice Corsa
  - Tipo Biglietto
  - Targa
- **Cronologia d’acquisto**
  - Generazione nuova ricevuta
  - Codice Cliente
  - Ricevuta

Codice Impiegato	Codice Corsa	Tipo Biglietto	Targa
<ul style="list-style-type: none"><li>• 1001</li><li>• 1002</li><li>• Valido &gt; 1002 [SINGLE]</li><li>• &lt;= 0 [ERROR]</li><li>• valido ma non presente nel DB [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• codice di un traghetto [TRA]</li><li>• codice di un aliscafo</li><li>• 103 [SINGLE]</li><li>• &lt;= 0 [ERROR]</li><li>• valido ma non presente nel DB [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• veicolo [if TRA] [VE]</li><li>• passeggero</li><li>• invalido [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valida</li><li>• invalida [if VE] [ERROR]</li></ul>

Generazione nuova ricevuta	Codice Cliente	Ricevuta
<ul style="list-style-type: none"><li>• Genera</li><li>• Non generare [NG]</li></ul>	<ul style="list-style-type: none"><li>• Valido</li><li>• &lt;= 0 [ERROR]</li><li>• valido ma non presente nel DB [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valida</li><li>• invalida [if NG] [ERROR]</li><li>• valida ma non presente nel DB [if NG] [ERROR]</li></ul>

**Osservazione:** i vincoli SINGLE sono stati scelti dopo un (ipotetico) confronto con il committente, che ha specificato che gli impiegati addetti all'operazione di emissione del biglietto sono prevalentemente due (codice 1001 e 1002) e la corsa con codice 103 (effettuata da un traghetto) viene selezionata raramente.

Numero di casi di test senza vincoli (prendendo un caso di test per test case specification):  
 $5 \times 5 \times 3 \times 2 \times 2 \times 3 \times 3 = 2700$ .

Numero di casi di test con vincoli errore e single (prendendo un caso di test per test case specification):  $(2 \times 2 \times 2 \times 1 \times 2 \times 1 \times 1) + 12 = 28$ .

Numero di casi di test con vincolo property [TRA]:  $(2 \times 2 \times 1 \times 1 \times 2 \times 1 \times 1) + 4 + 12 = 24$ .  
 Equivalentemente:  $(2 \times 1 \times (1+2) \times 1 \times 2 \times 1 \times 1) + 12 = 24$ .

**Osservazione:** dato che i restanti vincoli property sono legati a tre casi di errore, anche se li considerassi non diminuirebbe il numero di casi di test. Uso questi vincoli property per preparare adeguatamente i casi di test relativi (per testare la targa invalida, il tipo del biglietto deve essere "veicolo"; per testare la ricevuta invalida e la ricevuta valida ma non presente nel database, non deve essere generata una nuova ricevuta).

## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Test input validi #1 (property TRA)	I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 1, codice corsa 101, codice impiegato 1001 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato
2	Test input validi #2 (property TRA)	II classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono	- 1002 - 101 - Veicolo	Biglietto veicolo con codice biglietto 2, codice corsa 101, codice impiegato 1002 e targa	Il database è stato correttamente aggiornato

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe I classe I classe I classe	cronologie nel database	- AA000AA - Genera - 10001 - 10001VEI1019UO	AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	
3	Test input validi #3	I classe I classe Il classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto passeggero con codice biglietto 3, codice corsa 101 e codice impiegato 1001, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato
4	Test input validi #4	Il classe I classe Il classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1002 - 101 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto passeggero con codice biglietto 4, codice corsa 101 e codice impiegato 1002, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato
5	Test input validi #5	I classe Il classe Il classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 102 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto passeggero con codice biglietto 1, codice corsa 102 e codice impiegato 1001, cronologia inserita con codice corsa 102 e codice cliente 10001	Il database è stato correttamente aggiornato
6	Test input validi #6	Il classe Il classe Il classe Il classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1002 - 102 - Passeggero - AA000AA - Genera - 10001	Biglietto passeggero con codice biglietto 2, codice corsa 102 e codice impiegato 1002, cronologia inserita con codice corsa 102 e codice cliente 10001	Il database è stato correttamente aggiornato

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO



		I classe		- 10001VEI1019UO		
<b>7</b>	Test input validi #7 (property TRA)	I classe I classe I classe I classe II classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001VEI1019UO) è già stata inserita	- 1001 - 101 - Veicolo - AA000AA - Non generare - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 5, codice corsa 101, codice impiegato 1001 e targa AA000AA, cronologia aggiornata con codice impiegato 1001	Il database è stato correttamente aggiornato
<b>8</b>	Test input validi #8 (property TRA)	II classe I classe I classe I classe II classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001VEI1019UO) è già stata inserita	- 1002 - 101 - Veicolo - AA000AA - Non generare - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 6, codice corsa 101, codice impiegato 1002 e targa AA000AA, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente aggiornato
<b>9</b>	Test input validi #9	I classe I classe II classe I classe II classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001PAS1019UO) è già stata inserita	- 1001 - 101 - Passeggero - AA000AA - Non generare - 10001 - 10001PAS1019UO	Biglietto passeggero con codice biglietto 7, codice corsa 101 e codice impiegato 1001, cronologia aggiornata con codice impiegato 1001	Il database è stato correttamente aggiornato
<b>10</b>	Test input validi #10	II classe I classe II classe I classe II classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta	- 1002 - 101 - Passeggero - AA000AA - Non generare - 10001	Biglietto passeggero con codice biglietto 8, codice corsa 101 e codice impiegato 1002, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente aggiornato

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe	10001PAS1019UO) è già stata inserita	- 10001PAS1019UO		
<b>11</b>	Test input validi #11	I classe II classe II classe I classe II classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 102, ricevuta 10001PAS1029UO) è già stata inserita	- 1001 - 102 - Passeggero - AA000AA - Non generare - 10001 - 10001PAS1029UO	Biglietto passeggero con codice biglietto 3, codice corsa 102 e codice impiegato 1001, cronologia aggiornata con codice impiegato 1001	Il database è stato correttamente aggiornato
<b>12</b>	Test input validi #12	II classe II classe II classe II classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 102, ricevuta 10001PAS1029UO) è già stata inserita	- 1002 - 102 - Passeggero - AA000AA - Non generare - 10001 - 10001PAS1029UO	Biglietto passeggero con codice biglietto 4, codice corsa 102 e codice impiegato 1002, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente aggiornato
<b>13</b>	Test [SINGLE] #1	III classe I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1003 - 101 - Veicolo - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 9, codice corsa 101, codice impiegato 1003 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato
<b>14</b>	Test [SINGLE] #2	I classe III classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 103 - Veicolo - AA000AA - Genera - 10001	Biglietto veicolo con codice biglietto 1, codice corsa 103, codice impiegato 1001 e targa AA000AA, cronologia inserita con codice corsa 103 e codice cliente 10001	Il database è stato correttamente aggiornato

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe		- 10001VEI1019UO		
15	Test [ERROR] #1	IV classe I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- -1 - 101 - Veicolo - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto = null. Errore: il codice impiegato deve essere maggiore di 0	Nessuna modifica al database
16	Test [ERROR] #2	V classe I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 5000 - 101 - Veicolo - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto = null. Errore: l'impiegato selezionato non esiste	Nessuna modifica al database
17	Test [ERROR] #3	I classe IV classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - -1 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto = null. Errore: il codice corsa deve essere maggiore di 0	Nessuna modifica al database
18	Test [ERROR] #4	I classe V classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 500 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto = null. Errore: la corsa selezionata non esiste	Nessuna modifica al database
19	Test [ERROR] #5	I classe I classe	Il dipendente ha effettuato il login.	- 1001 - 101	Biglietto = null. Errore: il tipo di biglietto	Nessuna modifica al database

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		III classe I classe I classe I classe I classe	Non ci sono cronologie nel database	- Animale - AA000AA - Genera - 10001 - 10001VEI1019UO	inserito non è valido	
20	Test [ERROR] #6	I classe I classe I classe II classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - null - Genera - 10001 - 10001VEI1019UO	Biglietto = null. Errore: Errore: necessario inserire la targa	Nessuna modifica al database
21	Test [ERROR] #7	I classe I classe I classe I classe I classe II classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - AA000AA - Genera - -1 - 10001VEI1019UO	Biglietto = null. Errore: il codice cliente deve essere maggiore di 0	Nessuna modifica al database
22	Test [ERROR] #8	I classe I classe I classe I classe I classe III classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - AA000AA - Genera - 50000 - 10001VEI1019UO	Biglietto = null. Errore: il cliente selezionato non esiste	Nessuna modifica al database
23	Test [ERROR] #9	I classe I classe I classe I classe II classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101,	- 1001 - 101 - Veicolo - AA000AA - Non generare	Biglietto = null. Errore: necessario inserire la ricevuta	Nessuna modifica al database

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe II classe	ricevuta 10001VEI1019UO) è già stata inserita	- 10001 - null		
<b>24</b>	Test [ERROR] #10	I classe I classe I classe I classe II classe I classe III classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001VEI1019UO) è già stata inserita	- 1001 - 101 - Veicolo - AA000AA - Non generare - 10001 - 50001VEI1019UO	Biglietto = null. Errore: l'acquisto selezionato non esiste	Nessuna modifica al database

## PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Inserimento Corsa*”.

L’inserimento corsa si articola di 3 funzionalità testabili indipendentemente:

- ◆ Inserimento Corsa
- ◆ Inserimento Nave
- ◆ Inserimento Porto

In quanto per la creazione di una Corsa è necessaria una corrispondenza con una Nave e un Porto.

### INSERIMENTO CORSA

#### PARAMETRI E CATEGORIE

- **Corsa**
  - Porto Partenza
  - Porto Arrivo
  - Orario Partenza
  - Orario Arrivo
  - Prezzo
- **Nave**
  - Nome Nave

Porto Partenza	Porto Arrivo	Orario Partenza	Orario Arrivo	Prezzo	Nome Nave
<ul style="list-style-type: none"><li>• valido (<math>\leq 50</math>)</li><li>• non valido (<math>&gt; 50</math> caratteri) [ERROR]</li><li>• non valido (non esiste) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valido (<math>\leq 50</math>)</li><li>• non valido (<math>&gt; 100</math> caratteri) [ERROR]</li><li>• non valido (non esiste) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• formato e valori validi</li><li>• non valido (valori non validi) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• formato e valori validi</li><li>• non valido (valori non validi) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• formato e valori validi</li><li>• non valido (<math>&gt; 200</math>) [ERROR]</li><li>• non valido (<math>&lt; 5.00</math>) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valido (<math>\leq 50</math>)</li><li>• non valido (<math>&gt; 50</math> caratteri) [ERROR]</li></ul>

Numero di casi di test senza vincoli (prendendo un caso di test per test case specification):

$$3 * 3 * 2 * 2 * 3 * 2 = 216$$

Numero di casi di test con vincoli Errore (prendendo un caso di test per test case specification):

$$(1 * 1 * 1 * 1 * 1 * 1) + 9 = 10$$

## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Test input validi	I classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse ne navi nel database.	- Napoli - Ischia - 10:15 - 13:40 - 20.00 - Estate	Corsa con porto di partenza Napoli, porto di arrivo Ischia, orario di partenza 10:15, orario di arrivo 13:40, costo della corsa 20.00, nome della nave che effettua la corsa Estate con codice corsa 1.	Il database è stato correttamente aggiornato.
2	Test [ERROR] #1	II classe I classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Naaaaaaaaaappppppppppooooo oooooIIIIIIIIIIIIIIIIIIII ii - Ischia - 10:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.
3	Test [ERROR] #2	III classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Milano - Ischia - 10:15 - 13:40 - 20.00	Corsa = null. Errore: il porto inserito non è registrato.	Nessuna modifica al database.

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe		- Estate		
4	Test [ERROR] #3	I classe II classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - lsssssssscccccc ccchhhhhhhhhii iiiiiiiiaaaaaaaaaa aaaaa - 10:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.
5	Test [ERROR] #4	I classe III classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Milano - 10:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: il porto inserito non è registrato.	Nessuna modifica al database.
6	Test [ERROR] #5	I classe I classe II classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 15:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: orario inserito non è valido.	Nessuna modifica al database.
7	Test [ERROR] #6	I classe I classe I classe II classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15 - 40:40 - 20.00 - Estate	Corsa = null. Errore: valore non valido. <i>*Vedere Osservazione*</i>	Nessuna modifica al database.
8	Test [ERROR] #7	I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO



		I classe II classe I classe		- 13:40 - -1.00 - Estate		
9	Test [ERROR] #8	I classe I classe I classe I classe III classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15 - 13:40 - 250.00 - Estate	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.
10	Test [ERROR] #9	I classe I classe I classe I classe I classe II classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15 - 13:40 - 20.00 - Coooooooooooo sssssttttttttttt taaaaaaaaaaaaaa aaaaaaaaaaaaaaaa	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.

**Osservazione:** non occorre un caso di testing specifico per la gestione di questo tipo di error, in quanto il tipo di variabile definito 'LocalTime' effettua già un controllo per la validità del formato e del range di valori inseriti (non oltre 24 ore).

## INSERIMENTO NAVE

### PARAMETRI E CATEGORIE

- **Nave**
  - Nome Nave
  - Categoria
  - Capienza Passeggeri
  - Capienza Autoveicoli

Nome Nave	Categoria	Capienza Passeggeri	Capienza Autoveicoli	Codice Nave
<ul style="list-style-type: none"><li>• valido (<math>\leq 50</math>)</li><li>• non valido (<math>&gt; 50</math> caratteri) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• traghetto</li><li>• aliscafo</li><li>• non valido (ERROR)</li></ul>	<ul style="list-style-type: none"><li>• valido</li><li>• non valido (<math>&gt; 400</math>) [ERROR]</li><li>• non valido (valori non validi) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valido</li><li>• non valido (<math>&gt; 100</math>) [ERROR]</li><li>• non valido (valori non validi) [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valido</li><li>• non valido (<math>\leq 0</math>) [ERROR]</li></ul>

Numero di casi di test senza vincoli (prendendo un caso di test per test case specification):

$$2 * 3 * 3 * 3 * 2 = 108$$

Numero di casi di test con vincoli Errore (prendendo un caso di test per test case specification):

$$(1 * 2 * 1 * 1 * 1) + 7 = 9$$

## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Test input validi #1	I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 270	Nave di tipo traghetto, con nome Estate, capienza passeggeri 270, capienza	Il database è stato correttamente aggiornato.

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe I classe		- 30 - 101	autoveicoli 30, con codice corsa 101.	
2	Test input validi #2	I classe II classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Aliscafo - 270 - 102	Nave di tipo aliscafo, con nome Estate, e capienza passeggeri 270 con codice corsa 102.	Il database è stato correttamente aggiornato.
3	Test [ERROR] #1	II classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Coooooooooooo sssssttttttttttt taaaaaaaaaaaaaa aaaaaaaaaaaaaaa - Traghetto - 270 - 30 - 101	Nave = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.
4	Test [ERROR] #2	I classe III classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Battello - 270 - 30 - 101	Nave = null. Errore: valor inseriti non valido.	Nessuna modifica al database.
5	Test [ERROR] #3	I classe I classe II classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 500 - 30 - 101	Nave = null. Errore: valore non valido (max 400).	Nessuna modifica al database.
6	Test [ERROR] #4	I classe I classe III classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - -1 - 30 - 101	Nave = null. Errore: valore non valido.	Nessuna modifica al database.
7	Test [ERROR] #5	I classe	Il dipendente ha fatto il	- Estate	Nave = null.	Nessuna modifica

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe I classe II classe I classe	login(). Non ci sono navi nel database.	- Traghetto - 270 - 150 - 101	Errore: valore non valido (max 100).	al database.
8	Test [ERROR] #6	I classe I classe I classe III classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 270 - -1 - 101	Nave = null. Errore: valore non valido.	Nessuna modifica al database.
9	Test [ERROR] #7	I classe I classe I classe I classe II classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 270 - 30 - -1	Nave = null. Errore: il codice corsa deve essere > 0.	Nessuna modifica al database.

## INSERIMENTO PORTO

### PARAMETRI E CATEGORIE

- **Porto**
  - Città

#### Città

- valido
- non valido (> 50) [ERROR]
- non valido (non valida) [ERROR]

Numero di casi di test senza vincoli (prendendo un caso di test per test case specification): 3

Numero di casi di test con vincoli Errore (prendendo un caso di test per test case specification): 3

## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Test input validi	I classe	Il dipendente ha fatto il login(). Non ci sono porti nel database.	- Pozzuoli	Porto con città Pozzuoli.	Il database è stato correttamente aggiornato.
2	Test [ERROR] #1	Il classe	Il dipendente ha fatto il login(). Non ci sono porti nel database.	- Pooooooooooooo oooooooozzzzzzzzzz zzzzzzzzzzzzzzzzzz zzzzzzzzzzuuuuuuu uuuuuuooooooooo ooooooooooooooooo	Porto = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

<b>3</b>	Test [ERROR] #2	III classe	Il dipendente ha fatto il login(). Non ci sono porti nel database.	- Pozzuoli	Corsa = null. Errore: il porto inserito è già esistente.	Nessuna modifica al database.

## PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*ACQUISTO BIGLIETTO*”.

### PARAMETRI E CATEGORIE

- **Cliente**
  - Nome
  - Cognome
- **Corsa**
  - Codice Corsa
- **Biglietto**
  - Tipo Biglietto
  - Targa

Codice Corsa	Codice Cliente	Tipo Biglietto
<ul style="list-style-type: none"><li>• 101 [TRA]</li><li>• 102</li><li>• Valido &gt; 102 [SINGLE]</li><li>• &lt;= 0 [ERROR]</li><li>• valido ma non presente nel DB [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Valido</li><li>• &lt;= 0 [ERROR]</li><li>• valido ma non presente nel DB [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• veicolo [if TRA] [VE]</li><li>• passeggero</li><li>• invalido [ERROR]</li></ul>

Targa	Nome	Cognome	Tipologia Pagamento
<ul style="list-style-type: none"><li>• valida</li><li>• invalida [if VE] [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valido</li><li>• null [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valido</li><li>• null [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• valido</li><li>• null [ERROR]</li></ul>

**Osservazione:** I vincoli SINGLE sono stati scelti dopo un confronto con il committente (ipotetico) che ha specificato che le corse prevalentemente acquistate sono due (codice 101 e 102).

Numero di casi di test senza vincoli (prendendo un caso di test per test case specification):

$$5 * 3 * 2 * 3 * 2 * 2 = 720$$

Numero di casi di test con vincoli errore e single (prendendo un caso di test per test case specification):

$$(2 * 1 * 2 * 1 * 1 * 1 * 1) + 10 = 14$$

Numero di casi di test con vincolo property [TRA]:

$$(2 * 1 * 1 * 1 * 1 * 1 * 1) + 1 + 10 = (1 * 1 * (1 + 2) * 1 * 1 * 1 * 1) + 10 = 13$$

$$(2 * 2 * 1 * 1 * 2 * 1 * 1) + 4 + 12 = 24.$$

$$\text{Equivalentemente: } (2 * 1 * (1+2) * 1 * 2 * 1 * 1) + 12 = 24.$$

**Osservazione:** Il restante vincolo property è legato a un caso di errore; quindi, anche se considerato non diminuisce il numero di casi di test. Utilizzo i vincoli property per preparare i casi di test relativi (la *corsa* deve essere percorsa da un traghetto per poter ospitare veicoli e quindi permettere l'acquisto di biglietti di *tipo veicolo*).

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	TEST INPUT VALIDI #1 [PROPERTY TRA]	I classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto veicolo, codice corsa 101 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato
2	TEST INPUT VALIDI #2	I classe I classe II classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 passeggero AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto passeggero, codice corsa 101 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO



3	TEST INPUT VALIDI #3	II classe I classe II classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	102 10002 passeggero AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto passeggero, codice corsa 102 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato
4	TEST [SINGLE] #1	III classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	103 10001 veicolo AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto veicolo, codice corsa 103 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato
5	TEST [ERRORE] #1	III classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	-1 10001 veicolo AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il codice corsa deve essere > 0.	Nessuna modifica al database
6	TEST [ERRORE] #2	IV classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	500 10001 veicolo AA000AA Leonardo Catello contante	Ricevuta = null. Errore: la corsa selezionata non esiste.	Nessuna modifica al database

<b>7</b>	TEST [ERRORE] #3	I classe III classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 -1 veicolo AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il codice cliente deve essere > 0.	Nessuna modifica al database
<b>8</b>	TEST [ERRORE] #4	I classe IV classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 50000 veicolo AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il cliente selezionato non esiste.	Nessuna modifica al database
<b>9</b>	TEST [ERRORE] #5	I classe I classe III classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 null AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il tipo di biglietto inserito non è valido.	Nessuna modifica al database
<b>10</b>	TEST [ERRORE] #6	I classe I classe I classe II classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo null Leonardo Catello contante	Ricevuta = null. Errore: necessario inserire la targa.	Nessuna modifica al database

<b>11</b>	TEST [ERRORE] #7	I classe I classe I classe I classe II classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA null Catello contante	Ricevuta = null. Errore: necessario inserire il nome.	Nessuna modifica al database
<b>12</b>	TEST [ERRORE] #8	I classe I classe I classe I classe II classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA Leonardo null contante	Ricevuta = null. Errore: necessario inserire il cognome.	Nessuna modifica al database
<b>13</b>	TEST [ERRORE] #9	I classe I classe I classe I classe I classe II classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA Leonardo Catello null	Ricevuta = null. Errore: necessario inserire la tipologia del pagamento.	Nessuna modifica al database



## 5.2 Diagrammi di sequenza

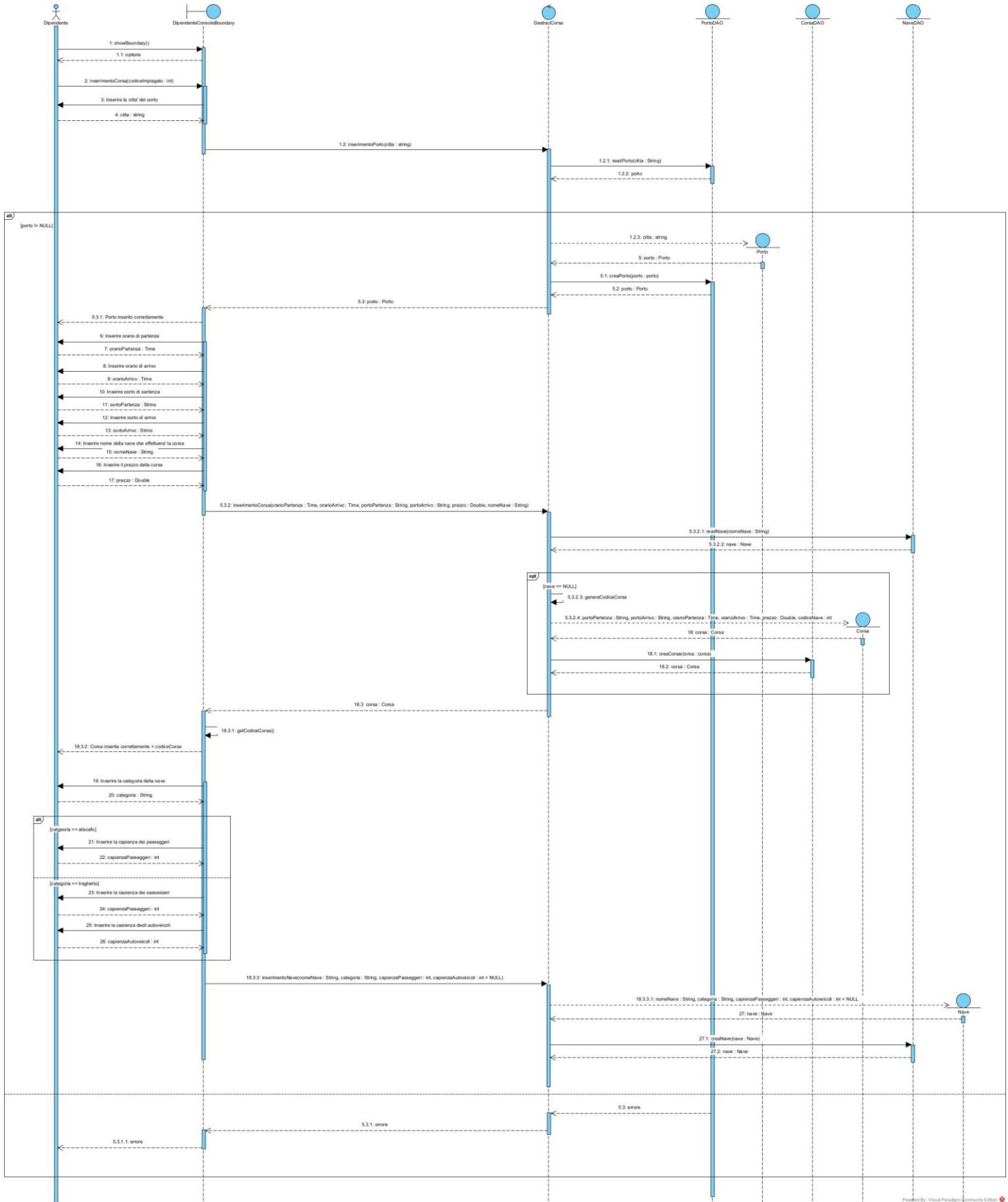


Figura 9 Sequence Diagram di Progettazione del caso d'uso Inserimento Corsa



FRANCESCO DI SERIO



## **6. Testing**

### **6.1 Test funzionale**

Si riportano i risultati dell'esecuzione dei test funzionali precedentemente pianificati.



### ESECUZIONE TEST FUNZIONALE PER LA FUNZIONALITÀ “Emissione Biglietto”

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (FAIL, PASS)
1	Test input validi #1 (property TRA)	I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 1, codice corsa 101, codice impiegato 1001 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente e aggiornato	Biglietto veicolo con codice biglietto 1, codice corsa 101, codice impiegato 1001 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato	PASS
2	Test input validi #2 (property TRA)	Il classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1002 - 101 - Veicolo - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 2, codice corsa 101, codice impiegato 1002 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente e aggiornato	Biglietto veicolo con codice biglietto 2, codice corsa 101, codice impiegato 1002 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato	PASS
3	Test input validi #3	I classe I classe Il classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto passeggero con codice biglietto 3, codice corsa 101 e codice impiegato 1001, cronologia inserita con codice corsa 101 e codice	Il database è stato correttamente e aggiornato	Biglietto passeggero con codice biglietto 3, codice corsa 101 e codice impiegato 1001, cronologia inserita con codice corsa 101 e codice	Il database è stato correttamente aggiornato	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

					cliente 10001		cliente 10001		
4	Test input validi #4	Il classe I classe Il classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1002 - 101 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto passeggero con codice biglietto 4, codice corsa 101 e codice impiegato 1002, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente e aggiornato	Biglietto passeggero con codice biglietto 4, codice corsa 101 e codice impiegato 1002, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato	PASS
5	Test input validi #5	I classe Il classe Il classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 102 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto passeggero con codice biglietto 1, codice corsa 102 e codice impiegato 1001, cronologia inserita con codice corsa 102 e codice cliente 10001	Il database è stato correttamente e aggiornato	Biglietto passeggero con codice biglietto 1, codice corsa 102 e codice impiegato 1001, cronologia inserita con codice corsa 102 e codice cliente 10001	Il database è stato correttamente aggiornato	PASS
6	Test input validi #6	Il classe Il classe Il classe Il classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1002 - 102 - Passeggero - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto passeggero con codice biglietto 2, codice corsa 102 e codice impiegato 1002, cronologia inserita con codice corsa 102 e codice cliente 10001	Il database è stato correttamente e aggiornato	Biglietto passeggero con codice biglietto 2, codice corsa 102 e codice impiegato 1002, cronologia inserita con codice corsa 102 e codice cliente 10001	Il database è stato correttamente aggiornato	PASS
7	Test input validi #7 (property TRA)	I classe I classe I classe I classe Il classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente	- 1001 - 101 - Veicolo - AA000AA - Non generare	Biglietto veicolo con codice biglietto 5, codice corsa 101, codice impiegato 1001 e targa	Il database è stato correttamente e aggiornato	Biglietto veicolo con codice biglietto 5, codice corsa 101, codice impiegato 1001 e targa	Il database è stato correttamente aggiornato	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe I classe	10001, corsa 101, ricevuta 10001VEI1019UO ) è già stata inserita	- 10001 - 10001VEI1019UO	AA000AA, cronologia aggiornata con codice impiegato 1001		AA000AA, cronologia aggiornata con codice impiegato 1001		
8	Test input validi #8 (property TRA)	Il classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001VEI1019UO ) è già stata inserita	- 1002 - 101 - Veicolo - AA000AA - Non generare - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 6, codice corsa 101, codice impiegato 1002 e targa AA000AA, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente e aggiornato	Biglietto veicolo con codice biglietto 6, codice corsa 101, codice impiegato 1002 e targa AA000AA, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente aggiornato	PASS
9	Test input validi #9	I classe I classe Il classe I classe Il classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001PAS1019UO ) è già stata inserita	- 1001 - 101 - Passeggero - AA000AA - Non generare - 10001 - 10001PAS1019UO	Biglietto passeggero con codice biglietto 7, codice corsa 101 e codice impiegato 1001, cronologia aggiornata con codice impiegato 1001	Il database è stato correttamente e aggiornato	Biglietto passeggero con codice biglietto 7, codice corsa 101 e codice impiegato 1001, cronologia aggiornata con codice impiegato 1001	Il database è stato correttamente aggiornato	PASS
10	Test input validi #10	Il classe I classe Il classe I classe Il classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001PAS1019UO ) è già stata	- 1002 - 101 - Passeggero - AA000AA - Non generare - 10001 - 10001PAS1019UO	Biglietto passeggero con codice biglietto 8, codice corsa 101 e codice impiegato 1002, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente e aggiornato	Biglietto passeggero con codice biglietto 8, codice corsa 101 e codice impiegato 1002, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente aggiornato	PASS

Elaborato di Ingegneria del Software

Studente:

LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

			inserita						
11	Test input validi #11	I classe II classe II classe I classe II classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 102, ricevuta 10001PAS1029UO) è già stata inserita	- 1001 - 102 - Passeggero - AA000AA - Non generare - 10001 - 10001PAS1029UO	Biglietto passeggero con codice biglietto 3, codice corsa 102 e codice impiegato 1001, cronologia aggiornata con codice impiegato 1001	Il database è stato correttamente aggiornato	Biglietto passeggero con codice biglietto 3, codice corsa 102 e codice impiegato 1001, cronologia aggiornata con codice impiegato 1001	Il database è stato correttamente aggiornato	PASS
12	Test input validi #12	II classe II classe II classe II classe I classe I classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 102, ricevuta 10001PAS1029UO) è già stata inserita	- 1002 - 102 - Passeggero - AA000AA - Non generare - 10001 - 10001PAS1029UO	Biglietto passeggero con codice biglietto 4, codice corsa 102 e codice impiegato 1002, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente aggiornato	Biglietto passeggero con codice biglietto 4, codice corsa 102 e codice impiegato 1002, cronologia aggiornata con codice impiegato 1002	Il database è stato correttamente aggiornato	PASS
13	Test [SINGLE] #1	III classe I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1003 - 101 - Veicolo - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto veicolo con codice biglietto 9, codice corsa 101, codice impiegato 1003 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato	Biglietto veicolo con codice biglietto 9, codice corsa 101, codice impiegato 1003 e targa AA000AA, cronologia inserita con codice corsa 101 e codice cliente 10001	Il database è stato correttamente aggiornato	PASS
14	Test [SINGLE] #2	I classe III classe	Il dipendente ha effettuato il	- 1001 - 103	Biglietto veicolo con codice biglietto	Il database è stato	Biglietto veicolo con codice biglietto	Il database è stato	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe I classe I classe I classe I classe	login. Non ci sono cronologie nel database	<ul style="list-style-type: none"> <li>- Veicolo</li> <li>- AA000AA</li> <li>- Genera</li> <li>- 10001</li> <li>- 10001VEI1019UO</li> </ul>	1, codice corsa 103, codice impiegato 1001 e targa AA000AA, cronologia inserita con codice corsa 103 e codice cliente 10001	correttamente e aggiornato	1, codice corsa 103, codice impiegato 1001 e targa AA000AA, cronologia inserita con codice corsa 103 e codice cliente 10001	correttamente aggiornato	
15	Test [ERROR] #1	IV classe I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	<ul style="list-style-type: none"> <li>- -1</li> <li>- 101</li> <li>- Veicolo</li> <li>- AA000AA</li> <li>- Genera</li> <li>- 10001</li> <li>- 10001VEI1019UO</li> </ul>	Biglietto = null. Errore: il codice impiegato deve essere maggiore di 0	Nessuna modifica al database	Biglietto = null. Errore: il codice impiegato deve essere maggiore di 0	Nessuna modifica al database	PASS
16	Test [ERROR] #2	V classe I classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	<ul style="list-style-type: none"> <li>- 5000</li> <li>- 101</li> <li>- Veicolo</li> <li>- AA000AA</li> <li>- Genera</li> <li>- 10001</li> <li>- 10001VEI1019UO</li> </ul>	Biglietto = null. Errore: l'impiegato selezionato non esiste	Nessuna modifica al database	Biglietto = null. Errore: eccezione SQL	Nessuna modifica al database	PASS
17	Test [ERROR] #3	I classe IV classe I classe I classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	<ul style="list-style-type: none"> <li>- 1001</li> <li>- -1</li> <li>- Passeggero</li> <li>- AA000AA</li> <li>- Genera</li> <li>- 10001</li> <li>- 10001VEI1019UO</li> </ul>	Biglietto = null. Errore: il codice corsa deve essere maggiore di 0	Nessuna modifica al database	Biglietto = null. Errore: il codice corsa deve essere maggiore di 0	Nessuna modifica al database	PASS
18	Test [ERROR] #4	I classe V classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	<ul style="list-style-type: none"> <li>- 1001</li> <li>- 500</li> <li>- Passeggero</li> <li>- AA000AA</li> </ul>	Biglietto = null. Errore: la corsa selezionata non esiste	Nessuna modifica al database	Biglietto = null. Errore: non ci sono più biglietti disponibili di	Nessuna modifica al database	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe I classe I classe	database	- Genera - 10001 - 10001VEI1019UO			questo tipo per questa corsa.		
19	Test [ERROR] #5	I classe I classe III classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Animale - AA000AA - Genera - 10001 - 10001VEI1019UO	Biglietto = null. Errore: il tipo di biglietto inserito non è valido	Nessuna modifica al database	Biglietto = null. Errore: il tipo di biglietto inserito non è valido	Nessuna modifica al database	PASS
20	Test [ERROR] #6	I classe I classe I classe II classe I classe I classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - null - Genera - 10001 - 10001VEI1019UO	Biglietto = null. Errore: Errore: necessario inserire la targa	Nessuna modifica al database	Biglietto = null. Errore: Errore: necessario inserire la targa	Nessuna modifica al database	PASS
21	Test [ERROR] #7	I classe I classe I classe I classe II classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - AA000AA - Genera - -1 - 10001VEI1019UO	Biglietto = null. Errore: il codice cliente deve essere maggiore di 0	Nessuna modifica al database	Biglietto = null. Errore: il codice cliente deve essere maggiore di 0	Nessuna modifica al database	PASS
22	Test [ERROR] #8	I classe I classe I classe I classe I classe III classe I classe	Il dipendente ha effettuato il login. Non ci sono cronologie nel database	- 1001 - 101 - Veicolo - AA000AA - Genera - 50000 - 10001VEI1019UO	Biglietto = null. Errore: il cliente selezionato non esiste	Nessuna modifica al database	Biglietto = null. Errore: eccezione SQL	Il database è stato modificato con l'aggiunta del biglietto	FAIL
23	Test [ERROR] #9	I classe I classe	Il dipendente ha effettuato il	- 1001 - 101	Biglietto = null. Errore: necessario	Nessuna modifica al	Biglietto = null. Errore: necessario	Il database è stato	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		I classe I classe II classe I classe II classe	login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001VEI1019UO ) è già stata inserita	- Veicolo - AA000AA - Non generare - 10001 - null	inserire la ricevuta	database	inserire la ricevuta	modificato con l'aggiunta del biglietto	
24	Test [ERROR] #10	I classe I classe I classe I classe II classe I classe III classe	Il dipendente ha effettuato il login. La cronologia di acquisto (cliente 10001, corsa 101, ricevuta 10001VEI1019UO ) è già stata inserita	- 1001 - 101 - Veicolo - AA000AA - Non generare - 10001 - 50001VEI1019UO	Biglietto = null. Errore: l'acquisto selezionato non esiste	Nessuna modifica al database	Biglietto = null. Errore: L'acquisto selezionato non esiste	Nessuna modifica al database	PASS

A seguito del caso di test numero 22 con esito FAIL, considerando anche i messaggi di errore diversi da quelli attesi per i casi di test numero 16 e numero 18, ho effettuato il debug del metodo *emissioneBiglietto* della classe *gestisciCorsa* del package *poseidon.control* andando ad inserire, tra il controllo dei valori di input, tre accessi al database:

- Il primo accesso verifica se il codice dell'impiegato inserito corrisponde ad un dipendente presente all'interno del database.
- Il secondo accesso verifica se il codice della corsa inserito corrisponde ad una corsa presente all'interno del database.
- Il terzo accesso verifica se il codice del cliente inserito corrisponde ad un cliente presente all'interno del database.

In caso di assenza nella base di dati dell'istanza selezionata, ho implementato la stampa di un corretto messaggio di errore e la terminazione del metodo con la restituzione di un riferimento nullo.

Inoltre, ho deciso di aggiungere anche un quarto accesso al database per verificare, nel caso in cui non si debba generare una nuova ricevuta di acquisto, se la ricevuta, il codice del cliente, il codice della corsa, il tipo del biglietto ed eventualmente la targa inseriti corrispondono ad una cronologia di acquisto effettivamente presente all'interno del database.

Infine, posso constatare che un test strutturale, effettuato quindi a partire dal codice, difficilmente sarebbe riuscito a rilevare tali difetti, essendo particolarmente legati ai valori di input del metodo e non relativi al suo flusso di controllo.

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

## ESECUZIONE TEST FUNZIONALE PER LA FUNZIONALITÀ "Inserimento Corsa"

## INSERIMENTO CORSA

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (FAIL, PASS)
1	Test input validi	I classe I classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse ne navi nel database.	- Napoli - Ischia - 10:15 - 13:40 - 20.00 - Estate	Corsa con porto di partenza Napoli, porto di arrivo Ischia, orario di partenza 10:15, orario di arrivo 13:40, costo della corsa 20.00, nome della nave che effettua la corsa Estate con codice corsa 1.	Il database è stato correttamente aggiornato.	Corsa con porto di partenza Napoli, porto di arrivo Ischia, orario di partenza 10:15, orario di arrivo 13:40, costo della corsa 20.00, nome della nave che effettua la corsa Estate con codice corsa 1.	Il database è stato correttamente aggiornato.	PASS
2	Test [ERROR] #1	II classe I classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Naaaaaaaaaapppp ppppppooooooooo oolllllllllllllllllllll - Ischia - 10:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	Corsa = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	PASS
3	Test [ERROR] #2	III classe I classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Milano - Ischia - 10:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: il porto inserito non è registrato.	Nessuna modifica al database.	Corsa = null. Errore: il porto inserito non è registrato.	Nessuna modifica al database.	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO



4	Test [ERROR] #3	I classe II classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - lsssssssscccccccc cchhhhhhhhhiiiiiii iiiiaaaaaaaaaaaaa a - 10:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	Corsa = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	PASS
5	Test [ERROR] #4	I classe III classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Milano - 10:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: il porto inserito non è registrato.	Nessuna modifica al database.	Corsa = null. Errore: il porto inserito non è registrato.	Nessuna modifica al database.	PASS
6	Test [ERROR] #5	I classe I classe II classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 15:15 - 13:40 - 20.00 - Estate	Corsa = null. Errore: orario inserito non è valido.	Nessuna modifica al database.	Corsa = null. Errore: orario inserito non è valido.	Nessuna modifica al database.	PASS
7	Test [ERROR] #6 <i>*Osservazione*</i>	I classe I classe I classe II classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15 - 40:40 - 20.00 - Estate	Corsa = null. Errore: valore non valido. <i>*Vedere Osservazione*</i>	Nessuna modifica al database.	Corsa = null. Errore: valore non valido. <i>*Vedere Osservazione*</i>	Nessuna modifica al database.	PASS
8	Test [ERROR] #7	I classe I classe I classe I classe II classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15 - 13:40 - -1.00 - Estate	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

9	Test [ERROR] #8	I classe I classe I classe I classe III classe I classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15 - 13:40 - 250.00 - Estate	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.	PASS
10	Test [ERROR] #9	I classe I classe I classe I classe I classe II classe	Il dipendente ha fatto il login(). Non ci sono corse nel database.	- Napoli - Ischia - 10:15 - 13:40 - 20.00 - Cooooooooooooosssss ssssttttttttttttttaa aaaaaaaaaaaaaaaaaa aaaaaaaaaaaa	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.	Corsa = null. Errore: valore inserito non valido.	Nessuna modifica al database.	PASS

**Osservazione:** non occorre un caso di testing specifico per la gestione di questo tipo di error, in quanto il tipo di variabile definito 'LocalTime' effettua già un controllo per la validità del formato e del range di valori inseriti (non oltre 24 ore).

### INSERIMENTO NAVE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (FAIL, PASS)
1	Test input validi #1 (property TRA)	I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 270 - 30 - 101	Nave di tipo traghetto, con nome Estate, capienza passeggeri 270, capienza autoveicoli 30, con codice corsa 101.	Il database è stato correttamente aggiornato.	Nave di tipo traghetto, con nome Estate, capienza passeggeri 270, capienza autoveicoli 30, con codice corsa 101.	Il database è stato correttamente aggiornato.	PASS
2	Test input validi #2 (property TRA)	I classe II classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Aliscafo - 270 - 102	Nave di tipo aliscafo, con nome Estate, e capienza passeggeri 270 con codice corsa 102.	Il database è stato correttamente aggiornato.	Nave di tipo aliscafo, con nome Estate, e capienza passeggeri 270 con codice corsa 102.	Il database è stato correttamente aggiornato.	PASS
3	Test [ERROR] #1	II classe I classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Cooooooooooooossssssstttttttttttttaa aaaaaaaaaaaaaaaaa aaaaaaaaa - Traghetto - 270 - 30 - 101	Nave = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	Nave = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	PASS
4	Test [ERROR] #2	I classe III classe I classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Battello - 270 - 30 - 101	Nave = null. Errore: valor inseriti non valido.	Nessuna modifica al database.	Nave = null. Errore: valor inseriti non valido.	Nessuna modifica al database.	PASS
5	Test [ERROR] #3	I classe I classe	Il dipendente ha fatto il login().	- Estate - Traghetto	Nave = null. Errore: valore non	Nessuna modifica al	Nave = null. Errore: valore non	Nessuna modifica al	PASS

Elaborato di Ingegneria del Software

Studente: LEONARDO CATELLO

DAIANA CIPOLLARO

FRANCESCO DI SERIO

		Il classe I classe I classe	Non ci sono navi nel database.	- 500 - 30 - 101	valido (max 400).	database.	valido (max 400).	database.	
6	Test [ERROR] #4	I classe I classe III classe I classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - -1 - 30 - 101	Nave = null. Errore: valore non valido.	Nessuna modifica al database.	Nave = null. Errore: valore non valido.	Nessuna modifica al database.	PASS
7	Test [ERROR] #5	I classe I classe I classe II classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 270 - 150 - 101	Nave = null. Errore: valore non valido (max 100).	Nessuna modifica al database.	Nave = null. Errore: valore non valido (max 100).	Nessuna modifica al database.	PASS
8	Test [ERROR] #6	I classe I classe I classe III classe I classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 270 - -1 - 101	Nave = null. Errore: valore non valido.	Nessuna modifica al database.	Nave = null. Errore: valore non valido.	Nessuna modifica al database.	PASS
9	Test [ERROR] #7	I classe I classe I classe I classe II classe	Il dipendente ha fatto il login(). Non ci sono navi nel database.	- Estate - Traghetto - 270 - 30 - -1	Nave = null. Errore: il codice corsa deve essere > 0.	Nessuna modifica al database.	Nave = null. Errore: il codice corsa deve essere > 0.	Nessuna modifica al database.	PASS

## INSERIMENTO PORTO

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (FAIL, PASS)
1	Test input validi	I classe	Il dipendente ha fatto il login(). Non ci sono porti nel database.	- Pozzuoli	Porto con città Pozzuoli.	Il database è stato correttamente e aggiornato.	Porto con città Pozzuoli.	Il database è stato correttamente aggiornato.	PASS
2	Test [ERROR] #1	II classe	Il dipendente ha fatto il login(). Non ci sono porti nel database.	- Poooooooooooooo ooooozzzzzzzzzzzz zzzzzzzzzzzzzzzzzz zzzzzzuuuuuuuuuuu uuooooooooooooooo o       iiiiiii iiiiiii	Porto = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	Porto = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	PASS
3	Test [ERROR] #2	III classe	Il dipendente ha fatto il login(). Non ci sono porti nel database.	- Pozzuoli	Porto = null. Errore: il porto inserito è già esistente.	Nessuna modifica al database.	Porto = null. Errore: caratteri inseriti non validi.	Nessuna modifica al database.	PASS

### ESECUZIONE TEST FUNZIONALE PER LA FUNZIONALITÀ "Acquista Biglietto"

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito
<u>1</u>	TEST INPUT VALIDI #1 [PROPERTY TRA]	I classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto veicolo, codice corsa 101, targa AA000AA, da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	Ricevuta per l'acquisto di un biglietto veicolo, codice corsa 101 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	PASS
<u>2</u>	TEST INPUT VALIDI #2	I classe I classe Il classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 passeggero AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto passeggero, codice corsa 101 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	Ricevuta per l'acquisto di un biglietto passeggero, codice corsa 101 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	PASS
<u>3</u>	TEST INPUT VALIDI #3	Il classe I classe Il classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	102 10002 passeggero AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto passeggero, codice corsa 102 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	Ricevuta per l'acquisto di un biglietto passeggero, codice corsa 102 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	PASS

Studente:

LEONARDO CATELLO

Elaborato di Ingegneria del Software

DAIANA CIPOLLARO

FRANCESCO DI SERIO

<u>4</u>	TEST [SINGLE] #1	III classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	103 10001 veicolo AA000AA Leonardo Catello contante	Ricevuta per l'acquisto di un biglietto veicolo, codice corsa 103, targa AA000AA, da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	Ricevuta per l'acquisto di un biglietto veicolo, codice corsa 103 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	PASS
<u>5</u>	TEST [ERRORE] #1	III classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	-1 10001 passeggero AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il codice corsa deve essere > 0.	Nessuna modifica al database	Ricevuta = null. Errore: il codice corsa deve essere > 0.	Nessuna modifica al database	PASS
<u>6</u>	TEST [ERRORE] #2	IV classe I classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	500 10001 passeggero AA000AA Leonardo Catello contante	Ricevuta = null. Errore: la corsa selezionata non esiste.	Nessuna modifica al database	Ricevuta = null. Errore: prezzo non valido.	Nessuna modifica al database	PASS
<u>7</u>	TEST [ERRORE] #3	I classe III classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 -1 passeggero AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il codice cliente deve essere > 0.	Nessuna modifica al database	Ricevuta = null. Errore: il codice cliente deve essere > 0.	Nessuna modifica al database	PASS

<u>8</u>	TEST [ERRORE] #4	I classe IV classe I classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 50000 passeggero AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il cliente selezionato non esiste.	Nessuna modifica al database	Ricevuta = null. Errore: il cliente selezionato non esiste.	Nessuna modifica al database	PASS
<u>9</u>	TEST [ERRORE] #5	I classe I classe III classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 null AA000AA Leonardo Catello contante	Ricevuta = null. Errore: il tipo di biglietto inserito non è valido.	Nessuna modifica al database	Ricevuta = null. Errore: il tipo di biglietto inserito non è valido.	Nessuna modifica al database	PASS
<u>10</u>	TEST [ERRORE] #6	I classe I classe I classe II classe I classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo null Leonardo Catello contante	Ricevuta = null. Errore: necessario inserire la targa.	Nessuna modifica al database	Ricevuta per l'acquisto di un biglietto veicolo, codice corsa 101 da parte del cliente con codice 10001.	Il database è stato correttamente aggiornato	FAIL
<u>11</u>	TEST [ERRORE] #7	I classe I classe I classe I classe II classe I classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA null Catello contante	Ricevuta = null. Errore: necessario inserire il nome.	Nessuna modifica al database	Ricevuta = null. Errore: necessario inserire il nome.	Nessuna modifica al database	PASS



<u>12</u>	TEST [ERRORE] #8	I classe I classe I classe I classe I classe II classe I classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA Leonardo null contante	Ricevuta = null. Errore: necessario inserire il cognome.	Nessuna modifica al database	Ricevuta = null. Errore: necessario inserire il cognome.	Nessuna modifica al database	PASS
<u>13</u>	TEST [ERRORE] #9	I classe I classe I classe I classe I classe II classe	Il cliente ha effettuato il login. Non ci sono cronologie nel database	101 10001 veicolo AA000AA Leonardo Catello null	Ricevuta = null. Errore: necessario inserire la tipologia del pagamento.	Nessuna modifica al database	Ricevuta = null. Errore: necessario inserire la tipologia del pagamento.	Nessuna modifica al database	PASS

Successivamente al FAIL del caso di test 10 e al messaggio di errore errato nel caso di test 6, raccogliendo maggiori informazioni tramite debug del metodo *acquistaBiglietto* della classe *gestisciCorsa* del package *poseidon.control*, ho aggiunto due casi di controllo ulteriori per i valori di input *targa* e per la corsa selezionata tramite *codiceCorsa* con relativi messaggi di errore e terminazione del metodo con *ricevuta* nulla.

Questo ha permesso di rendere il codice più robusto ad eventuali errori in input che difficilmente sarebbero stati rilevati senza il precedente test strutturale.