



CORSO DI LAUREA IN INGEGNERIA INFORMATICA

CORSO DI INGEGNERIA DEL SOFTWARE

PROF. S. RUSSO - A.A. 2024 - 25

Progetto

Franchising noleggio scooter “TouristScooters”



GRUPPO DI LAVORO

“CiùlsMeglChUan”

Aldo Manzo N46007037

Gianmarco La Salvia N46007055

Sommario

Capitolo 1 – Specifiche informali	4
Capitolo 2 – Analisi e specifica dei requisiti	5
2.1 Analisi nomi-verbi	5
2.2 Revisione dei requisiti	6
2.3 Glossario dei termini	7
2.4 Classificazione dei requisiti	8
2.4.1 Requisiti funzionali	8
2.4.2 Requisiti sui dati	9
2.4.3 Vincoli/Altri requisiti	9
2.5 Modellazione dei casi d'uso	10
2.5.1 Attori e casi d'uso	10
2.5.2 Diagramma dei casi d'uso	12
2.5.3 Scenari	12
2.6 Modellazione dei dati	15
2.6.1 Progettazione concettuale	15
2.7 Diagramma delle classi	15
2.7.1 Diagramma delle classi di analisi	15
2.7.2 Diagramma delle classi raffinato	16
2.8 Diagrammi di sequenza	16
2.9 Verifica della completezza dei requisiti	18
Capitolo 3 - Stima dei costi	19
3.1 Stima dei costi	19
Capitolo 4 - Piano di test funzionale	21
4.1 Piano di test funzionale per il caso d'uso "RicercaScooter"	21
4.2 Piano di test funzionale per il caso d'uso "PrenotazioneScooter"	24
Capitolo 5 - Progettazione	31
5.1 Progettazione della base di dati	31
5.1.1 Progettazione logica	31
5.2 Diagramma delle classi	31
5.3 Diagrammi di sequenza	31
Capitolo 6 - Implementazione	34
6.1 Dipendenze di progetto	34

6.2 Descrizione della struttura dell'applicazione	34
6.3 JavaDoc	34
6.4 Costi.....	35
6.4.1 Confronto con stima dei costi effettuata	35
Capitolo 7 - Testing	37
7.1 Test strutturale	37
7.1.1 Complessità ciclomatica	37
7.1.2 Test di unità.....	39
7.2 Test funzionale	52



Capitolo 1 – Specifiche informali

Franchising noleggio scooter “TouristScooters”

Si vuole realizzare una applicazione web per la catena “TouristScooters” di agenzie di noleggio scooter in località turistiche, tipicamente isolate. Sul sistema agiscono: il direttore della catena; i titolari delle agenzie in franchising, che noleggiavano gli scooter ai turisti; i turisti, che possono verificare la disponibilità (e i prezzi) e procedere al noleggio.

Il direttore della catena registra le agenzie, con località, indirizzo, titolare (cui il sistema assegna automaticamente username e password), e-mail e numero di telefono.

Il titolare di una agenzia registra per ogni scooter: la targa, la cilindrata, la tipologia (tradizionale, elettrico), e il prezzo per giorno-noleggio, diverso in bassa stagione rispetto alla alta stagione. Poiché il sistema deve tener traccia nel tempo di tutti gli scooter, non è possibile rimuoverli dal sistema ma i titolari possono alterare lo stato degli scooter da “in-servizio” a “dismesso”.

I clienti possono ricercare sul sistema gli scooter in base alla località e all'intervallo temporale di noleggio (data ritiro e data consegna). Il sistema mostra loro i veicoli disponibili nell'agenzia della località selezionata nel periodo indicato, e il costo complessivo del noleggio. Uno scooter è effettivamente disponibile per il noleggio se al momento della ricerca risulta “in-servizio” e se non esistono altre prenotazioni che lo impegnano anche in parziale sovrapposizione con le date indicate per la prenotazione.

Per effettuare una prenotazione, un turista che non possiede le credenziali deve registrarsi sulla piattaforma specificando: nome, cognome, data di nascita, e-mail. Un cliente registrato può effettivamente noleggiare uno scooter, dopo la ricerca, specificando anche eventuali accessori da includere nella prenotazione.

Gli accessori si distinguono in servizi assicurativi (opzione per secondo guidatore, assicurazione infortuni del guidatore) e optional (es. casco, bauletto). Gli accessori hanno una descrizione testuale e un prezzo prefissato che contribuisce a determinare il costo totale della prenotazione. Il costo della prenotazione è dato dal prodotto del prezzo giorno noleggio per i giorni di prenotazione, sommato con i prezzi dei servizi accessori scelti.

Ogni mese il direttore della catena riceve automaticamente dal sistema un report dei noleggi di ciascuna agenzia, per poter fatturare (in percentuale rispetto ai noleggi) a ciascuna agenzia i costi dei servizi di promozione, pubblicità e gestione della applicazione web. I titolari delle agenzie possono richiedere al sistema un report del numero di giorni di noleggio per ogni mese per ciascuno scooter, al fine di valutare l'acquisto di nuovi scooter, tenendo conto delle preferenze dei clienti.

Capitolo 2 – Analisi e specifica dei requisiti

2.1 Analisi nomi-verbi

Si vuole realizzare una applicazione web per la catena “TouristScooters” di agenzie di noleggio scooter in località turistiche, tipicamente isolate. Sul sistema agiscono: il direttore della catena; i titolari delle agenzie in franchising, che noleggianno gli scooter ai turisti; i turisti, che possono verificare la disponibilità (e i prezzi) e procedere al noleggio.

Il **direttore della catena** **registra le agenzie**, con **località**, **indirizzo**, **titolare** (cui il sistema **assegna automaticamente username e password**), **e-mail** e **numero di telefono**.

Il **titolare di una agenzia** **registra per ogni scooter**: la **targa**, la **cilindrata**, la **tipologia** (tradizionale, elettrico), e il **prezzo per giorno-noleggio**, diverso in bassa stagione rispetto alla alta stagione. Poiché il sistema deve tener traccia nel tempo di tutti gli scooter, non è possibile rimuoverli dal sistema ma i **titolari** **possono alterare lo stato** degli scooter da “in-servizio” a “dismesso”.

I **clienti** **possono ricercare sul sistema gli scooter in base alla località e all’intervallo temporale di noleggio** (**data ritiro** e **data consegna**). Il sistema **mostra loro i veicoli disponibili nell’agenzia della località selezionata nel periodo indicato, e il costo complessivo del noleggio**. Uno scooter è effettivamente disponibile per il noleggio se al momento della ricerca risulta “in-servizio” e se non esistono altre prenotazioni che lo impegnano anche in parziale sovrapposizione con le date indicate per la prenotazione.

Per effettuare una **prenotazione**, un **turista** che non possiede le credenziali **deve registrarsi sulla piattaforma specificando: nome, cognome, data di nascita, e-mail**. Un **cliente registrato** **può effettivamente noleggiare uno scooter, dopo la ricerca, specificando anche eventuali accessori da includere nella prenotazione**.

Gli **accessori** si distinguono in **servizi assicurativi** (opzione per secondo guidatore, assicurazione infortuni del guidatore) e **optional** (es. casco, bauletto). Gli accessori hanno una **descrizione testuale** e un **prezzo** prefissato che contribuisce a determinare il costo totale della prenotazione. Il costo della prenotazione è dato dal prodotto del prezzo giorno noleggio per i giorni di prenotazione, sommato con i prezzi dei servizi accessori scelti.

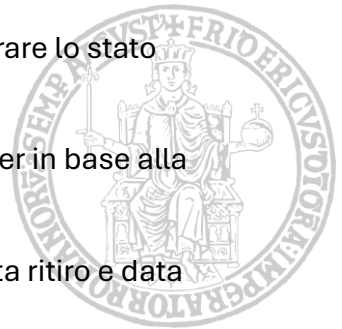
Ogni mese il direttore della catena riceve automaticamente dal sistema un report dei noleggi di ciascuna agenzia, per poter fatturare (in percentuale rispetto ai noleggi) a ciascuna agenzia i costi dei servizi di promozione, pubblicità e gestione della applicazione web. I **titolari delle agenzie** **possono richiedere al sistema un report del numero di giorni di noleggio per ogni mese per ciascuno scooter**, al fine di valutare l’acquisto di nuovi scooter, tenendo conto delle preferenze dei clienti.

LEGENDA:

Classe	Funzionalità	Attributo	Attore	Classe-Attore
--------	--------------	-----------	--------	---------------

2.2 Revisione dei requisiti

1. Il sistema deve offrire al Direttore una funzionalità per registrare le agenzie.
2. Di ogni agenzia si vuole memorizzare località, indirizzo, Titolare, e-mail e numero di telefono.
3. Il sistema assegna automaticamente username e password ad ogni Titolare al momento della registrazione dell'agenzia.
4. Il sistema deve offrire al Titolare di un'agenzia una funzionalità per registrare uno scooter.
5. Di ogni scooter si vuole memorizzare targa, cilindrata, tipologia e prezzo per giorno-noleggio.
6. Il sistema deve distinguere la tipologia di scooter tra "tradizionale" ed "elettrico".
7. Il prezzo per giorno-noleggio di uno scooter deve essere diverso in bassa stagione rispetto alla alta stagione.
8. Il sistema non deve offrire una funzionalità per rimuovere gli scooter.
9. Il sistema deve offrire una funzionalità al Titolare di un'agenzia per alterare lo stato degli scooter da "in-servizio" a "dismesso".
10. Il sistema deve offrire al Cliente una funzionalità per ricercare gli scooter in base alla località e all'intervallo temporale di noleggio.
11. L'intervallo temporale di noleggio di uno scooter è caratterizzato da data ritiro e data consegna dello stesso.
12. Uno scooter è effettivamente disponibile per il noleggio se al momento della ricerca risulta "in-servizio" e se non esistono altre prenotazioni che lo impegnano.
13. Il sistema deve richiedere che il Cliente sia registrato per poter effettuare una prenotazione.
14. Il sistema deve offrire al Cliente una funzionalità per registrarsi.
15. Di ogni Cliente registrato si vuole memorizzare nome, cognome, data di nascita, e-mail.
16. Il sistema deve offrire la funzionalità al Cliente per autenticarsi.
17. Il sistema deve offrire una funzionalità al Cliente registrato per noleggiare uno scooter, specificando eventuali accessori da includere nella prenotazione.
18. Il sistema deve calcolare un costo totale per ogni prenotazione.



19. Gli accessori si distinguono in servizi assicurativi (opzione per secondo guidatore, assicurazione infortuni del guidatore) e optional (es. casco, bauletto).
20. Di ogni accessorio si vuole memorizzare una descrizione testuale e un prezzo prefissato.
21. Il costo della prenotazione è dato dal prodotto del prezzo giorno noleggio per i giorni di prenotazione, sommato con i prezzi dei servizi accessori scelti.
22. Il sistema deve inviare un report dei noleggi di ciascuna agenzia al Direttore della catena con cadenza mensile.
23. Il sistema deve offrire al Titolare di un'agenzia una funzionalità per richiedere un report del numero di giorni di noleggio per ogni mese di ciascuno scooter.

2.3 Glossario dei termini

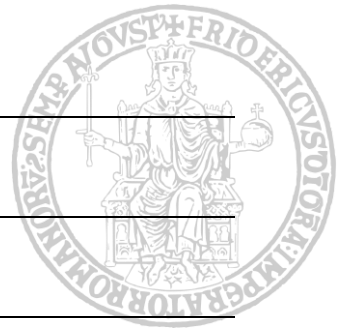
Si riporta il glossario dei termini.

Termine	Descrizione	Sinonimi
Agenzia	Un punto di noleggio scooter in una località turistica, gestito da un titolare.	
Scooter	Veicolo a due ruote registrato da un'agenzia, disponibile per il noleggio.	
Titolare	Responsabile di una specifica agenzia in franchising.	Titolare di una agenzia
Direttore	Figura che gestisce l'intera catena "TouristScooters". Si occupa della registrazione delle agenzie.	Direttore della catena
Cliente	Un generico cliente che utilizza la piattaforma per cercare scooter disponibili senza essere registrato.	Turista
Cliente registrato	Cliente che ha completato la registrazione. Può effettuare una prenotazione	
Prenotazione	Operazione effettuata da un cliente registrato per noleggiare uno scooter in un intervallo temporale, con eventuali accessori.	Noleggio
Accessorio	Servizi o optional selezionabili dal cliente durante la prenotazione (es. casco, assicurazione).	

2.4 Classificazione dei requisiti

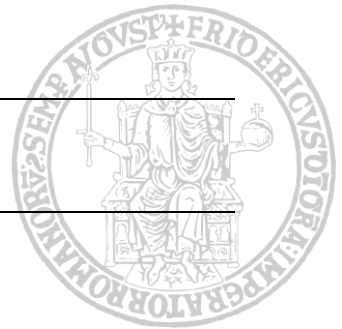
2.4.1 Requisiti funzionali

ID	Requisito	Origine (n. frase dei requisiti revisionati)
RF ₀₁	Il sistema deve offrire al Direttore una funzionalità per registrare le agenzie.	1
RF ₀₂	Il sistema assegna automaticamente username e password ad ogni Titolare al momento della registrazione dell'agenzia.	3
RF ₀₃	Il sistema deve offrire al Titolare di un'agenzia una funzionalità per registrare uno scooter.	4
RF ₀₄	Il sistema deve offrire una funzionalità al Titolare di un'agenzia per alterare lo stato degli scooter da "in-servizio" a "dismesso".	9
RF ₀₅	Il sistema deve offrire al Cliente una funzionalità per ricercare gli scooter in base alla località e all'intervallo temporale di noleggio.	10
RF ₀₆	Il sistema deve offrire al Cliente una funzionalità per registrarsi.	14
RF ₀₇	Il sistema deve offrire al cliente una funzionalità per autenticarsi	16
RF ₀₈	Il sistema deve offrire una funzionalità al Cliente registrato per noleggiare uno scooter, specificando eventuali accessori da includere nella prenotazione.	17
RF ₀₉	Il sistema deve calcolare un costo totale per ogni prenotazione.	18
RF ₁₀	Il sistema deve inviare un report dei noleggi di ciascuna agenzia al Direttore della catena con cadenza mensile.	22
RF ₁₁	Il sistema deve offrire al Titolare di un'agenzia una funzionalità per richiedere un report del numero di giorni di noleggio per ogni mese di ciascuno scooter.	23



2.4.2 Requisiti sui dati

ID	Requisito	Origine (n. frase dei requisiti revisionati)
RD ₀₁	Ogni agenzia è caratterizzata da località, indirizzo, Titolare, e-mail e numero di telefono.	2
RD ₀₂	Di ogni scooter si vuole memorizzare targa, cilindrata, tipologia e prezzo per giorno-noleggio.	5
RD ₀₃	Il sistema deve distinguere la tipologia di scooter tra “tradizionale” ed “elettrico”.	6
RD ₀₄	Il prezzo per giorno-noleggio di uno scooter deve essere diverso in bassa stagione rispetto alla alta stagione.	7
RD ₀₅	L’intervallo temporale di noleggio di uno scooter è caratterizzato da data ritiro e data consegna dello stesso.	11
RD ₀₆	Di ogni Cliente registrato si vuole memorizzare nome, cognome, data di nascita, e-mail.	15
RD ₀₇	Gli accessori si distinguono in servizi assicurativi (opzione per secondo guidatore, assicurazione infortuni del guidatore) e optional (es. casco, bauletto).	19
RD ₀₈	Di ogni accessorio si vuole memorizzare una descrizione testuale e un prezzo prefissato.	20
RD ₀₉	Il costo della prenotazione è dato dal prodotto del prezzo giorno noleggio per i giorni di prenotazione, sommato con i prezzi dei servizi accessori scelti.	21



2.4.3 Vincoli/Altri requisiti

ID	Requisito	Origine (n. frase dei requisiti revisionati)
V ₀₁	Il sistema non deve offrire funzionalità per rimuovere gli scooter.	8

V ₀₂	Uno scooter è effettivamente disponibile per il noleggio se al momento della ricerca risulta “in-servizio” e se non esistono altre prenotazioni che lo impegnano.	12
V ₀₃	Il sistema deve richiedere che il Cliente sia registrato per poter effettuare una prenotazione.	13

2.5 Modellazione dei casi d'uso

2.5.1 Attori e casi d'uso

Attori Primari:

- Direttore
- Titolare
- Cliente
- Cliente registrato
- Tempo

Attori Secondari:

- Servizio assegnazione credenziali
- Servizio notificazione

Casi d'uso:

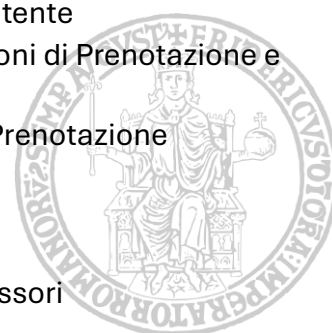
- UC1: Registra agenzie
- UC2: Registra scooter
- UC3: Modifica stato scooter
- UC4: Ricerca scooter
- UC5: Registrazione cliente
- UC6: Prenotazione scooter
- UC7: Genera report mensile per agenzia
- UC8: Genera report mensile per scooter

Casi d'uso di inclusione:

- UC9: Autenticazione utente
- UC10: Visualizza Opzioni di Prenotazione e costi
- UC11: Calcolo costo Prenotazione

Casi d'uso di estensione:

- UC12: Selezione Accessori

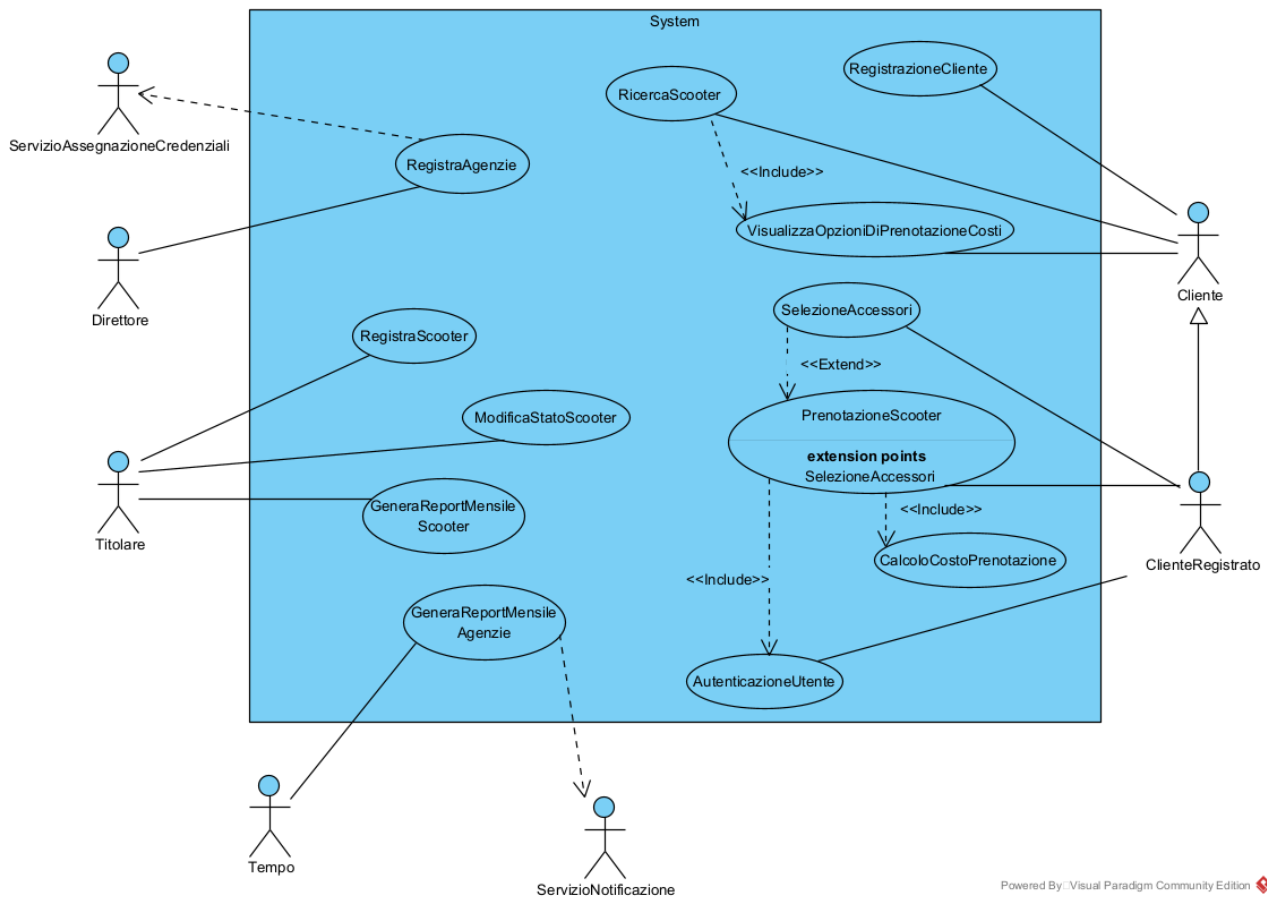


Caso d'Uso	Attori Primari	Attori Secondari	Incl. / Ext.	Requisiti Corrispondenti
UC1: Registra agenzie	Direttore	Servizio assegnazione credenziali	-	RF01, RF02
UC2: Registra scooter	Titolare	-	-	RF03
UC3: Modifica stato scooter	Titolare	-	-	RF04
UC4: Ricerca scooter	Cliente	-	Include UC10	RF05
UC5: Registrazione cliente	Cliente	-	-	RF06

UC6: Prenotazione scooter	Cliente registrato	-	Include UC9 e UC11 ed esteso da UC12	RF08
UC7: Genera report mensile per agenzia	Tempo	Servizio notificazione	-	RF10
UC8: Genera report mensile per scooter	Titolare	-	-	RF11
UC9: Autenticazione utente	Cliente registrato	-	Incluso in PrenotazioneScooter	RF07
UC10: Visualizza Opzioni di Prenotazione e Costi	Cliente	-	Incluso in Ricerca Scooter	RF05
UC11: Calcolo costo Prenotazione	-	-	Incluso in Prenotazione scooter	RF09
UC12: Selezione Accessori	Cliente registrato	-	Estende Prenotazione Scooter	RF08



2.5.2 Diagramma dei casi d'uso



2.5.3 Scenari

Caso d'uso:	RicercaScooter
Attore primario	Cliente
Attore secondario	-
Descrizione	Il cliente ricerca sul sistema gli scooter in base alla località e all'intervallo temporale di noleggio.
Pre-Condizioni	-
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il Cliente accede alla funzione di ricerca scooter <ol style="list-style-type: none"> 1.1 Il Cliente inserisce la località di ritiro, la data di ritiro e la data di consegna 2. Il sistema elabora i risultati in base ai criteri di ricerca inseriti 3. Il sistema mostra gli scooter disponibili nella località e nel periodo selezionati 4. Il Cliente seleziona uno scooter dalla lista dei risultati 5. «include» (VisualizzaOpzioniDiPrenotazioneCosti)
Post-condizioni	Vengono visualizzate le opzioni di prenotazione per lo scooter selezionato.
Casi d'uso correlati	VisualizzaOpzioniDiPrenotazioneCosti

Sequenza di eventi alternativi	-
---------------------------------------	---

Caso d'uso	PrenotazioneScooter
Attore primario	Cliente registrato
Attore secondario	-
Descrizione	Il cliente registrato prenota uno scooter selezionando date, optional e confermando il noleggio.
Pre-Condizioni	<ul style="list-style-type: none"> - Il cliente si è registrato al sistema - Sono disponibili scooter nella località selezionata - Il cliente ha ricercato lo scooter e visualizzato le opzioni di prenotazione disponibili
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il Cliente prosegue alla prenotazione dello scooter 2. Il cliente registrato sceglie se aggiungere degli accessori <ol style="list-style-type: none"> 2.1 «extends» (SelezioneAccessori) 3. Il cliente inserisce email e password 4. «include» (AutenticazioneUtente) 5. «include» (CalcolaCostoPrenotazione) 6. Il cliente conferma la prenotazione
Post-Condizioni	<ul style="list-style-type: none"> - Una prenotazione è registrata nel sistema - Lo scooter prenotato risulta occupato per il periodo selezionato e non sarà più disponibile
Casi d'uso correlati	AutenticazioneUtente, RicercaScooter, SelezioneAccessori, CalcoloCostoPrenotazione
Sequenza di eventi alternativi	4a. Se l'autenticazione fallisce il sistema mostra un messaggio di errore

Caso d'uso	AutenticazioneUtente
Attore primario	Cliente registrato
Attore secondario	-
Descrizione	L'utente inserisce le credenziali. Il sistema verifica l'identità e, se valida, consente l'accesso.
Pre-Condizioni	-
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'utente inserisce le proprie credenziali (username/email e password) 2. Il sistema verifica la correttezza delle credenziali 3. Se le credenziali sono corrette, l'utente può proseguire
Post-Condizioni	L'utente può proseguire nella prenotazione
Casi d'uso correlati	-
Sequenza di eventi alternativi	3a. Credenziali errate: il sistema mostra un messaggio di errore

Caso d'uso	CalcolaCostoPrenotazione
-------------------	---------------------------------

Attore primario	-
Attore secondario	-
Descrizione	Il sistema calcola automaticamente il costo totale della prenotazione, considerando la durata del noleggio dello scooter e gli accessori selezionati.
Pre-Condizioni	<ul style="list-style-type: none"> - Lo scooter è stato selezionato - Il periodo di prenotazione (data ritiro e consegna) è stato specificato - Gli accessori opzionali sono stati eventualmente selezionati - Lo scooter risulta disponibile per il periodo richiesto
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il sistema recupera il prezzo giornaliero dello scooter (diverso in alta/bassa stagione) 2. Il sistema calcola il numero di giorni di noleggio 3. Il sistema calcola il costo base (prezzo giornaliero × giorni di noleggio) 4. Il sistema calcola il costo degli accessori selezionati 5. Il sistema somma il costo base e il costo degli accessori 6. Il sistema restituisce il costo totale della prenotazione
Post-Condizioni	-
Casi d'uso correlati	-
Sequenza di eventi alternativi	2a. Il periodo attraversa sia alta che bassa stagione: il sistema calcola separatamente i giorni in ciascuna stagione e applica i rispettivi prezzi

Caso d'uso	VisualizzaOpzioniDiPrenotazioneCosti
Attore primario	Cliente
Attore secondario	-
Descrizione	Il sistema mostra a video i dettagli disponibili per uno scooter
Pre-Condizioni	Il cliente ha ricercato sul sistema uno scooter
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il sistema recupera i dettagli dello scooter selezionato 2. Il sistema mostra a video i dettagli disponibili per quello scooter
Post-Condizioni	-
Casi d'uso correlati	-
Sequenza di eventi alternativi	-

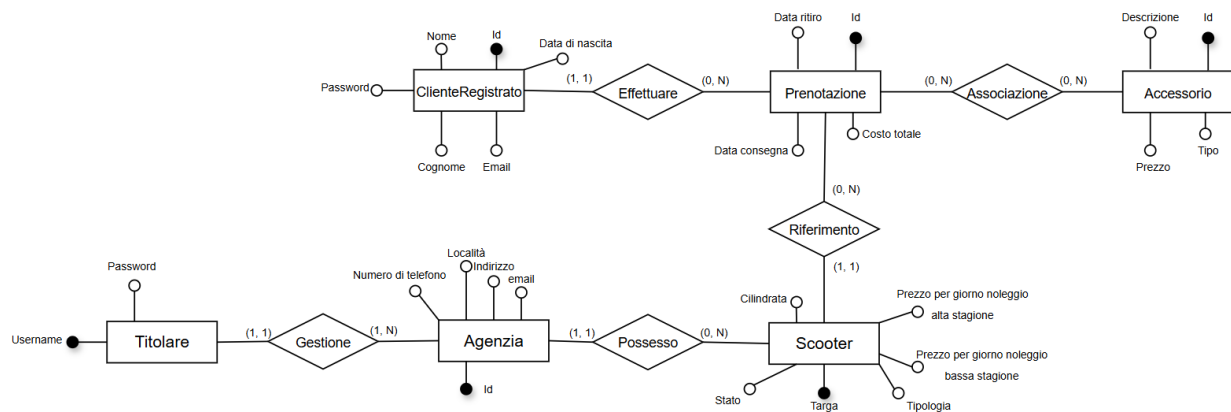
Caso d'uso	SelezioneAccessori
Attore primario	ClienteRegistrato
Attore secondario	-
Descrizione	Il sistema permette di selezionare gli accessori disponibili per un determinato scooter
Pre-Condizioni	- Il cliente ha ricercato uno scooter

Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il sistema presenta al ClienteRegistrato la lista degli accessori disponibili suddivisi in: servizi assicurativi e optional. 2. Il ClienteRegistrato seleziona gli accessori desiderati fra quelli visualizzati
Post-Condizioni	Gli accessori selezionati sono associati alla prenotazione
Casi d'uso correlati	-
Sequenza di eventi alternativi	-

2.6 Modellazione dei dati

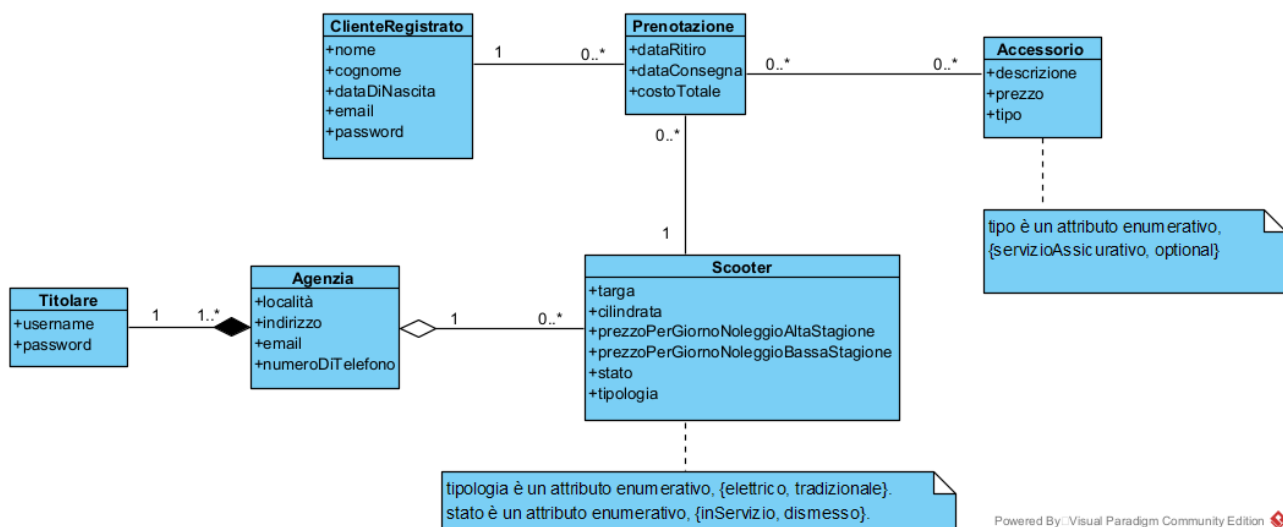
2.6.1 Progettazione concettuale

Si riporta di seguito il diagramma E/R che modella a livello concettuale i dati di interesse.

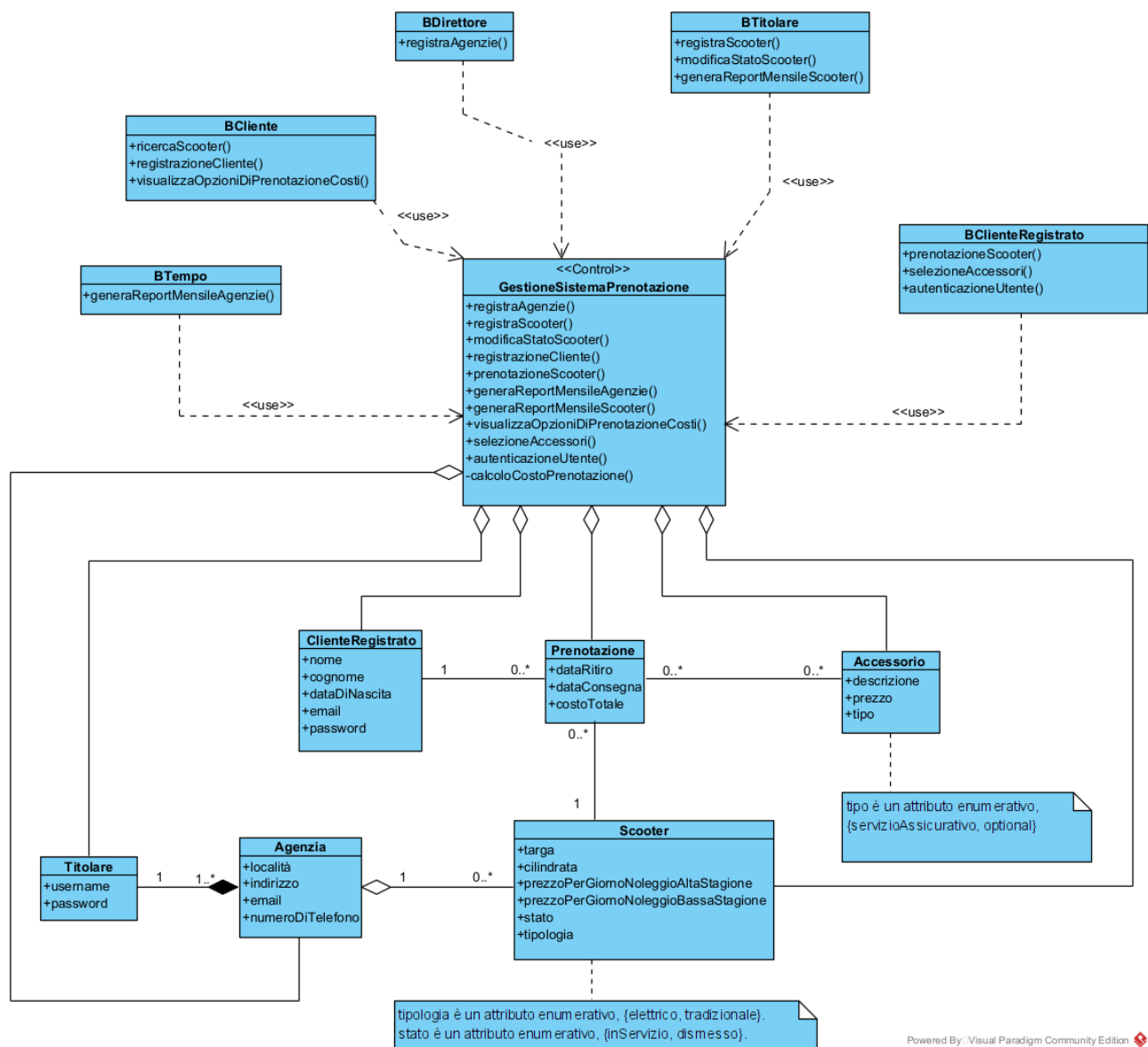


2.7 Diagramma delle classi

2.7.1 Diagramma delle classi di analisi



2.7.2 Diagramma delle classi raffinato

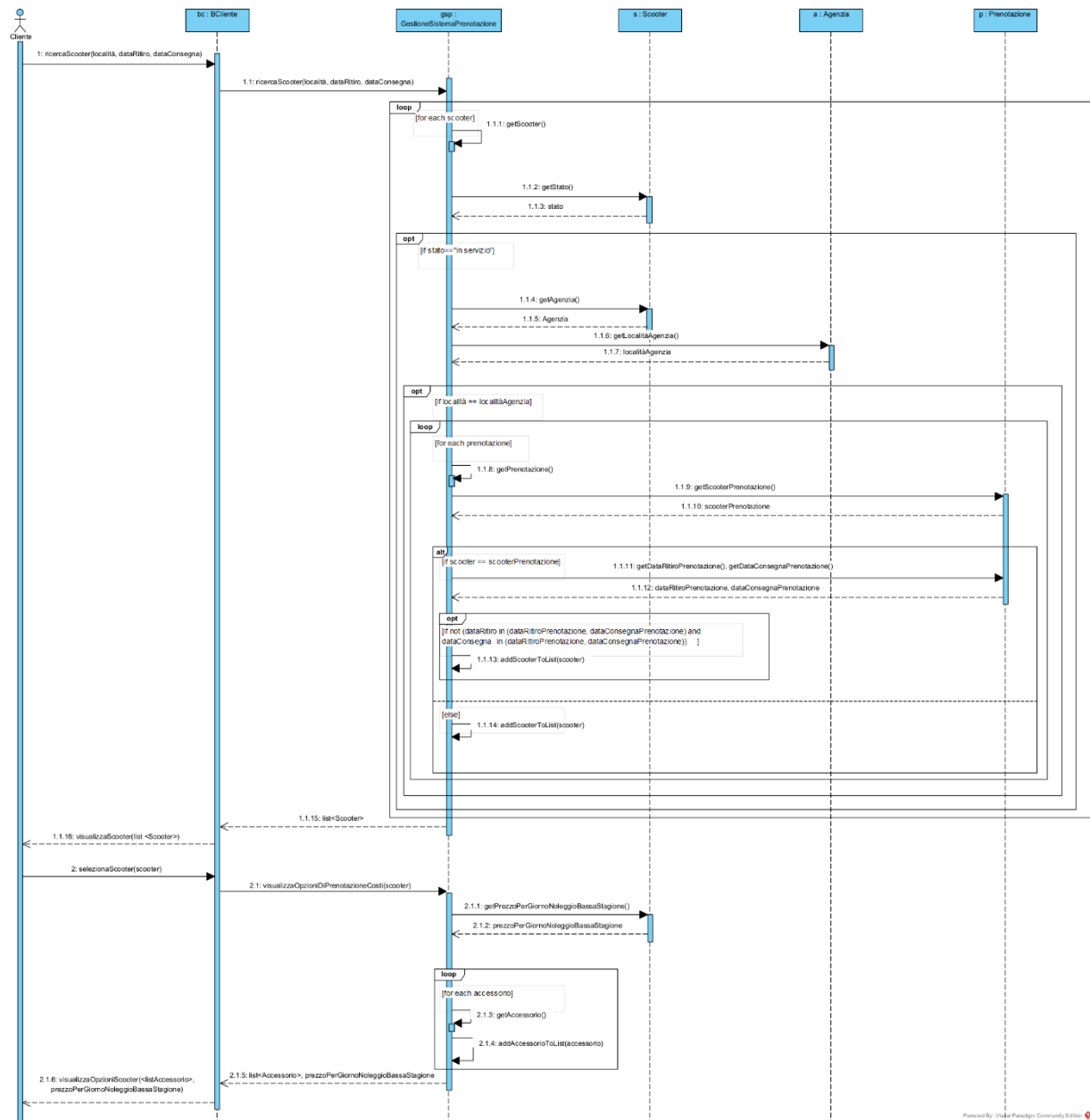


2.8 Diagrammi di sequenza

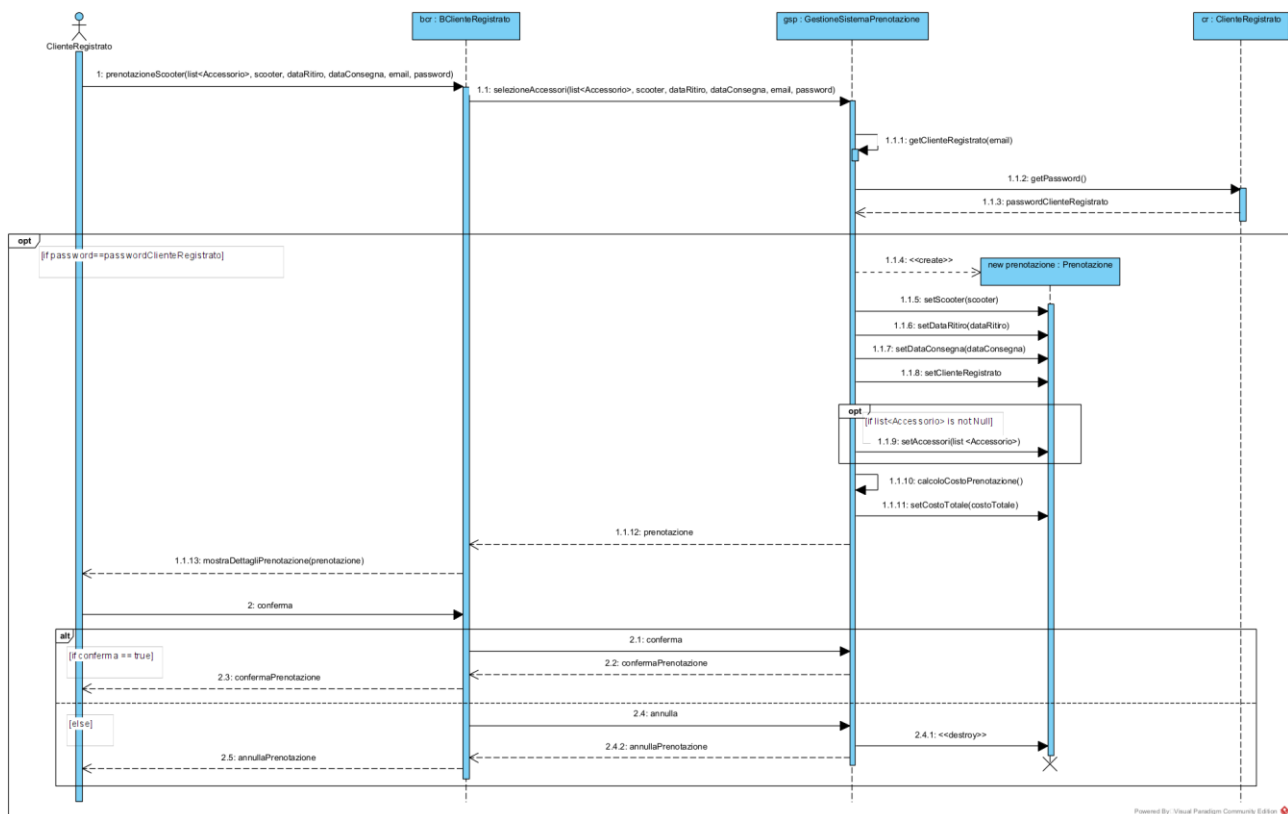
Si riportano di seguito i diagrammi di sequenza di analisi per i casi d'uso RicercaScooter e PrenotazioneScooter.

Caso d'uso RicercaScooter

U



Caso d'uso PrenotazioneScooter



2.9 Verifica della completezza dei requisiti

Legenda UCD = Use Case Diagram, CD = Class Diagram, SD = Sequence Diagram

- RF₀₁ è modellato nell'UCD con l'attore "Direttore" e con il caso d'uso UC1
- RF₀₂ è modellato nell'UCD con l'attore secondario "ServizioAssegnazioneCredenziali" e con il caso d'uso UC1
- RF₀₃ è modellato nell'UCD con l'attore "Titolare" e con il caso d'uso UC2
- RF₀₄ è modellato nell'UCD con l'attore "Titolare" e con il caso d'uso UC3
- RF₀₅ è modellato nell'UCD con l'attore "Cliente" e con il caso d'uso UC4
- RF₀₆ è modellato nell'UCD con l'attore "Cliente" e con il caso d'uso UC5
- RF₀₇ è modellato nell'UCD con l'attore "Cliente" e con il caso d'uso UC9
- RF₀₈ è modellato nell'UCD con l'attore "ClienteRegistrato" e con il caso d'uso UC6
- RF₀₉ è modellato nell'UCD e con il caso d'uso UC11
- RF₁₀ è modellato nell'UCD con il "Tempo" come attore e con il caso d'uso UC7
- RF₁₁ è modellato nell'UCD con l'attore "Titolare" e con il caso d'uso UC8
- RD₀₁ è modellato nel CD con la classe "Agenzia"
- RD₀₂ è modellato nel CD con la classe "Scooter"
- RD₀₃ è modellato nel CD con l'attributo "tipologia" della classe "Scooter"
- RD₀₄ è modellato nel CD con gli attributi "prezzoPerGiornoNoleggioAltaStagione", "prezzoPerGiornoNoleggioBassaStagione" della classe "Scooter"

- RD₀₅ è modellato nel CD con gli attributi “dataRitiro” e “dataConsegna” della classe Prenotazione
- RD₀₆ è modellato nel CD con la classe “ClienteRegistrato”
- RD₀₇ è modellato nel CD con l’attributo “tipo” della classe “Accessorio”
- RD₀₈ è modellato nel CD con la classe “Accessorio”
- RD₀₉ è modellato nel SD con la funzione “CalcolaCostoPrenotazione”

Capitolo 3 - Stima dei costi

3.1 Stima dei costi

Si riporta la stima dei costi secondo il metodo dei Punti Funzione. La tabella di riferimento per le complessità di dati e transazioni è la seguente.

	SEMPLICE	MEDIO	COMPLESSO
NILF	7	10	15
NEIF	5	7	10
NEI	3	4	6
NEO	4	5	7
NEQ	3	4	6

RicercaScooter

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	3		10		30
NEIF	0				0
NEI	3	3			9
NEO	1		5		5
NEQ	1		4		4

NILF: Le tabelle consultate dal sistema sono Scooter, Agenzie e Prenotazioni, le identifichiamo come ILF. [3 medio]

NEI: Località, DataRitiro, DataConsegna. [3 semplici]

NEO: Visualizzazione degli scooter. [1 medio]

NEQ: Interrogazione per il prelievo dei dati. [1 medio]

FATTORI CORRETTIVI

COMUNICAZIONE DATI	1
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	2
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	4
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITA' ELABORATIVA	0
RIUSABILITA'	3
FACILITA' INSTALLAZIONE	2
FACILITA' GESTIONE OPERATIVA	2
Molteplicità DI SITI	0
FACILITA' DI MODIFICA	3
	28

Indicatore Valore

UFP	48
LLOC	2554

Indicatore Valore

FP	44.64
LLOC	2365

PrenotazioneScooter

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	4		10		40
NEIF	0				0
NEI	5	3			15
NEO	3		5		15
NEQ	4		4		16

NILF: Le tabelle consultate dal sistema sono Accessori, ClienteRegistrati, Prenotazioni e Prenotazioni_Accessori, le identifichiamo come ILF. [4 medio]

NEI: Accessori, Scooter, DataRitiro, DataConsegna, Email e Password. [5 semplici]

NEO: Visualizzazione dei dettagli sulla prenotazione. [3 medio]

NEQ: Interrogazione per il prelievo dei dati. [4 medio]

FATTORI CORRETTIVI

COMUNICAZIONE DATI	1
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	2
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	3
EFFICIENZA PER L'UTENTE FINALE	4
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITA' ELABORATIVA	0
RIUSABILITA'	3
FACILITA' INSTALLAZIONE	2
FACILITA' GESTIONE OPERATIVA	2
Molteplicità DI SITI	0
FACILITA' DI MODIFICA	3
	28

Indicatore Valore

UFP	86
LLOC	4558

Indicatore Valore

FP	79.98
LLOC	4238

Capitolo 4 - Piano di test funzionale

4.1 Piano di test funzionale per il caso d'uso "RicercaScooter"

Località	DataRitiro	DataConsegna	ElementoDiSistemaDatabase
<ul style="list-style-type: none">- Stringa non vuota con solo caratteri alfabetici- Stringa vuota o nulla [ERROR]- Stringa con caratteri speciali [ERROR]- Stringa con numeri [ERROR]- Stringa di lunghezza > 100 caratteri [ERROR]	<ul style="list-style-type: none">- Data in formato valido (aaaa-mm-gg)- Data con formato non valido [ERROR]- Data mancante [ERROR]	<ul style="list-style-type: none">- Data in formato valido aaaa-mm-gg), successiva a DataRitiro- Data in formato valido (aaaa-mm-gg), precedente a DataRitiro [ERROR]- Data con formato non valido, [ERROR]- Data mancante [ERROR]	<ul style="list-style-type: none">- Esiste almeno un'agenzia per quella località- Nessuna agenzia per quella località [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $5 \times 3 \times 4 \times 2 = 120$

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 10 (4 per Località, 2 per DataRitiro, 3 per DataConsegna, 1 per ElementoDiSistemaDatabase). Il numero di test risultante è: $(1 \times 1 \times 1 \times 1) + 10 = 11$.

Test Suite

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-Condizioni	Input	Output attesi	Post-condizioni attese
1	Tutti input validi	Località valida, DataRitiro valida, DataConsegna valida,	Il database è inizializzato correttamente, il database contiene	{Località: "Napoli", DataRitiro: "2025-06-07", DataConsegna: "2025-06-10"}	Scooter disponibili	Il sistema mostra gli scooter disponibili al cliente

6	DataRitiro con formato non valido	DataRitiro con formato non valido [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "Napoli", DataRitiro: "07/06/2025", DataConsegna: "2025-06-10"}	Error: DataRitiro in formato non valido	Il sistema richiede al cliente di reinserire i dati di ricerca
7	DataConsegna con formato non valido	DataConsegna con formato non valido [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "Napoli", DataRitiro: "2025-06-07", DataConsegna: "2025/06/2025"}	Error: DataConsegna in formato non valido	Il sistema richiede al cliente di reinserire i dati di ricerca
8	Data in formato valido (gg/mm/aaaa), precedente a DataRitiro	Data in formato valido (gg/mm/aaaa), precedente a DataRitiro [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "Napoli", DataRitiro: "2025-06-07", DataConsegna: "2025-06-05"}	Error: DataConsegna precedente a DataRitiro	Il sistema richiede al cliente di reinserire i dati di ricerca
9	DataRitiro mancante	DataRitiro mancante [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "Napoli", DataRitiro: "", DataConsegna: "2025-06-05"}	Error: DataRitiro mancante	Il sistema richiede al cliente di reinserire i dati di ricerca
10	DataConsegna mancante	DataConsegna mancante [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "Napoli", DataRitiro: "2025-06-07", DataConsegna: ""}	Error: DataConsegna mancante	Il sistema richiede al cliente di reinserire i dati di ricerca
11	Nessuna agenzia per quella località	Nessuna agenzia per quella località [ERROR]	L'utente ha selezionato la funzionalità di ricerca. Non esiste alcuna agenzia a Foggia	{Località: "Foggia", DataRitiro: "2025-06-07", DataConsegna: "2025-06-10"}	Error: non sono stati trovati scooter disponibili per i criteri di ricerca selezionati.	Il sistema richiede al cliente di reinserire i dati di ricerca

4.2 Piano di test funzionale per il caso d'uso "PrenotazioneScooter"

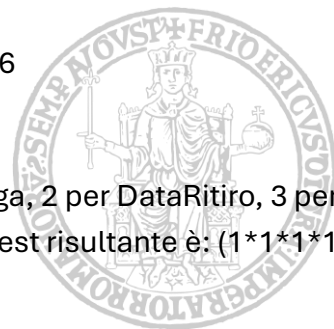
Si riporta di seguito il piano di test funzionali per la funzione di PrenotazioneScooter

Targa	DataRitiro	DataConsegna	Accessori	Email	Password	ElementoDiSistemaDatabase
- Stringa non vuota, con formato valido (e.g. AB204BA) - Stringa vuota o nulla [ERROR] - Stringa con formato errato [ERROR]	- Data in formato valido (aaaa-mm-gg) - Data con formato non valido [ERROR] - Data mancante [ERROR]	- Data in formato valido (aaaa-mm-gg), successiva a DataRitiro - Data in formato valido (aaaa-mm-gg), precedente a DataRitiro [ERROR] - Data con formato non valido, [ERROR] - Data mancante [ERROR]	- Lista di ID interi validi (es. [1, 2]) - Lista vuota [SINGLE] - Lista con caratteri non numerici [ERROR]	- Stringa con formato email valido - Stringa con formato email non valido [ERROR] - Stringa vuota o nulla [ERROR]	- Stringa non vuota, corretta per l'email fornita - Stringa vuota o nulla [ERROR] - Stringa non corretta per l'email fornita [ERROR]	- Esiste uno scooter per la targa indicata e tutti gli accessori indicati esistono nel sistema - Targa non presente nel sistema [ERROR] - Uno o più accessori non esistono nel sistema [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $3*3*4*3*3*3*3=2916$

Introduciamo i vincoli [ERROR], [SINGLE].

Il numero di test da eseguire per testare singolarmente i vincoli è 15 (2 per Targa, 2 per DataRitiro, 3 per DataConsegna, 2 per Accessori, 2 per Email, 2 per Password, 2 per ElementoDiSistemaDatabase). Il numero di test risultante è: $(1*1*1*1*1*1*1) + 15 = 16$.



Test Suite

Si noti che per migliorare l'esperienza utente e garantire una maggiore sicurezza, si è deciso che in alcuni casi specifici gli input errati genereranno semplicemente una pagina di errore. Questo comportamento evita ambiguità e impedisce l'elaborazione di richieste non valide o potenzialmente dannose.

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-Condizioni	Input	Output attesi	Post-condizioni attese
1	Tutti input validi	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	Il database è inizializzato correttamente, il database contiene uno scooter per la targa indicata, gli accessori per gli id indicati e un cliente registrato per l'email indicata. L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Prenotazione completata con successo	Il sistema registra la prenotazione
2	Stringa targa vuota o nulla	Stringa targa vuota o nulla [ERROR], DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Error: targa non valida	Il sistema mostra al cliente una pagina di errore

3	Stringa targa con formato errato	Stringa con formato errato [ERROR], DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "123-ABC", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1], Email: "user@example.com", Password: "CorrectPwd123"}	Error: targa non valida	Il sistema mostra al cliente una pagina di errore
4	DataRitiro con formato non valido	Targa valida, DataRitiro con formato non valido [ERROR], DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "01-06-2025", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Error: formato DataRitiro non valido	Il sistema mostra al cliente una pagina di errore
5	DataRitiro mancante	Targa valida, DataRitiro mancante [ERROR], DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Error: DataRitiro mancante	Il sistema mostra al cliente una pagina di errore

6	DataConsegna precedente a DataRitiro	Targa valida, DataRitiro valida, DataConsegna in formato valido ma precedente a DataRitiro [ERROR], Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-11", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Error: DataConsegna deve essere successiva a DataRitiro	Il sistema mostra al cliente una pagina di errore
7	DataConsegna con formato non valido	Targa valida, DataRitiro valida, DataConsegna con formato non valido [ERROR], Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-110", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Error: formato DataConsegna non valido	Il sistema mostra al cliente una pagina di errore
8	DataConsegna mancante	Targa valida, DataRitiro valida, DataConsegna mancante [ERROR], Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Error: DataConsegna mancante	Il sistema mostra al cliente una pagina di errore

9	Lista accessori vuota	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [], Email: "user@example.com", Password: "CorrectPwd123"}	Prenotazione completata con successo	Il sistema registra la prenotazione
10	Lista con caratteri non numerici	Targa valida, DataRitiro valida, DataConsegna valida, Lista accessori con caratteri non numerici [ERROR], Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [a, b], Email: "user@example.com", Password: "CorrectPwd123"}	Error: lista accessori con caratteri non numerici	Il sistema mostra al cliente una pagina di errore
11	Stringa email con formato non valido	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Stringa email con formato non valido [ERROR], Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "userexample.com", Password: "CorrectPwd123"}	Error: stringa email con formato errato	Il sistema chiede di reinserire l'indirizzo email

12	Stringa email vuota o nulla	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Stringa email vuota o nulla [ERROR], Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "", Password: "CorrectPwd123"}	Error: stringa email mancante	Il sistema chiede di reinserire l'indirizzo email
13	Stringa password vuota o nulla	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Stringa password vuota o nulla [ERROR], ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: ""}	Error: stringa password mancante	Il sistema chiede di reinserire la password
14	Stringa password non corretta per l'email fornita	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Stringa password non corretta per l'email fornita [ERROR],	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "IncorrectPwd123"}	Error: credenziali errate	Il sistema mostra al cliente una pagina di errore

		ElementoDiSistema Database valida				
15	Targa non presente nel sistema	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, Targa non presente nel sistema [ERROR]	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "AB405AB", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Error: non è presente alcuno scooter con la targa indicata nel sistema	Il sistema mostra al cliente una pagina di errore
16	Uno o più accessori non esistono nel sistema	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, Uno o più accessori non esistono nel sistema [ERROR]	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "AB405AB", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [9999, 10000], Email: "user@example.com", Password: "CorrectPwd123"}	Error: non è presente alcun accessorio con l'id indicato nel sistema	Il sistema mostra al cliente una pagina di errore



Capitolo 5 - Progettazione

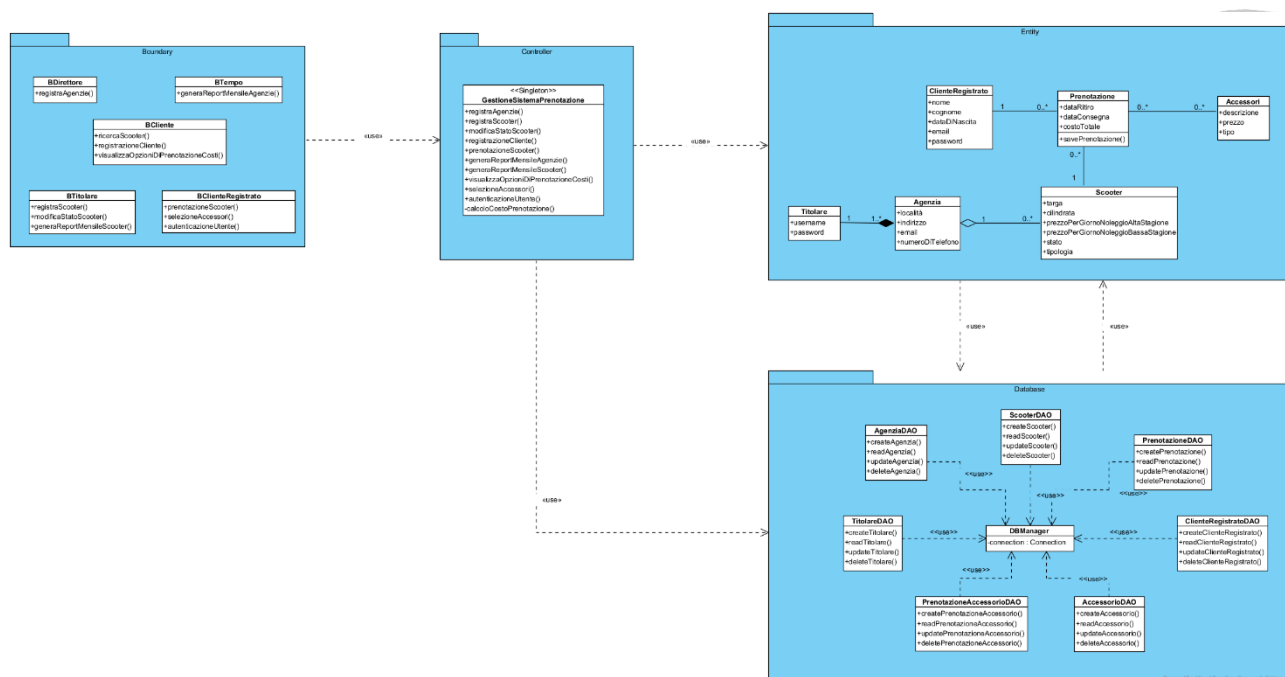
5.1 Progettazione della base di dati

5.1.1 Progettazione logica



5.2 Diagramma delle classi

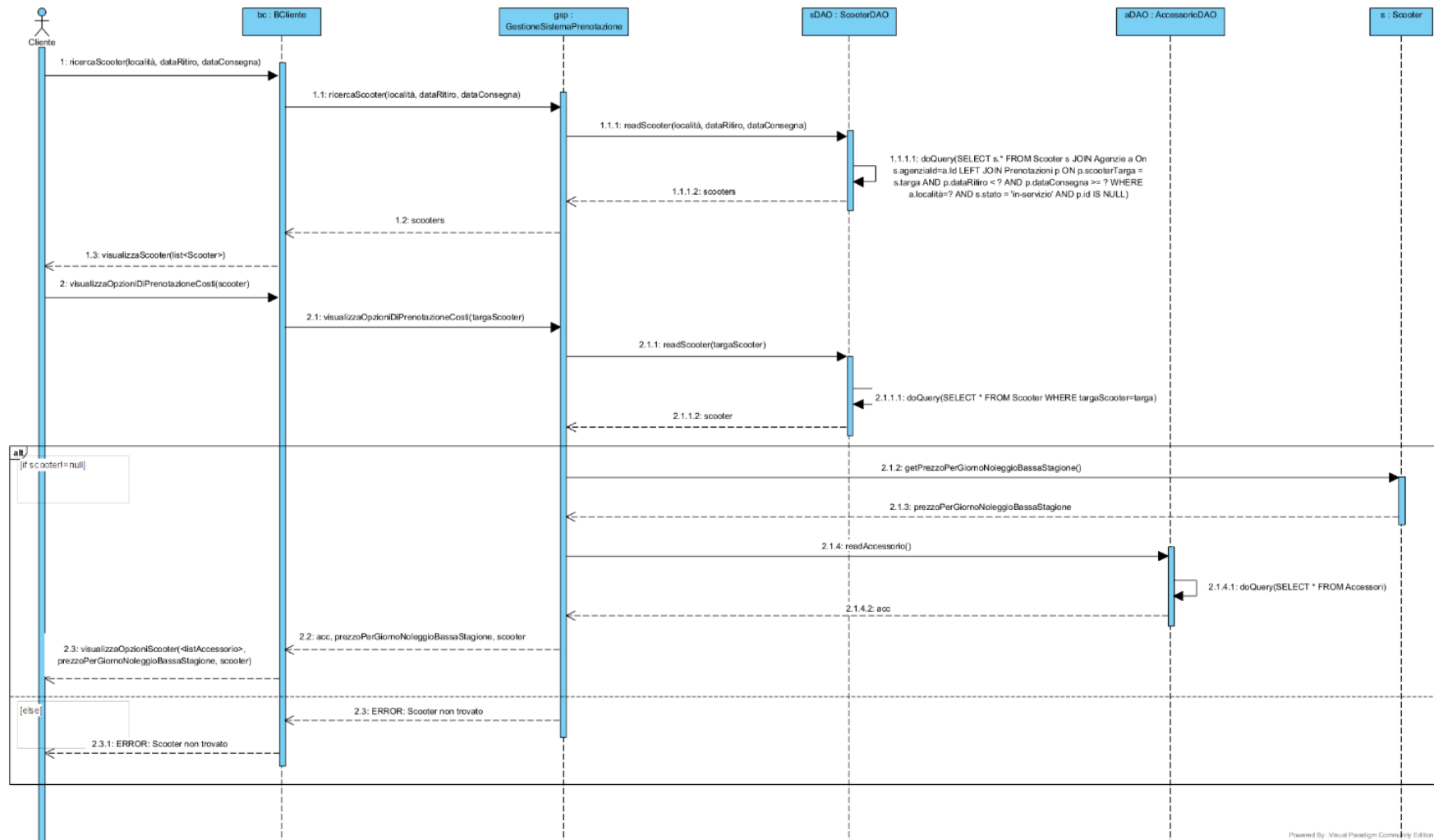
Si riporta di seguito il diagramma delle classi di progettazione.



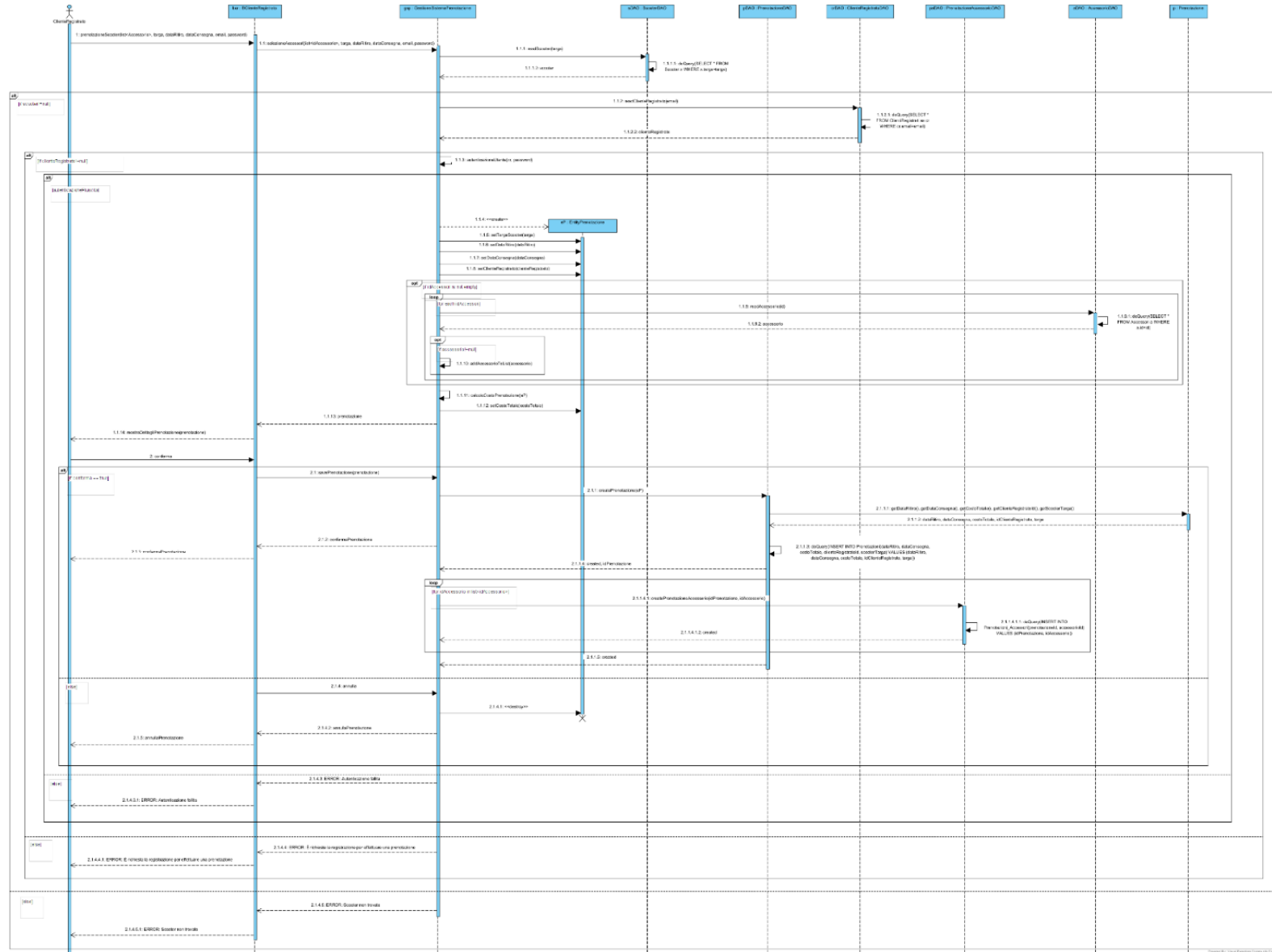
5.3 Diagrammi di sequenza

Si riportano di seguito i diagrammi di sequenza di progettazione per i casi d'uso RicercaScooter e PrenotazioneScooter.

Caso d'uso RicercaScooter



Caso d'uso Prenotazione Scooter



Capitolo 6 - Implementazione

6.1 Dipendenze di progetto

Il progetto **ciulsMeglCheUan** utilizza le seguenti dipendenze tecnologiche:

- **Spring Boot 2.7.18** - Framework principale per lo sviluppo dell'applicazione web Java;
- **Spring Boot Starter Web** - Supporto per applicazioni web e servizi RESTful con Spring MVC;
- **Spring Boot Starter Thymeleaf** - Template engine per la generazione di pagine HTML dinamiche;
- **MySQL Connector/J 9.3.0** - Driver JDBC per la connessione al database MySQL;
- **Mockito Inline 5.2.0** - Framework di mocking avanzato per test unitari;
- **Spring Boot Starter Test** - Suite di testing integrata (con esclusioni per Mockito);
- **Mockito Core 4.5.1** - Framework per la creazione di oggetti mock nei test;
- **Java 11** - Versione del linguaggio Java utilizzata;
- **Spring Boot Maven Plugin** - Plugin per build e packaging dell'applicazione.

6.2 Descrizione della struttura dell'applicazione

I package principali sono application, boundary, control, database, entity ed exception.

Il package application contiene la classe di avvio dell'applicazione. Nel package boundary troviamo le classi che gestiscono l'interazione con l'utente, cioè BCliente, BDirettore, BTempo, BTitolare e BClienteRegistrato. Il package control ospita la classe GestioneSistemaPrenotazione che centralizza la logica applicativa.

Il package entity raccoglie le entità del dominio, tra cui Agenzia, Scooter, Prenotazione, ClienteRegistrato, Titolare e Accessori. Nel package database sono presenti le classi DAO per ogni entità (AgenziaDAO, ScooterDAO, PrenotazioneDAO, ClienteRegistratoDAO, TitolareDAO, AccessoriDAO, PrenotazioneAccessoriDAO), oltre alla classe DBManager per la gestione della connessione al database.

Infine, nel package exception troviamo le eccezioni personalizzate come DAOException, DBConnectionException e OperationException, usate per gestire gli errori specifici del sistema.

6.3 JavaDoc

Si riporta di seguito la JavaDoc per i metodi ricercaScooter e selezioneAccessori.

Metodo ricercaScooter

```
1. /**
2.  * Ricerca gli scooter disponibili per una specifica località e periodo.
3.  *
4.  * <p>Effettua la validazione della località e delle date prima di procedere
5.  * con la ricerca nel database. Le date devono essere nel formato yyyy-MM-dd
6.  * e la data di ritiro deve essere precedente o uguale alla data di consegna.</p>
7.  *
```

```

8.  * @param localita La località dove cercare gli scooter disponibili
9.  * @param dataRitiro Data di inizio noleggio (formato yyyy-MM-dd)
10. * @param dataConsegna Data di fine noleggio (formato yyyy-MM-dd)
11. * @return Lista degli scooter disponibili per il periodo specificato
12. * @throws OperationException se la località non è valida, le date non sono valide
13. *                               o si verifica un errore nel database
14. *
15. * @see #isValidLocalita(String)
16. * @see #validaIntervalloDate(String, String)
17. */

```

Metodo selezioneAccessori

```

1. /**
2.  * Crea una prenotazione selezionando gli accessori desiderati e autenticando l'utente.
3.  *
4.  * <p>Questo metodo:</p>
5.  * <ol>
6.  *   <li>Valida la targa dello scooter e l'intervallo di date</li>
7.  *   <li>Verifica l'esistenza dello scooter</li>
8.  *   <li>Autentica il cliente con email e password</li>
9.  *   <li>Crea una nuova prenotazione</li>
10. *   <li>Aggiunge gli accessori selezionati (se presenti)</li>
11. *   <li>Calcola il costo totale includendo stagionalità</li>
12. * </ol>
13. *
14. * @param accId Lista degli ID degli accessori selezionati (può essere vuota)
15. * @param dataRitiro Data di ritiro nel formato yyyy-MM-dd
16. * @param dataConsegna Data di consegna nel formato yyyy-MM-dd
17. * @param email Email del cliente registrato
18. * @param password Password del cliente per l'autenticazione
19. * @param targa Targa dello scooter da prenotare
20. * @return La prenotazione creata con il costo calcolato
21. * @throws OperationException se la targa non è valida, le date non sono valide,
22. *                               lo scooter non esiste, l'autenticazione fallisce
23. *                               o un accessorio non viene trovato
24. *
25. * @see #isTargaValida(String)
26. * @see #validaIntervalloDate(String, String)
27. * @see #autenticazioneUtente(String, String)
28. * @see #calcoloCostoPrenotazione(Prenotazione)
29. */

```



Si noti che la JavaDoc è disponibile, nella sua interezza, nel repository del progetto al link https://github.com/Ae0nix/Progetto_Ingegneria_del_Software.git.

6.4 Costi

6.4.1 Confronto con stima dei costi effettuata

Per il conteggio di LOC e LLOC si è utilizzato un tool apposito, cloc. Il risultato dell'analisi dei LOC sul progetto è, dunque, riportato di seguito.

Language

Files

Blank

Comment

Code

Java	25	452	521	1602
CSS	7	166	26	1222
HTML	8	36	31	420
XML	8	0	0	398
Bourne Shell	1	26	48	185
DOS Batch	1	15	0	134
Maven	1	8	0	85
Properties	3	1	16	8
Text	1	0	0	4
SUM:	55	704	642	4058

Nel nostro caso, la stima dei costi è stata effettuata attraverso l'analisi dei Punti Funzione e delle linee di codice attese (LLOC) per ciascuna funzionalità prevista, come RicercaScooter e PrenotazioneScooter. Tali valori rappresentano una previsione teorica dell'effort necessario per lo sviluppo.

Tuttavia, il confronto con i dati effettivi riportati nella tabella delle tecnologie (suddivisi per linguaggio di programmazione) evidenzia delle discrepanze, dovute al fatto che la stima iniziale non sempre riflette in modo puntuale la reale distribuzione e quantità di codice prodotto.

Ciò può dipendere da diversi fattori, quali l'adozione di librerie esterne o l'ottimizzazione del codice in fase di implementazione. In questo modo, la tabella delle tecnologie fornisce una fotografia dettagliata del prodotto finito, mentre la stima dei costi offre un'indicazione utile nella fase di pianificazione e previsione progettuale.

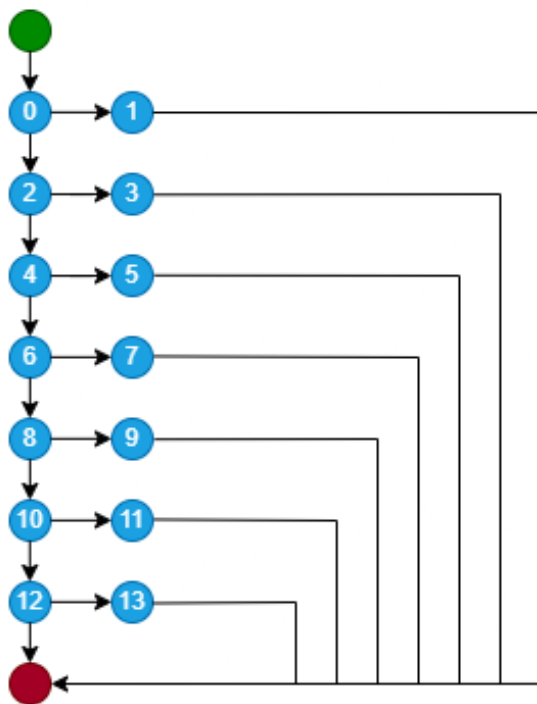
Capitolo 7 - Testing

7.1 Test strutturale

7.1.1 Complessità ciclomatica

Metodo ricercaScooter

Si mostra di seguito il Control Flow Graph per il metodo ricercaScooter() della classe GestioneSistemaPrenotazione.



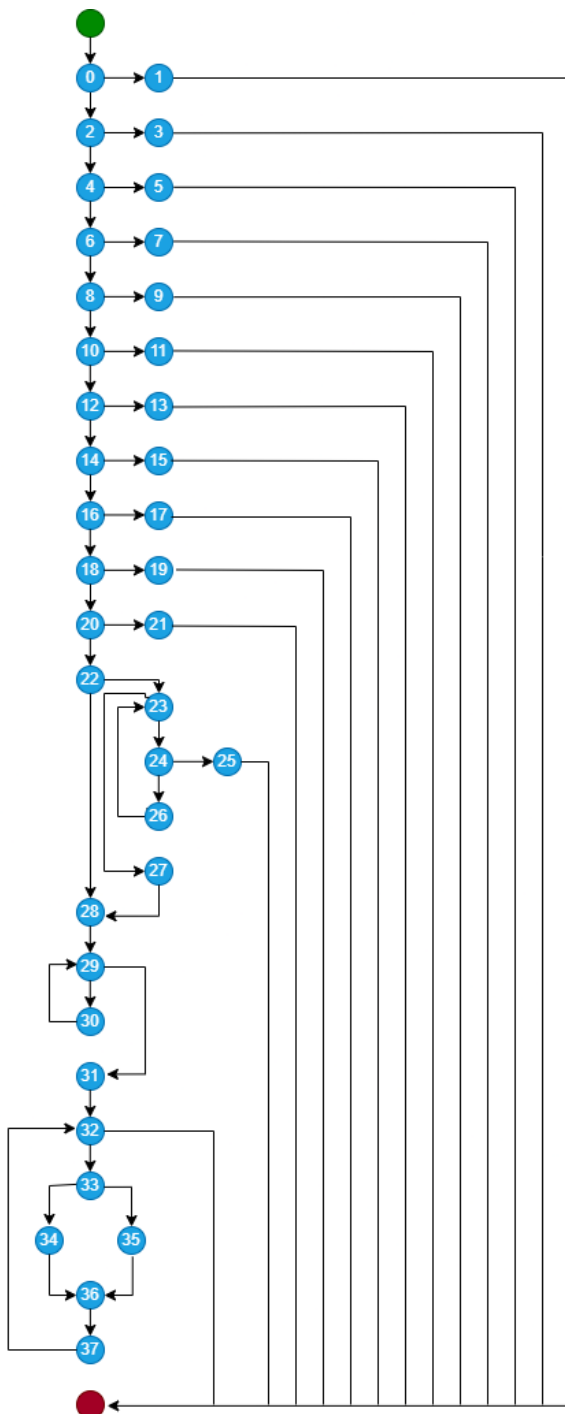
Numero ciclomatico:

- Numero di regioni del grafo = **8**
- Numero di nodi predicato (0,2,4,6,8,10,12) + 1 = **8**
- Numero di archi – Numero di nodi + 2 * Componenti del grafo disconnesse = (22 - 16) + 2 = **8**

Cammini di base:

- 1.0-1
- 2.0-2-3
- 3.0-2-4-5
- 4.0-2-4-6-7
- 5.0-2-4-6-8-9
- 6.0-2-4-6-8-10-11
- 7.0-2-4-6-8-10-12
- 8.0-2-4-6-8-10-12-13



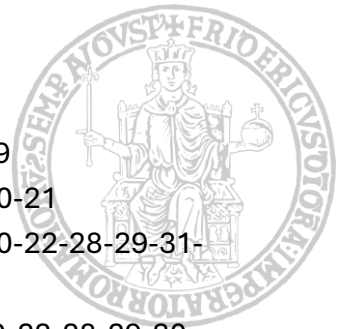


Numero cicломatico:

- Numero di regioni del grafo: **18**
 - Numero nodi predicato (0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 30, 33) + 1 = **18**
 - Numero di archi – Numero di nodi + 2 *
- Componenti del grafo disconnesse = $(54 - 39) + 2 = 18$

Cammini di base:

1. 0-1
2. 0-2-3
3. 0-2-4-5
4. 0-2-4-6-7
5. 0-2-4-6-8-9
6. 0-2-4-6-8-10-11
7. 0-2-4-6-8-10-12-13
8. 0-2-4-6-8-10-12-14-15
9. 0-2-4-6-8-10-12-14-16-17
10. 0-2-4-6-8-10-12-14-16-18-19
11. 0-2-4-6-8-10-12-14-16-18-20-21
12. 0-2-4-6-8-10-12-14-16-18-20-22-28-29-31-32-33-34-36-37-32
13. 0-2-4-6-8-10-12-14-16-18-20-22-28-29-30-29-31-32-33-35-36-37-32
14. 0-2-4-6-8-10-12-14-16-18-20-22-28-29-30-29-31-32-33-34-36-37-32
15. 0-2-4-6-8-10-12-14-16-18-20-22-23-24-25
16. 0-2-4-6-8-10-12-14-16-18-20-22-23-27-28-29-31-32-33-35-36-37-32
17. 0-2-4-6-8-10-12-14-16-18-20-22-28-29-31-32-33-35-36-37-32
18. 0-2-4-6-8-10-12-14-16-18-20-22-23-24-26-23-27-28-29-31-32-33-34-36-37-32



7.1.2 Test di unità

Si definisce una test suite secondo il criterio di adeguatezza della copertura dei cammini di base per ciascuno dei metodi della sottosezione precedente.

Metodo ricercaScooter()

Test case ID	Descrizione	Pre-Condizioni	Input	Output attesi
1	Località vuota o null	Sistema inizializzato correttamente	{Località: "", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può essere vuota."
2	Località troppo lunga	Sistema inizializzato correttamente	{Località: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può superare i 100 caratteri."
3	Località con caratteri numerici	Sistema inizializzato correttamente	{Località: "Roma123", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può contenere numeri."
4	Località con caratteri speciali	Sistema inizializzato correttamente	{Località: "Roma@#\$%", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località può contenere solo lettere, spazi e apostrofi."
5	Data con formato non valido	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "15/01/2025", DataConsegna: "2025-01-20"}	OperationException: "Formato data non valido (deve essere yyyy-MM-dd)."
6	Data ritiro successiva a data consegna	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "2025-01-25", DataConsegna: "2025-01-20"}	OperationException: "La data di ritiro deve essere precedente o uguale alla data di consegna."
7	Ricerca scooter con successo	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	Lista di scooter disponibili
8	Errore database	Database non disponibile o errore di connessione	{Località: "Roma", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "Errore database"

Si riporta di seguito il risultato dell'esecuzione dei test di unità per il metodo `ricercaScooter()`.

Test case ID	Descrizione	Pre-Condizioni	Input	Output attesi	Output ottenuti	Esito (FAIL, PASS)
1	Località vuota o null	Sistema inizializzato correttamente	{Località: "", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può essere vuota."	OperationException: "La località non può essere vuota."	PASS
2	Località troppo lunga	Sistema inizializzato correttamente	{Località: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può superare i 100 caratteri."	OperationException: "La località non può superare i 100 caratteri."	PASS
3	Località con caratteri numerici	Sistema inizializzato correttamente	{Località: "Roma123", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può contenere numeri."	OperationException: "La località non può contenere numeri."	PASS
4	Località con caratteri speciali	Sistema inizializzato correttamente	{Località: "Roma@#\$", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località può contenere solo lettere, spazi e apostrofi."	OperationException: "La località può contenere solo lettere, spazi e apostrofi."	PASS
5	Data con formato non valido	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "15/01/2025", DataConsegna: "2025-01-20"}	OperationException: "Formato data non valido (deve essere yyyy-MM-dd)."	OperationException: "Formato data non valido (deve essere yyyy-MM-dd)."	PASS

6	Data ritiro successiva a data consegna	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "2025-01-25", DataConsegna: "2025-01-20"}	OperationException: "La data di ritiro deve essere precedente o uguale alla data di consegna."	OperationException: "La data di ritiro deve essere precedente o uguale alla data di consegna."	PASS
7	Ricerca scooter con successo	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	Lista di scooter disponibili	Lista di scooter disponibili	PASS
8	Errore database	Database non disponibile o errore di connessione	{Località: "Roma", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "Errore database"	OperationException: "Errore database"	PASS



Metodo selezioneAccessori

Test case ID	Descrizione	Pre-Condizioni	Input	Output attesi
1	Località vuota o null	Sistema inizializzato correttamente	{Località: "", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può essere vuota."
2	Località troppo lunga	Sistema inizializzato correttamente	{Località: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può superare i 100 caratteri."
3	Località con caratteri numerici	Sistema inizializzato correttamente	{Località: "Roma123", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può contenere numeri."
4	Località con caratteri speciali	Sistema inizializzato correttamente	{Località: "Roma@#\$%", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località può contenere solo lettere, spazi e apostrofi."
5	Data con formato non valido	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "15/01/2025", DataConsegna: "2025-01-20"}	OperationException: "Formato data non valido (deve essere yyyy-MM-dd)."
6	Data ritiro successiva a data consegna	Sistema inizializzato correttamente	{Località: "Roma", DataRitiro: "2025-01-25", DataConsegna: "2025-01-20"}	OperationException: "La data di ritiro deve essere precedente o uguale alla data di consegna."
7	Targa di uno scooter non presente nel sistema	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345", targa: "ZZ999CW"}	OperationException: "Scooter non trovato"
8	Il cliente non è registrato	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26",	OperationException: "È richiesta la registrazione per

	nel sistema		email: "pippo@pippo.ito", password: "pippo", targa: "AB123CD" }	effettuare una prenotazione"
9	La password immessa è sbagliata	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: " 2025-01-26", email: "alex.marino@email.it", password: "pippoPippo.ito", targa: "AB123CD"}	OperationExceptio n: "Credenziali errate"
10	Errore database	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: " 2025-01-26", email: "alex.marino@email.it", password: "pippoPippo.ito", targa: "AB123CD"}	OperationExceptio n: "Errore lettura ClienteRegistrato"
11	Errore database	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: " 2025-01-26", email: "alex.marino@email.it", password: "pippoPippo.ito", targa: "AB123CD"}	OperationExceptio n: "Errore connessione al database"
12	Lista di id Accessori vuota < o nulla e alta stagione	Sistema inizializzato correttamente	{Lista_accessori: , dataRitiro: "2025-07-25", dataConsegna: " 2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione
13	Lista di id Accessori vuota o nulla e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: , dataRitiro: "2025-01-25", dataConsegna: " 2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione
14	Prenotazione corretta in alta stagione	Sistema inizializzato correttamente	{Lista_accessori: [1], dataRitiro: "2025-07-25", dataConsegna: " 2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione
15	Id di un accessorio non presente in database	Sistema inizializzato correttamente	{Lista_accessori: [30], dataRitiro: "2025-07-25", dataConsegna: " 2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	OperationExceptio n: "Accessorio non trovato"



16	Lista accessori vuota e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: [30], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione
17	Lista di id Accessori vuota o nulla e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: , dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione
18	Prenotazione corretta in alta stagione con accessori presenti	Sistema inizializzato correttamente	{Lista_accessori: [1], dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione



7	Targa di uno scooter non presente nel sistema	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345", targa: "ZZ999CW"}	OperationException: "Scooter non trovato"	OperationException: "Scooter non trovato"	PASS
8	Il cliente non è registrato nel sistema	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "pippo@pippozzo.it", password: "pippo", targa: "AB123CD"}	OperationException: "È richiesta la registrazione per effettuare una prenotazione"	OperationException: "È richiesta la registrazione per effettuare una prenotazione"	PASS
9	La password immessa è sbagliata	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "pippoPippozzo!", targa: "AB123CD"}	OperationException: "Credenziali errate"	OperationException: "Credenziali errate"	PASS
10	Errore database	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "pippoPippozzo!", targa: "AB123CD"}	OperationException: "Errore lettura ClienteRegistrato"	OperationException: "Errore lettura ClienteRegistrato"	PASS
11	Errore database	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "pippoPippozzo!", targa: "AB123CD"}	OperationException: "Errore connessione al database"	OperationException: "Errore connessione al database"	PASS

12	Lista di id Accessori vuota < o nulla e alta stagione	Sistema inizializzato correttamente	{Lista_accessori: , dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	FAIL
13	Lista di id Accessori vuota o nulla e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: , dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	FAIL
14	Prenotazione corretta in alta stagione	Sistema inizializzato correttamente	{Lista_accessori: [1], dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS
15	Id di un accessorio o non presente in database	Sistema inizializzato correttamente	{Lista_accessori: [30], dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	OperationException: "Accessorio non trovato"	OperationException: "Accessorio non trovato"	PASS
16	Lista accessori vuota e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: [], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS

17	Lista di id Accessori vuota o nulla e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: [], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS
18	Prenotazione corretta in alta stagione con accessori presenti	Sistema inizializzato correttamente	{Lista_accessori: [1], dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS

A seguito dell'esecuzione di alcuni test con esito negativo, è stata intrapresa un'attività di debugging approfondita che ha consentito di individuare un'anomalia nel codice. L'errore era dovuto a un controllo non sicuro sulla lista di interi associata agli accessori, inizialmente gestito mediante la condizione `!accessori.isEmpty()`. Tuttavia, in presenza di una lista non inizializzata, tale controllo causava una *NullPointerException*.

Per risolvere il problema, la condizione è stata sostituita con `accessori != null`, garantendo una verifica preventiva dell'inizializzazione della lista prima di accedere ai suoi metodi.

Si riportano dunque di seguito i risultati dell'attività di testing effettuata dopo la correzione del difetto sopracitato.

Test case ID	Descrizione	Pre-Condizioni	Input	Output attesi	Output ottenuti	Esito (FAIL, PASS)
1	Località vuota o null	Sistema inizializzato correttamente	{Località: "", DataRitiro: "2025-01-15", DataConsegna: "2025-01-20"}	OperationException: "La località non può essere vuota."	OperationException: "La località non può essere vuota."	PASS

8	Il cliente non è registrato nel sistema	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "pippo@pippozzo.it", password: "pippo", targa: "AB123CD" }	OperationException: "È richiesta la registrazione per effettuare una prenotazione"	OperationException: "È richiesta la registrazione per effettuare una prenotazione"	PASS
9	La password immessa è sbagliata	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "pippoPippozzo!", targa: "AB123CD" }	OperationException: "Credenziali errate"	OperationException: "Credenziali errate"	PASS
10	Errore database	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "pippoPippozzo!", targa: "AB123CD" }	OperationException: "Errore lettura ClienteRegistrato"	OperationException: "Errore lettura ClienteRegistrato"	PASS
11	Errore database	Sistema inizializzato correttamente	{Lista_accessori: [1,2,3], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "pippoPippozzo!", targa: "AB123CD" }	OperationException: "Errore connessione al database"	OperationException: "Errore connessione al database"	PASS
12	Lista di id Accessori vuota < o nulla e alta stagione	Sistema inizializzato correttamente	{Lista_accessori: , dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD" }	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS

13	Lista di id Accessori vuota o nulla e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: , dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS
14	Prenotazione corretta in alta stagione	Sistema inizializzato correttamente	{Lista_accessori: [1], dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS
15	Id di un accessorio o non presente in database	Sistema inizializzato correttamente	{Lista_accessori: [30], dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	OperationException: "Accessorio non trovato"	OperationException: "Accessorio non trovato"	PASS
16	Lista accessori vuota e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: [], dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS
17	Lista di id Accessori vuota o nulla e bassa stagione	Sistema inizializzato correttamente	{Lista_accessori: [] , dataRitiro: "2025-01-25", dataConsegna: "2025-01-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS

18	Prenotazione corretta in alta stagione con accessori presenti	Sistema inizializzato correttamente	{Lista_accessori: [1], dataRitiro: "2025-07-25", dataConsegna: "2025-07-26", email: "alex.marino@email.it", password: "alex345!", targa: "AB123CD"}	Riepilogo Prenotazione	Riepilogo Prenotazione	PASS
----	---------------------------------------------------------------	-------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------	------------------------	------------------------	------

7.2 Test funzionale

Si descrivono di seguito i risultati dei test funzionali precedentemente specificati.

Caso d'uso RicercaScooter

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-Condizioni	Input	Output attesi	Output ottenuti	Post-condizioni attese	Post-condizioni e ottenute	Esito (FAIL, PASS)
1	Tutti input validi	Località valida, DataRitiro valida, DataConsegna valida, ElementoDiSistemaDatabase valido	Il database è inizializzato correttamente, il database contiene un'agenzia della località indicata	{Località: "Napoli", DataRitiro: "2025-06-07", DataConsegna: "2025-06-10"}	Scooter disponibili	Scooter disponibili	Il sistema mostra gli scooter disponibili al cliente	Il sistema mostra gli scooter disponibili al cliente	PASS
2	Stringa località vuota o nulla	Stringa località vuota o nulla [ERROR], DataRitiro	L'utente ha selezionato la	{Località: "", DataRitiro: "2025-06-07",	Error: stringa località	Error: località non specificata	Il sistema richiede al cliente di reinserire i	Il sistema richiede al cliente di	PASS

		valida,vuota DataConsegna valida, ElementoDiSiste maDatabase valido	funzionalità di ricerca	DataConsegna: "2025-06-10"}	non specificata		dati di ricerca	reinserire i dati di ricerca	
3	Stringa località con caratteri speciali	Stringa località con caratteri speciali [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "Napol!", DataRitiro: "2025-06-07", DataConsegna: "2025-06-10"}	Error: la località può contenere solo lettere, spazi e apostrofi.	Error: la località può contenere solo lettere, spazi e apostrofi.	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS
4	Stringa località con numeri	Stringa località con numeri [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "N4p0l1", DataRitiro: "2025-06-07", DataConsegna: "2025-06-10"}	Error: stringa località non può contenere numeri	Error: stringa località non può contenere numeri	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS
5	Stringa località di lunghezza > 100 caratteri	Stringa località di lunghezza > 100 caratteri [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: " aaaaaaaaaaaaaa aaaaaaaaaaaaaa aaaa aaaaaaaaaaaaaa aaaaaaaaaaaaaa aaaa aaaaaaaaaaaaaa aaaaaaaaaaaaaa aaaaaaaaaaaaaa	Error: stringa località non può superare i 100 caratteri	Error: stringa località non può superare i 100 caratteri	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS

				aaaa aaaaa”, DataRitiro: “2025-06-07”, DataConsegna: “2025-06-10”}					
6	DataRitiro con formato non valido	DataRitiro con formato non valido [ERROR]	L’utente ha selezionato la funzionalità di ricerca	{Località: ”Napoli”, DataRitiro: “07/06/2025”, DataConsegna: “2025-06-10”}	Error: DataRitiro in formato non valido	Error: DataRitiro in formato non valido	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS
7	DataConse gna con formato non valido	DataConsegna con formato non valido [ERROR]	L’utente ha selezionato la funzionalità di ricerca	{Località: ”Napoli”, DataRitiro: “2025-06-07”, DataConsegna: “2025/06/2025”}	Error: DataConse gna in formato non valido	Error: DataConse gna in formato non valido	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS
8	Data in formato valido (gg/mm/aa aa), precedente a DataRitiro	Data in formato valido (gg/mm/aaaa), precedente a DataRitiro [ERROR]	L’utente ha selezionato la funzionalità di ricerca	{Località: ”Napoli”, DataRitiro: “2025-06-07”, DataConsegna: “2025-06-05”}	Error: DataConse gna precedente a DataRitiro	Error: DataConse gna precedente a DataRitiro	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS
9	DataRitiro mancante	DataRitiro mancante [ERROR]	L’utente ha selezionato la	{Località: ”Napoli”, DataRitiro: “”,	Error: DataRitiro mancante	Error: DataRitiro mancante	Il sistema richiede al cliente di	Il sistema richiede al cliente	PASS

			funzionalità di ricerca	DataConsegna: "2025-06-05"}			reinserire i dati di ricerca	di reinserire i dati di ricerca	
10	DataConsegna mancante	DataConsegna mancante [ERROR]	L'utente ha selezionato la funzionalità di ricerca	{Località: "Napoli", DataRitiro: "2025-06-07", DataConsegna: ""}	Error: DataConsegna mancante	Error: DataConsegna mancante	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS
11	Nessuna agenzia per quella località	Nessuna agenzia per quella località [ERROR]	L'utente ha selezionato la funzionalità di ricerca. Non esiste alcuna agenzia a Foggia	{Località: "Foggia", DataRitiro: "2025-06-07", DataConsegna: "2025-06-10"}	Error: non sono stati trovati scooter disponibili per i criteri di ricerca selezionati.	Error: non sono stati trovati scooter disponibili per i criteri di ricerca selezionati.	Il sistema richiede al cliente di reinserire i dati di ricerca	Il sistema richiede al cliente di reinserire i dati di ricerca	PASS



Caso d'uso PrenotazioneScooter

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-Condizioni	Input	Output attesi	Output ottenuti	Post-condizioni attese	Post-condizioni ottenute	Esito(FAIL, PASS)
1	Tutti input validi	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	Il database è inizializzato correttamente, il database contiene uno scooter per la targa indicata, gli accessori per gli id indicati e un cliente registrato per l'email indicata. L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.com", Password: "CorrectPwd123"}	Riepilogo prenotazione	Riepilogo prenotazione	Il sistema mostra la prenotazione	Il sistema mostra la prenotazione	PASS
2	Stringa targa vuota o nulla	Stringa targa vuota o nulla [ERROR], DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida,	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "user@example.co	Error: targa non valida	Error: targa non valida	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS

		ElementoDiSistema Database valida		m”, Password: “CorrectPwd123”}					
3	Stringa targa con formato errato	Stringa con formato errato [ERROR], DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L’utente ha selezionato la funzionalità di prenotazione	{Targa: “123-ABC”, DataRitiro: “2025-06-01”, DataConsegna: “2025-06-10”, Accessori: [1], Email: “luca.ferrari@email.it”, Password: “luca123”}	Error: targa non valida	Error: targa non valida	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
4	DataRitiro con formato non valido	Targa valida, DataRitiro con formato non valido [ERROR], DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L’utente ha selezionato la funzionalità di prenotazione	{Targa: “KL890MN”, DataRitiro: “01-06-2025”, DataConsegna: “2025-06-10”, Accessori: [1, 2], Email: “luca.ferrari@email.it”, Password: “luca123”}	Error: formato DataRitiro non valido	Error: formato DataRitiro non valido	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
5	DataRitiro mancante	Targa valida, DataRitiro mancante [ERROR], DataConsegna valida, Accessori valida, Email valida,	L’utente ha selezionato la funzionalità di prenotazione	{Targa: “KL890MN”, DataRitiro: “”, DataConsegna: “2025-06-10”, Accessori: [1, 2], Email: “luca.ferrari@email	Error: DataRitiro mancante	Error: DataRitiro mancante	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS

		Password valida, ElementoDiSistema Database valida		.it", Password: "luca123"}					
6	DataConsegna precedente a DataRitiro	Targa valida, DataRitiro valida, DataConsegna in formato valido ma precedente a DataRitiro [ERROR], Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-11", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "luca.ferrari@email.it", Password: "luca123"}	Error: DataConsegna deve essere successiva a DataRitiro	Error: DataConsegna deve essere successiva a DataRitiro	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
7	DataConsegna con formato non valido	Targa valida, DataRitiro valida, DataConsegna con formato non valido [ERROR], Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-110", Accessori: [1, 2], Email: "luca.ferrari@email.it", Password: "luca123"}	Error: formato DataConsegna non valido	Error: formato DataConsegna non valido	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
8	DataConsegna mancante	Targa valida, DataRitiro valida, DataConsegna mancante [ERROR], Accessori valida, Email valida, Password valida,	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "", Accessori: [1, 2], Email: "luca.ferrari@email	Error: DataConsegna mancante	Error: DataConsegna mancante	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS

		ElementoDiSistema Database valida		.it", Password: "luca123"}					
9	Lista accessori vuota	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [], Email: "luca.ferrari@email.it", Password: "luca123"}	Riepilogo prenotazione	Riepilogo prenotazione	Il sistema mostra la prenotazione	Il sistema mostra la prenotazione	PASS
10	Lista con caratteri non numerici	Targa valida, DataRitiro valida, DataConsegna valida, Lista accessori con caratteri non numerici [ERROR], Email valida, Password valida, ElementoDiSistema Database valida	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [a, b], Email: "luca.ferrari@email.it", Password: "luca123"}	Error: lista accessori con caratteri non numerici	Error: lista accessori con caratteri non numerici	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
11	Stringa email con formato non valido	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Stringa email con formato non valido [ERROR], Password valida,	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "KL890MN", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "luca.ferrariemail.it	Error: è richiesta la registrazione per proseguire	Error: è richiesta la registrazione e per proseguire	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS

		ElementoDiSistema Database valida		”, Password: “luca123”}					
12	Stringa email vuota o nulla	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Stringa email vuota o nulla [ERROR], Password valida, ElementoDiSistema Database valida	L’utente ha selezionato la funzionalità di prenotazione	{Targa: “KL890MN”, DataRitiro: “2025- 06-01”, DataConsegna: “2025-06-10”, Accessori: [1, 2], Email: “”, Password: “CorrectPwd123”}	Error: è richiesta la registraz ione per prosegu ire	Error: è richie sta la regist razion e per prose guire	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
13	Stringa password vuota o nulla	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Stringa password vuota o nulla [ERROR], ElementoDiSistema Database valida	L’utente ha selezionato la funzionalità di prenotazione	{Targa: “KL890MN”, DataRitiro: “2025- 06-01”, DataConsegna: “2025-06-10”, Accessori: [1, 2], Email: “user@example.co m”, Password: “”}	Error: è richiesta la registraz ione per prosegu ire	Error: è richie sta la regist razion e per prose guire	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
14	Stringa password non corretta per l’email fornita	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Stringa password non corretta per	L’utente ha selezionato la funzionalità di prenotazione	{Targa: “KL890MN”, DataRitiro: “2025- 06-01”, DataConsegna: “2025-06-10”, Accessori: [1, 2], Email: “luca.ferrari@email	Error: è richiesta la registraz ione per prosegu ire	Error: è richie sta la regist razion e per prose guire	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS

		l'email fornita [ERROR], ElementoDiSistema Database valida		.it", Password: "luca"}					
15	Targa non presente nel sistema	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, Targa non presente nel sistema [ERROR]	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "AB405AB", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [1, 2], Email: "luca.ferrari@email.it", Password: "luca123"}	Error: non è presente e alcuno scooter con la targa indicata nel sistema	Error: non è presente alcuno scooter con la targa indicata nel sistema	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS
16	Uno o più accessori non esistono nel sistema	Targa valida, DataRitiro valida, DataConsegna valida, Accessori valida, Email valida, Password valida, Uno o più accessori non esistono nel sistema [ERROR]	L'utente ha selezionato la funzionalità di prenotazione	{Targa: "AB405AB", DataRitiro: "2025-06-01", DataConsegna: "2025-06-10", Accessori: [9999, 10000], Email: "luca.ferrari@email.it", Password: "luca123"}	Error: non è presente e alcuno accessorio con l'id indicato nel sistema	Error: non è presente alcuno accessorio con l'id indicato nel sistema	Il sistema mostra al cliente una pagina di errore	Il sistema mostra al cliente una pagina di errore	PASS