

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
CORSO DI LAUREA IN INGEGNERIA INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE
PROF. S. RUSSO - A.A. 2020 - 21

Progetto

Applicazione per il noleggio di auto

Gruppo: **Turbo F**

Studenti:

Luca Petracca

Gianluca Pepe

Riccardo Motta

Indice

1. SPECIFICHE INFORMALI.....	3
1.1 SPECIFICHE INFORMALI D'UTENTE.....	3
2. ANALISI E SPECIFICA DEI REQUISITI	4
2.1 ANALISI NOMI-VERBI	4
2.2 REVISIONE DEI REQUISITI	4
2.3 GLOSSARIO DEI TERMINI.....	5
2.4 CLASSIFICAZIONE DEI REQUISITI.....	6
2.4.1 REQUISITI FUNZIONALI	6
2.4.2 REQUISITI SUI DATI	7
2.4.3 VINCOLI / ALTRI REQUISITI	8
2.5 MODELLAZIONE DEI CASI D'USO	8
2.5.1 ATTORI E CASI D'USO.....	8
2.6 DIAGRAMMA DELLE CLASSI	12
2.6.1 DIAGRAMMA DELLE CLASSI RAFFINATO	13
2.7 DIAGRAMMI DI SEQUENZA	14
3. STIMA DEI COSTI	17
4. PIANO DI TEST FUNZIONALE.....	20
5. PROGETTAZIONE	30
5.1 DIAGRAMMA DELLE CLASSI	30
5.2 DIAGRAMMI DI SEQUENZA	32
6. IMPLEMENTAZIONE	35
7. TESTING	36
7.1 TEST FUNZIONALE.....	36
7.2 TEST STRUTTURALE	44
7.2.1 COMPLESSITÀ CICLOMATICA.....	44

1. Specifiche Informali

Specifiche informali d'utente

Si vuole realizzare un sistema informatico per una società di noleggio auto. Sul sistema agiscono gli operatori della società di noleggio, che gestiscono il parco auto, e i guidatori, che possono verificare la disponibilità (e i prezzi) delle auto e procedere al noleggio. Il sistema deve registrare per ogni veicolo: la targa, il prezzo fissato per giorno-noleggio, il numero di passeggeri per cui l'auto è omologata, l'alimentazione (benzina/gasolio) e la potenza del motore. La compagnia gestisce i segmenti di veicoli B, C e D, che corrispondono alle auto small, medium e large.

Gli operatori possono aggiungere o modificare i dati relativi ai veicoli della compagnia. Poiché il sistema deve registrare tutte le auto che sono state impiegate per il noleggio, non è possibile rimuovere le auto dal sistema ma i funzionari all'atto di modifica possono alterare lo stato dei veicoli tra in-servizio o dismesso.

Gli utenti possono ricercare sul sistema le auto in base all'intervallo temporale di noleggio (data ritiro e data consegna) e al segmento desiderato. Il sistema deve mostrare loro le auto per il segmento richiesto che sono disponibili nel periodo indicato: un'auto è effettivamente disponibile per il noleggio se risulta al momento della ricerca come in-servizio e se non esistono altre prenotazioni che la impegnano anche parzialmente con le date indicate per la prenotazione.

Un utente che non possiede le credenziali può registrarsi in ogni momento sulla piattaforma specificando: nome, cognome, data di nascita, email, anno di conseguimento, scadenza e numero della patente di guida.

Un utente registrato può interagire col sistema per noleggiare un'auto. Il noleggio consiste di una fase di ricerca di un'auto, seguita dalla specifica di accessori da includere alla prenotazione: gli accessori si distinguono in servizi assicurativi e optional auto. Un utente può scegliere al più un servizio assicurativo accessorio, perché tali servizi vanno in conflitto tra loro (ad esempio un servizio assicurativo prevede una riduzione della franchigia, un altro la elimina); d'altro canto, non esistono limiti di scelta per l'utente riguardo gli optional auto.

Tutti 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 di tutti i servizi accessori scelti.

Infine, all'atto della prenotazione è possibile aggiungere gratuitamente un ulteriore utente tra quelli registrati nel sistema associandolo alla prenotazione come guidatore supplementare.

2. Analisi e specifica dei requisiti

2.1 Analisi nomi-verbi

Si vuole realizzare un sistema informatico per una società di noleggio **auto**. Sul sistema agiscono gli operatori della società di noleggio, che gestiscono il parco auto, e i guidatori, che possono **verificare la disponibilità** (e i prezzi) delle auto e **procedere al noleggio**.

Il sistema deve **registrare** per ogni veicolo: **la targa, il prezzo fissato per giorno-noleggio, il numero di passeggeri per cui l'auto è omologata, l'alimentazione (benzina/gasolio) e la potenza del motore**. La compagnia gestisce i **segmenti di veicoli B, C e D**, che corrispondono alle auto small, medium e large. Gli **operatori** possono **aggiungere o modificare i dati** relativi ai veicoli della compagnia. Poiché il sistema deve registrare tutte le auto che sono state impiegate per il **noleggio**, non è possibile rimuovere le auto dal sistema ma i funzionari all'atto di modifica possono **alterare lo stato dei veicoli tra in-servizio o dismesso**. Gli **utenti** possono **ricercare sul sistema le auto** in base all'intervallo temporale di noleggio (**data ritiro e data consegna**) e al segmento desiderato. Il sistema deve mostrare loro le auto per il segmento richiesto che sono disponibili nel periodo indicato: un'auto è effettivamente disponibile per il noleggio se risulta al momento della ricerca come in-servizio e se non esistono altre prenotazioni che la impegnano anche parzialmente con le date indicate per la prenotazione.

Un utente che non possiede le credenziali **può registrarsi in ogni momento sulla piattaforma** specificando: **nome, cognome, data di nascita, email, anno di conseguimento, scadenza e numero della patente di guida**. Un **utente registrato** può **interagire col sistema per noleggiare un'auto**. Il noleggio consiste di una fase di **ricerca di un'auto**, seguita dalla **specifica di accessori** da includere alla prenotazione: gli **accessori** si distinguono in **servizi assicurativi e optional auto**. Un utente può scegliere al più un servizio assicurativo accessorio, perché tali servizi vanno in conflitto tra loro (ad esempio un servizio assicurativo prevede una riduzione della **franchigia**, un altro la elimina); d'altro canto, non esistono limiti di scelta per l'utente riguardo gli optional auto.

Tutti 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 di tutti i servizi accessori scelti. Infine, all'atto della prenotazione è possibile **aggiungere gratuitamente un ulteriore utente** tra quelli registrati nel sistema associandolo alla prenotazione come **guidatore supplementare**.

LEGENDA:

Classe

Attributo

Funzionalità

Attore

Classe-Attore

2.2 Revisione dei requisiti

1. La società di noleggio da gestire è un sistema informatico.
2. La società di noleggio mette a disposizione veicoli.
3. Un veicolo ha una targa, un prezzo per ogni giorno di noleggio, un numero di passeggeri per cui l'auto è omologata, l'alimentazione, la potenza del motore e un segmento.
4. Sul sistema agiscono gli operatori della società di noleggio.
5. Sul sistema agiscono i guidatori.
6. Il sistema deve offrire al guidatore una funzionalità per registrarsi sulla piattaforma.
7. Un utente registrato specifica nome, cognome, data di nascita, email, anno di conseguimento, scadenza e numero della patente di guida.
8. Il sistema deve offrire all'operatore una funzionalità per registrare un veicolo.

9. Il sistema deve offrire all'operatore una funzionalità per modificare i dati del veicolo e gestire lo stato (in-servizio/dismesso) del veicolo.
10. Il sistema deve offrire all'utente registrato una funzionalità per verificare la disponibilità e i prezzi delle auto.
11. Il sistema deve offrire all'utente registrato una funzionalità per ricercare le auto in base alla data di ritiro e di consegna e al segmento del veicolo.
12. Il sistema deve offrire al guidatore una funzionalità per procedere al noleggio dell'auto.
13. Un noleggio ha una data di ritiro, una data di consegna e un costo totale.
14. Il sistema deve offrire all'utente registrato una funzionalità per eventualmente specificare accessori all'atto del noleggio.
15. Un accessorio si divide in servizi assicurativi ed optional.
16. Il sistema deve negare all'utente registrato la possibilità di specificare più di un servizio assicurativo.
17. Tutti gli accessori hanno una descrizione testuale e un prezzo prefissato.
18. Il sistema deve offrire all'operatore una funzionalità per calcolare il prezzo totale del noleggio moltiplicando il prezzo per giorno del veicolo scelto per il numero di giorni richiesti, aggiungendo i costi degli accessori eventualmente scelti.
19. Il sistema deve offrire all'utente registrato la possibilità di aggiungere un guidatore supplementare.
20. Un guidatore supplementare è un utente registrato alla piattaforma.

2.3 Glossario dei termini

Termine	Descrizione	Sinonimi
Auto	Un'auto che è possibile noleggiare presso la società di noleggio	Veicolo
Targa	La targa dell'auto	
Prezzo per giorno-noleggio	Quanto costa noleggiare un'auto per un giorno	
Numero di passeggeri per cui l'auto è omologata	Quante persone un'auto può contenere	
Alimentazione	L'alimentazione dell'auto	
Potenza del motore	La potenza del motore dell'auto	
Segmenti del veicolo	Simbolo per categorizzare le auto in base alle dimensioni dell'auto	
Operatori	Dipendenti della società di noleggio	
Noleggio	La prenotazione di un'auto presso la società di noleggio	Prenotazione
Stato dei veicoli	Stato che indica se il veicolo è ancora disponibile o meno	
Utente	Gli utenti del servizio noleggio	Guidatore
Utente registrato	Un utente del servizio noleggio che ha effettuato la registrazione	
Data ritiro & Data consegna	Le date in cui il noleggio inizia e finisce	
Nome	Il nome dell'utente	

Cognome	Il cognome dell'utente	
Data di nascita	Data di nascita dell'utente	
E-mail	E-mail dell'utente	
Anno di conseguimento della patente	Anno in cui l'utente ha conseguito la patente	
Anno scadenza della patente	Anno in cui scade la patente dell'utente	
Numero patente di guida	Numero della patente dell'utente	
Accessorio	Accessorio offerto dal servizio per ogni auto	
Servizio assicurativo	Le opzioni assicurative offerte dal servizio noleggio	Assicurazione
Optional	Optional offerti dal servizio noleggio per le auto	
Guidatore supplementare	Un eventuale guidatore supplementare da indicare nel noleggio tra gli utenti registrati	
Franchigia	La parte di danno che resta a carico dell'assicurato	
Descrizione accessorio	Un termine che definisce l'accessorio	Nome accessorio

2.4 Classificazione dei requisiti

2.4.1 Requisiti funzionali

ID	Requisito	Origine
RF01	Sul sistema agiscono gli operatori della società di noleggio.	4
RF02	Sul sistema agiscono i guidatori.	5
RF03	Il sistema deve offrire al guidatore una funzionalità per registrarsi sulla piattaforma.	6
RF04	Il sistema deve offrire all'operatore una funzionalità per registrare un veicolo.	8
RF05	Il sistema deve offrire all'operatore una funzionalità per modificare i dati del veicolo e gestire lo stato (in-servizio/dismesso) del veicolo.	9
RF06	Il sistema deve offrire all'utente registrato una funzionalità per verificare la disponibilità e i prezzi delle auto.	10

RF07	Il sistema deve offrire all'utente registrato una funzionalità per ricercare le auto in base alla data di ritiro e di consegna e al segmento del veicolo.	11
RF08	Il sistema deve offrire al guidatore una funzionalità per procedere al noleggio dell'auto.	12
RF09	Il sistema deve offrire all'utente registrato una funzionalità per eventualmente specificare accessori all'atto del noleggio.	14
RF10	Il sistema deve negare all'utente registrato la possibilità di specificare più di un servizio assicurativo.	16
RF11	Il sistema deve offrire all'operatore una funzionalità per calcolare il prezzo totale del noleggio moltiplicando il prezzo per giorno del veicolo scelto per il numero di giorni richiesti, aggiungendo i costi degli accessori eventualmente scelti.	18
RF12	Il sistema deve offrire all'utente registrato la possibilità di aggiungere un guidatore supplementare.	19

2.4.2 Requisiti sui dati

ID	Requisito	Origine
RD01	La società di noleggio da gestire è un sistema informatico.	1
RD02	La società di noleggio mette a disposizione veicoli.	2
RD03	Un veicolo ha una targa, un prezzo per ogni giorno di noleggio, un numero di passeggeri per cui l'auto è omologata, l'alimentazione, la	3

	potenza del motore e un segmento.	
RD04	Un utente registrato specifica nome, cognome, data di nascita, email, anno di conseguimento, scadenza e numero della patente di guida.	7
RD05	Un noleggio ha una data di ritiro, una data di consegna e un costo totale.	13
RD06	Un accessorio si divide in servizi assicurativi ed optional.	15
RD07	Tutti gli accessori hanno una descrizione testuale e un prezzo prefissato.	17
RD08	Un guidatore supplementare è un utente registrato alla piattaforma.	20

2.4.3 Vincoli / Altri requisiti

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

RNF02: Il sistema deve gestire decine di accessi contemporanei, prevalentemente nei periodi estivi.

2.5 Modellazione dei casi d'uso

2.5.1 Attori e casi d'uso

Attori primari: Utente, UtenteRegistrato, Operatore

Attori secondari: NONE

Casi d'uso:

Registrazione

RegistraVeicolo:

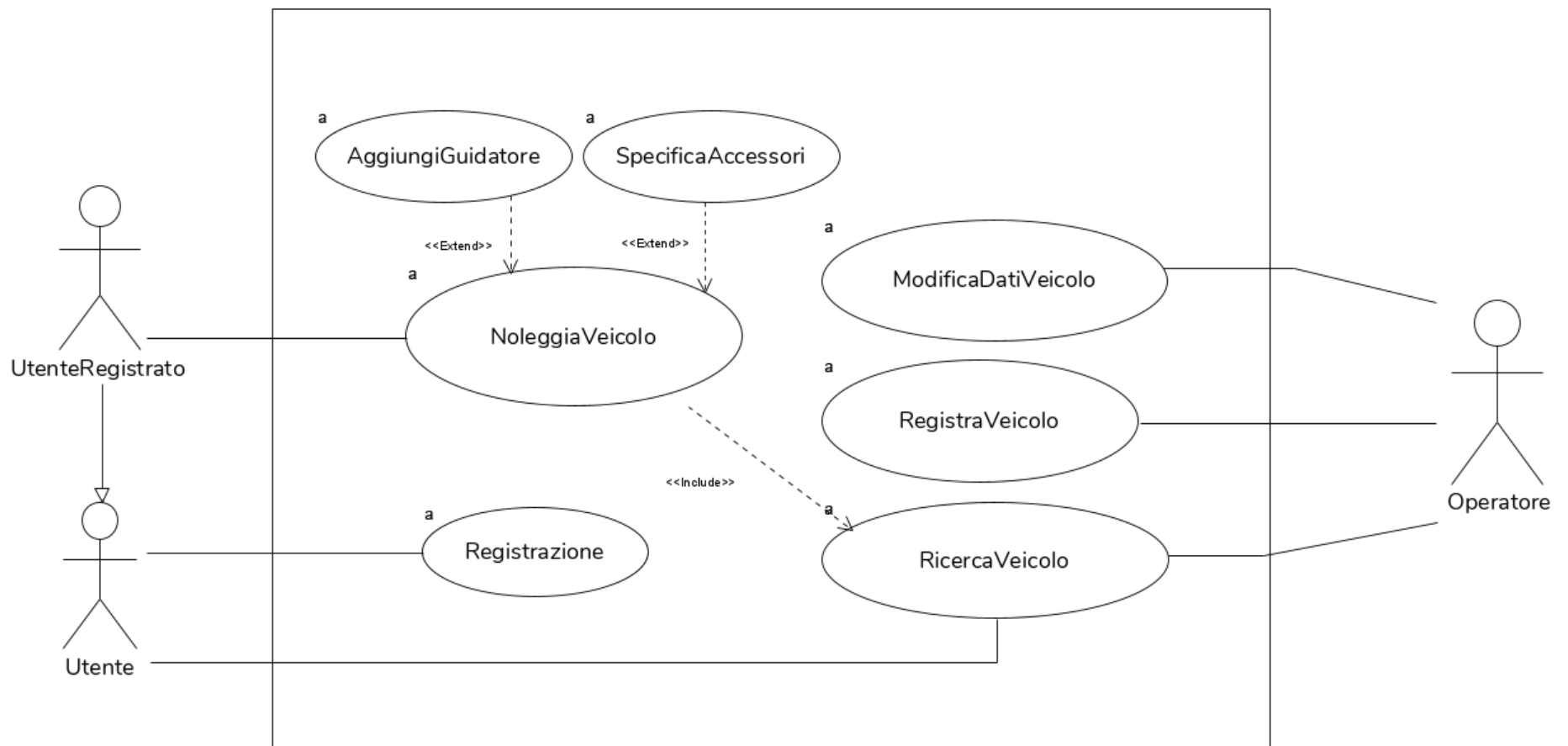
ModificaDatiVeicolo;

RicercaVeicolo:

NoleggiaVeicolo: include RicercaVeicolo;

SpecificaAccessori: extends NoleggiaVeicolo;

AggiungiGuidatore: extends NoleggiaVeicolo;



Scenari:

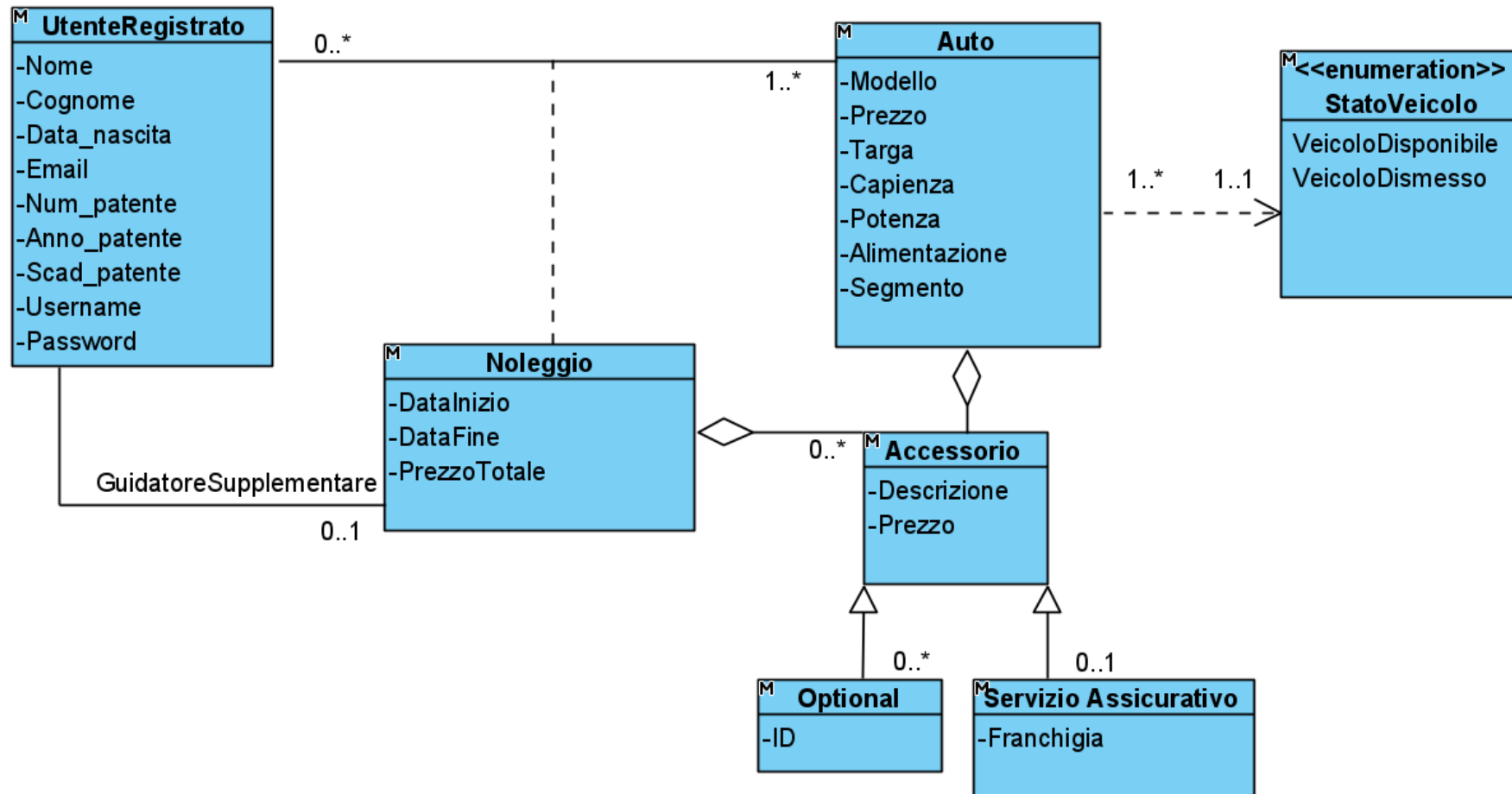
Nome caso d'uso: NoleggiaVeicolo
ID: 1
Breve descrizione: Consente ad un utente registrato di noleggiare un veicolo.
Attori Primari: UtenteRegistrato
Attori Secondari: Nessuno
Pre-Condizione: L'utente registrato deve aver effettuato il "Login"
Sequenza di Eventi: <ol style="list-style-type: none">1. Il caso d'uso inizia quando l'utente registrato clicca "NoleggiaVeicolo".2. L'utente registrato inserisce le date di noleggio e il segmento del veicolo desiderato e clicca su "RicercaVeicolo".3. Il sistema mostra i veicoli disponibili e chiede all'utente registrato di specificare gli optional.4. L'utente registrato specifica gli optional e al più un servizio assicurativo mediante la "SpecificaAccessori".5. L'utente registrato scegli eventualmente un guidatore aggiuntivo mediante "AggiungiGuidatore".6. La prenotazione viene effettuata.
Post Condizioni: Database aggiornato con il noleggio effettuato e veicolo reso non disponibile nelle date scelte.
Sequenza di eventi alternativa: La RicercaVeicolo non produce risultati.
Post Condizione Alternativa: Nessuna

Nome caso d'uso: RegistraVeicolo
ID: 2
Breve descrizione: Consente ad un operatore di registrare i dati di un nuovo veicolo non esistente nel sistema.
Attori Primari: Operatore
Attori Secondari: Nessuno
Pre-Condizione: L'operatore deve aver effettuato il "Login"
Sequenza di Eventi: <ol style="list-style-type: none">1. Il caso d'uso inizia quando l'operatore clicca su "RegistraVeicolo"2. L'operatore inserisce i dati del veicolo che vuole registrare3. I dati del veicolo vengono inseriti nel database
Post Condizioni: Database aggiornato con i dati del nuovo veicolo.
Sequenza di eventi alternativa: Nessuna
Post Condizione Alternativa: Nessuna

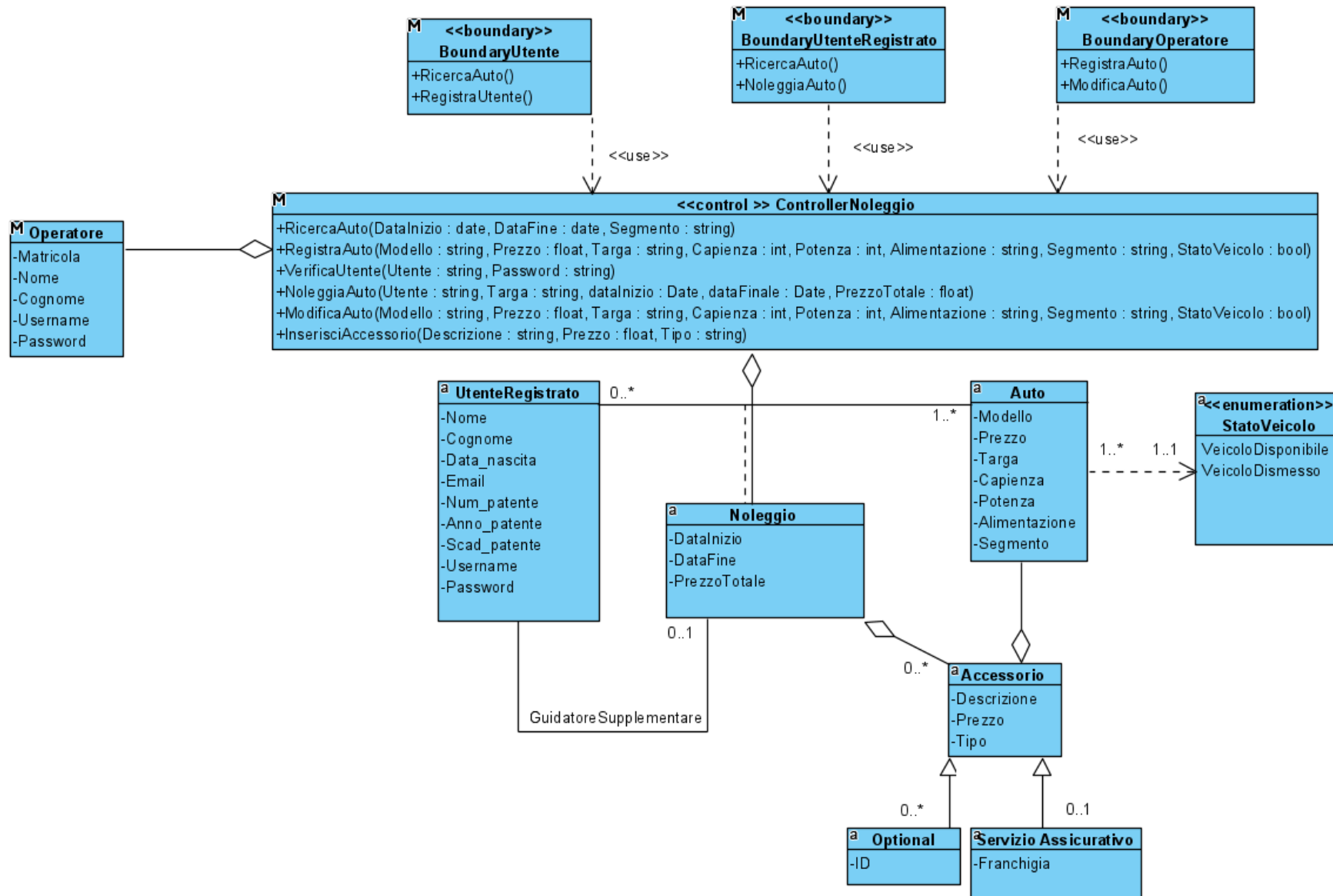
Nome caso d'uso: ModificaDatiVeicolo
ID: 3
Breve descrizione: Consente di modificare i dati di un veicolo già esistente nel sistema. È possibile modificare parzialmente alcuni dati lasciandone altri immutati.
Attori Primari: Operatore
Attori Secondari: Nessuno
Pre-Condizione: Il veicolo deve essere presente nel Database. L'operatore deve aver effettuato il "Login"
Sequenza di Eventi: <ol style="list-style-type: none"> 4. Il caso d'uso inizia quando l'operatore clicca su "ModificaDatiVeicolo" 5. L'operatore inserisce i dati del veicolo che vuole modificare 6. I dati del veicolo modificati vengono inseriti nel database
Post Condizioni: Database aggiornato con i nuovi dati inseriti del veicolo
Sequenza di eventi alternativa: Nessuna
Post Condizione Alternativa: Nessuna

2.6 Diagramma delle classi

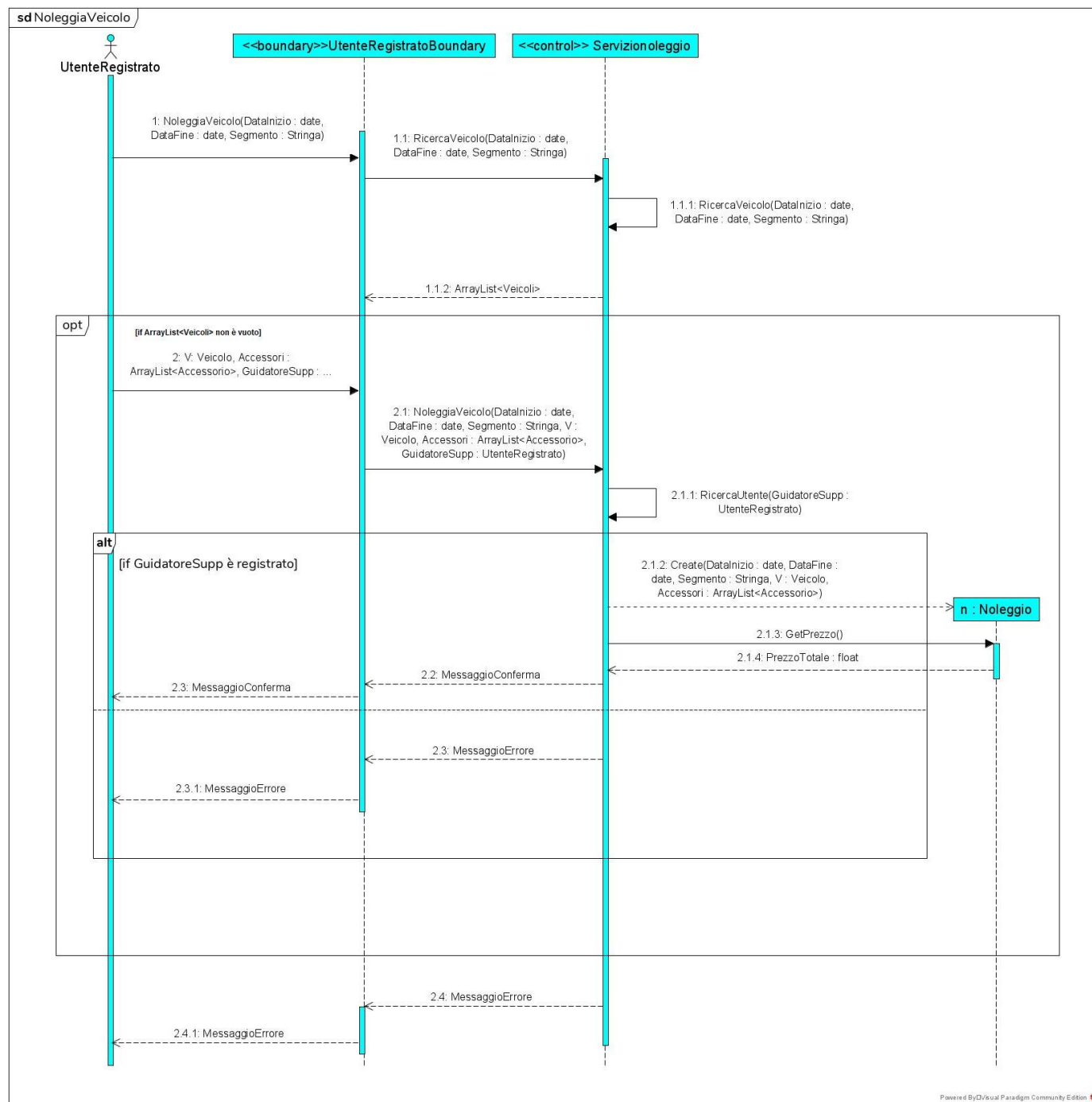
Primo diagramma delle classi di analisi:

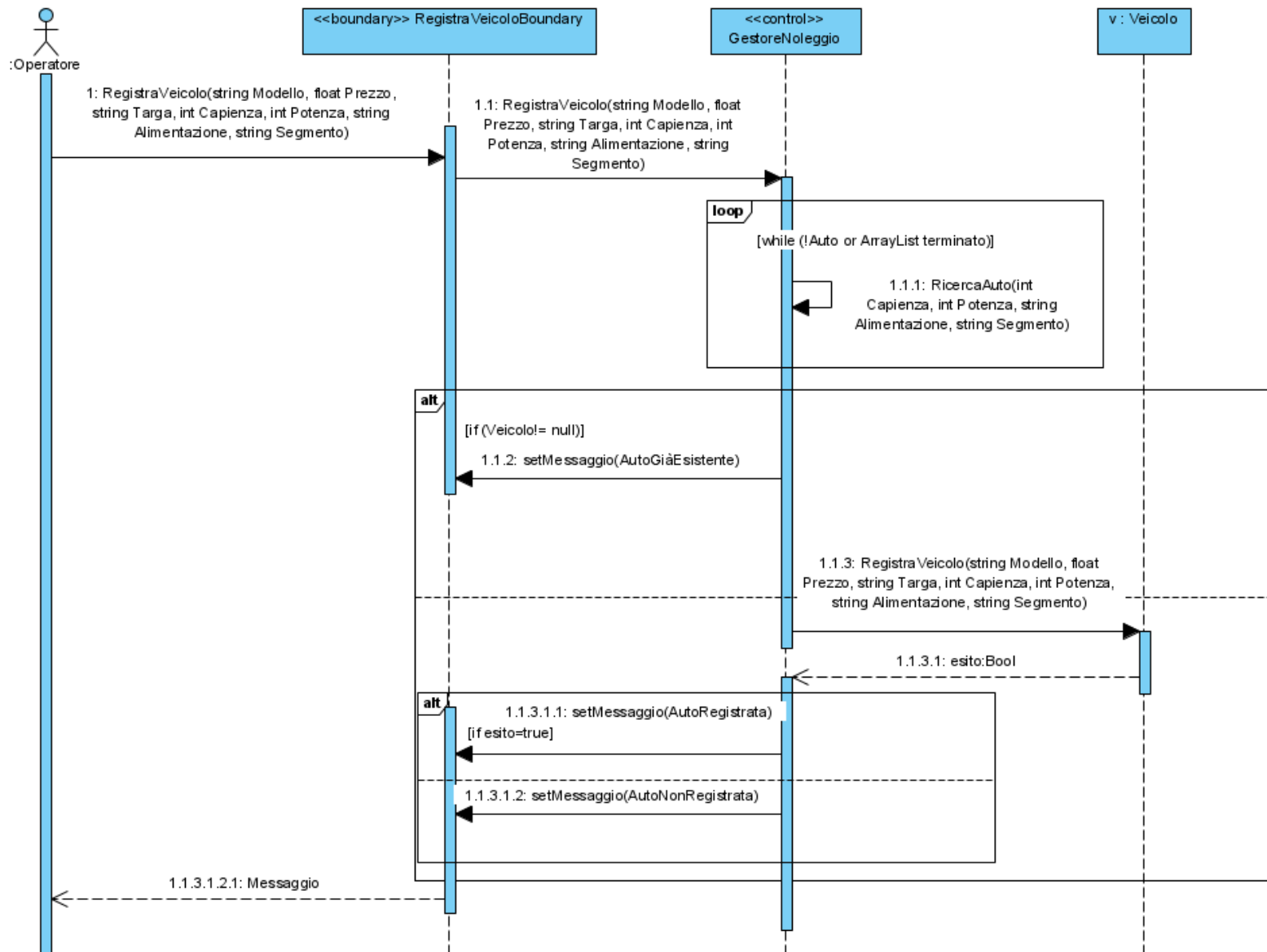


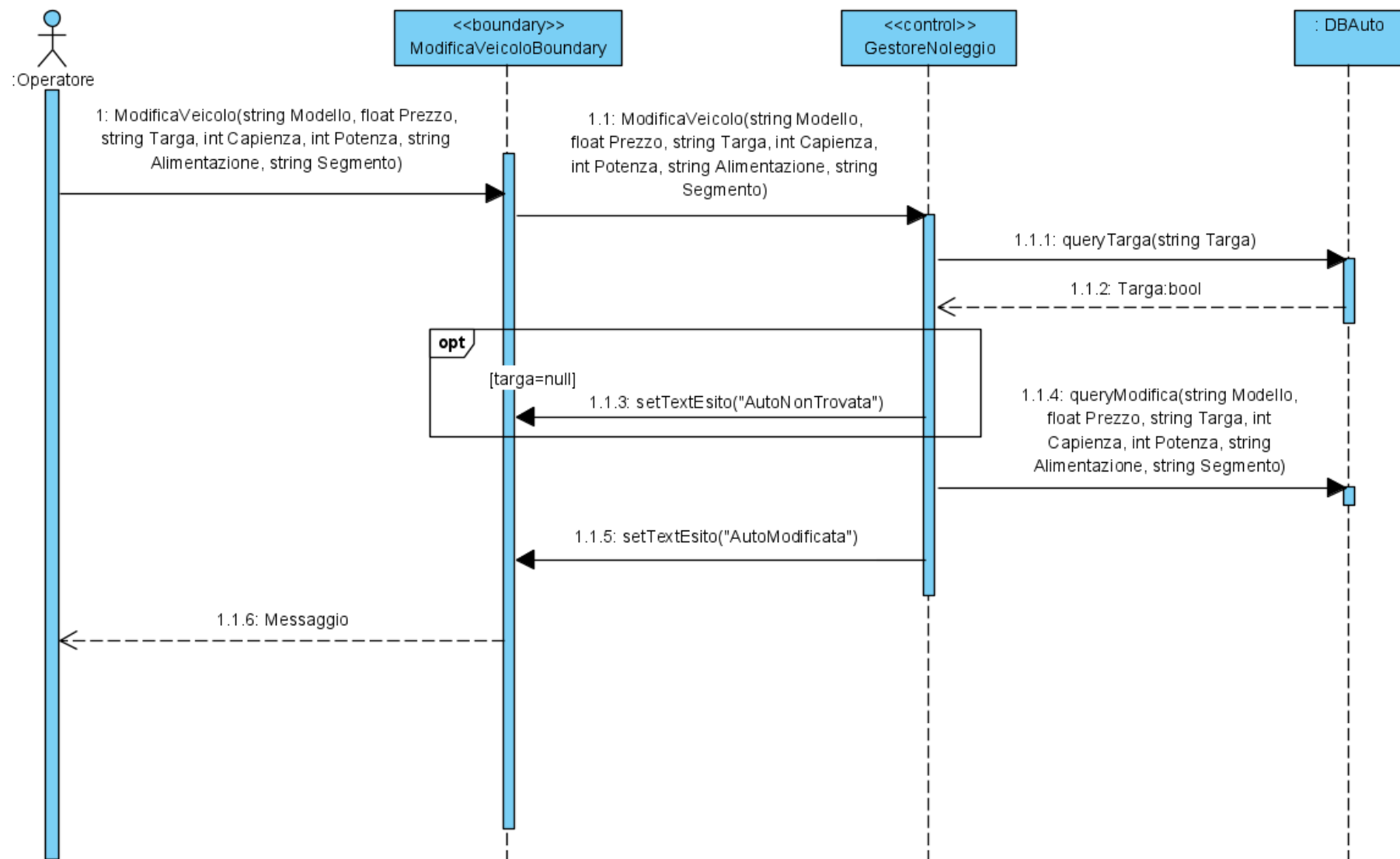
2.6.1 Diagramma delle classi di analisi raffinato:



2.7 Diagrammi di sequenza







3. Stima dei costi

Stima dei Costi - NoleggiaVeicolo

ILF: Tabella Veicoli (Semplice)
Tabella Noleggi (Semplice)
 $NILF=2 \quad PI=6 \Rightarrow VPI = 12$

EIF: Interfaccia Veicoli Disponibili (Semplice)
 $VPI = 4$

EI: Dati Noleggio forniti dall'utente (Medio)
 $VPI = 4$

EO: Calcolo Prezzo (Medio)
 $VPI = 10$

EQ: Informazioni su Veicoli Disponibili per un determinato periodo/luogo (Medio)
 $VPI = 7$

$UFP = 37$

Fattori Correttivi

Comunicazione dati: 3
Distribuzione elaborazione: 4
Prestazioni: 2
Utilizzo intensivo configurazione: 1
Frequenza delle transazioni: 4
Inserimento dati interattivo: 4
Efficienza per l'utente finale: 5
Aggiornamento interattivo: 5
Complessità elaborativa: 1
Riusabilità: 1
Facilità installazione: 3
Facilità gestione operativa: 3
Molteplicità di siti: 0
Facilità di modifica: 3

$FP = UFP \times (0,65 + 0,01 \times \text{SumFi})$
 $FP = 37 \times (0,65 + 0,01 \times 39) = 9,38$

Costo in Java = $9.38 \times 6 = 56$
Costo Finale = $9.38 \times 53 = 497$

Stima dei Costi - RegistraVeicolo

ILF: Tabella Veicoli (Semplice)

NILF = 1 Peso: 3

EI: Dati veicolo da inserire (Semplice)

NEI = 1 Peso: 4

UFP = 7

Comunicazione dati: 2

Distribuzione elaborazione: 2

Prestazioni: 2

Utilizzo intensivo configurazione: 1

Frequenza delle transazioni: 1

Inserimento dati interattivo: 2

Efficienza per l'utente finale: 3

Aggiornamento interattivo: 1

Complessità elaborativa: 2

Riusabilità: 2

Facilità installazione: 3

Facilità gestione operativa: 2

Molteplicità di siti: 0

Facilità di modifica: 2

$FP = UFP \times (0,65 + 0,01 \times \text{SumFi})$

$FP = 7 \times (0,65 + 0,01 \times 25) = 7,21$

Costo in Java = 43.26

Costo Finale = 382.13

Stima dei Costi – ModificaDatiVeicolo

ILF: Tabella Veicoli (Semplice)

NILF = 1 Peso: 3

EI: Dati veicolo da modificare (Semplice)

NEI = 1 Peso: 4

UFP = 7

Fattori Correttivi

Comunicazione dati: 3

Distribuzione elaborazione: 3

Prestazioni: 3

Utilizzo intensivo configurazione: 1

Frequenza delle transazioni: 2

Inserimento dati interattivo: 2

Efficienza per l'utente finale: 3

Aggiornamento interattivo: 3

Complessità elaborativa: 2

Riusabilità: 2

Facilità installazione: 3

Facilità gestione operativa: 3

Molteplicità di siti: 0

Facilità di modifica: 4

$FP = UFP \times (0,65 + 0,01 \times \text{SumFi})$

$FP = 7 \times (0,65 + 0,01 \times 34) = 7,99$

Costo in Java = 42

Costo Finale = 371

4. Piano di test funzionale

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

Categoria	Valore
<ul style="list-style-type: none">• Data Inizio• Data Fine• Segmento• Veicoli disponibili	<ul style="list-style-type: none">- selezione data con formato valido-nessuna data inserita [ERROR]- selezione data con formato valido-nessuna data inserita [ERROR]- Segmento tra B, C, D- Segmento non valido [ERROR]-0-1 o molti

Ci sono 4 categorie tutte con 2 classi di valori.

Numero di test senza vincoli: $2 \times 2 \times 2 \times 2 = 16$

Ci sono 3 vincoli error.

Numero di test con vincoli error: $1 \times 1 \times 1 \times 2 + 3 = 5$

TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	L'utente inserisce correttamente le date, il segmento scelto; vengono trovati dei veicoli corrispondenti alla ricerca.	<ul style="list-style-type: none"> Data Inizio: data con formato valido Data Fine: data con formato valido Segmento: uno tra B-C-D Veicoli disponibili: 1 o molti 	<ul style="list-style-type: none"> Esistono dei veicoli corrispondenti ai parametri della ricerca nel database 	<ul style="list-style-type: none"> Data Inizio Data Fine Segmento 	<ul style="list-style-type: none"> Lista veicoli disponibili 	<ul style="list-style-type: none"> Stampa della lista dei veicoli disponibili
2	L'utente inserisce correttamente le date, il segmento scelto, ma non vengono trovati dei veicoli corrispondenti alla ricerca.	<ul style="list-style-type: none"> Data Inizio: data con formato valido Data Fine: data con formato valido Segmento: uno tra B-C-D Veicoli disponibili: 0 	<ul style="list-style-type: none"> Nel DB non è presente nessun'auto corrispondente alla ricerca 	<ul style="list-style-type: none"> Data Inizio Data Fine Segmento 	<ul style="list-style-type: none"> Lista veicoli disponibili vuota 	<ul style="list-style-type: none"> Nessuna
3	L'utente inserisce correttamente le date, ma un segmento invalido	<ul style="list-style-type: none"> Data Inizio: data con formato valido Data Fine: data con formato valido Segmento invalido 	<ul style="list-style-type: none"> Nessuna 	<ul style="list-style-type: none"> Data Inizio Data Fine Segmento 	<ul style="list-style-type: none"> Nessuno 	<ul style="list-style-type: none"> Nessuna

4	L'utente inserisce un segmento valido, ma come Data Fine una data precedente alla Data Inizio.	<ul style="list-style-type: none"> • Data Inizio: data con formato valido • Data Fine: data con formato valido • Segmento: uno tra B-C-D 	<ul style="list-style-type: none"> • Nessuna 	<ul style="list-style-type: none"> • Data Inizio • Data Fine • Segmento 	<ul style="list-style-type: none"> • Nessuno 	<ul style="list-style-type: none"> • Nessuna
5	L'utente inserisce un segmento valido, ma come Data Inizio una data precedente alla data in cui avviene la ricerca.	<ul style="list-style-type: none"> • Data Inizio: data con formato valido • Data Fine: data con formato valido • Segmento: uno tra B-C-D 	<ul style="list-style-type: none"> • Nessuna 	<ul style="list-style-type: none"> • Data Inizio • Data Fine • Segmento 	<ul style="list-style-type: none"> • Nessuno 	<ul style="list-style-type: none"> • Nessuna

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

Categoria	Valore
<ul style="list-style-type: none">• Modello• Targa• PrezzoPerGiorno• Capienza• Potenza• Segmento• Alimentazione	<ul style="list-style-type: none">- stringa di caratteri- stringa di caratteri- Float- Valore non numerico [ERROR]- Intero- Valore non numerico [ERROR]- Intero- Valore non numerico [ERROR]- Segmento tra B, C, D- Segmento non valido [ERROR]- Alimentazione tra benzina e gasolio- Alimentazione non valida [ERROR]

Ci sono 7 categorie, di cui: 5 categorie con 2 classi di valori, 2 con 1 classe di valori.

Numero di test senza vincoli: $2 \times 2 \times 2 \times 2 \times 2 \times 1 \times 1 = 32$

Ci sono 5 vincoli error.

Numero di test con vincoli error: $1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 + 5 = 6$

TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, la capienza, la potenza, il segmento e l'alimentazione del veicolo.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: variabile di tipo float • Capienza: Intero • Potenza: Intero • Segmento: uno tra B-C-D • Alimentazione: uno tra "benzina" e "gasolio" 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza • Segmento • Alimentazione 	Nessuno	<ul style="list-style-type: none"> • Veicolo aggiunto al DB
2	L'operatore inserisce un modello di veicolo e una targa valida, ma un prezzo non valido.	<ul style="list-style-type: none"> • Modello: stringa di caratteri alfabetici • Targa: stringa 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa 	Nessuno	Nessuna

		di caratteri nel giusto formato • Prezzo: non float		• PrezzoPerGiorno		
3	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, ma una capienza passeggeri fuori dall'intervallo valido.	• Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza diversa da 4,5,6	L'operatore ha effettuato il login	• Modello • Targa • PrezzoPerGiorno • Capienza	Nessuno	Nessuna
4	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza valida ma una potenza non valida.	• Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore non intero	L'operatore ha effettuato il login	• Modello • Targa • PrezzoPerGiorno • Capienza • Potenza	Nessuno	Nessuna
5	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza	• Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float	L'operatore ha effettuato il login	• Modello • Targa	Nessuno	Nessuna

	e una potenza valida, ma un segmento diverso da quelli validi.	<ul style="list-style-type: none"> • Capienza tra 4,5,6 • Potenza: valore intero • Segmento diverso da B-C-D 		<ul style="list-style-type: none"> • PrezzoPerGiorno • Capienza • Potenza • Segmento 		
6	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza e una potenza valida, un segmento tra B-C-D ma un'alimentazione non valida.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore intero • Segmento tra B-C-D • Alimentazione diversa da benzina/gasolio 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza • Segmento • Alimentazione 	Nessuno	Nessuna

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

Categoria	Valore
<ul style="list-style-type: none">• Modello• Targa• PrezzoPerGiorno• Capienza• Potenza• Segmento• Alimentazione• StatoVeicolo	<ul style="list-style-type: none">- stringa di caratteri- stringa di caratteri- Float- Valore non numerico [ERROR]- Intero- Valore non numerico [ERROR]- Intero- Valore non numerico [ERROR]- Segmento tra B, C, D- Segmento non valido [ERROR]- Alimentazione tra benzina e gasolio- Alimentazione non valida [ERROR]- 0- 1

Ci sono 7 categorie, di cui: 5 categorie con 2 classi di valori, 2 con 1 classe di valori.

Numero di test senza vincoli: $2 \times 2 \times 2 \times 2 \times 2 \times 1 \times 1 = 64$.

Ci sono 5 vincoli error.

Numero di test con vincoli error: $1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 + 5 = 6$

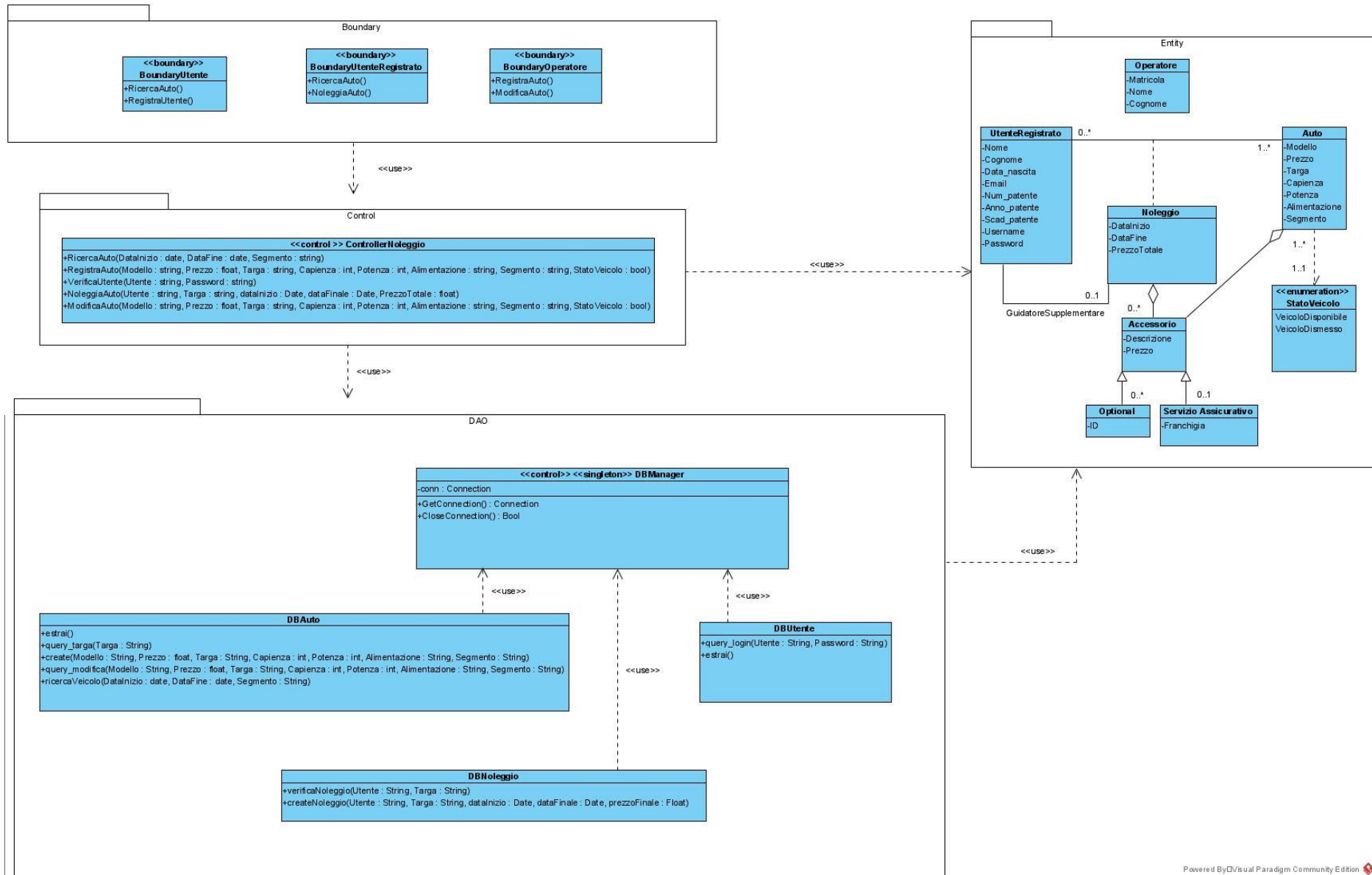
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	L'operatore modifica correttamente il modello, la targa del veicolo, il prezzo per giorno, la capienza, la potenza, il segmento, l'alimentazione del veicolo e altera lo stato.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: variabile di tipo float • Capienza: Intero • Potenza: Intero • Segmento: uno tra B-C-D • Alimentazione: uno tra "benzina" e "gasolio" • StatoVeicolo tra dismesso/disponibile 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza • Segmento • Alimentazione • StatoVeicolo 	Nessuno	<ul style="list-style-type: none"> • Veicolo modificato sul DB
2	L'operatore inserisce un modello di veicolo e una targa valida, ma un prezzo non valido.	<ul style="list-style-type: none"> • Modello: stringa di caratteri alfabetici • Targa: stringa di caratteri nel giusto formato • Prezzo: non float 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno 	Nessuno	Nessuna

3	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, ma una capienza passeggeri fuori dall'intervallo valido.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza diversa da 4,5,6 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza 	Nessuno	Nessuna
4	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza valida ma una potenza non valida.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore non intero 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza 	Nessuno	Nessuna
5	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore intero 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno 	Nessuno	Nessuna

	capienza e una potenza valida, ma un segmento diverso da quelli validi.	<ul style="list-style-type: none"> • Segmento diverso da B-C-D 		<ul style="list-style-type: none"> • Capienza • Potenza • Segmento 		
6	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza e una potenza valida, un segmento tra B-C-D ma un'alimentazione non valida.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore intero • Segmento tra B-C-D • Alimentazione diversa da benzina/gasolio 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza • Segmento • Alimentazione 	Nessuno	Nessuna

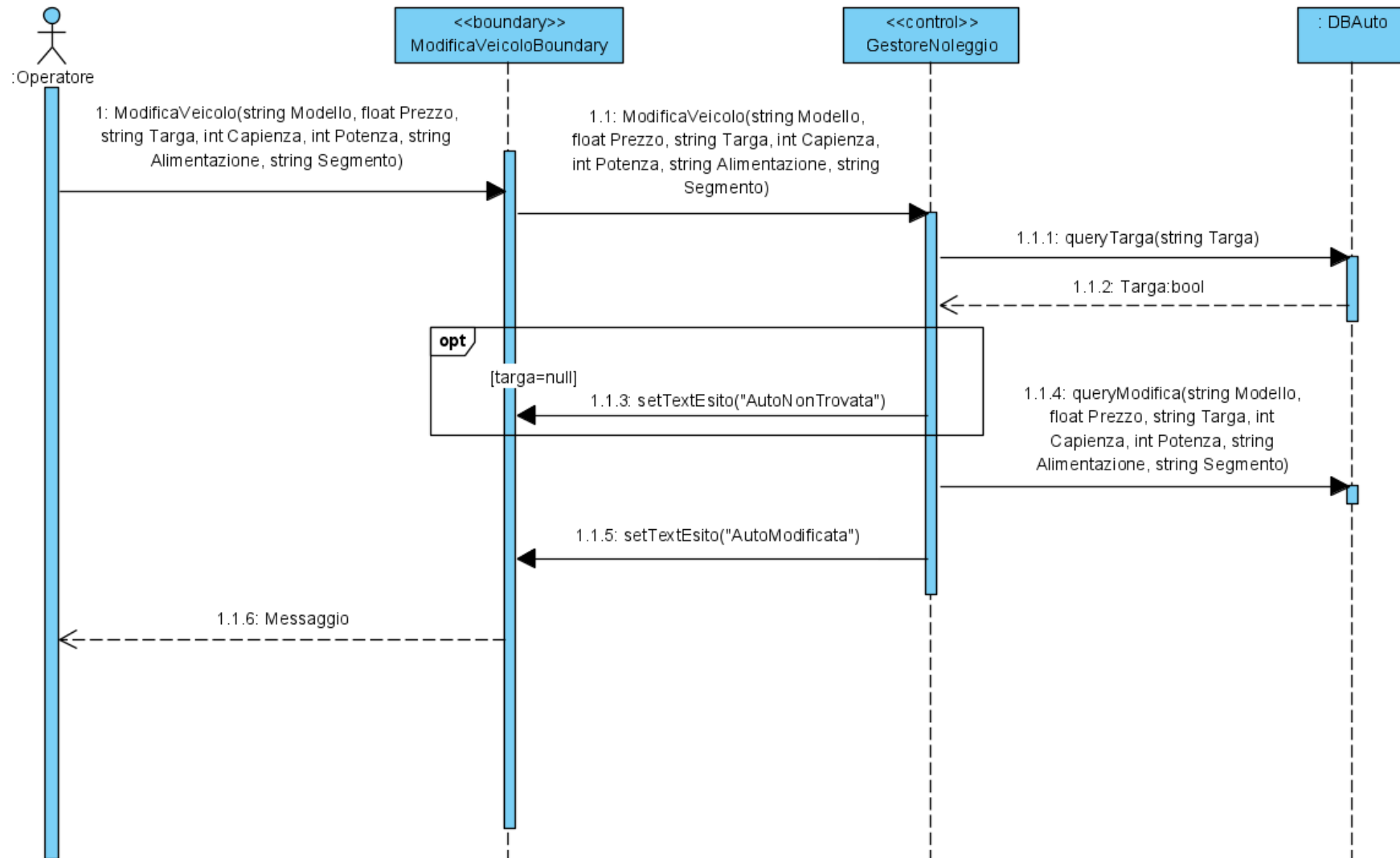
5. Progettazione

5.1 Diagramma delle classi

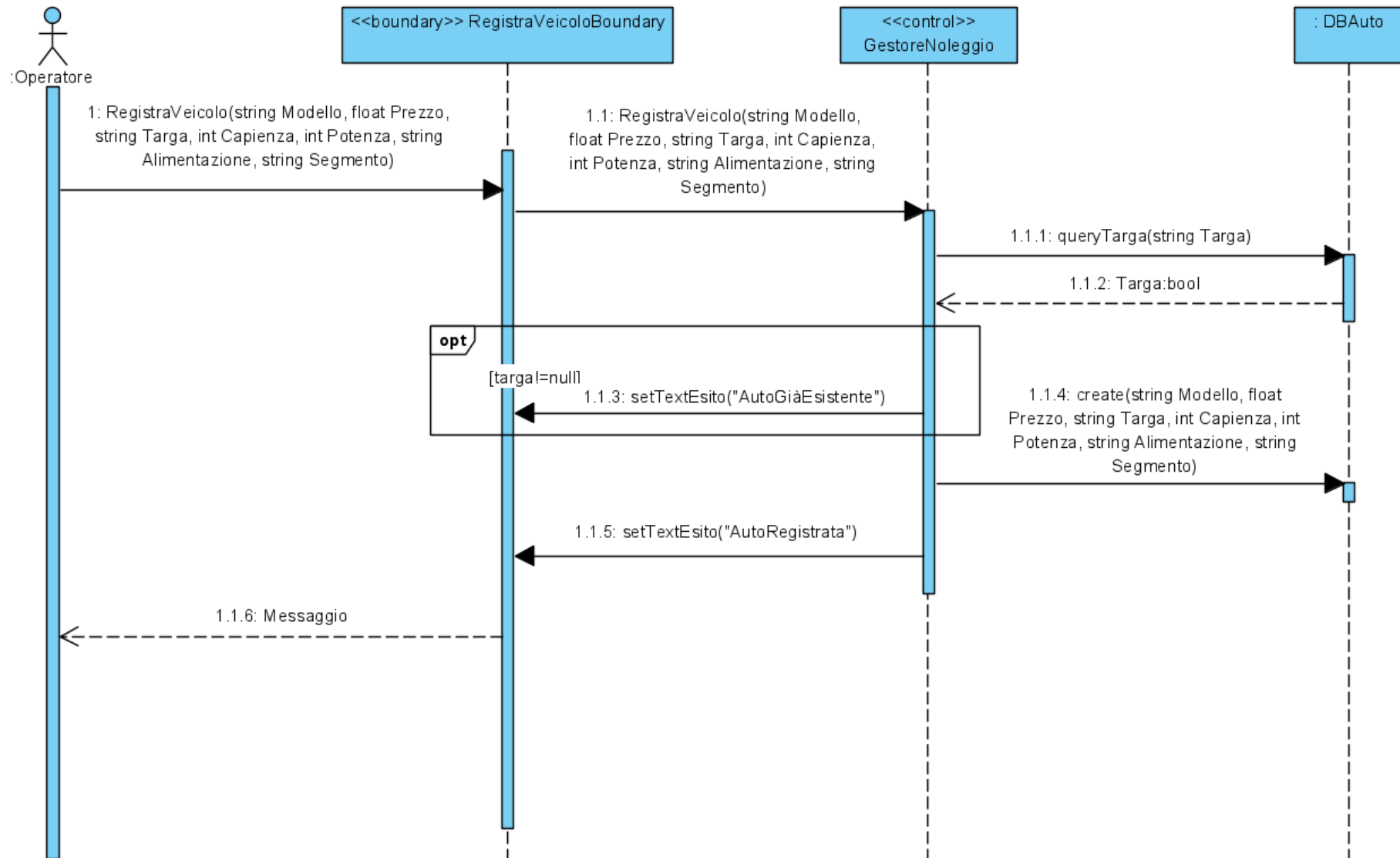


5.2 Diagrammi di sequenza

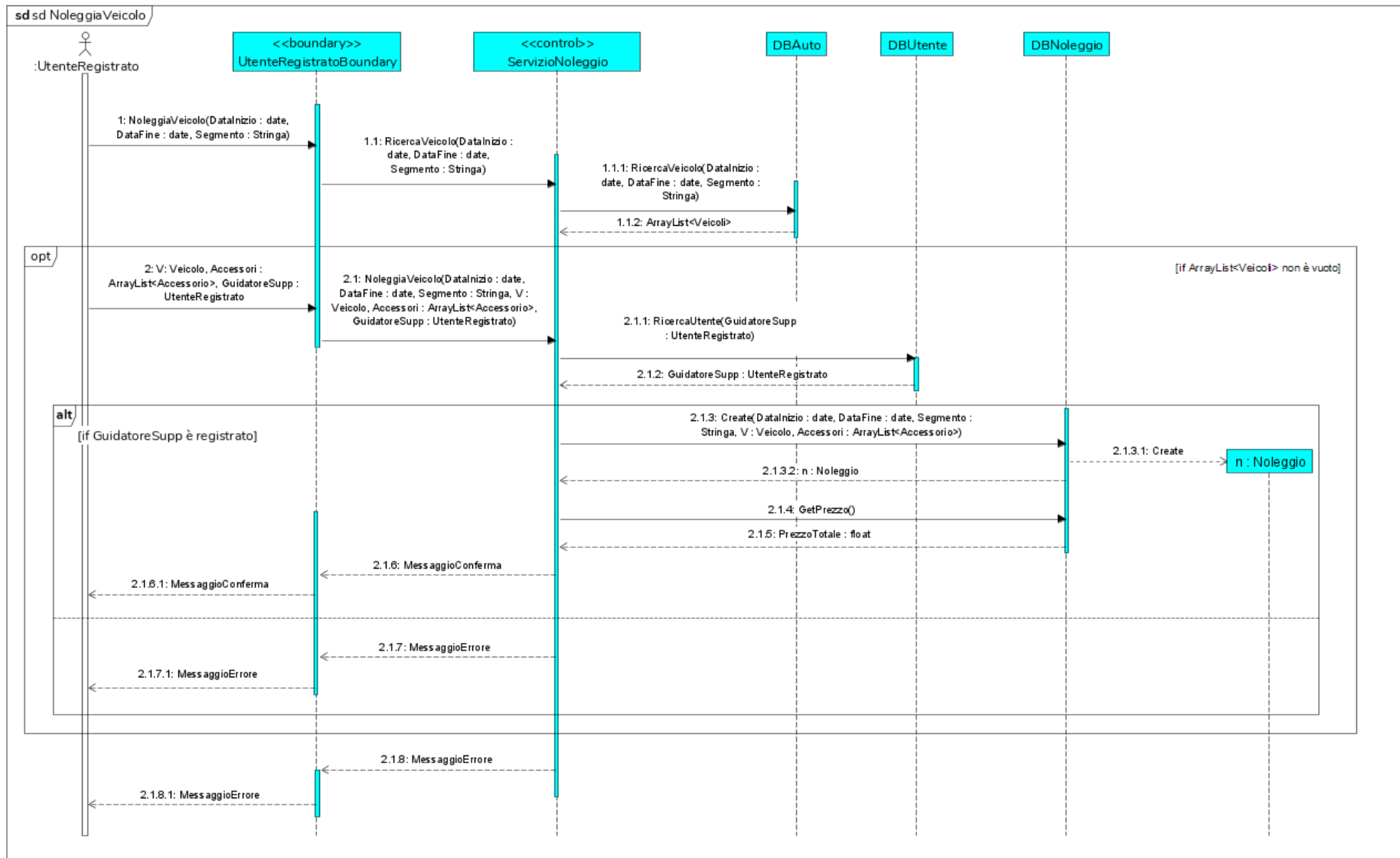
ModificaVeicolo:



RegistraVeicolo:



NoleggiaVeicolo:



6 Implementazione

Package: Entity, Database, Control, Boundary

Classi:

- Entity: Auto, Utente, Accessorio, Noleggio
- Database: DBAuto, DBUtente, DBNoleggio, DBAccessorio, DBManager
- Control: GestoreNoleggio
- Boundary: UtenteBoundary, UtenteRegistratoBoundary, OperatoreBoundary

Righe di codice: 2231

- Entity: Auto (63), Utente (37), Accessorio (3), Noleggio (49)
- Database: DBAuto (191), DBUtente (65), DBNoleggio (85), DBAccessorio (3), DBManager (27)
- Control: GestoreNoleggio (120)
- Boundary: UtenteBoundary (498), UtenteRegistratoBoundary (311), OperatoreBoundary (622)

Stima dei costi:

NoleggiaVeicolo: previste 56 righe di codice in Java -> Totale righe di codice effettive: 59

RegistraVeicolo: previste 43 righe di codice in Java -> Totale righe di codice effettive: 47

ModificaVeicolo: previste 42 righe di codice in Java -> Totale righe di codice effettive: 50

7. Testing

7.1 Test funzionale: "RicercaVeicolo"

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito (FAIL, PASS)
1	L'utente inserisce correttamente le date, il segmento scelto; vengono trovati dei veicoli corrispondenti alla ricerca.	<ul style="list-style-type: none">• Data Inizio: data con formato valido• Data Fine: data con formato valido• Segmento: uno tra B-C-D• Veicoli disponibili: 1 o molti	<ul style="list-style-type: none">• Esistono dei veicoli corrispondenti ai parametri della ricerca nel database	<ul style="list-style-type: none">• Data Inizio• Data Fine• Segmento	<ul style="list-style-type: none">• Lista veicoli disponibili	<ul style="list-style-type: none">• Stampa della lista dei veicoli disponibili	PASS

2	L'utente inserisce correttamente le date, il segmento scelto, ma non vengono trovati dei veicoli corrispondenti alla ricerca.	<ul style="list-style-type: none"> • Data Inizio: data con formato valido • Data Fine: data con formato valido • Segmento: uno tra B-C-D • Veicoli disponibili: 0 	Nel DB non è presente nessun'auto corrispondente alla ricerca	<ul style="list-style-type: none"> • Data Inizio • Data Fine • Segmento 	<ul style="list-style-type: none"> • Lista veicoli disponibili vuota 	<ul style="list-style-type: none"> • Nessuna 	PASS
3	L'utente inserisce correttamente le date, ma un segmento invalido	<ul style="list-style-type: none"> • Data Inizio: data con formato valido • Data Fine: data con formato valido • Segmento invalido 	<ul style="list-style-type: none"> • Nessuna 	<ul style="list-style-type: none"> • Data Inizio • Data Fine • Segmento 	<ul style="list-style-type: none"> • Nessuno 	<ul style="list-style-type: none"> • Nessuna 	FAIL
4	L'utente inserisce un segmento valido, ma come Data Fine una data precedente alla Data Inizio.	<ul style="list-style-type: none"> • Data Inizio: data con formato valido • Data Fine: data con formato valido • Segmento: uno tra B-C-D 	<ul style="list-style-type: none"> • Nessuna 	<ul style="list-style-type: none"> • Data Inizio • Data Fine • Segmento 	<ul style="list-style-type: none"> • Nessuno 	<ul style="list-style-type: none"> • Nessuna 	FAIL

Debugging:

Test case con ID 3 ha dato esito FAIL: la chiamata alla funzione RicercaVeicolo avveniva solo se il segmento in ingresso era uno tra B-C-D
 Debugging effettuato: abbiamo implementato un controllo sul segmento in ingresso con un ciclo if che restituisce un messaggio di errore se il segmento non è tra quelli contemplati, e ti permette di inserirlo nuovamente.

Test case con ID 4 ha dato esito FAIL: l'eccezione DateTimeException non veniva gestita correttamente.

Debugging effettuato: inserito un blocco try-catch che gestisce l'eccezione.

“RegistraVeicolo”.

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito (FAIL, PASS)
1	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, la capienza, la potenza, il segmento e l'alimentazione del veicolo.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: variabile di tipo float • Capienza: Intero • Potenza: Intero • Segmento: uno tra B-C-D • Alimentazione: uno tra “benzina” e “gasolio” 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza • Segmento • Alimentazione 	Nessuno	<ul style="list-style-type: none"> • Veicolo aggiunto al DB 	PASS
2	L'operatore inserisce un modello di veicolo e una targa valida, ma	<ul style="list-style-type: none"> • Modello: stringa di caratteri alfabetici 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa 	Nessuno	Nessuna	PASS

	un prezzo non valido.	<ul style="list-style-type: none"> • Targa: stringa di caratteri nel giusto formato • Prezzo: non float 		<ul style="list-style-type: none"> • PrezzoPerGiorno 			
3	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, ma una capienza passeggeri fuori dall'intervallo valido.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza diversa da 4,5,6 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza 	Nessuno	Nessuna	PASS
4	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza valida ma una potenza non valida.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore non intero 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza 	Nessuno	Nessuna	PASS
5	L'operatore inserisce correttamente il modello, la targa del	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa 	Nessuno	Nessuna	PASS

	veicolo, il prezzo per giorno, una capienza e una potenza valida, ma un segmento diverso da quelli validi.	<ul style="list-style-type: none"> • di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore intero • Segmento diverso da B-C-D 		<ul style="list-style-type: none"> • PrezzoPerGiorno • Capienza • Potenza • Segmento 			
6	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza e una potenza valida, un segmento tra B-C-D ma un'alimentazione non valida.	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore intero • Segmento tra B-C-D • Alimentazione diversa da benzina/gasoli o 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza • Segmento • Alimentazione 	Nessuno	Nessuna	PASS

“ModificaDatiVeicolo”.

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito (FAIL, PASS)
1	L'operatore modifica correttamente il modello, la targa del veicolo, il prezzo per giorno, la capienza, la potenza, il segmento, l'alimentazione del veicolo e altera lo stato.	<ul style="list-style-type: none"> Modello: stringa di caratteri Targa: stringa di caratteri Prezzo: variabile di tipo float Capienza: Intero Potenza: Intero Segmento: uno tra B-C-D Alimentazione: uno tra “benzina” e “gasolio” StatoVeicolo tra dismesso/disponibile 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> Modello Targa PrezzoPerGiorno Capienza Potenza Segmento Alimentazione StatoVeicolo . 	Nessuno	<ul style="list-style-type: none"> Veicolo modificato sul DB 	PASS
2	L'operatore inserisce un modello di	<ul style="list-style-type: none"> Modello: stringa di caratteri alfabetici Targa: stringa di caratteri 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> Modello Targa 	Nessuno	Nessuna	PASS

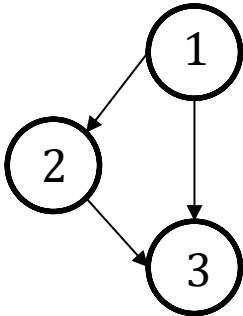
	veicolo e una targa valida, ma un prezzo non valido.	nel giusto formato Prezzo: non float		<ul style="list-style-type: none"> PrezzoPerGiorno 			
3	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, ma una capienza passeggeri fuori dall'intervallo valido.	<ul style="list-style-type: none"> Modello: stringa di caratteri Targa: stringa di caratteri Prezzo: float Capienza diversa da 4,5,6 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> Modello Targa PrezzoPerGiorno Capienza 	Nessuno	Nessuna	PASS
4	L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza valida ma una potenza non valida.	<ul style="list-style-type: none"> Modello: stringa di caratteri Targa: stringa di caratteri Prezzo: float Capienza tra 4,5,6 Potenza: valore non intero 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> Modello Targa PrezzoPerGiorno Capienza Potenza 	Nessuno	Nessuna	PASS
5	L'operatore inserisce correttamente il	<ul style="list-style-type: none"> Modello: stringa di caratteri Targa: stringa di caratteri 	L'operatore ha effettuato il login	<ul style="list-style-type: none"> Modello 	Nessuno	Nessuna	PASS

	<p>modello, la targa del veicolo, il prezzo per giorno, una capienza e una potenza valida, ma un segmento diverso da quelli validi.</p>	<ul style="list-style-type: none"> • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore intero • Segmento diverso da B-C-D 		<ul style="list-style-type: none"> • Targa • PrezzoPerGiorno • Capienza • Potenza <p>Segmento</p>			
6	<p>L'operatore inserisce correttamente il modello, la targa del veicolo, il prezzo per giorno, una capienza e una potenza valida, un segmento tra B-C-D ma un'alimentazione non valida.</p>	<ul style="list-style-type: none"> • Modello: stringa di caratteri • Targa: stringa di caratteri • Prezzo: float • Capienza tra 4,5,6 • Potenza: valore intero • Segmento tra B-C-D • Alimentazione diversa da benzina/gasolio 	<p>L'operatore ha effettuato il login</p>	<ul style="list-style-type: none"> • Modello • Targa • PrezzoPerGiorno • Capienza • Potenza • Segmento • Alimentazione 	Nessuno	Nessuna	PASS

7.2 Test strutturale

7.2.1 Complessità ciclomatica

Login

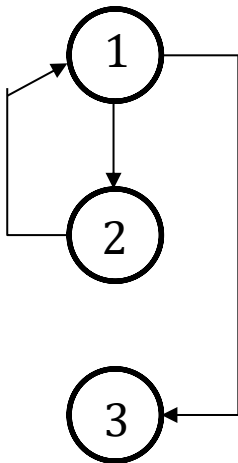


```
public static boolean query_login (String utente,String password) {  
    try {  
        Connection conn = DBManager.getConnection();  
  
        String query = "SELECT * FROM OPERATORE WHERE USERNAME=? AND PASSWORD=?";  
  
        PreparedStatement pstmt = conn.prepareStatement(query);  
  
        pstmt.setString(1, utente);  
        pstmt.setString(2, password);  
  
        ResultSet rs = pstmt.executeQuery();  
  
        if(rs.next())  
            return true;  
  
    } catch (Exception e) {  
  
    }  
    return false;  
}
```

Numero di complessità ciclomatica: 2

Cammini linearmente indipendenti: 1-2-3;1-3;

RicercaVeicolo



```
static public ArrayList<Auto> RicercaAutoPerData(Date Inizio,Date Fine,String segm){
    ArrayList<Auto> ArrayAuto = new ArrayList<Auto>();

    try {
        java.sql.Date sqlDate1 = new java.sql.Date(Inizio.getTime());
        java.sql.Date sqlDate2 = new java.sql.Date(Fine.getTime());
        Connection conn = DBManager.getConnection();
        String query ="SELECT
MODELLO,PREZZO,veicolo.TARGA,CAPIENZA,POTENZA,ALIMENTAZIONE,SEGMENTO FROM
VEICOLO where targa not in (select targa from noleggio where
((DataInizio<='"+sqlDate1.toString()+"' and DataFine>='"+sqlDate1.toString()+"') or
(DataInizio<='"+sqlDate2.toString()+"' and DataFine>='"+sqlDate2.toString()+"')) and
segmento= '"+segm+"' and segmento='"+segm+"' and StatoVeicolo='TRUE'»;
        PreparedStatement pstmt = conn.prepareStatement(query);
        ResultSet rs = pstmt.executeQuery();

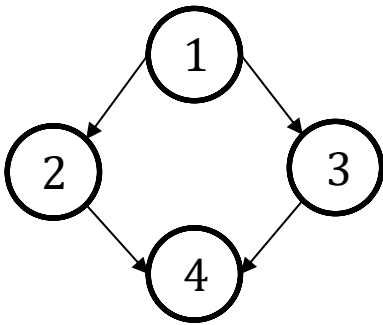
        while(rs.next()) {
            Auto a=new Auto(0,rs.getString("Modello"), rs.getFloat("Prezzo"),
rs.getString("Targa"), rs.getInt("Capienza"), rs.getInt("Potenza"),
rs.getString("Alimentazione"), rs.getString("Segmento"), true);
            ArrayAuto.add(a);
        }
    }
    catch(Exception e) {}

    return ArrayAuto;}
}
```

Numero di complessità ciclomatica: 2

Cammini linearmente indipendenti: 1-2-1-3

ModificaVeicolo



```
public static void query_modifica (String Targa, boolean StatoVeicolo ) {  
    try {  
        Connection conn = DBManager.getConnection();  
        String sqlquery;  
        if(StatoVeicolo)  
            sqlquery = "update veicolo set StatoVeicolo= 'TRUE' where targa='"+Targa+"'";  
        else sqlquery = "update veicolo set StatoVeicolo= 'FALSE' where targa='"+Targa+"'";  
        PreparedStatement stmt = conn.prepareStatement(sqlquery);  
        stmt.executeUpdate();  
        DBManager.closeConnection();  
    } catch (Exception e) {  
  
    }  
  
    DBManager.closeConnection();  
}
```

Numero di complessità ciclomatica: 2

Cammini linearmente indipendenti: 1-2-4; 1-3-4;