UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

CORSO DI LAUREA IN INGEGNERIA INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE
PROF. A.R. FASOLINO - A.A. 2021 - 22

Progetto

Stazione di Sostituzione e ricarica di batterie per auto elettriche

Studente:	Nome	Cognome	Matricola	Email
Giovanni	Cesarano	N46006030	giovan.cesarano@studenti.unina.it
Valeria	Coraggio	N46005125	vale.coraggio@studenti.unina.it
Fabio	Fernandez	N46005127	fa.fernandez@studenti.unina.it
Carmine	Ambrosino	N46004492	carmi.ambrosino@studenti.unina.it

Versione del *12/09/2022*

INDICE

1. Specifiche informali

La stazione fornisce batterie per specifiche autovetture elettriche. Ogni autovettura è caratterizzata dalla marca del fornitore e dal modello (es. Renault Zoe, Fiat 500E).
Ciascuna batteria è identificata da un codice univoco, un costo di sostituzione, e può essere sottoposta a un numero massimo di cicli di carica. Ogni batteria è specifica per un particolare modello di autovettura.
Per consentire ai clienti di usufruire dei servizi forniti dalla stazione, il sistema deve permettere loro di registrarsi fornendo: nome (stringa non vuota di max 15 caratteri), cognome (stringa non vuota di max 20 caratteri), data di nascita (data valida secondo il formato gg-mm-aaaa). Ai clienti registrati viene fornito un badge, caratterizzato da un codice univoco e dal credito residuo. Il badge viene utilizzato dai clienti per poter usufruire dei servizi offerti dalla stazione. Il sistema permette ai clienti registrati di poter ricaricare il credito del proprio badge.
Ad ogni ingresso della stazione è posizionato un terminale avente un lettore di badge e un display interattivo.
Un cliente registrato, per poter sostituire la propria batteria, deve prenotare una batteria. A tal fine, accede ad uno dei terminali, inserisce il proprio badge nel lettore e tramite il display seleziona il modello della propria autovettura. Il sistema verifica sia la disponibilità della batteria per il modello di autovettura selezionato, sia che il credito presente sul badge del cliente sia sufficiente per la sostituzione. In caso di esito negativo dei controlli, viene mostrato al cliente uno specifico messaggio di errore. In caso di esito positivo, una delle batterie compatibili disponibili sarà prenotata, il suo codice sarà restituito in output al cliente, e il credito del cliente verrà aggiornato sottraendo il costo di sostituzione della batteria.
Dopo la prenotazione, il cliente dovrà effettuare la sostituzione della batteria prenotata.
La sostituzione verrà eseguita da un addetto il quale, dopo aver installato la nuova batteria, dovrà registrare nel sistema l'avvenuta sostituzione, inserendo il codice della batteria installata, il codice del cliente, nonchè la data e l'ora dell'avvenuta sostituzione. Qualora il cliente non abbia eseguito la sostituzione entro 60 minuti dalla prenotazione, esso riceverà un ulteriore addebito di una penale sul proprio credito, pari al 25% del costo di sostituzione.
Dopo l'installazione, il sistema non manterrà più traccia della assegnazione esistente tra la vecchia batteria e il cliente.
Dopo aver eseguito la sostituzione, l'operatore potrà verificare se la batteria prelevata dall'autovettura può sostenere un ulteriore ciclo di carica. In caso affermativo la batteria sarà sottoposta ad una ricarica e successivamente ritornerà ad essere disponibile; in caso contrario, la batteria verrà dismessa e non sarà più disponibile.
Il sistema deve consentire al gestore della stazione di: visualizzare la disponibilità delle batterie per un dato modello di autovettura; caricare nel sistema nuove batterie; visualizzare la data e l'ora dell'ultimo ricambio effettuato da un dato cliente registrato; visualizzare l'elenco delle batterie quasi esauste (ossia che possono sostenere un numero di ricariche inferiore a 5).

2. Analisi e specifica dei requisiti

2.1 Analisi nomi-verbi

La stazione fornisce batterie per specifiche autovetture elettriche. Ogni autovettura è caratterizzata dalla marca del fornitore e dal modello (es. Renault Zoe, Fiat 500E).
Ciascuna batteria è identificata da un codice univoco, un costo di sostituzione, e può essere sottoposta a un numero massimo di cicli di carica. Ogni batteria è specifica per un particolare modello di autovettura.
Per consentire ai clienti di usufruire dei servizi forniti dalla stazione, il sistema deve permettere loro di registrarsi fornendo: nome (stringa non vuota di max 15 caratteri), cognome (stringa non vuota di max 20 caratteri), data di nascita (data valida secondo il formato gg-mm-aaaa). Ai clienti registrati viene fornito un badge, caratterizzato da un codice univoco e dal credito residuo. Il badge viene utilizzato dai clienti per poter usufruire dei servizi offerti dalla stazione. Il sistema permette ai clienti registrati di poter ricaricare il credito del proprio badge.
Ad ogni ingresso della stazione è posizionato un terminale avente un lettore di badge e un display interattivo.
Un cliente registrato, per poter sostituire la propria batteria, deve prenotare una batteria. A tal fine, accede ad uno dei terminali, inserisce il proprio badge nel lettore e tramite il display seleziona il modello della propria autovettura. Il sistema verifica sia la disponibilità della batteria per il modello di autovettura selezionato, sia che il credito presente sul badge del cliente sia sufficiente per la sostituzione. In caso di esito negativo dei controlli, viene mostrato al cliente uno specifico messaggio di errore. In caso di esito positivo, una delle batterie compatibili disponibili sarà prenotata, il suo codice sarà restituito in output al cliente, e il credito del cliente verrà aggiornato sottraendo il costo di sostituzione della batteria.
Dopo la prenotazione, il cliente dovrà effettuare la *sostituzione* della batteria prenotata.
La sostituzione verrà eseguita da un addetto il quale, dopo aver installato la nuova batteria, dovrà registrare nel sistema l'avvenuta sostituzione, inserendo il codice della batteria installata, il codice del cliente, nonchè la data e l'ora dell'avvenuta sostituzione. Qualora il cliente non abbia eseguito la sostituzione entro 60 minuti dalla prenotazione, esso riceverà un ulteriore addebito di una penale sul proprio credito, pari al 25% del costo di sostituzione.
Dopo l'installazione, il sistema non manterrà più traccia della assegnazione esistente tra la vecchia batteria e il cliente.
Dopo aver eseguito la sostituzione, l'operatore potrà verificare se la batteria prelevata dall'autovettura può sostenere un ulteriore ciclo di carica. In caso affermativo la batteria sarà sottoposta ad una ricarica e successivamente ritornerà ad essere disponibile; in caso contrario, la batteria verrà dismessa e non sarà più disponibile.
Il sistema deve consentire al gestore della stazione di: visualizzare la disponibilità delle batterie per un dato modello di autovettura; caricare nel sistema nuove batterie; visualizzare la data e l'ora dell'ultimo ricambio effettuato da un dato cliente registrato; visualizzare l'elenco delle batterie quasi esauste (ossia che possono sostenere un numero di ricariche inferiore a 5).

LEGENDA:
Classe
Attributo
Funzionalità
Attore
Classe-Attore

2.2 Revisione dei requisiti

1. *Di ogni Autovettura si vuole memorizzare nome, marca e modello*
2. *Il sistema deve offrire al Cliente una funzionalità per registrarsi.*
3. *Di ogni Cliente registrato si vuole memorizzare nome, cognome, data di nascita*
4. *Il sistema deve memorizzare un badge per ogni nuovo cliente registrato*
5. *Di ogni badge si vuole memorizzare un codice univoco ed il credito residuo*
6. *Il sistema deve offrire al Cliente una funzionalità per ricaricare il proprio credito*
7. *Il sistema deve offrire al Cliente una funzionalità per prenotare una nuova batteria*
8. *Il sistema deve offrire al Cliente una funzionalità per richiedere la sostituzione della batteria*
9. *Il sistema deve offrire all'Operatore una funzionalità per registrare l'avvenuta sostituzione*
10. *Di ogni Sostituzione si vuole memorizzare il codice cliente, il codice della batteria e la data*
11. *Il sistema deve applicare una penale sul credito del cliente pari al 25% del costo di sostituzione, qualora il cliente non abbia eseguito la sostituzione entro 60 minuti dalla prenotazione*
12. *Il sistema deve offrire all'Operatore la funzionalità di riutilizzo di una batteria*
13. *Il sistema deve offrire al Gestore della stazione la funzionalità di inserimento di nuove batterie*
14. *Di ogni Batteria si vuole memorizzare il codice, il costo ed il numero di cicli di ricarica*
15. *Il sistema deve offrire al Gestore della stazione la funzionalità di visualizzazione della disponibilità di batterie per autovettura*
16. *Il sistema deve offrire al Gestore della stazione la funzionalità di visualizzazione dell'ultima sostituzione effettuata da uno specifico Cliente*
17. *Il sistema deve offrire al Gestore della stazione la funzionalità di visualizzazione delle batterie quasi esauste*

2.3 Glossario dei termini

Termine	Descrizione	Sinonimi
Cliente registrato	Un cliente che ha effettuato la procedura di registrazione	
Operatore	Un impiegato della stazione che effettua manualmente le sostituzioni delle batterie	Addetto
Gestore della stazione	Un impiegato della stazione che si occupa delle funzionalità di amministrazione del sistema	

2.4 Classificazione dei requisiti

2.4.1 Requisiti funzionali

ID	Requisito	Origine (n. frase dei requisiti revisionati)
RF01	Il sistema deve offrire al Cliente una funzionalità per registrarsi	2
RF02	Il sistema deve memorizzare un badge per ogni nuovo cliente registrato	4
RF03	Il sistema deve offrire al Cliente una funzionalità per ricaricare il proprio credito residuo	5
RF04	Il sistema deve offrire al Cliente una funzionalità per prenotare una nuova batteria	7
RF05	Il sistema deve offrire al Cliente una funzionalità per richiedere la sostituzione della batteria	8
RF06	Il sistema deve offrire all'Operatore una funzionalità per registrare l'avvenuta sostituzione	9
RF07	Il sistema deve applicare una penale sul credito del cliente pari al 25% del costo di sostituzione, qualora il cliente non abbia eseguito la sostituzione entro 60 minuti dalla prenotazione	11
RF08	Il sistema deve offrire all'Operatore la funzionalità di riutilizzo di una batteria	12
RF09	Il sistema deve offrire al Gestore della stazione la funzionalità di inserimento di nuove batterie	13
RF10	Il sistema deve offrire al Gestore della stazione la funzionalità di visualizzazione della disponibilità di batterie per autovettura	15
RF11	Il sistema deve offrire al Gestore della stazione la funzionalità di visualizzazione dell'ultima sostituzione effettuata da uno specifico Cliente	16
RF12	Il sistema deve offrire al Gestore della stazione la funzionalità di visualizzazione delle batterie quasi esauste	17

2.4.2 Requisiti sui dati

ID	Requisito	Origine (n. frase dei requisiti revisionati)
RD01	Di ogni Autovettura si vuole memorizzare nome, marca e modello	1
RD02	Di ogni Cliente registrato si vuole memorizzare nome, cognome, data di nascita	3
RD03	Di ogni badge si vuole memorizzare un codice univoco ed il credito residuo	5
RD04	Di ogni Sostituzione si vuole memorizzare il codice cliente e la data	10
RD05	Di ogni Batteria si vuole memorizzare il codice, il costo ed il numero di cicli di ricarica	14

2.4.3 Vincoli / Altri requisiti

ID	Requisito	Origine (n. frase dei requisiti revisionati)
V01		
RNF01		

2.5 Modellazione dei casi d'uso

2.5.1 Attori e casi d'uso

Attori Primari:

- Cliente
- Cliente registrato
- Operatore
- Gestore della stazione

Attori Secondari:

- Nessuno

Casi d'uso:

- UC1: Effettua Registrazione
- UC2: Assegna Badge
- UC3: Ricarica Credito
- UC4: Prenota Batteria
- UC5: Richiedi Sostituzione
- UC6: Effettua Sostituzione
- UC7: Verifica Batteria
- UC8: Visualizza Disponibilità Batterie

- UC9: Carica Batteria
- UC10: Visualizza Ultima Sostituzione Cliente
- UC11: Visualizza Batterie Esauste

Casi d' uso di inclusione:

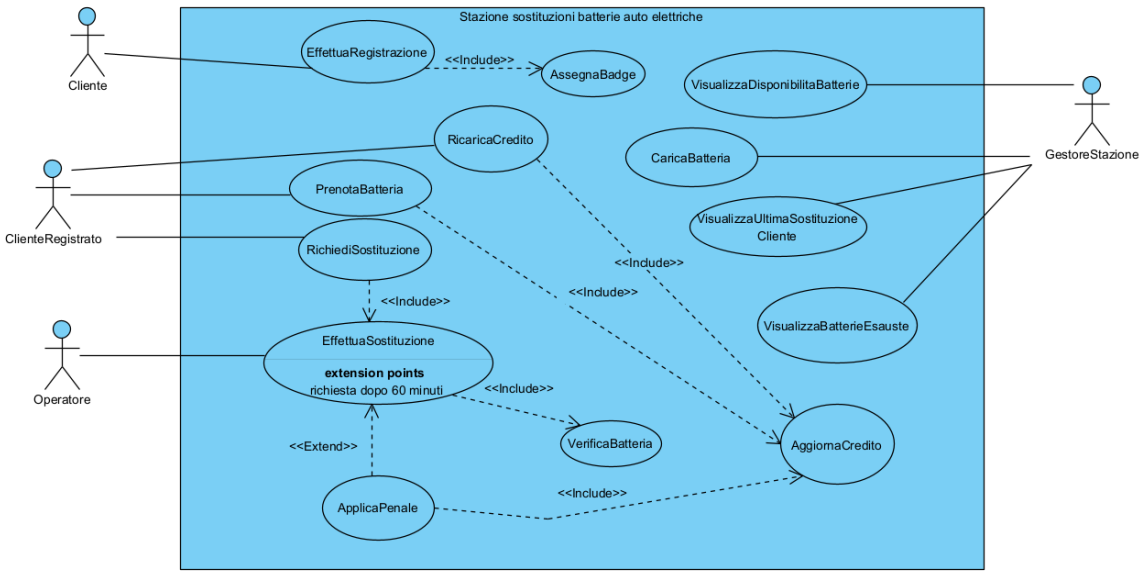
- UC12: Aggiorna Credito

Casi d' uso di estensione:

- UC13: Applica Penale

Caso d'uso	**Attori Primari**	**Attori Secondari**	**Incl. / Ext.**	**Requisiti corrispondenti**
UC1: Effettua Registrazione	Cliente	-	Include Effettua Registrazione	RF01
UC1: Assegna Badge	Cliente	-	Incluso in Effettua Registrazione	RF02
UC3: RicaricaCredito	Cliente Registrato	-	Include AggiornaCredito	RF03
UC4: PrenotaBatteria	Cliente Registrato	-	-	RF04
UC5: RichiediSostituzione	Cliente Registrato	-	-	RF05
UC6: EffettuaSostituzione	Operatore	-	Include VerificaBatteria	RF06
UC7: VerificaBatteria	Operatore	-	Incluso in EffettuaSostituzione	RF08
UC8:Visualizza Disponibilita Batterie	Gestore Stazione	-	-	RF10
UC9: CaricaBatteria	Gestore Stazione	-	-	RF09
UC10: Visualizza Ultima Sostituzione Cliente	Gestore Stazione	-	-	RF11
UC11: Visualizza Batterie Esauste	Gestore Stazione	-	-	RF12
UC12: AggiornaCredito	Cliente Registrato	-	Incluso in Ricarica Credito	RF03
UC12: AggiornaCredito	Cliente Registrato	-	Incluso in Prenota Batteria	RF04
UC12: AggiornaCredito	Operatore	-	Incluso in Applica Penale	RF06
UC13: ApplicaPenale	Operatore	-	Estensione di Effettua Sostituzione	RF07

2.5.2 Diagramma dei casi d'uso



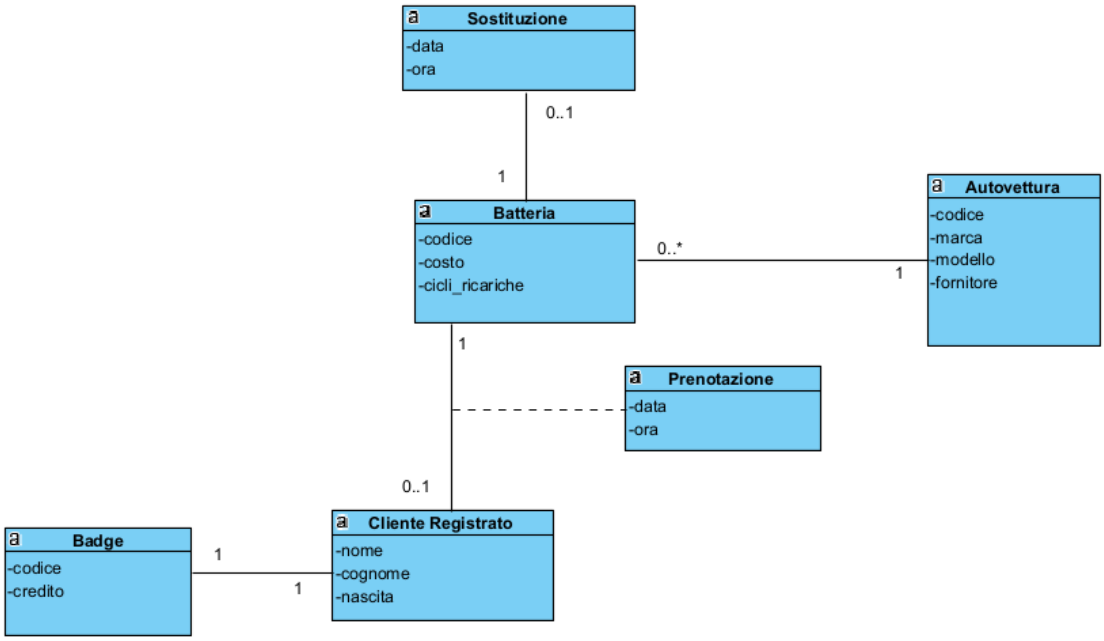
2.5.3 Scenari

Caso d'uso:	**EffettuaSostituzione**
Attore primario	Operatore
Attore secondario	-
Descrizione	Un operatore della stazione effettua la sostituzione della batteria prenotata dal Cliente
Pre-Condizioni	Lo stato della sostituzione deve essere richiesto dal cliente
Sequenza di eventi principale	1. Il caso d'uso inizia dopo che il Cliente richiede la sostituzione 2. L'Operatore inserisce il codice della batteria, codice cliente, data e ora dell'avvenuta sostituzione 3. SE il codice batteria o il codice cliente non è valido, 3.1. il sistema restituisce un ERRORE all'operatore, 4. ALTRIMENTI 4.1. Il sistema verifica quanto tempo è trascorso dalla prenotazione del Cliente alla richiesta di sostituzione. 4.2. SE è passato un tempo in minuti superiore a 60 minuti 4.2.1. Viene applicata una penale al Cliente pari al 25% del costo della batteria richiesta 4.3. include(VerificaBatteria)
Post-Condizioni	La sostituzione effettuata viene inserita della base di dati.
Casi d'uso correlati	*nessuno*
Sequenza di eventi alternativi	-
Attore primario	Operatore
Attore secondario	-

Caso d'uso:	VerificaBatteria
Attore primario	Operatore
Attore secondario	-
Descrizione	Un operatore della stazione verifica se la batteria prelevata dall'autovettura può sostenere un ulteriore ciclo di carica
Pre-Condizioni	L'operatore deve avere prelevato la batteria dall'autovettura
Sequenza di eventi principale	1. Il caso d'uso inizia dopo che l'operatore ha prelevato la batteria 2. L'operatore verifica lo stato della batteria 4. SE è riutilizzabile 4.1. L'operatore effettuerà un ciclo di ricarica 4.2. L'operatore aggiorna la lista delle batterie disponibili 5. ALTRIMENTI 5.1. L'operatore rende la batteria non più disponibile
Post-Condizioni	La sostituzione effettuata viene inserita della base di dati.
Casi d'uso correlati	*nessuno*
Sequenza di eventi alternativi	-
Attore primario	Operatore
Attore secondario	-

2.6 Diagramma delle classi

Diagramma delle classi di analisi.



2.6 Diagrammi di sequenza

Diagramma di sequenza di analisi per il caso d'uso effettuaRegistrazione.

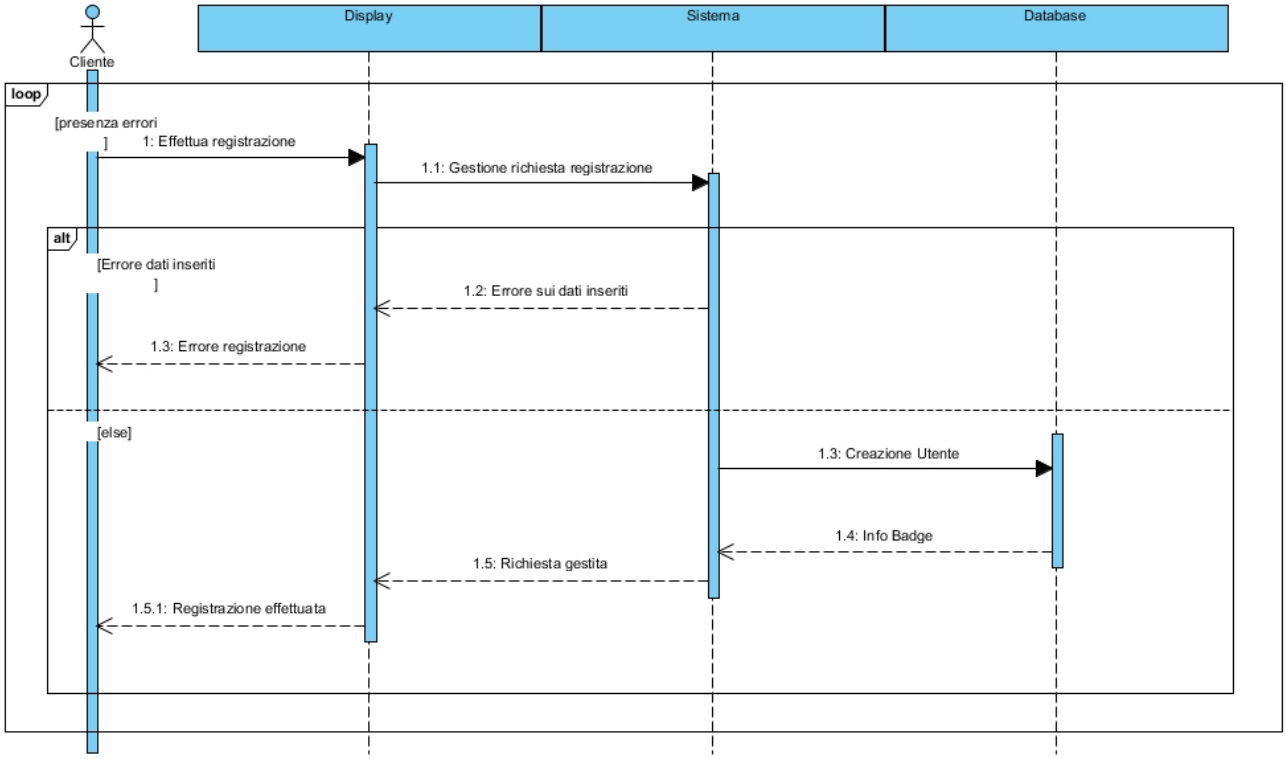
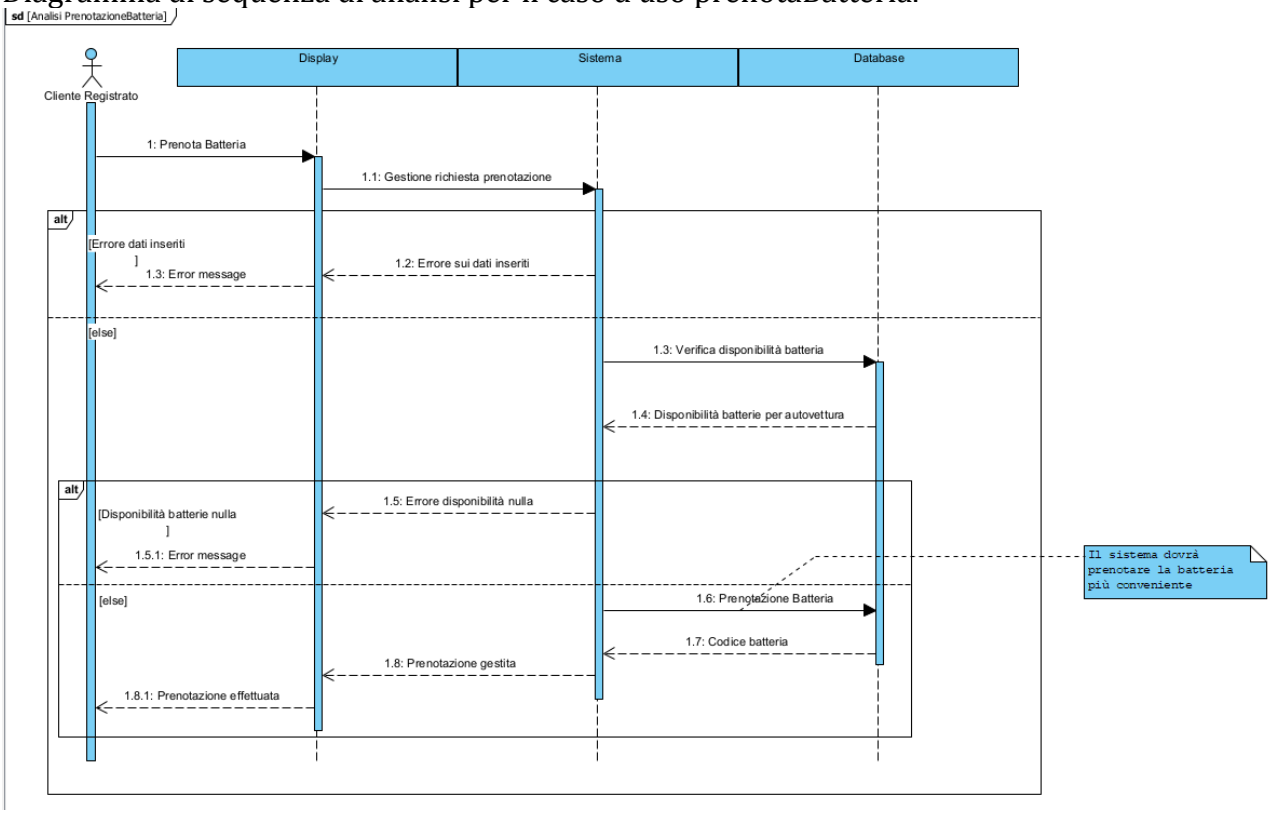


Diagramma di sequenza di analisi per il caso d'uso prenotaBatteria.



2.7 Verifica della completezza dei requisiti

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

- **RF01** è modellato nell'UCD con l'attore "Cliente" e con il caso d'uso UC1
- **RF02** è modellato nell'UCD con l'attore "Cliente" e con il caso d'uso UC2
- **RF03** è modellato nell'UCD con l'attore "Cliente Registrato" e con il caso d'uso UC3
- **RF04** è modellato nell'UCD con l'attore "Cliente Registrato" e con il caso d'uso UC4
- **RF05** è modellato nell'UCD con l'attore "Cliente Registrato" e con il caso d'uso UC5
- **RF06** è modellato nell'UCD con l'attore "Operatore" e con il caso d'uso UC6
- **RF07** è modellato nell'UCD con l'attore "Operatore" e con il caso d'uso UC12
- **RF08** è modellato nell'UCD con l'attore "Operatore" e con il caso d'uso UC7
- **RF09** è modellato nell'UCD con l'attore "Gestore della stazione" con il caso d'uso UC9
- **RF10** è modellato nell'UCD con l'attore "Gestore della stazione" con il caso d'uso UC8
- **RF11** è modellato nell'UCD con l'attore "Gestore della stazione" con il caso d'uso UC10
- **RF12** è modellato nell'UCD con l'attore "Gestore della stazione" e il caso d'uso UC11
- **RD01** è modellato nel CD con gli attributi "nome", "marca" e "modello" della classe "Autovettura"
- **RD02** è modellato nel CD con gli attributi "nome", "cognome", "data di nascita" della classe "Cliente"
- **RD03** è modellato nel CD con gli attributi "codice univoco" ed il "credito residuo" della classe "Badge"
- **RD04,** è modellato nel CD con gli attributi "codice cliente" e la "data" della classe "Sostituzione"
- **RD05** è modellato nel CD con gli attributi il "codice", il "costo" ed il "numero di cicli di ricarica" della classe "Batteria"

3. Stima dei costi

- Tabella di riferimento per le complessità di dati e transazioni

	SEMPLICE	MEDIO	COMPLESSO
NILF	3	6	10
NEIF	2	4	7
NEI	1	3	5
NEO	2	4	7
NEQ	1	3	6

- Tabella elenco dei fattori correttivi (il cui valore è compreso tra 0 e 5)

FATTORI CORRETTIVI
COMUNICAZIONE DATI
DISTRIBUZIONE ELABORAZIONE
PRESTAZIONI

UTILIZZO INTENSIVO CONFIGURAZIONE
FREQUENZA DELLE TRANSAZIONI
INSERIMENTO DATI INTERATTIVO
EFFICIENZA PER L'UTENTE FINALE
AGGIORNAMENTO INTERATTIVO
COMPLESSITA' ELABORATIVA
RIUSABILITA'
FACILITA' INSTALLAZIONE
FACILITA' GESTIONE OPERATIVA
MOLTEPLICITA' DI SITI
FACILITA' DI MODIFICA

EFFETTUAREGISTRAZIONE

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	2	3			6
NEIF	0				
NEI	4	1			4
NEO	1	2			2
NEQ	0				

NILF: 2 di tabelle del db relazionale (Cliente e Badge)
NEI: 4 input (Nome, Cognome, Nascita, Credito)
NEO: il calcolo del codice badge

UFP=	**12**
LLOC/FP =	**53**

FATTORI CORRETTIVI

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

FP=	10,2
JAVA =	540

	20

FP = 12 * (0,65 + 0,01 * 20)
LLOC = FP * 53 = 10,2* 53 =540

PRENOTAZIONEBATTERIA

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	3	3			9
NEIF	0				
NEI	2	1			2
NEO	2	2			4
NEQ	2	1			2

UFP=	17
LLOC/FP =	53

NILF: 3 di tabelle del db relazionale (Badge, Prenotazione, Batteria)
NEI: 2 input (codice_badge, codice_autovettura)
NEO: elaborazione per la ricerca della batteria, elaborazione per fornire codice_prenotazione
NEQ: 2 query eseguite per il recupero dei dati (nei caricaBadge() e caricaAutovettura());

FATTORI CORRETTIVI

COMUNICAZIONE DATI	1
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	1
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	1
INSERIMENTO DATI INTERATTIVO	0
EFFICIENZA PER L'UTENTE FINALE	2
AGGIORNAMENTO INTERATTIVO	0
COMPLESSITA' ELABORATIVA	2
RIUSABILITA'	1
FACILITA' INSTALLAZIONE	0
FACILITA' GESTIONE OPERATIVA	0
Molteplicità DI SITI	0
FACILITA' DI MODIFICA	1
	9

FP = 17 * (0,65 + 0,01 * 9)
LLOC = FP * 53 = 12,6* 53 =666

FP=	12,6
JAVA =	666

EFFETTUASOSTITUZIONE

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	3	3			9
NEIF	0				
NEI	1	1			1
NEO	3	2			6
NEQ	3	1			3

UFP=	**19**
LLOC/FP =	**53**

NILF: 3 di tabelle del db relazionale (Sostituzione,Batteria,Badge)
NEI: 1 input (codice_prenotazione)
NEO: applica penale, batteria non riutilizzabile, sostituzione effettuata
NEQ: 3 query eseguite per il recupero dei dati (caricaSostituzione(), caricaPrenotazione(), caricaBatteriaByCodice());

FATTORI CORRETTIVI

COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	1
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	1
INSERIMENTO DATI INTERATTIVO	0
EFFICIENZA PER L'UTENTE FINALE	1
AGGIORNAMENTO INTERATTIVO	0
COMPLESSITA' ELABORATIVA	2
RIUSABILITA'	1
FACILITA' INSTALLAZIONE	0
FACILITA' GESTIONE OPERATIVA	0
Molteplicità DI SITI	0
FACILITA' DI MODIFICA	1
	9

FP = 19 * (0,65 + 0,01 * 9)
LLOC = FP * 53 = 14* 53 =745

FP=	**14**
JAVA =	**745**

4. Piano di test funzionale

PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ "*EffettuaRegistrazione*".

NOME	COGNOME	NASCITA	CREDITO
• Stringa di caratteri di lunghezza >0 e <=15 • Stringa di caratteri di lunghezza nulla [ERROR] • Stringa di caratteri di lunghezza > 15 [ERROR] • Stringa che contiene simboli che non sono caratteri [ERROR]	• Stringa di caratteri di lunghezza >0 e <= 20 • Stringa di caratteri di lunghezza nulla [ERROR] • Stringa di caratteri di lunghezza > 20 [ERROR] • Stringa che contiene simboli che non sono caratteri [ERROR]	• Data con formato valido(gg-mm-aaaa) • Data con formato non valido [ERROR]	• Valore decimale >= 0 • Valore decimale < 0 [ERROR] • Valore non numerico [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $4 * 4 * 2 * 3 = 96$
Introduciamo i vincoli [ERROR].
Il numero di test da eseguire per testare singolarmente i vincoli è 9 (3 per Nome, 3 per Cognome, 1 per Nascita, 2 per Credito).
Il numero di test risultante è: (1*1*1*1) + 9 = 10

TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	Nome valido Cognome valido Nascita valida Credito valido		{Nome: “Giovanni”, Cognome: “Cesarano”, Nascita: “04-04-1999”, Credito: 1000}	Registrazione effettuata con successo.	Viene aggiunto un cliente nel database con un rispettivo codice_badge.
2	Nome stringa nulla	Nome stringa nulla[ERROR] Cognome valido Nascita valida Credito valido		{Nome: “ ”, Cognome: “Fernandez” Data: “25/12/2021”, Credito: 500}	Nome non valido.	
3	Nome stringa > 15 caratteri	Nome stringa > 15 caratteri[ERROR] Cognome valido Nascita valida Credito valido		{Nome: “aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaa”, Cognome: “Fernandez” Data: “25/12/2021”, Credito: 500}	Nome non valido.	
4	Nome stringa con simboli	Nome stringa con simboli [ERROR], Cognome valido Nascita valida Credito valido		{Nome: “§ç£”, Cognome: “Coraggio” Nascita: “25/12/2021”, Credito: 500}	Nome non valido.	
5	Cognome stringa nulla	Nome valido Cognome stringa nulla[ERROR] Nascita valida		{Nome: “Fabio” Cognome: “ ”, Data: “25/12/2021”, Credito: 500}	Cognome non valido.	

		Credito valido				
6	Cognome stringa > 20 caratteri	Nome valido Cognome stringa > 20 caratteri[ERROR] Nascita valida Credito valido		{Nome: “Fabio” Cognome: “bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb”, Data: “25/12/2021”, Credito: 500}	Cognome non valido.	
7	Cognome stringa con simboli	Nome valido Cognome stringa con simboli [ERROR], Nascita valida Credito valido		{Nome: ”Valeria” Cognome: “§ç£”, Nascita: “25/12/2021”, Credito: 500}	Cognome non valido.	
8	Data di nascita con formato non valido	Nome valido, Cognome valido formato nascita non valido [ERROR], Credito valido		{Nome: “Eva”, Cognome: “Rossi” Data: “225/1332/2021”, Credito: 500}	Formato data non valido.	
9	Credito < 0	Nome valido, Cognome valido Nascita valida Credito < 0 non valido [ERROR]		{Nome: “Giovanni”, Cognome: “Cesarano”, Nascita: “04-04-1999”, Credito: -100}	Credito non valido.	
10	Credito non numerico	Nome valido, Cognome valido Nascita valida Credito non numerico non valido [ERROR]		{Nome: “Giovanni”, Cognome: “Cesarano”, Nascita: “04-04-1999”, Credito: “sjdfw”}	Credito non valido.	

PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ "P*renotazioneBatteria*".

CODICE_BADGE	CODICE_AUTOVETTURE
• Valore intero >0 • Valore intero <=0 [ERROR] • Valore non numerico [ERROR]	• Valore intero >0 • Valore intero <=0 [ERROR] • Valore non numerico [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $3 * 3 = 9$
Introduciamo i vincoli [ERROR].
Il numero di test da eseguire per testare singolarmente i vincoli è 4 (2 per Codice_Badge, 2 per Codice_Autovetture).
È necessario inoltre effettuare un controllo su due pre-condizioni di sistema:

1. Se il codice del badge risulta assegnato ad un cliente per poter effettuare una prenotazione
2. Se il credito del cliente è sufficiente

Il numero di test risultante è: (1*1) + 4 +2 = 7

TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	codice_badge valido codice_autovettura valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = 5043 }	Prenotazione della batteria effettuata con successo.	La prenotazione è aggiunta al database

2	Tutti input validi, errore per la pre-condizione badge	codice_badge valido codice_autovettura valido	codice_badge non è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = 5043 }	Badge non trovato.	
3	Tutti input validi, errore per la pre-condizione credito	codice_badge valido codice_autovettura valido	codice_badge è presente nel database, il credito non è sufficiente	{codice_badge = 35, codice_autovettura = 5043 }	Credito non sufficiente.	
4	codice_badge < =0	codice_badge <=0 [ERROR] codice_autovettura valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = -35, codice_autovettura = 5043 }	Badge non valido.	
5	codice_badge non numerico	codice_badge non int [ERROR] codice_autovettura valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = “sdsge”, codice_autovettura = 5043 }	Badge non valido.	
6	codice_autovettura < =0	codice_autovettura <=0 [ERROR] codice_badge valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = -5043 }	Codice autovettura non valido.	
7	codice_autovettura non numerico	codice_autovettura non int [ERROR] codice_badge valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = “sges” }	Codice autovettura non valido.	

PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ "*EffettuaSostituzione*".

CODICE_BATTERIA	CODICE_BADGE	TIMESTAMP
• Valore intero > 0 • Valore intero <=0 [ERROR] • Valore non numerico [ERROR]	• Valore intero > 0 • Valore intero <=0 [ERROR] • Valore non numerico [ERROR]	• Timestamp valido • Timestamp non valido[ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: 3 + 3 + 2
Introduciamo i vincoli [ERROR].
Il numero di test da eseguire per testare singolarmente i vincoli è 5(2 per codice_batteria, 2 per codice_badge, 1 per timestamp).
È necessario inoltre effettuare un controllo su due pre-condizioni di sistema:

1. Se è stata prenotata una batteria con codiceBatteria.
2. Se la sostituzione su un codice_batteria è già stata eseguita.

Il numero di test risultante è: (1*1*1) + 5 + 2 = 8

TEST SUITE

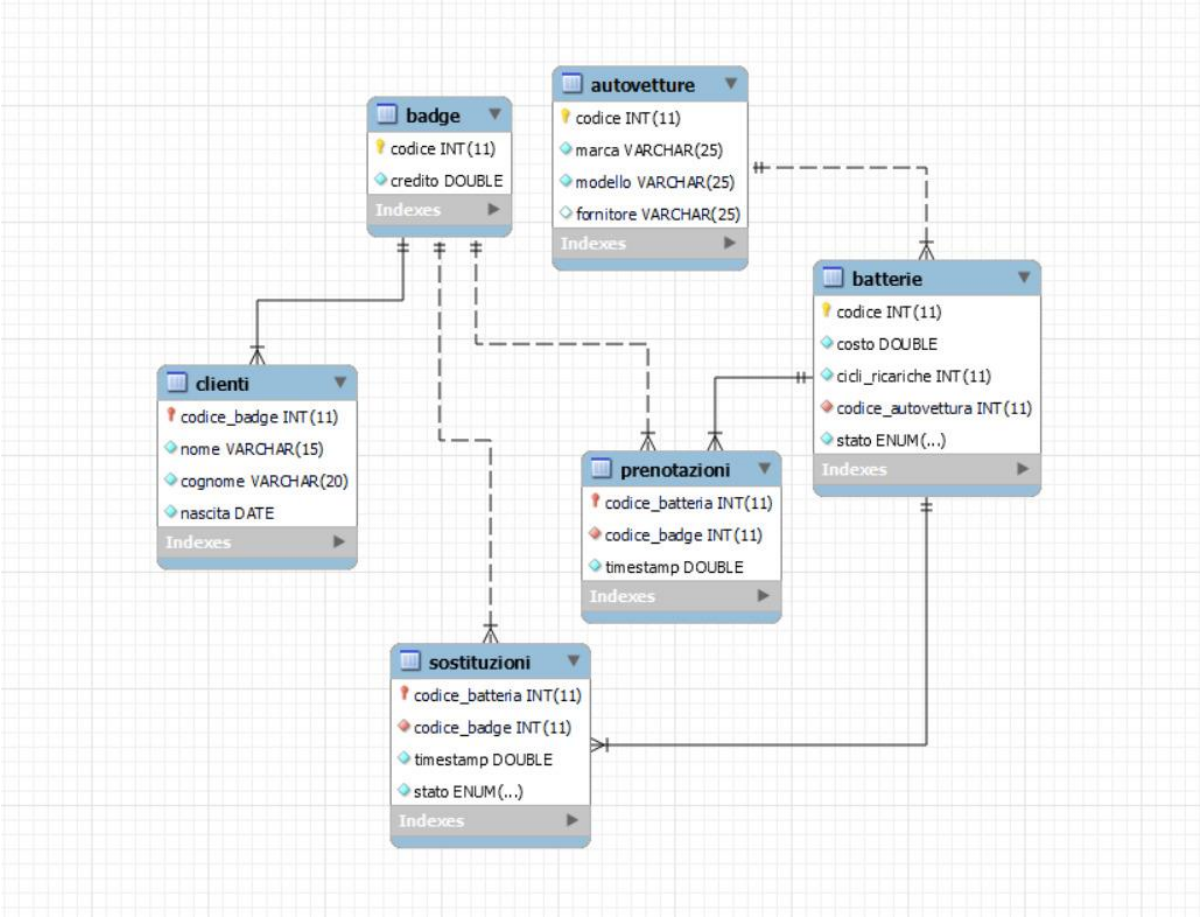
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi, precondizioni valide	codice_batteria valido codice_badge valido	La batteria è stata prenotata	{codice_batteria: 1234, codice_badge: 123,	Sostituzione effettuata con successo.	La sostituzione è aggiunta al database

		timestamp valido	La batteria non è stata già sostituita	timestamp: 1663071611132 }		
2	Input validi, errore di pre-condizione, batteria non è stata prenotata	codice_batteria valido codice_badge valido timestamp valido	La batteria non è stata prenotata La batteria non è stata sostituita	{codice_batteria: 1234, codice_badge: 123, timestamp: 1663071611132 }	La batteria deve essere stata prenotata per poterla sostituire	
3	Input valido, errore di pre-condizione, la batteria è già stata sostuita	codice_batteria valido codice_badge valido timestamp valido	La batteria è stata prenotata È stata già eseguita una sostituzione con codice_batteria	{codice_batteria: 1234, codice_badge: 123, timestamp: 1663071611132 }	La batteria può essere ssotituita una sola volta	
4	codice_batteria<= 0	codice_batteria <= 0 [ERROR] codice_badge valido timestamp valido	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: -1234, codice_badge: 123, timestamp: 1663071611132 }	Codice batteria non valido	
5	codice_batteria non numerico	codice_batteria non numerico [ERROR] codice_badge valido timestamp valido	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: “wow”, codice_badge: 123, timestamp: 1663071611132 }	Codice batteria non valido.	
6	codice_badge<=0	codice_batteria valido codice_badge<=0[ERROR]	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: 1234, codice_badge: -123,	Codice badge non valido.	

		timestamp valido		timestamp: 1663071611132 }		
7	codice_badge non numerico	codice_batteria valido codice_badge non numerico [ERROR] timestamp valido	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: 1234, codice_badge: “wow”, timestamp: 1663071611132 }	Codice batteria non valido.	
8	timestamp non valido	codice_batteria valido codice_badge valido timestamp non valido[ERROR]	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: 1234, codice_badge: 123, timestamp: “wow”}	Timestamp non valido	

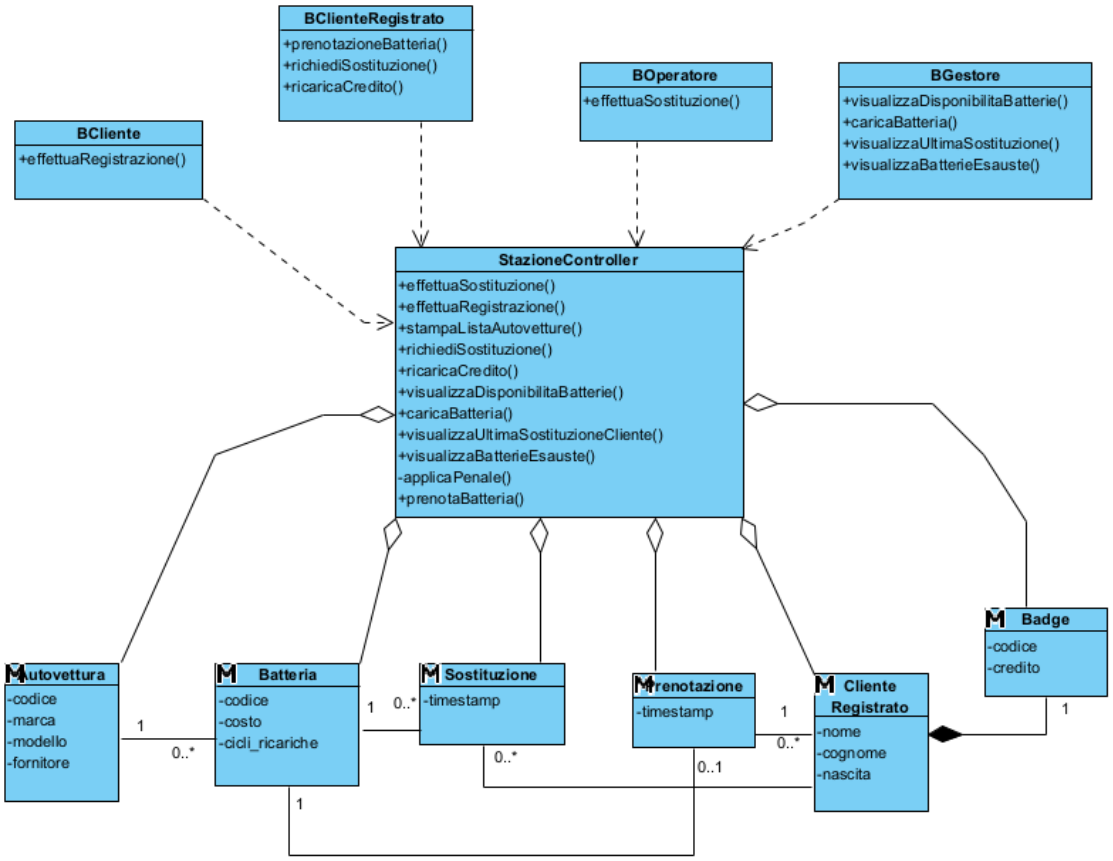
5. Progettazione

Schema base dati:

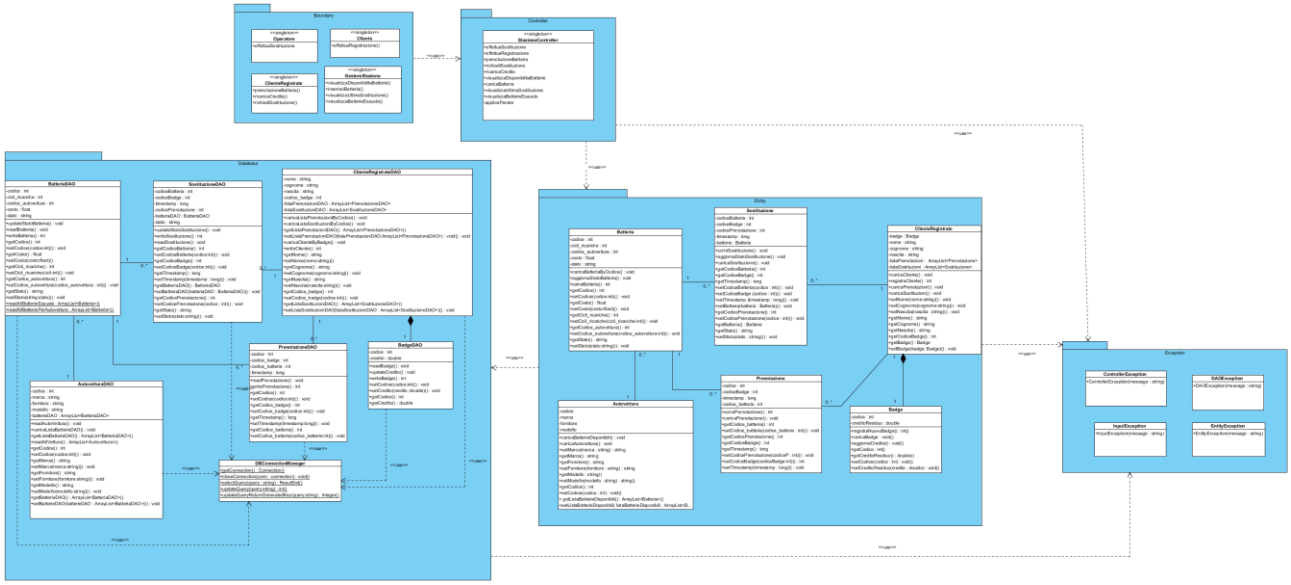


5.1 Diagramma delle classi

Diagramma delle classi raffinato (con classi Control e Boundary).



Package Diagram:



5.2 Diagrammi di sequenza

Diagramma di sequenza del caso d'uso EffettuaRegistrazione:

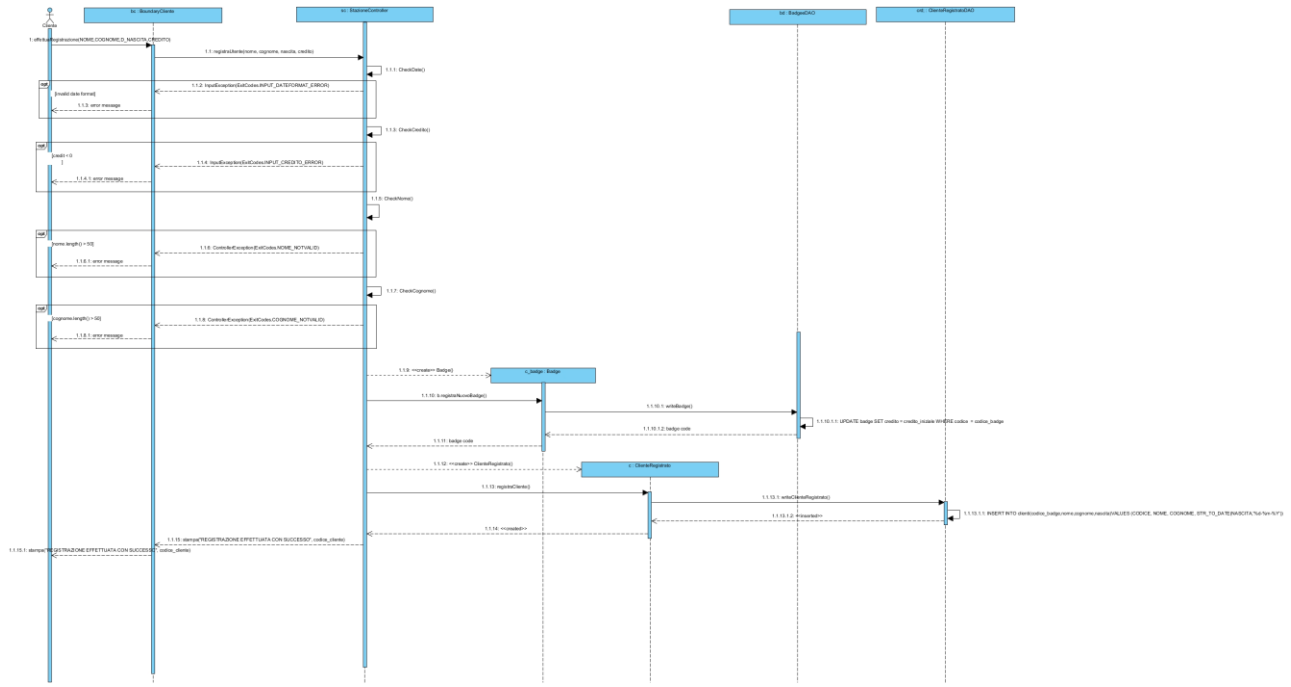
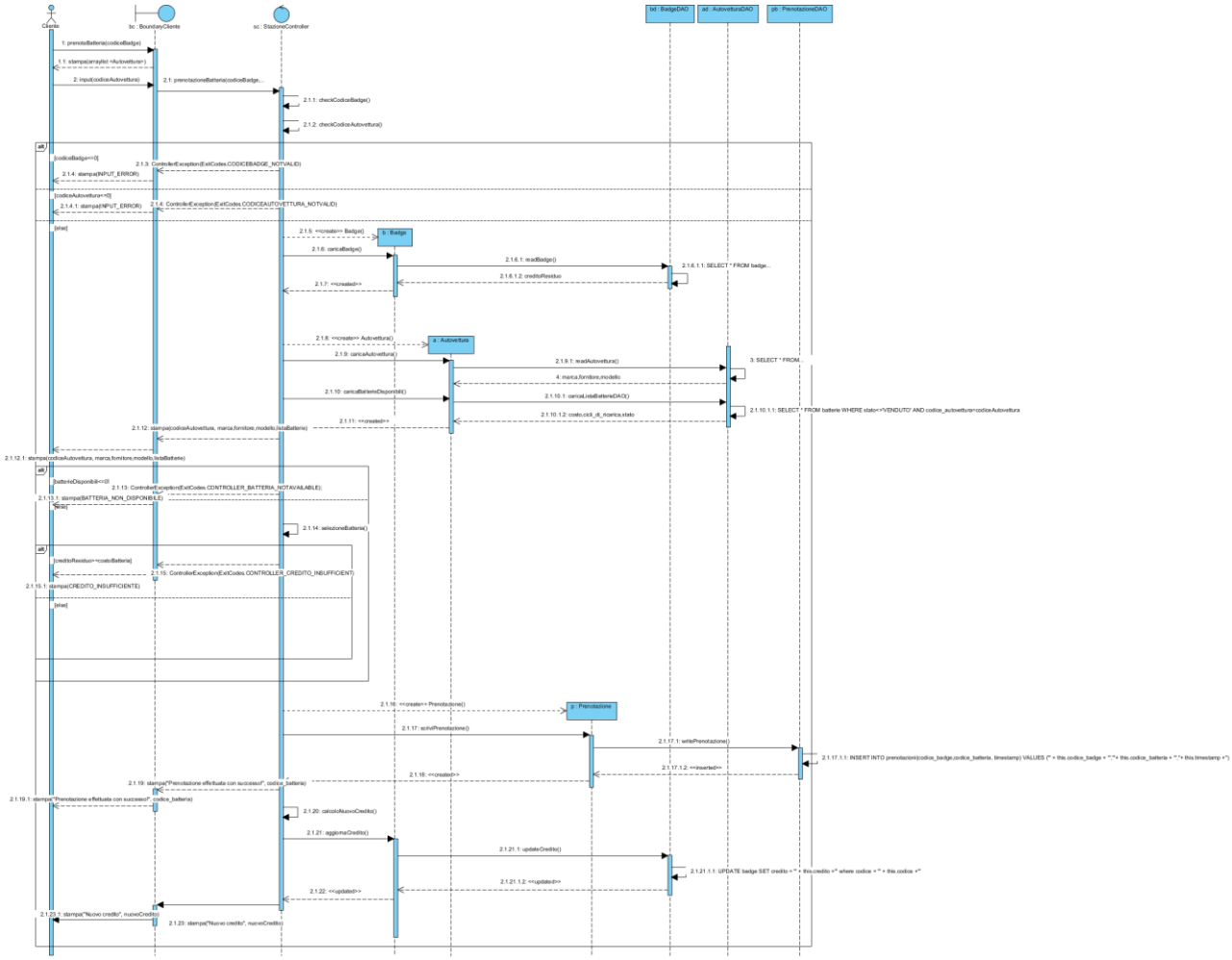


Diagramma di sequenza del caso d'uso PrenotaBatteria:

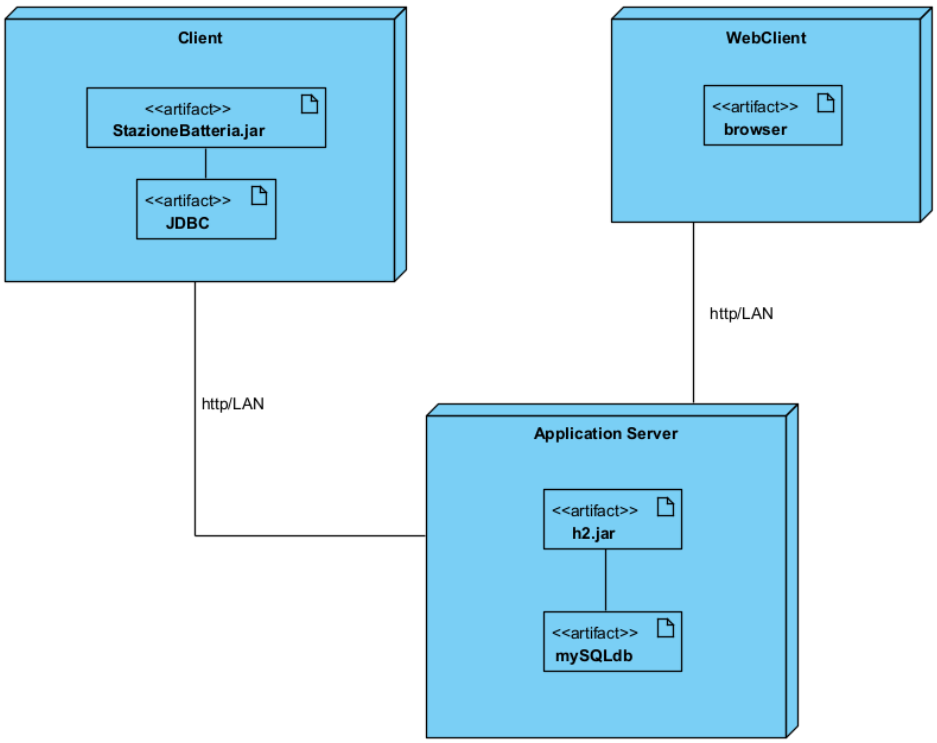


6. Implementazione

Package: Boundary, Controller, Database, Entity, Exception, Test
Classi: Badge, ClienteRegistrato, Autovettura, Prenotazione, Sostituzione, Batteria, BadgeDAO, ClienteRegistratoDAO, AutovetturaDAO, PrenotazioneDAO, SostituzioneDAO, BatteriaDAO, StazioneController, Boundary, DateValidator, DBCollectionManager, ExitCode, TestEffettuaRegistrazione, TestEffettuaSostituzione, TestEffettuaPrenotazioneBatteria
Tipi di eccezione: DAOexception, ControllerException, BoundaryException, EntityException, InputException

Elenco artefatti: Stazione.jar, h2.jar, driverh2, db.sql

Deployment diagram:



LOC complessive: 1952

Stima LLOC in Java per EffettuaRegistrazione(): 540
Stima LLOC in Java per PrenotaBatteria(): 665
Stima LLOC in Java per EffettuaSostituzione(): 745

LLOC in Java per EffettuaRegistrazione(): 200
LLOC in Java per PrenotaBatteria(): 272
LLOC in Java per EffettuaSostituzione(): 407

7. Testing

7.1 Test strutturale

7.1.1 Complessità ciclomatica

Metodo effettuaRegistrazione()

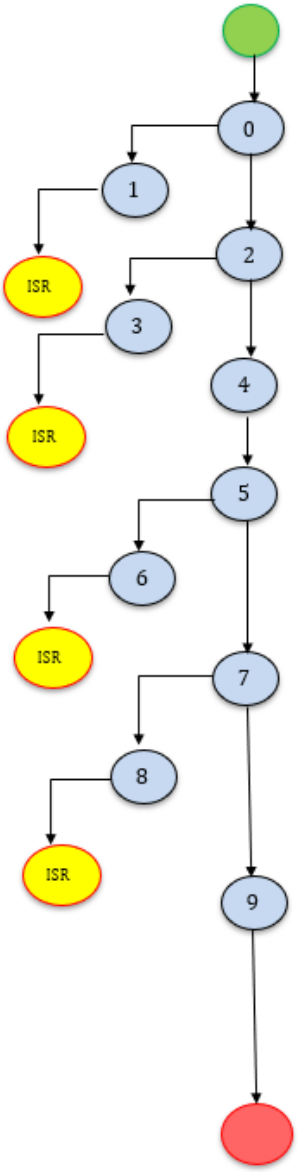
```java
public int effettuaRegistrazione(String nome, String cognome, String nascita, float credito) throws
InputException, EntityException {

        if(nome.length() > 15 || nome.length() == 0) {
          throw new InputException(ExitCodes.NOME_NOTVALID);
        }
        if(cognome.length() > 20 || cognome.length() == 0) {
          throw new InputException(ExitCodes.COGNOME_NOTVALID);
        }


        DateValidator validator = new DateValidatorUsingDateFormat("dd-mm-yyyy");
        if(!validator.isValid(nascita)) throw new InputException(ExitCodes.INPUT_DATEFORMAT_ERROR);

        if(credito < 0) throw new InputException(ExitCodes.INPUT_CREDITO_ERROR);

        ClienteRegistrato c = new ClienteRegistrato();
        Badge b = new Badge();
        b.setCreditoResiduo(credito);
        int codice = b.registraNuovoBadge();
        c.setBadge(b);

        c.setNome(nome);
        c.setCognome(cognome);
        c.setNascita(nascita);

        c.registraCliente();

        /*ClienteDAO cd = new ClienteDAO(new Cliente(nome,cognome,nascita, credito));
        cd.writeCliente();*/

        System.out.println("Registrazione effettuata con successo!\nCodice_badge: " + codice);
        return codice;

    }
```

NUMERO CICLOMATICO: 5
numero di regioni chiuse del grafo +1 = 5
numero di nodi predicati (4) +1 = 5
archi – # nodi + 2 = (13 - 10) + 2 = 5

CAMMINI:

1. 0-1
2. 0-2-3
3. 0-2-4-5-6
4. 0-2-4-5-7-8
5. 0-2-4-5-7-9

Control Flow Graph



Metodo prenotazioneBatteria()

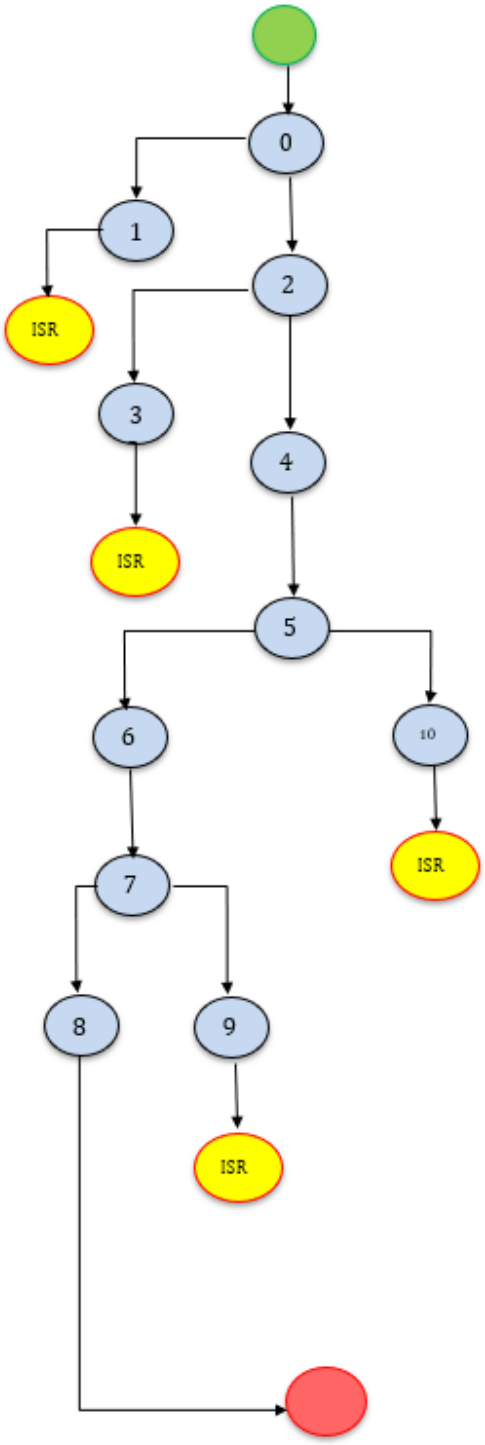
```
public int prenotaBatteria(int codice_badge, int codice_autovettura) throws ControllerException,
DAOException {

        if(codice_badge <= 0) {
          throw new ControllerException(ExitCodes.CODICEBADGE_NOTVALID);
        }

        if(codice_autovettura <= 0) {
          throw new ControllerException(ExitCodes.CODICEAUTOVETTURA_NOTVALID);
        }

        Badge b = new Badge();
        b.setCodice(codice_badge);
        b.caricaBadge();

        Autovettura av = new Autovettura();
        av.setCodice(codice_autovettura);
        av.caricaAutovettura();
        av.caricaBatterieDisponibili();
        System.out.println(av);


        if(av.getListaBatterieDisponibili().size() > 0) {

            System.out.println(b);

            Batteria batt = av.cercaBatteriaMenoEsausta();

            if(b.getCreditoResiduo() >= batt.getCosto()) {
                System.out.println("Prenotata:" + batt);
                batt.aggiornaStatoBatteria();
                Prenotazione p = new Prenotazione();
                p.setCodiceBadge(codice_badge);
                p.setCodice_batteria(batt.getCodice());
                int codice = p.scriviPrenotazione();
                System.out.println("prenotazione effettuata con successo!");
                System.out.println("CodiceBatteriaPrenotata: "+batt.getCodice());
                double nuovoCredito = b.getCreditoResiduo() - batt.getCosto();
                b.setCreditoResiduo(nuovoCredito);
                b.aggiornaCredito();
                System.out.println("Nuovo credito: " + nuovoCredito);
            }else {
                throw new ControllerException(ExitCodes.CONTROLLER_CREDITO_INSUFFICIENT);
                //System.err.println("Credito non sufficiente!");
            }
        }else {
            throw new ControllerException(ExitCodes.CONTROLLER_BATTERIA_NOTAVAILABLE);
            //System.err.println("Batteria non disponibile per questo modello!");
        }

        return 0;
    }
```

NUMERO CICLOMATICO: 5
numero di regioni chiuse del grafo +1 = 5
numero di nodi predicati (4) +1 = 5
archi – # nodi + 2 = (14- 11) + 2 = 5

CAMMINI:

1. 0-2-4-5-6-7-8
2. 0-1
3. 0-2-4-5-10
4. 0-2-4-5-6-7-9
5. 0-2-3

Control Flow Graph



Metodo effettuaSostituzione()

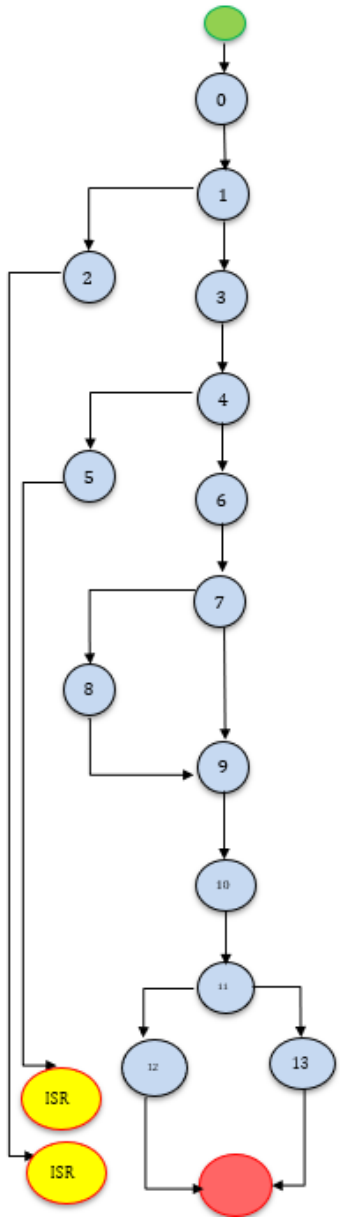
```java
public void effettuaSostituzione(int codice_prenotazione) throws DAOException, ControllerException {

        Sostituzione sos = new Sostituzione();
        sos.setCodicePrenotazione(codice_prenotazione);
        sos.caricaSostituzione();
        if(codice_prenotazione <= 0) {
          throw new ControllerException(ExitCodes.CODICEPRENOTAZIONE_NOTVALID);
        }
        System.out.println(sos);
        if(sos.getStato().equalsIgnoreCase("sostituito")) {
            throw new ControllerException(ExitCodes.CONTROLLER_SOSTITUZIONE_ALRDONE);
        }

        sos.aggiornaStatoSostituzione();
        Prenotazione p = new Prenotazione();
        p.setCodiceP(codice_prenotazione);
        p.caricaPrenotazione();
        double minutes = (sos.getTimestamp() - p.getTimestamp()) / 60.0f;

        Batteria b = new Batteria();
        b.setCodice(p.getCodice_batteria());
        b.caricaBatteriaByCodice();

        if(minutes > 60) { //applico una penale
            applicaPenale(b, p);
        }

        //salvo la nuova batteria nel db;
        Batteria batt = new Batteria();
        int cicli = new Random().nextInt(5);
        batt.setCicli_ricariche(cicli);
        batt.setCosto(10 * cicli);
        batt.setStato("USATO");
        System.out.println("codice autovettura:" + b.getCodice_autovettura());
        batt.setCodice_autovettura(b.getCodice_autovettura());
        if(cicli > 0) batt.salvaBatteria();
        else System.out.println("batteria non riutilizzabile!");

    }
```

NUMERO CICLOMATICO: 5
numero di regioni chiuse del grafo +1 = 5
numero di nodi predicati (4) +1 = 5
archi – # nodi + 2 = (17- 14) + 2 = 5

CAMMINI:

6. 0-1-2
7. 0-1-3-4-5
8. 0-1-3-4-6-7-9-10-11-12
9. 0-1-3-4-6-7-8-9-10-11-12
10. 0-1-3-4-6-7-9-10-11-13

Control Flow Graph



7.2 Test funzionale

EFFETTUA_REGISTRAZIONE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (*FAIL*, *PASS*)
1	Tutti input validi	Nome valido Cognome valido Nascita valida Credito valido		{Nome: "Giovanni", Cognome: "Cesarano", Nascita: "04-04-1999", Credito: 1000}	Registrazione effettuata con successo.	Viene aggiunto un cliente nel database con un rispettivo codice_badge.	Registrazione effettuata con successo.	Viene aggiunto un cliente nel database con un rispettivo codice_badge.	PASS
2	Nome stringa nulla	Nome stringa nulla[ERROR] Cognome valido Nascita valida Credito valido		{Nome: " ", Cognome: "Fernandez" Data: "25/12/2021", Credito: 500}	Nome non valido.		Nome non valido.		PASS
3	Nome stringa > 15 caratteri	Nome stringa > 15 caratteri[ERROR] Cognome valido		{Nome: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	Nome non valido.		Nome non valido.		PASS

		Nascita valida Credito valido		aaaaaaaaaaaaaa aaaaaaaaaaa”, Cognome: “Fernandez” Data: “25/12/2021”, Credito: 500}					
4	Nome stringa con simboli	Nome stringa con simboli [ERROR], Cognome valido Nascita valida Credito valido		{Nome: “§ç£”, Cognome: “Coraggio” Nascita: “25/12/2021”, Credito: 500}	Nome non valido.		Nome non valido.		PASS
5	Cognome stringa nulla	Nome valido Cognome stringa nulla[ERROR] Nascita valida Credito valido		{Nome: “Fabio” Cognome: “ ”, Data: “25/12/2021”, Credito: 500}	Cognome non valido.		Cognome non valido.		PASS
6	Cognome stringa > 20 caratteri	Nome valido Cognome stringa > 20 caratteri[ERROR]		{Nome: “Fabio” Cognome: “bbbbbbbbbbbbb bbbbbbbbbbbbb bbbbbbbbbbbbb bbbbbbbbbbbbb	Cognome non valido.		Cognome non valido.		PASS

		Nascita valida Credito valido		bbbbbbbbbbbbb”, Data: “25/12/2021”, Credito: 500}					
7	Cognome stringa con simboli	Nome valido Cognome stringa con simboli [ERROR], Nascita valida Credito valido		{Nome: “Valeria” Cognome: “§ç£”, Nascita: “25/12/2021”, Credito: 500}	Cognome non valido.		Cognome non valido.		PASS
8	Data di nascita con formato non valido	Nome valido, Cognome valido formato nascita non valido [ERROR], Credito valido		{Nome: “Eva”, Cognome: “Rossi” Data: “225/1332/2021”, Credito: 500}	Formato data non valido.		Formato data non valido.		PASS
9	Credito < 0	Nome valido, Cognome valido Nascita valida		{Nome: “Giovanni”, Cognome: “Cesarano”, Nascita: “04-04-1999”, Credito: -100}	Credito non valido.		Credito non valido.		PASS

		Credito < 0 non valido [ERROR]							
10	Credito non numerico	Nome valido, Cognome valido Nascita valida Credito non numerico non valido [ERROR]		{Nome: “Giovanni”, Cognome: “Cesarano”, Nascita: “04-04-1999”, Credito: “sjdfw”}	Credito non valido.		Credito non valido.		PASS

PRENOTAZIONE_BATTERIA

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (*FAIL*, *PASS*)
1	Tutti input validi	codice_badge valido codice_autovettura valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = 5043 }	Prenotazione della batteria effettuata con successo.	La prenotazione è aggiunta al database	Prenotazione della batteria effettuata con successo.	La prenotazione è aggiunta al database	PASS
2	Tutti input validi, errore per la pre-	codice_badge valido codice_autovettura valido	codice_badge non è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = 5043 }	Badge non trovato.		Badge non trovato		PASS

	condizione badge								
3	Tutti input validi, errore per la pre-condizione credito	codice_badge valido codice_autovettura valido	codice_badge è presente nel database, il credito non è sufficiente	{codice_badge = 35, codice_autovettura = 5043 }	Credito non sufficiente.		Credito non sufficiente.		PASS
4	codice_badge <=0	codice_badge <=0 [ERROR] codice_autovettura valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = -35, codice_autovettura = 5043 }	Badge non valido.		Badge non valido.		PASS
5	codice_badge non numerico	codice_badge non int [ERROR] codice_autovettura valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = “sdsge”, codice_autovettura = 5043 }	Badge non valido.		Badge non valido.		PASS
6	codice_autovettura < =0	codice_autovettura <=0 [ERROR] codice_badge valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = -5043 }	Codice autovettura non valido.		Codice autovettura non valido.		PASS
7	codice_autovettura non numerico	codice_autovettura non int [ERROR] codice_badge valido	codice_badge è presente nel database, il credito è sufficiente	{codice_badge = 35, codice_autovettura = “sges” }	Codice autovettura non valido.		Codice autovettura non valido.		PASS

EFFETTUA_SOSTITUZIONE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Output Ottenuti	Post-condizioni Ottenute	Esito (*FAIL*, *PASS*)
1	Tutti input validi, precondizioni valide	codice_batteria valido codice_badge valido timestamp valido	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: 1234, codice_badge: 123, timestamp: 1663071611132 }	Sostituzione effettuata con successo.	La sostituzione è aggiunta al database	Sostituzione effettuata con successo.	La sostituzione è aggiunta al database	PASS
2	Input validi, errore di pre-condizione, batteria non è stata prenotata	codice_batteria valido codice_badge valido timestamp valido	La batteria non è stata prenotata La batteria non è stata sostituita	{codice_batteria: 1234, codice_badge: 123, timestamp: 1663071611132 }	La batteria deve essere stata prenotata per poterla sostituire		La batteria deve essere stata prenotata per poterla sostituire		PASS
3	Input valido, errore di pre-condizione, la batteria è già stata sostuita	codice_batteria valido codice_badge valido timestamp valido	La batteria è stata prenotata È stata già eseguita una sostituzione con codice_batteria	{codice_batteria: 1234, codice_badge: 123, timestamp: 1663071611132 }	La batteria può essere ssotituita una sola volta		La batteria può essere ssotituita una sola volta		PASS
4	codice_batteria<= 0	codice_batteria <= 0 [ERROR] codice_badge valido	La batteria è stata prenotata	{codice_batteria: -1234, codice_badge: 123,	Codice batteria non valido		Codice batteria non valido		PASS

		timestamp valido	La batteria non è stata già sostituita	timestamp: 1663071611132 }					
5	codice_batteria non numerico	codice_batteria non numerico [ERROR] codice_badge valido timestamp valido	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: “wow”, codice_badge: 123, timestamp: 1663071611132 }	Codice batteria non valido.		Codice batteria non valido.		PASS
6	codice_badge<=0	codice_batteria valido codice_badge<=0[ERROR] timestamp valido	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: 1234, codice_badge: -123, timestamp: 1663071611132 }	Codice badge non valido.		Codice badge non valido.		PASS
7	codice_badge non numerico	codice_batteria valido codice_badge non numerico [ERROR] timestamp valido	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: 1234, codice_badge: “wow”, timestamp: 1663071611132 }	Codice batteria non valido.		Codice batteria non valido.		PASS
8	timestamp non valido	codice_batteria valido codice_badge valido timestamp non valido[ERROR]	La batteria è stata prenotata La batteria non è stata già sostituita	{codice_batteria: 1234, codice_badge: 123, timestamp: “wow”}	Timestamp non valido		Timestamp non valido		PASS