

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
CORSO DI LAUREA IN INGEGNERIA INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE
PROF. R. PIETRANTUONO- A.A. 2022 – 23

Progetto

Negozi di videogiochi

Studente: Francesco Pio Fontana N46005532

francescop.fontana@studenti.unina.it

Versione 1 del 9/07/2023

Indice

| | |
|---|------------------------------|
| 1. SPECIFICHE INFORMALI..... | 1 |
| 2. ANALISI E SPECIFICA DEI REQUISITI | 2 |
| 2.1 ANALISI NOMI-VERBI | ERROR! BOOKMARK NOT DEFINED. |
| 2.2 REVISIONE DEI REQUISITI | ERROR! BOOKMARK NOT DEFINED. |
| 2.3 GLOSSARIO DEI TERMINI..... | 4 |
| 2.4 CLASSIFICAZIONE DEI REQUISITI..... | 5 |
| 2.4.1 REQUISITI FUNZIONALI | 5 |
| 2.4.2 REQUISITI SUI DATI | 6 |
| 2.4.3 VINCOLI / ALTRI REQUISITI | 7 |
| 2.5 MODELLAZIONE DEI CASI D'USO | 8 |
| 2.5.1 ATTORI E CASI D'USO | 8 |
| 2.5.2 DIAGRAMMA DEI CASI D'USO | 2 |
| 2.5.3 SCENARI | 3 |
| 2.6 DIAGRAMMA DELLE CLASSI | 4 |
| 2.7 DIAGRAMMI DI SEQUENZA | 5 |
| 3. STIMA DEI COSTI | 7 |
| 4. PIANO DI TEST FUNZIONALE..... | 12 |
| 5. PROGETTAZIONE | 18 |
| 5.1 DIAGRAMMA DELLE CLASSI | 18 |
| 5.2 DIAGRAMMI DI SEQUENZA | 18 |
| 6.TESTING | 21 |
| 6.1 TEST STRUTTURALE | ERROR! BOOKMARK NOT DEFINED. |
| 6.1.1 COMPLESSITÀ CICLOMATICA..... | ERROR! BOOKMARK NOT DEFINED. |
| 6.2 TEST FUNZIONALE..... | 21 |

1. Specifiche informali

Si vuole realizzare un sistema informatico per la gestione di un negozio per la vendita di videogiochi. I giochi sono disponibili in formato digitale e in formato fisico. Essi sono caratterizzati da un codice identificativo, un titolo, la console di riferimento, un codice PEGI ed il prezzo. I giochi in formato fisico sono caratterizzati anche dalla quantità disponibile in magazzino. Il proprietario del negozio può accedere al sistema utilizzando le proprie credenziali (username e password). Una volta acceduto può aggiungere al sistema dei dipendenti, specificandone il nome, il cognome e il codice fiscale. Il proprietario può anche emettere sconti destinati ai clienti del sistema: ogni sconto è caratterizzato da un codice e da una percentuale, che rappresenta lo sconto da apportare sul totale della spesa, applicabile una singola volta. Ogni volta che uno sconto viene emesso, il cliente destinatario riceve una e-mail con il codice sconto. Ogni dipendente può registrarsi al sistema inserendo il proprio codice fiscale, un indirizzo e-mail e configurando una password. I clienti del sistema possono registrarsi specificando nome, cognome, indirizzo, e-mail, username, password e numero della carta di credito. Una volta acceduto al sistema, un utente può visualizzare una lista dei titoli disponibili, specificando la console (PS4, XboxOne, PC, ...). Per procedere con l'acquisto il cliente deve inserire una lista di giochi con le relative quantità e specificare per ogni copia se acquistare la copia fisica o quella digitale. Al termine della procedura d'acquisto il cliente può applicare un codice sconto, dopodiché il sistema detrae il totale dalla carta di credito del cliente. Se il gioco viene acquistato in formato fisico il sistema invia una notifica ad un dipendente per effettuare la spedizione e decrementa la quantità disponibile in magazzino. Se il gioco viene acquistato in formato digitale il sistema genera un codice univoco della copia, che il cliente può utilizzare per il download direttamente dalla console. Ogni primo giorno del mese il sistema genera un report di tutti gli acquisti effettuati nel mese precedente. Il proprietario può visualizzare gli N clienti che hanno effettuato più acquisti nell'ultimo mese per l'emissione degli sconti.

2. Analisi e specifica dei requisiti

2.1 Analisi nomi – verbi

Si vuole realizzare un sistema informatico per la gestione di un negozio per la vendita di videogiochi. I giochi sono disponibili in formato digitale e in formato fisico. Essi sono caratterizzati da un codice identificativo, un titolo, la console di riferimento, un codice PEGI ed il prezzo. I giochi in formato fisico sono caratterizzati anche dalla quantità disponibile in magazzino. Il proprietario del negozio può accedere al sistema utilizzando le proprie credenziali (username e password). Una volta acceduto può aggiungere al sistema dei dipendenti, specificandone il nome, il cognome e il codice fiscale. Il proprietario può anche emettere sconti destinati ai clienti del sistema: ogni sconto è caratterizzato da un codice e da una percentuale, che rappresenta lo sconto da apportare sul totale della spesa, applicabile una singola volta. Ogni volta che uno sconto viene emesso, il cliente destinatario riceve una e-mail con il codice sconto. Ogni dipendente può registrarsi al sistema inserendo il proprio codice fiscale, un indirizzo e-mail e configurando una password. I clienti del sistema possono registrarsi specificando nome, cognome, indirizzo, e-mail, username, password e numero della carta di credito. Una volta acceduto al sistema, un utente può visualizzare una lista dei titoli disponibili, specificando la console (PS4, XboxOne, PC, ...). Per procedere con l'acquisto il cliente deve inserire una lista di giochi con le relative quantità e specificare per ogni copia se acquistare la copia fisica o quella digitale. Al termine della procedura d'acquisto il cliente può applicare un codice sconto, dopodiché il sistema detrae il totale dalla carta di credito del cliente. Se il gioco viene acquistato un dipendente in formato fisico il sistema invia una notifica ad per effettuare la spedizione e decrementa la quantità disponibile in magazzino. Se il gioco viene acquistato in formato digitale il sistema genera un codice univoco della copia, che il cliente può utilizzare per il download direttamente dalla console. Ogni primo giorno del mese il sistema genera un report di tutti gli acquisti effettuati nel mese precedente. Il proprietario può visualizzare gli N clienti che hanno effettuato più acquisti nell'ultimo mese per l'emissione degli sconti.

LEGENDA:

FUNZIONALITA'

ATTORE

ATTRIBUTO

CLASSE

CLASSE-ATTORE

Revisione dei requisiti

1. Il sistema deve offrire al Proprietario del negozio una funzionalità di accesso al sistema;
2. Il sistema deve offrire al Proprietario del negozio una funzionalità di registrazione dei dipendenti
3. L'aggiunta dei dipendenti consiste nell'inserire: nome, cognome e codice fiscale;
4. Il sistema deve offrire al Proprietario del negozio una funzionalità di emissione di sconti per clienti scelti;
5. Ciascuno sconto deve avere un codice identificativo e una percentuale;
6. Il sistema deve offrire al Proprietario del negozio una funzionalità per generare codici sconto per i propri clienti;
7. Il sistema deve inviare un'e-mail al Cliente con i dati relativi al codice sconto;
8. Il sistema deve offrire ai Dipendenti una funzionalità di registrazione nel sistema;
9. La registrazione di un Dipendente consiste nell'inserire il proprio: codice fiscale, indirizzo e-mail e password;
10. La registrazione di un Cliente consiste nell'inserire: nome, cognome, indirizzo e-mail, username, password e numero di carta di credito;
11. Il sistema deve offrire al Cliente una funzionalità per la visualizzazione della lista dei giochi disponibili specificandone la console di appartenenza;
12. Per effettuare un acquisto il Cliente deve specificare il numero di copie di ogni gioco scelto e il formato fisico o digitale per tali copie;
13. Il sistema deve offrire al Cliente una funzionalità per l'applicazione del codice sconto;
14. Il sistema deve detrarre automaticamente il totale della spesa dalla carta di credito del Cliente;
15. Il sistema deve generare automaticamente una notifica al Dipendente se in un ordine sono compresi copie di giochi in formato fisico;
16. Il sistema deve decrementare automaticamente le quantità dei giochi disponibili nel magazzino se in un ordine sono compresi copie in formato fisico;
17. Il sistema deve generare automaticamente un codice univoco per la ogni copia di gioco acquistato in formato digitale;
18. Il sistema deve generare automaticamente un report di tutti gli acquisti effettuati nel mese precedente;
19. Il sistema deve offrire al Proprietario del negozio una funzionalità per la visualizzazione degli N clienti che hanno effettuato più acquisti nell'ultimo mese.
20. Il cliente può applicare un unico codice sconto;
21. Il sistema deve garantire la sicurezza delle informazioni personali dei dipendenti e dei clienti;
22. Il sistema deve essere disponibile 24 su 24, 7 giorni su 7;
23. Il sistema deve essere in grado di gestire un grande numero di acquisti fatti in contemporanea;
24. Il sistema deve essere in grado di gestire acquisti contenenti giochi di diverso tipo e formati di diverso tipo.

Glossario dei termini

| Termine | Descrizione | Sinonimi |
|-----------------------|--|--------------------------------------|
| Negozi di videogiochi | Si riferisce all'attività commerciale che vende videogiochi in formato digitale e fisico. | |
| Gioco | Attività divertenti e intrattenitrice che coinvolge una o più persone | |
| Dipendente | Persona che lavora nel negozio di videogiochi | Impiegato |
| Cliente | Utente che può effettuare una registrazione | |
| Cliente registrato | Utente che può effettuare acquisti all'interno del sistema | |
| Registrazione | Processo mediante il quale vengono inseriti i dati relativi ai Dipendenti o ai Clienti all'interno del sistema per la loro corretta archiviazione. | Iscrizione |
| Autenticazione | Processo mediante il quale un utente accede al sistema inserendo le proprie credenziali di accesso (username e password). | Login |
| Proprietario | Persona responsabile del negozio di videogiochi | Titolare |
| Sistema | Software utilizzato per gestire tutte le funzionalità riguardo il negozio di videogiochi | Struttura informatica |
| Acquisto | Processo mediante il quale un cliente seleziona i giochi desiderati | |
| Sconto | Riduzione di prezzo applicabile ai Clienti per incentivare un acquisto | Offerta Promozione |
| E-mail | Sistema di comunicazione digitale per lo scambio di messaggi digitali attraverso l'internet | Indirizzo di posta elettronica |
| Visualizzazione | Processo mediante il quale il Proprietario o il Cliente accede al sistema per ottenere informazioni rispettivamente sui giochi disponibili o sui clienti con il maggior numero di acquisti | Consultazione |
| Report | Documento che fornisce informazioni di riepilogo su tutti gli acquisti del mese precedente, inviato mensilmente al proprietario del negozio | Resoconto |
| Console | Piattaforma su cui il gioco può essere eseguito | Sistema di gioco |
| Notifica | Avviso inviato per informare i dipendenti sugli acquisti di giochi in formato fisico | Comunicazione Avvertimento |
| Lista dei giochi | Elenco contenente l'insieme dei giochi disponibili | Catalogo |
| Database | Collezione organizzata per gestire, archiviare e interrogati tutti i dati relativi al negozio di videogiochi | Base di dati |

2.4 Classificazione dei requisiti

Classificare i requisiti (funzionali, sui dati, vincoli/altri requisiti), riportando le frasi dei requisiti (revisionati) che li esprimono (ciascuna frase deve esprimere un singolo requisito), numerandoli secondo una codifica (es.: RF01, RF02, RFn per i requisiti funzionali; RD01, RD02, RDm per i requisiti sui dati). Verificare che i requisiti siano chiari, completi e verificabili.

2.4.1 Requisiti funzionali

| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
|------|--|--|
| RF01 | Il sistema deve offrire al Proprietario del negozio una funzionalità di accesso al sistema | 1 |
| RF02 | Il sistema deve offrire al Proprietario del negozio una funzionalità di registrazione dei dipendenti | 2 |
| RF03 | Il sistema deve offrire al Proprietario del negozio una funzionalità di emissione di sconti per clienti scelti | 4 |
| RF04 | Il sistema deve offrire al Proprietario del negozio una funzionalità per generare codici sconto per i propri clienti | 6 |
| RF05 | Il sistema deve inviare un'e-mail al Cliente con i dati relativi al codice sconto | 7 |
| RF06 | Il sistema deve offrire ai Dipendenti una funzionalità di registrazione nel sistema | 8 |
| RF07 | Il sistema deve offrire al Cliente una funzionalità per la visualizzazione della lista dei giochi disponibili specificandone la console di appartenenza | 11 |
| RF8 | Il sistema deve offrire al Cliente una funzionalità per l'applicazione del codice sconto | 13 |
| RF9 | Il sistema deve detrarre automaticamente il totale della spesa dalla carta di credito del Cliente | 14 |
| RF10 | Il sistema deve generare automaticamente una notifica al Dipendente se in un ordine sono compresi copie di giochi in formato fisico | 15 |
| RF11 | Il sistema deve decrementare automaticamente le quantità dei giochi disponibili nel magazzino se in un ordine sono compresi copie in formato fisico | 16 |
| RF12 | Il sistema deve generare automaticamente un codice univoco per la ogni copia di gioco acquistato in formato digitale | 17 |
| RF13 | Il sistema deve generare automaticamente un report di tutti gli acquisti effettuati nel mese precedente | 18 |
| RF14 | Il sistema deve offrire al Proprietario del negozio una funzionalità per la visualizzazione degli N clienti che hanno effettuato più acquisti nell'ultimo mese | 19 |

2.4.2 Requisiti sui dati

| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
|------|--|--|
| RD01 | L'aggiunta di un dipendente consiste nell'inserire: nome, cognome e codice fiscale | 3 |
| RD02 | Ciascuno sconto deve avere un codice identificativo e una percentuale | 5 |
| RD03 | La registrazione di un Dipendente consiste nell'inserire il proprio: codice fiscale, indirizzo e-mail e password | 9 |
| RD04 | La registrazione di un Cliente consiste nell'inserire: nome, cognome, indirizzo e-mail, username, password e numero di carta di credito | 10 |
| RD05 | Per effettuare un acquisto il Cliente deve specificare il numero di copie di ogni gioco scelto e il formato fisico o digitale per tali copie | 12 |

2.4.3 Vincoli / Altri requisiti

Descrivere eventuali vincoli o requisiti non funzionali.

Per es., stimare il volume dei dati e delle transazioni (Quanti utenti si stimano? Quante operazioni si prevedono al giorno per ogni utente? Qual è la frequenza relativa dei vari tipi di operazioni (es: interrogazioni 100/h, modifiche 5/g, ...))

Es.:

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

RNF02: Per l'invio mensile della newsletter, deve essere disponibile un server di posta elettronica esterno al sistema.

RNF02: Il sistema deve gestire decine di accessi contemporanei, prevalentemente alla sera, per gli ordini in consegna al mattino successivo.

| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
|------|---|--|
| Vo1 | Il cliente può applicare un unico codice sconto | 20 |
| RDo1 | Il sistema deve garantire la sicurezza delle informazioni personali dei dipendenti e dei clienti | 21 |
| RDo2 | Il sistema deve essere disponibile 24 su 24, 7 giorni su 7 | 22 |
| RDo3 | Il sistema deve essere in grado di gestire un grande numero di acquisti fatti in contemporanea | 23 |
| RDo4 | Il sistema deve essere in grado di gestire acquisti contenenti giochi di diverso tipo e formati di diverso tipo | 24 |

Modellazione dei casi d'uso

Attori e casi d'uso

Prima elencare attori primari, attori secondari, casi d'uso, ed eventuali casi d'uso di inclusione e di estensione. Poi riportare in tabella i casi d'uso indicando per ciascuno gli attori e i relativi casi d'uso inclusi o di estensione.

Attori Primari:

- Proprietario
- Cliente
- Dipendente
- Tempo

Attori Secondari:

- Servizio mail

Casi d'uso:

- UC1: Registrazione Dipendente
- UC2: Registrazione Cliente
- UC3: Emissione Sconti
- UC4: Inserimento Dipendenti
- UC5: Visualizzazione titoli disponibili
- UC6: Acquisto di giochi
- UC7: Acquisto di giochi fisici
- UC8: Acquisto di giochi digitali
- UC9: Gestione Spedizione
- UC10: Generazione report mensili
- UC11: Visualizzazione clienti migliori

Casi d'uso di inclusione:

- UC12: Notifica di spedizione
- UC13: Notifica sconto
- UC14: Generazione codice univoco

Casi d'uso di estensione:

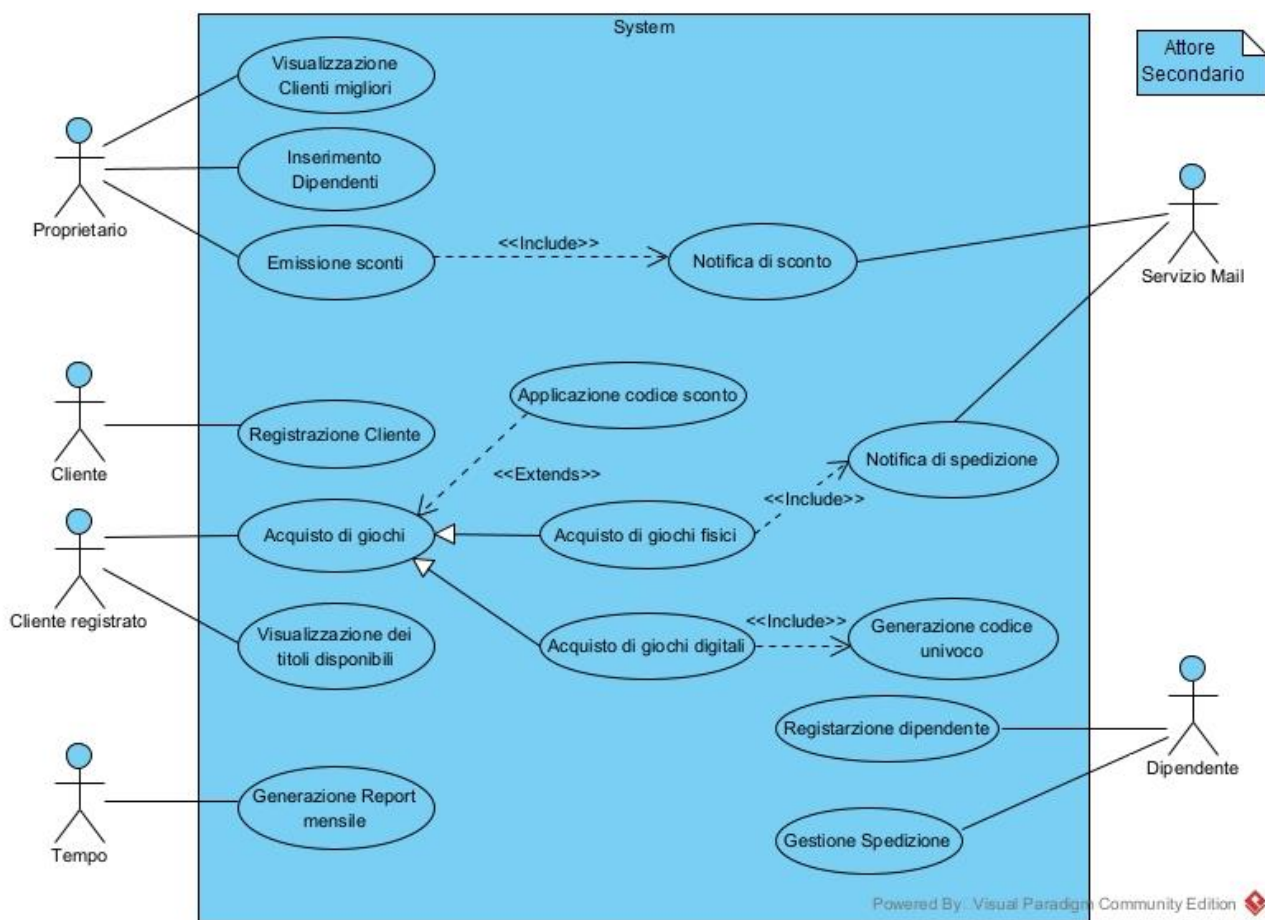
- UC15: Applicazione codice sconto

| Caso d'uso | Attori Primari | Attori Secondari | Incl. / Ext. |
|---|----------------|------------------|---|
| UC1: Registrazione Dipendente | Dipendente | - | - |
| UC2: Registrazione Cliente | Cliente | - | - |
| UC3: Emissione Sconti | Proprietario | Servizio Mail | Include Notifica di sconto |
| UC4: Inserimento Dipendenti | Proprietario | - | - |
| UC5: Visualizzazione titoli disponibili | Cliente | - | - |
| UC6: Acquisto di giochi | Cliente | - | Generalizzazione di Acquisto di giochi fisici e Acquisto di giochi digitali |
| UC7: Acquisto di giochi fisici | Cliente | Servizio Mail | Include Notifica di spedizione |

| | | | |
|--|--------------|---------------|--|
| UC8: Acquisto di giochi digitali | Cliente | - | Include Generazione codice univoco |
| UC9: Gestione Spedizione | Dipendente | - | - |
| UC9: Generazione report mensili | Tempo | - | - |
| UC11: Visualizzazione clienti migliori | Proprietario | - | - |
| UC12: Notifica di spedizione | - | Servizio Mail | Incluso in Acquisto di giochi fisici |
| UC13: Notifica di sconto | - | Servizio Mail | Incluso in Emissione sconti |
| UC14: Generazione codice univoco | - | - | Incluso in Acquisto di giochi digitali |
| UC15: Applicazione codice sconto | Cliente | - | Estensione di Acquisto di giochi |

Diagramma dei casi d'uso

Riportare il diagramma dei casi d'uso.



Scenari

Selezionare un caso d'uso (dunque una funzionalità) per ogni membro del gruppo, da sviluppare fino alla codifica in Java (dunque, diagramma di sequenza di analisi raffinato, diagramma di sequenza di progettazione, implementazione e test del caso d'uso scelto). Riportare in questa sezione lo scenario principale per il caso d'uso scelto, come nel seguente esempio

| Caso d'uso: | ApplicazioneCodiceSconto |
|--------------------------------|---|
| Attore primario | Cliente |
| Attore secondario | - |
| Descrizione | Il cliente può decidere se applicare o meno lo sconto ricevuto |
| Pre-Condizioni | Il cliente ha acquistato almeno un gioco Il cliente deve essere registrato al sistema Lo sconto è applicabile una sola volta |
| Sequenza di eventi principale | <ol style="list-style-type: none"> 1. Il caso d'uso inizia ogni volta che il cliente conferma l'acquisto di uno o più giochi <<Extends>> AcquistoDiGiochi 2. Il sistema controlla che il cliente appartenga all'anagrafica dei clienti registrati attraverso l'email inserita 3. Il sistema prevede l'eventuale inserimento in una casella di testo di un codice sconto 4. Il sistema controlla che il codice sconto sia stato effettivamente emesso 5. Il sistema calcola la percentuale riferita dal codice sconto inserito rispetto al totale 6. Il sistema sottrae dal totale il risultato dell'operazione precedente |
| Post-Condizioni | Il sistema detrae dalla carta di credito del cliente il totale con sconto applicato |
| Casi d'uso correlati | <i>AcquistoDiGiochi</i> |
| Sequenza di eventi alternativi | <ol style="list-style-type: none"> 2.1 Se l'email non è stato trovata: <ul style="list-style-type: none"> • Il sistema procederà con l'acquisto calcolando il prezzo intero 4.1 Se il codice sconto non è valido: <ul style="list-style-type: none"> • Il sistema restituisce al cliente un ERRORE che indica che il codice inserito è errato o non più valido • Il sistema esorta il cliente ad inserire un codice che sia corretto |

| Caso d'uso: | InserimentoDipendenti |
|-------------------------------|--|
| Attore primario | Proprietario |
| Attore secondario | - |
| Descrizione | Il proprietario può decidere di aggiungere n dipendenti al negozio |
| Pre-Condizioni | Il proprietario deve essere acceduto al sistema |
| Sequenza di eventi principale | <ol style="list-style-type: none"> 1. Il caso d'uso inizia ogni volta che il proprietario desidera aggiungere nuovi dipendenti 2. Il proprietario seleziona la quantità n di dipendenti da aggiungere 3. Il sistema predispone l'inserimento dei dati dei nuovi dipendenti <ol style="list-style-type: none"> 3.1 Il proprietario inserisce per ciascuno nome, cognome e codice fiscale 4. Il sistema salva i dati inseriti all'interno di un database che contiene tutti i dipendenti del negozio |
| Post-Condizioni | Il sistema aggiunge nuovi dipendenti nel database dei dipendenti |
| Casi d'uso correlati | <i>nessuno</i> |

Sequenza di eventi alternativi

- 2.1 Se la quantità n di dipendenti da aggiungere è ≤ 0
 - Il sistema richiede l'inserimento di una quantità $n > 0$
- 3.2 Se il codice fiscale inserito è già presente nell'elenco dei dipendenti:
 - Il sistema notifica al proprietario che il codice fiscale inserito esiste già nel sistema

Diagramma delle classi

Riportare il diagramma delle classi di prima analisi ed il diagramma di analisi "raffinato" secondo il modello BCE, ossia comprensivo di classi "Control" e "Boundary".

Diagramma delle classi di prima analisi.

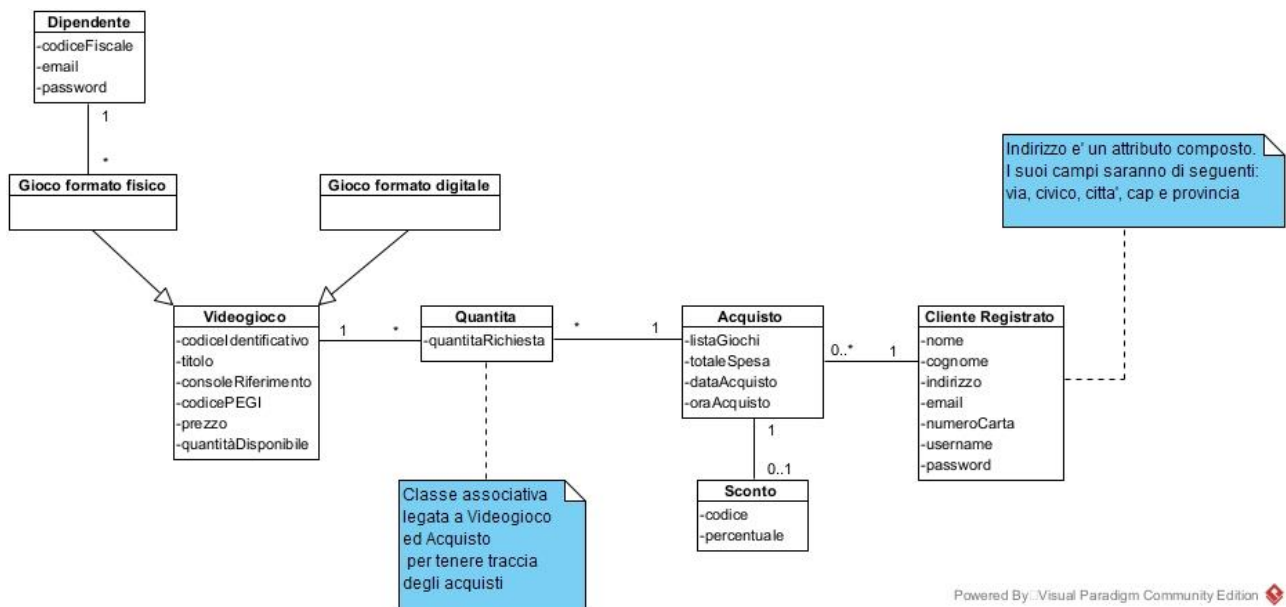
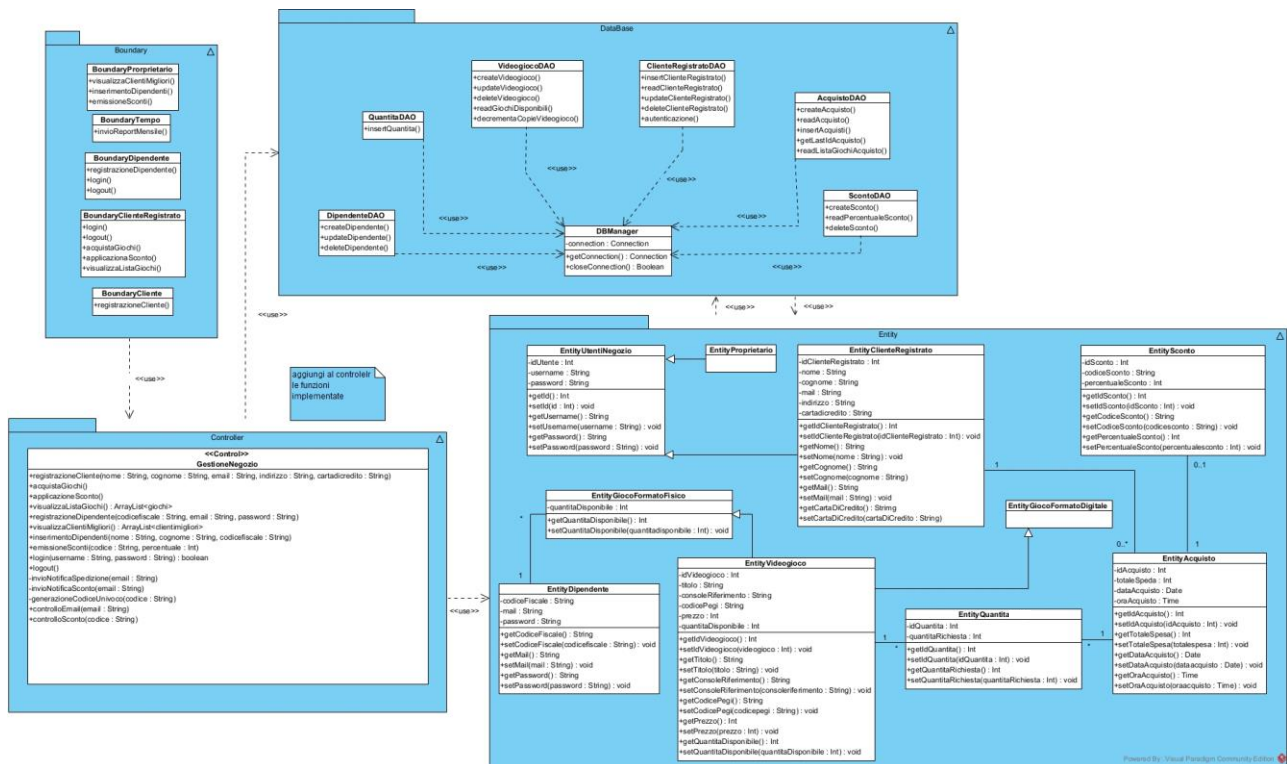


Diagramma delle classi raffinato (con classi Control e Boundary).



Diagrammi di sequenza

Riportare il diagramma di sequenza di analisi raffinato (ossia con “Control” e “Boundary”) per le funzionalità (ossia i casi d’uso) da implementare (una per ogni membro del gruppo) e che saranno sviluppate fino alla codifica in Java ed al test.

Diagramma di sequenza per il caso d’uso “Applicazione sconto”

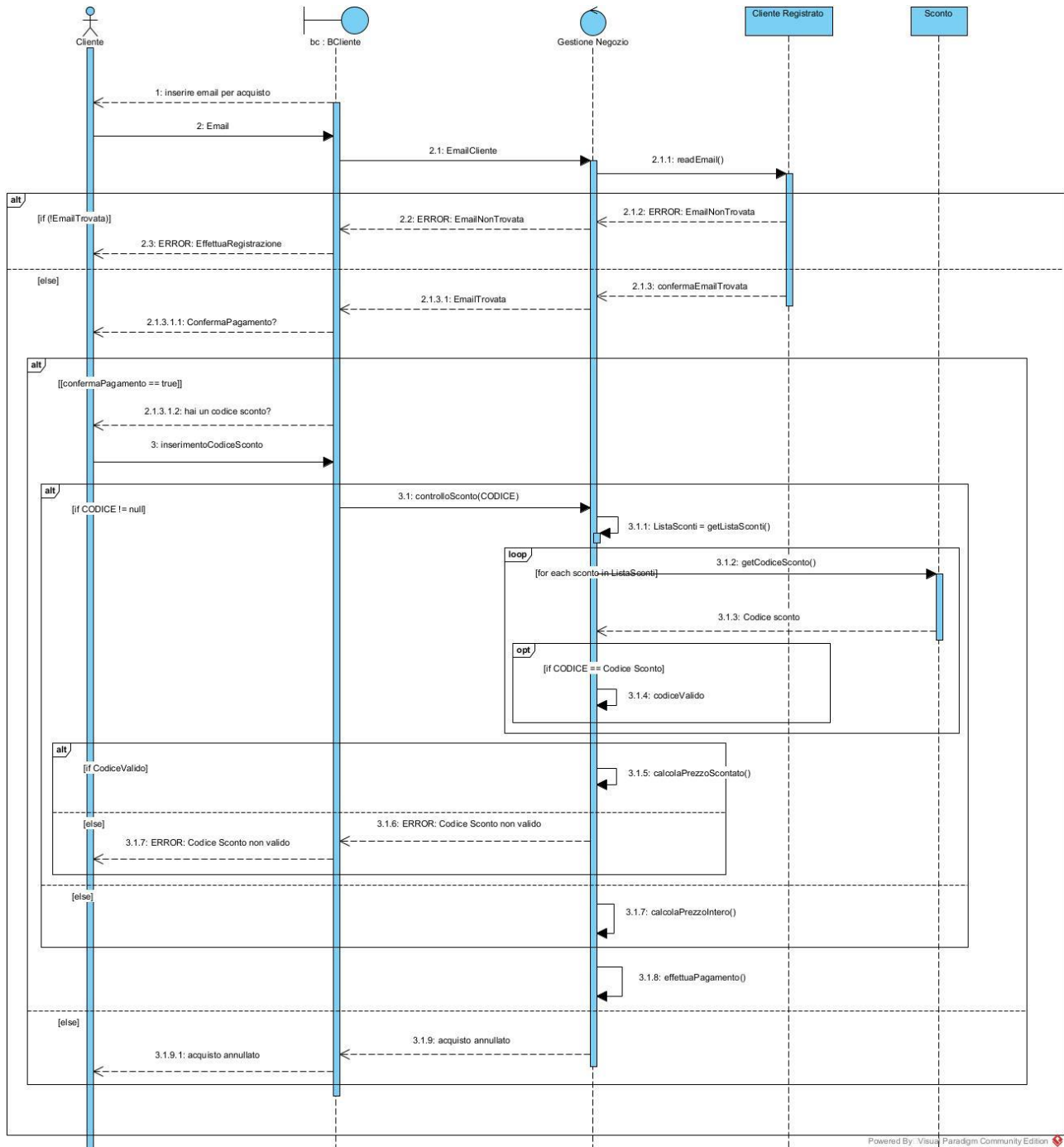
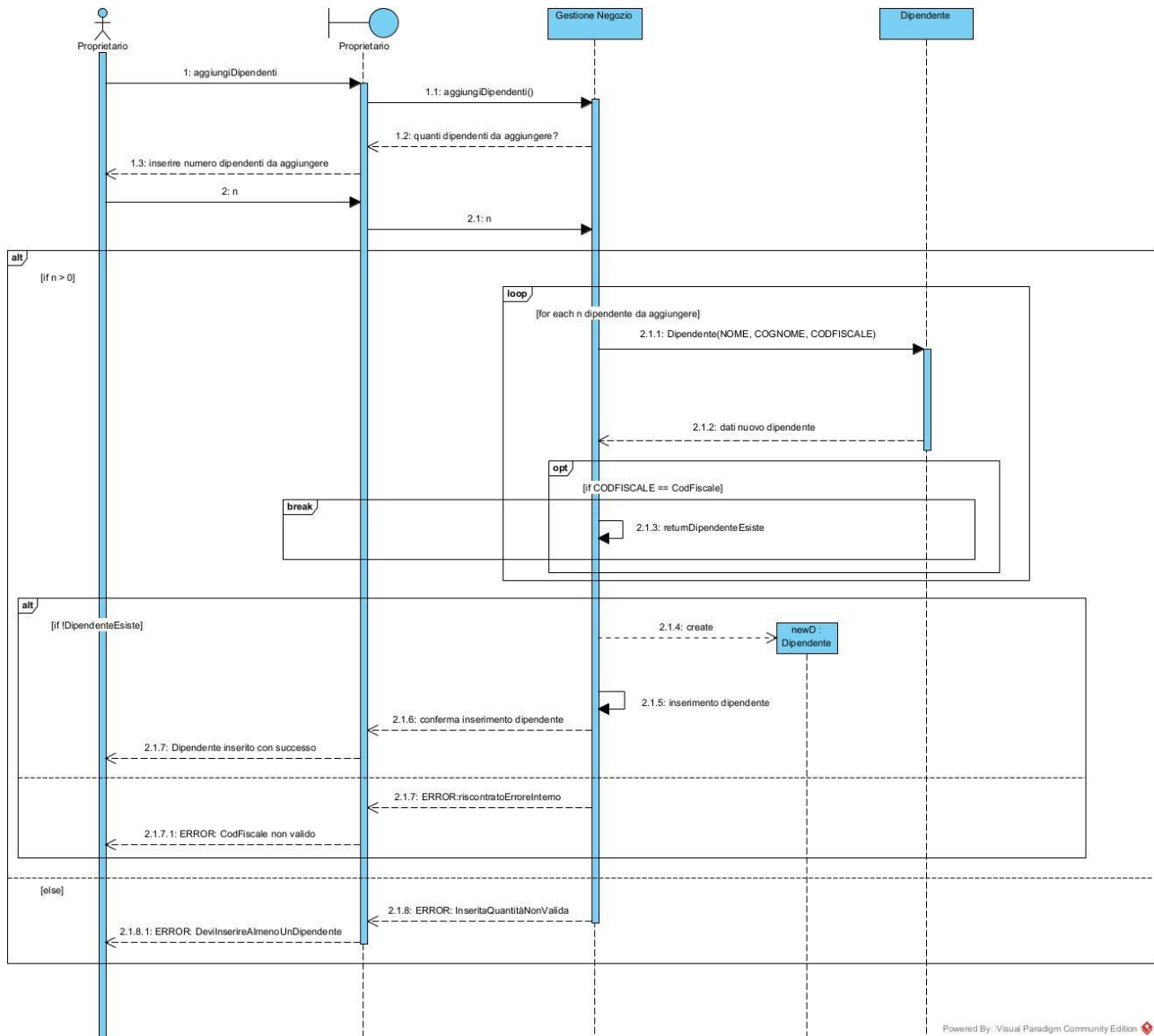


Diagramma di sequenza per il caso d'uso "Inserimento dipendenti"



3. Stima dei costi

[OPZIONALE] Riportare la stima dei costi secondo il metodo dei Punti Funzione.

Sulla base delle informazioni relative al software da realizzare possiamo identificare i seguenti tipi di funzionalità:

➤ **Funzionalità**

Il proprietario può effettuare operazioni di:

1. Aggiunta dei dipendenti
2. Emissione degli sconti

Il cliente può effettuare l'operazione di registrazione.

Il cliente registrato può effettuare operazioni di:

1. Autenticazione
2. Visualizzazione titoli disponibili
3. Acquisto giochi

Il dipendente può effettuare l'operazione di registrazione.

Altre operazioni:

1. Generazione report acquisti
2. Generazione lista clienti migliori

➤ **Identificazione delle funzioni dati**

L'ambito di conteggio è sviluppo di progetto, in quanto siamo in fase di analisi.

Non sono presenti external interface file (EIF) poiché il sistema non coopera con altri sistemi esterni.

È presente il database contenente tutte le informazioni relative ai clienti, ai dipendenti, ai giochi ed agli sconti, quindi, considerando come Internal Logical File ogni entità dal punto di vista utente avremo NILF = 4.

Il rating della complessità degli ILF è basato sul numero di elementi di dati (DET) e tipi di record (RET), dunque:

DET:

- Anagrafica cliente registrato: id, nome, cognome, telefono, e-mail, via, numero civico, città, provincia, username, password, carta di credito (DET = 12).
- Anagrafica dipendete: id, codice fiscale, e-mail e password. (DET=4).
- Anagrafico gioco: id, titolo, console di riferimento, codice univoco, codice PEGI, prezzo, formato e quantità disponibile (DET = 8).
- Anagrafico sconto: codice sconto, percentuale sconto, descrizione sconto (DET = 3).

RET:

- Cliente (RET = 1)

- Dipendente (RET = 1)
- Gioco (RET = 1)
- Sconto (RET = 1)

➤ **Tabella di riferimento per le complessità di dati e transazioni**

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

-APPLICAZIONE SCONTO

| | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
|-------------|--------|----------|-------|-----------|-----|
| NILF | 1 | 3 | - | - | 3 |
| NEIF | 0 | - | - | - | 0 |
| NEI | 1 | 3 | - | - | 3 |
| NEO | 1 | 7 | - | - | 7 |
| NEQ | 1 | 5 | - | - | 5 |

NILF (number of internal logical files): per l'inserimento di un codice sconto consideriamo l'interazione tra l'entità "Cliente", che gestisce l'inserimento del codice, con l'entità "Sconto", che contiene informazioni su tutti i codici sconto emessi dal proprietario all'interno del sistema, lo identifichiamo come ILF [1 low].

NEI (number of external inputs): per il numero di input esterni che il sistema riceve ed elabora consideriamo il codice inserito dal cliente. Quindi abbiamo NEI = 1 con complessità [low].

NEO (number of external outputs): per il numero di output esterni prodotto dal sistema consideriamo solo l'emissione di una conferma circa la validità del codice sconto inserito, quindi NEO = 1 con complessità [low].

NEQ (number of external inquiries): per il numero di richieste esterne consideriamo solo una verifica della validità del codice sconto inserito, quindi NEQ = 1 con complessità [low].

UFP = 18 FP = AFP*UFP = 16,74 ≈ 17
 JAVA = 901

-INSERIMENTO DIPENDENTI

| | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
|------|--------|----------|-------|-----------|-----|
| NILF | 1 | 3 | - | - | 3 |
| NEIF | 0 | - | - | - | 0 |
| NEI | 3 | 3 | - | - | 9 |
| NEO | 1 | 7 | - | - | 7 |
| NEQ | 1 | 5 | - | - | 5 |

NILF (number of internal logical files): per l'inserimento di un dipendente consideriamo l'interazione tra l'entità "Proprietario", che gestisce l'inserimento dei dati del nuovo dipendente, con l'entità "Dipendente", che contiene informazioni su tutti i dipendenti registrati all'interno del sistema, lo identifichiamo come ILF [1 low].

NEI (number of external inputs): consideriamo l'inserimento del nome, del cognome e del codice fiscale del dipendente da aggiungere al sistema lo identifichiamo come NEI = 3 con complessità [low].

NEO (number of external outputs): per il numero di output esterni prodotto dal sistema consideriamo solo la conferma di aggiunta del nuovo dipendente, quindi NEO = 1 con complessità [low].

NEQ (number of external inquiries): per il numero di richieste esterne consideriamo solo una verifica del codice fiscale per assicurarci che non appartenga già ad un altro dipendente precedentemente creato, quindi NEQ = 1 con complessità [low].

$$UFP = 24 \quad FP = AFP * UFP = 22,32 \approx 22$$

$$JAVA = 1166 \text{ (LOC x FP in JAVA = 53)}$$

$$AFP = 0.93$$

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

FATTORI CORRETTIVI

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

4. Piano di test funzionale

Progettare i casi di test funzionale con la tecnica del *Category Partition Testing* per le funzionalità scelte.

[OPZIONALE] Descrivere il procedimento di calcolo.

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

| Codice | Elemento di sistema Database |
|---|---|
| <ul style="list-style-type: none">• Stringa di caratteri di lunghezza ≤ 255• Stringa di caratteri di lunghezza > 255 [ERROR]• Stringa che contiene caratteri speciali [ERROR] | <ul style="list-style-type: none">• SCONTO inserito = sconto emesso• SCONTO inserito \neq sconto emesso [ERROR] |

Il numero di test da effettuarsi senza particolari vincoli è $3 * 2 = 6$

Introduciamo i vincoli [ERROR]

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (2 per codice, 1 per ElementoDiSistemaDatabase)

Il numero di test risultante è $(1 * 1) + 3 = 4$

TEST SUITE

| Test Case ID | Descrizione | Classi di equivalenza coperte | Pre-condizioni | Input | Output Attesi | Post-condizioni Attese |
|--------------|---------------------------------------|--|---|---------------------------------|---|---|
| 1 | Tutti input validi | Codice valido ElementoDiSistemaDatabase valido | Il cliente è registrato al sistema e ha selezionato la funzionalità applica sconto. Il sistema chiede al cliente di inserire un codice sconto | {Codice: "HOLIDAY10"} | Il sistema acquisisce la percentuale relativa allo sconto e la sottrae dal totale di acquisto | Il sistema detrae dalla carta di credito del cliente il totale con sconto applicato |
| 2 | Codice stringa > 255 caratteri | Codice stringa > 255 caratteri [ERROR] ElementoDiSistemaDatabase valido | Il cliente è registrato al sistema e ha selezionato la funzionalità applica sconto. Il sistema chiede al cliente di inserire un codice sconto | {Codice: "HOLIDAY1000000....."} | Codice sconto troppo lungo! | Il sistema chiede nuovamente di inserire un codice sconto valido |
| 3 | Codice stringa con caratteri speciali | Codice stringa con caratteri speciali [ERROR] ElementoDiSistemaDatabase valido | Il cliente è registrato al sistema e ha selezionato la funzionalità applica sconto. Il sistema chiede al cliente di inserire un codice sconto | {Codice: "§ç£10"} | Codice contiene caratteri speciali! | Il sistema chiede nuovamente di inserire un codice sconto valido |
| 4 | SCONTO inserito non emesso | Codice valido ElementoDiSistemaDatabase SCONTO inserito != sconto emesso [ERROR] | Il cliente ha inserito un codice sconto. Il sistema richiede al cliente l'inserimento di un codice sconto | {Codice: "HOLIDAY10"} | Il codice sconto non può essere usato! | Il sistema chiede nuovamente di inserire un codice sconto valido |

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

| Nome | Cognome | CodFiscale | Elemento di sistema Database |
|--|--|---|--|
| <ul style="list-style-type: none"> Stringa di caratteri di lunghezza ≤ 255 Stringa di caratteri di lunghezza >255 [ERROR] Stringa che contiene simboli che non sono caratteri [ERROR] | <ul style="list-style-type: none"> Stringa di caratteri di lunghezza ≤ 255 Stringa di caratteri di lunghezza >255 [ERROR] Stringa che contiene simboli che non sono caratteri [ERROR] | <ul style="list-style-type: none"> Stringa di caratteri di lunghezza = 16 Stringa di caratteri di lunghezza $\neq 16$ [ERROR] Stringa che contiene simboli che non sono caratteri [ERROR] | <ul style="list-style-type: none"> Codice fiscale inserito \neq codice già presente Codice fiscale inserito = codice già presente [ERROR] |

Il numero di test da effettuarsi senza particolari vincoli è: $3 * 3 * 3 * 2 = 54$

Introduciamo i vincoli [ERROR]

Il numero dei test da eseguire per testare singolarmente i vincoli è 7 (2 per Nome, 2 per Cognome, 2 per CodFiscale, 1 per ElementoDiSistemaDatabase)

Il numero di test risultante è $(1 * 1 * 1 * 1) + 7 = 8$

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, CodFiscale valido ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. E' possibile inserire un dipendente con nome e cognome inseriti. Il CodFiscale viene aggiunto al database. | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | Nuovo dipendente aggiunto | I dati del nuovo dipendente vengono aggiunti al database |
| 2 | Nome stringa > 255 caratteri | Nome stringa > 255 caratteri [ERROR], Cognome valido, CodFiscale valido, ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema. Il sistema richiede l'inserimento del nome. | {Nome: "aaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaa ...", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | Nome dipendente troppo lungo! | Il sistema chiede nuovamente di inserire un nome |
| 3 | Nome stringa con simboli | Nome stringa con simboli [ERROR], Cognome valido, CodFiscale valido, ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema. Il sistema richiede l'inserimento del nome. | {Nome: "Şcogn£", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | Nome contiene simboli! | Il sistema chiede nuovamente di inserire un nome |
| 4 | Cognome stringa > 255 caratteri | Nome valido, Cognome stringa > 255 caratteri [ERROR], CodFiscale valido, ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome. Il sistema richiede | {Nome: "Carlo", Cognome: "aaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaa ...", | Cognome dipendente troppo lungo! | Il sistema chiede nuovamente di inserire un cognome |

| | | | | | | |
|---|-------------------------------------|---|--|--|--|--|
| | | | l'inserimento del cognome. | CodFiscale: "SCGCRL01P23L359Q"} | | |
| 5 | Cognome stringa con simboli | Nome valido, Cognome stringa con simboli [ERROR], CodFiscale valido ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome. Il sistema richiede l'inserimento del cognome. | {Nome: "Carlo", Cognome: "Şcogn£miglio", CodFiscale: "SCGCRL01P23L359Q"} | Cognome contiene simboli! | Il sistema chiede nuovamente di inserire un cognome |
| 6 | CodFiscal e stringa != 16 caratteri | Nome valido, Cognome valido, CodFiscale stringa != 16 caratteri [ERROR] ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome e cognome. Il sistema richiede l'inserimento del codFiscale. | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01"} | CodFiscale troppo lungo o troppo corto | Il sistema chiede nuovamente di inserire un codFiscale |
| 7 | CodFiscal e stringa con simboli | Nome valido, Cognome valido, CodFiscale stringa con simboli [ERROR] ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome e cognome. Il sistema richiede l'inserimento del codFiscale. | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "ŞçGCRL01£23L359Q"} | CodFiscale contiene simboli! | Il sistema chiede nuovamente di inserire un codFiscale |

| | | | | | | |
|---|--------------------------------------|--|--|--|--|--|
| 8 | CodFiscal e già presente nel sistema | Nome valido, Cognome valido, CodFiscale valido ElementoDiSistemaDatabase Codice fiscale inserito = codice già presente [ERROR] | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome e cognome. Il sistema richiede l'inserimento del codFiscale. | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | CodFiscale inserito già esistente. Riprovare.. | Il sistema chiede nuovamente di inserire un codFiscale |
|---|--------------------------------------|--|--|--|--|--|

5. Progettazione

Diagramma delle classi

Riportare il diagramma delle classi di progettazione. Reificare eventuali classi associative del diagramma delle classi di analisi. Specificare argomenti e tipo di ritorno delle operazioni (per quelle più significative, coinvolte nei casi d'uso sviluppati fino alla implementazione). Includere classi del dominio della soluzione, come strutture dati e classi DAO. Raggruppare le classi in package secondo il paradigma BCED.

Diagrammi di sequenza

Riportare il diagramma di sequenza di progetto per le funzionalità (ossia i casi d'uso) scelte.

Diagramma di sequenza DAO per caso d'uso "Applicazione sconto"

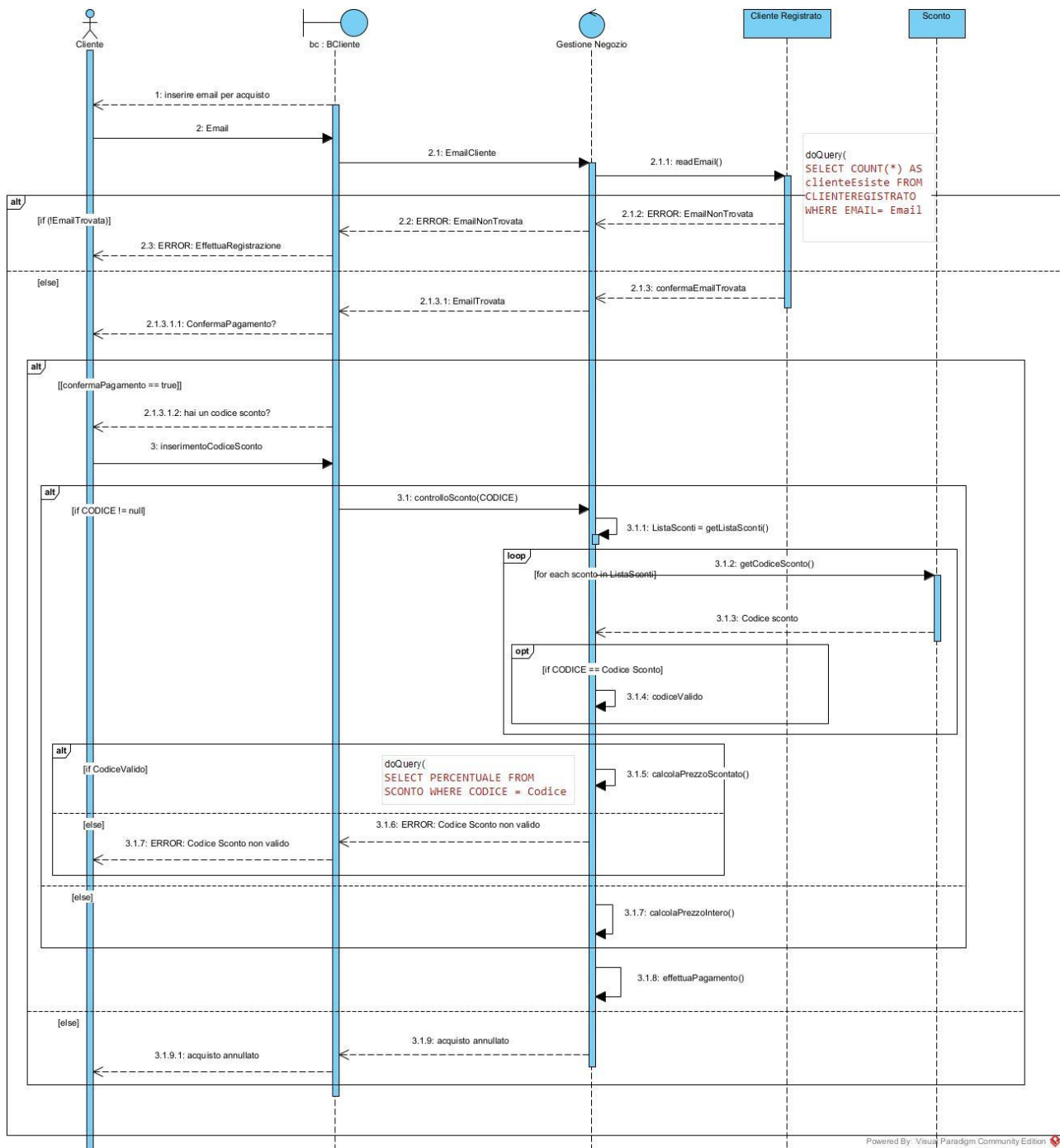
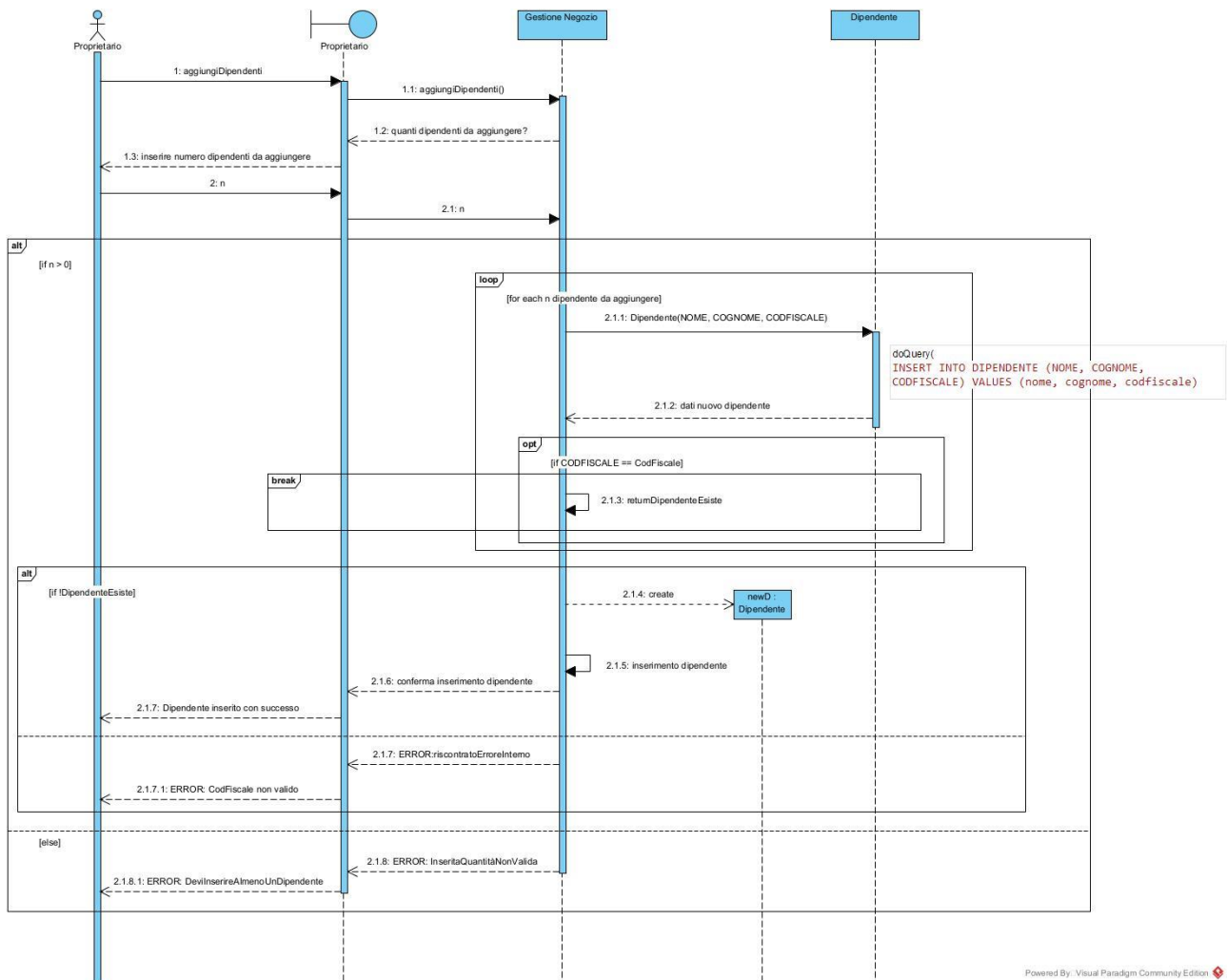


Diagramma di sequenza DAO per caso d'uso "Inserimento dipendente"



6. Testing

Test funzionale

Descrivere i risultati dell'esecuzione dei test funzionali precedentemente pianificati adoperando lo schema di tabella seguente.

Descrivere le eventuali azioni di *debugging* a seguito di casi di test con esito *FAIL*.

Commentare se gli eventuali difetti rilevati dal test funzionale potevano essere rilevati anche da un test strutturale.

Test per il caso d'uso "Applicazione sconto".

| Test Case ID | Descrizione | Classi di equivalenza coperte | Pre-condizioni | Input | Output Attesi | Post-condizioni Attese | Output ottenuti | Post condizioni ottenute | Esito |
|--------------|---------------------------------------|---|---|---------------------------------|---|---|---|---|-------|
| 1 | Tutti input validi | Codice valido ElementoDiSistemaDatabas e valido | Il cliente è registrato al sistema e ha selezionato la funzionalità applica sconto. Il sistema chiede al cliente di inserire un codice sconto | {Codice: "HOLIDAY10"} | Il sistema acquisisce la percentuale relativa allo sconto e la sottrae dal totale di acquisto | Il sistema detrae dalla carta di credito del cliente il totale con sconto applicato | Il sistema acquisisce la percentuale relativa allo sconto e la sottrae dal totale di acquisto | Il sistema detrae dalla carta di credito del cliente il totale con sconto applicato | PASS |
| 2 | Codice stringa > 255 caratteri | Codice stringa > 255 caratteri [ERROR] ElementoDiSistemaDatabas e valido | Il cliente è registrato al sistema e ha selezionato la funzionalità applica sconto. Il sistema chiede al cliente di inserire un codice sconto | {Codice: "HOLIDAY1000000....."} | Codice sconto troppo lungo! | Il sistema chiede nuovamente di inserire un codice sconto valido | Codice sconto troppo lungo! | Il sistema chiede nuovamente di inserire un codice sconto valido | PASS |
| 3 | Codice stringa con caratteri speciali | Codice stringa con caratteri speciali [ERROR] ElementoDiSistemaDatabas e valido | Il cliente è registrato al sistema e ha selezionato la funzionalità applica sconto. Il sistema chiede al cliente di inserire un codice sconto | {Codice: "§ç£10"} | Codice contiene caratteri speciali! | Il sistema chiede nuovamente di inserire un codice sconto valido | Codice contiene caratteri speciali! | Il sistema chiede nuovamente di inserire un codice sconto valido | PASS |

| | | | | | | | | | |
|---|-----------------------------------|---|---|-----------------------|--|--|--|--|------|
| 4 | Codice SCONTO inserito non emesso | Codice valido ElementoDiSistemaDatabase e Codice SCONTO inserito != codice emesso [ERROR] | Il cliente ha inserito un codice sconto. Il sistema richiede al cliente l'inserimento di un codice sconto | {Codice: "HOLIDAY10"} | Il codice sconto non può essere usato! | Il sistema chiede nuovamente di inserire un codice sconto valido | Il codice sconto non può essere usato! | Il sistema chiede nuovamente di inserire un codice sconto valido | PASS |
|---|-----------------------------------|---|---|-----------------------|--|--|--|--|------|

Test per il caso d'uso "Inserimento dipendenti".

| Test Case ID | Descrizione | Classi di equivalenza coperte | Pre-condizioni | Input | Output Attesi | Post-condizioni Attese | Output ottenuti | Post-condizioni ottenute | Esito |
|--------------|-------------|-------------------------------|----------------|-------|---------------|------------------------|-----------------|--------------------------|-------|
|--------------|-------------|-------------------------------|----------------|-------|---------------|------------------------|-----------------|--------------------------|-------|

| | | | | | | | | | |
|---|---------------------------------|---|---|---|----------------------------------|--|----------------------------------|--|------|
| 1 | Tutti input validi | Nome valido, Cognome valido, CodFiscale valido ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. E' possibile inserire un dipendente con nome e cognome inseriti. Il CodFiscale viene aggiunto al database. | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | Nuovo dipendente aggiunto | I dati del nuovo dipendente vengono aggiunti al database | Nuovo dipendente aggiunto | I dati del nuovo dipendente vengono aggiunti al database | PASS |
| 2 | Nome stringa > 255 caratteri | Nome stringa > 255 caratteri [ERROR], Cognome valido, CodFiscale valido, ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema. Il sistema richiede l'inserimento del nome. | {Nome: "aaaaaaaaaaaaaaaaaaaaaaaaaaa...", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | Nome dipendente troppo lungo! | Il sistema chiede nuovamente di inserire un nome | Nome dipendente troppo lungo! | Il sistema chiede nuovamente di inserire un nome | PASS |
| 3 | Nome stringa con simboli | Nome stringa con simboli [ERROR], Cognome valido, CodFiscale valido, ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema. Il sistema richiede l'inserimento del nome. | {Nome: "Şcogn£", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | Nome contiene simboli! | Il sistema chiede nuovamente di inserire un nome | Nome contiene simboli! | Il sistema chiede nuovamente di inserire un nome | PASS |
| 4 | Cognome stringa > 255 caratteri | Nome valido, Cognome stringa > 255 caratteri [ERROR], | Il database è inizializzato correttamente. Il proprietario è | {Nome: "Carlo", Cognome: "aaaaaaaaaaaaaaaaaaaaa | Cognome dipendente troppo lungo! | Il sistema chiede nuovamente di | Cognome dipendente troppo lungo! | Il sistema chiede nuovamente | PASS |

| | | | | | | | | | |
|---|------------------------------------|---|--|--|--|--|--|--|------|
| | | CodFiscale valido ElementoDiSistemaDatabase valido | registrato al sistema e ha inserito nome. Il sistema richiede l'inserimento del cognome. | aaaaaaaaaaaaaaaaaaa aaa...", CodFiscale: "SCGCRL01P23L359Q"} | | inserire un cognome | | di inserire un cognome | |
| 5 | Cognome stringa con simboli | Nome valido, Cognome stringa con simboli [ERROR], CodFiscale valido ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome. Il sistema richiede l'inserimento del cognome. | {Nome: "Carlo", Cognome: "Şcogn£miglio", CodFiscale: "SCGCRL01P23L359Q"} | Cognome contiene simboli! | Il sistema chiede nuovamente di inserire un cognome | Cognome contiene simboli! | Il sistema chiede nuovamente di inserire un cognome | PASS |
| 6 | CodFiscale stringa != 16 caratteri | Nome valido, Cognome valido, CodFiscale stringa != 16 caratteri [ERROR] ElementoDiSistemaDatabase valido | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome e cognome. Il sistema richiede l'inserimento del codFiscale. | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01"} | CodFiscale troppo lungo o troppo corto | Il sistema chiede nuovamente di inserire un codFiscale | CodFiscale troppo lungo o troppo corto | Il sistema chiede nuovamente di inserire un codFiscale | PASS |
| 7 | CodFiscale stringa con simboli | Nome valido, Cognome valido, CodFiscale stringa con simboli [ERROR] | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "ŞçGCRL01£23L359Q"} | CodFiscale contiene simboli! | Il sistema chiede nuovamente di inserire un codFiscale | CodFiscale contiene simboli! | Il sistema chiede nuovamente di inserire un codFiscale | PASS |

| | | | | | | | | | |
|---|-------------------------------------|---|--|--|--|--|---------------------------------------|--|------|
| | | ElementoDiSistemaDatabase valido | inserito nome e cognome. Il sistema richiede l'inserimento del codFiscale. | | | | | | |
| 8 | CodFiscale già presente nel sistema | Nome valido, Cognome valido, CodFiscale già presente nel sistema [ERROR] ElementoDiSistemaDatabase Codice fiscale = codice già presente [ERROR] | Il database è inizializzato correttamente. Il proprietario è registrato al sistema e ha inserito nome e cognome. Il sistema richiede l'inserimento del codFiscale. | {Nome: "Carlo", Cognome: "Scognamiglio", CodFiscale: "SCGCRL01P23L359Q"} | CodFiscale inserito è già esistente. Riprovare.. | Il sistema chiede nuovamente di inserire un codFiscale | inserito è già esistente. Riprovare.. | Il sistema chiede nuovamente di inserire un codFiscale | PASS |