

RELAZIONE PROGETTO

PROGRAMMAZIONE e MODELLAZIONE AD OGGETTI



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

CORSO DI LAUREA IN
INFORMATICA APPLICATA
SCUOLA DI
SCIENZE TECNOLOGIE E FILOSOFIA DELL'INFORMAZIONE

MATRICOLA:
Nardini Gianluca
291147

Sessione autunnale 2020/2021

SPECIFICA DEL PROBLEMA

Il progetto consiste in un software grafico per la gestione delle postepay dei diversi clienti. Il programma presenta diverse schermate che permettono di:

- Inserire il cliente che ha una nuova postepay
- Modificare alcune informazioni riguardanti il cliente
- Eliminare il cliente che possiede una postepay
- Visualizzare una pagina che contiene tutte le informazioni riguardanti il cliente
- Inserire nuovi saldi della carta nel tempo per ogni cliente

Ogni carta postepay ha le seguenti informazioni:

- Numero della carta
- Importo iniziale della carta

Inoltre sono presenti anche i dati personali di ogni cliente che intende avere una carta postepay.

I dati personali, che vengono salvati in un file JSON, comprendono:

- Nome
- Cognome
- Sesso
- Data di nascita
- Luogo di nascita
- Numero di telefono

STUDIO DEL PROBLEMA

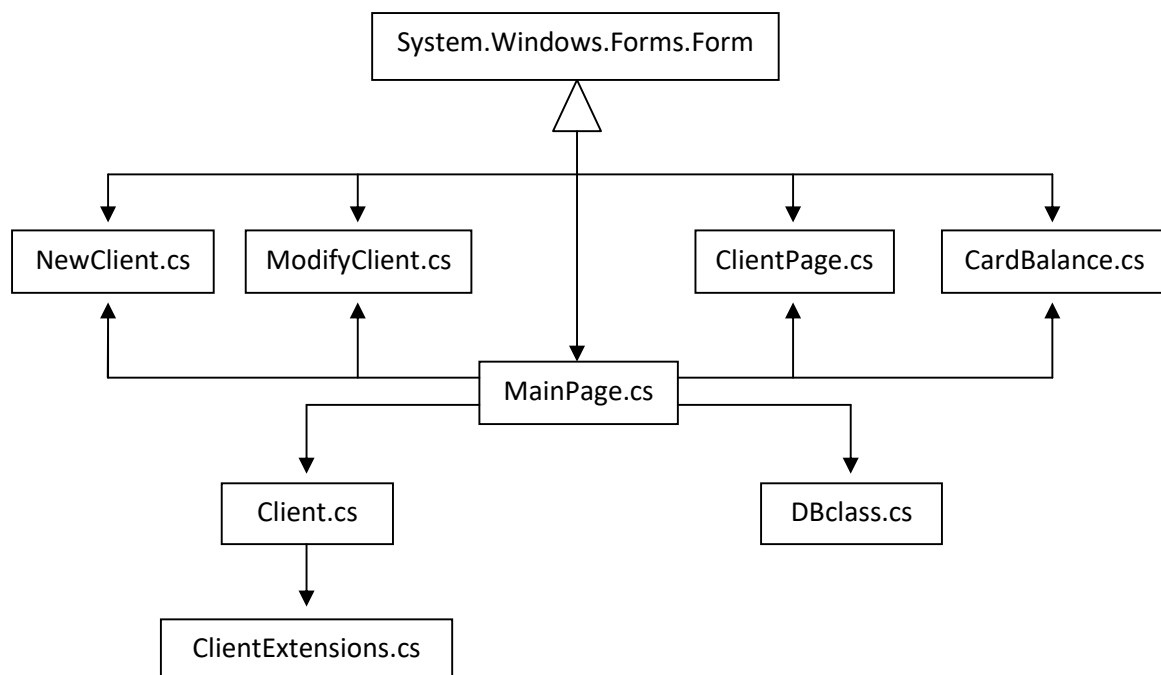
I punti critici del progetto sono i seguenti:

- Aggiungere un nuovo cliente, in questa fase l'utente inserisce i dati personali del cliente, ovvero nome, cognome, sesso, data di nascita, luogo di nascita e numero di telefono, inoltre deve inserire anche i dati relativi alla postepay, quindi numero della carta e importo iniziale della carta.
Una volta che sono stati inseriti tutti i dati il programma li elabora e crea un nuovo oggetto di tipo ListViewItem da inserire nell'apposita ListView.
- Modificare un cliente, l'utente per effettuare questa modifica deve obbligatoriamente cliccare su un elemento della ListViewClients, altrimenti si visualizza un messaggio di errore, successivamente dopo aver selezionato il cliente di cui si vuole effettuare la modifica si può cliccare sull'apposito button.
Una volta aperta la nuova finestra l'utente potrà modificare i campi del cliente e in seguito premere ok, il programma eliminerà il "vecchio" cliente e successivamente aggiungerà quello "nuovo", inserendolo nella stessa posizione della lista.
- Eliminare un cliente, l'utente per poter eliminare un cliente deve innanzitutto selezionarlo, altrimenti si visualizza un messaggio di errore.
Una volta selezionato il cliente si può cliccare sul button apposito, si apre una nuova finestra di avvertimento nella quale si deve confermare l'eliminazione del cliente, se si conferma il cliente viene definitivamente eliminato, altrimenti si torna alla pagina principale e non viene eseguita nessuna operazione.
- Visualizzare tutte le informazioni del cliente, per poter aprire questa finestra si deve necessariamente selezionare un cliente nella lista dei clienti e poi si può cliccare l'apposito button, nella finestra che si aprirà si trovano tutti i dati del cliente, come nome, cognome, sesso, data di nascita, ecc, e i dati della carta, ovvero il numero della carta.
- Inserire un nuovo saldo della carta ad un cliente già inserito in precedenza, per eseguire questa operazione l'utente deve selezionare un cliente.
Una volta selezionato e premuto sul button apposito si aprirà una nuova finestra nella quale si può immettere il nuovo saldo della carta in un certa data (quest'ultima deve essere necessariamente futura a quelle impostate precedentemente), premendo ok il programma aggiungerà questi due nuovi dati nelle rispettive liste, aggiornando inoltre il grafico situato nella pagina principale.
- Salvare e caricare il database dei clienti con tutte le loro rispettive informazioni, per effettuare queste operazioni il software utilizza un file in formato JSON , lo carica all'avvio dell'applicazione e lo salva durante la chiusura dell'applicazione.

SCELTE ARCHITETTURALI

Il programma presenta una classe principale (*MainPage.cs*) dove al suo interno ci sono tutti i collegamenti che si possono effettuare con le altre form, la listview ed il grafico del saldo della carta. Dalla schermata principale si possono aprire tutte le altre form, tra cui: *NewClient.cs*, *ModifyClient.cs*, *ClientPage.cs* e *CardBalance.cs*.

Sono presenti altre classi che sono: *Client.cs*, che contiene tutte le informazioni del cliente; *ClientExtensions.cs*, utilizzata per creare l'oggetto di tipo *ListViewItem*; *DBclass.cs*, utile per gestire il database dei clienti.

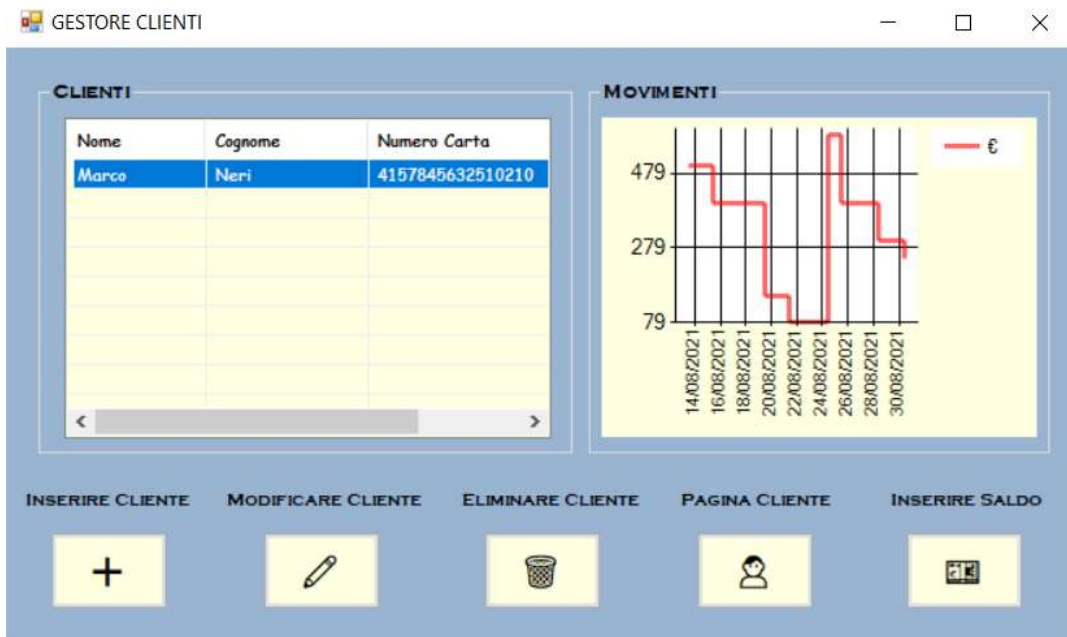


Descrizione di tutte le form

1- MainPage.cs

Le funzioni principali di questa prima form sono:

- Far visualizzare all'utente la lista dei clienti situata sulla sinistra della finestra e il relativo grafico adiacente alla tabella
- Dare la possibilità all'utente di aprire tutte le altre form



Cliccando su un cliente si visualizzano, tramite grafico, tutti i movimenti effettuati.

Inoltre, selezionando un cliente lo si può modificare, eliminare, visualizzare la pagina con tutti i suoi dati personali e infine inserire il saldo della carta in una certa data, in caso non si selezioni nessun cliente prima di cliccare il relativo button comparirà un messaggio d'errore. Invece per quanto riguarda il button dell'inserimento di un nuovo cliente non è necessario selezionare alcun cliente, infatti al click del button si aprirà direttamente la relativa finestra.

Attributi:

- Una lista di tipo *Client*, chiamata *clients*, che contiene tutti i clienti che sono di tipo *Client*, questa lista è inizializzata a *private* perché viene utilizzata solo all'interno di questa classe o viene passata come parametro nelle altre form.
- Un oggetto di tipo *ListViewItem*, chiamato *clientItem*, utile per aggiungere i clienti nella *listViewClients*, visto che ogni elemento di questa lista deve essere di tipo *ListViewItem*.
- Una variabile intera chiamata *index*, utilizzata per gestire l'inserimento dei clienti nella lista nella giusta posizione.
- Un oggetto di tipo *DBclass*, chiamato *dbHandler*, utile per salvare e caricare tutti i dati dei clienti nel file JSON (*DBclass* è una classe che è stata creata).

Metodi:

- *btnNew_Click* è il metodo che viene eseguito nel momento in cui si preme il button "Inserire cliente". In primo luogo si crea e si visualizza a schermo la form di tipo *NewClient* alla quale si passa come parametro la lista dei clienti, successivamente si trasforma il cliente che è stato inserito in un oggetto di tipo *ListViewItem* per inserirlo nella lista.
Viene effettuato un controllo sull'attributo *index*, se l'utente apre la form ma annulla l'inserimento di un nuovo cliente la lista ovviamente rimane invariata e il numero dei clienti sarà uguale all'*index* e quindi non si inserisce nessun elemento.
- *btnModify_Click* è il metodo che viene eseguito nel momento in cui si preme il button "Modificare cliente". Nel caso in cui un cliente è stato selezionato si crea e si visualizza a schermo la form di tipo *ModifyClient* alla quale si passa come parametro il cliente selezionato, nel momento in cui viene modificato il cliente dall'utente, il metodo rimuove il "vecchio" cliente e inserisce quello "nuovo" nella stessa posizione. Compare a schermo un messaggio d'errore nel caso in cui non venga selezionato alcun cliente.
- *btnCancel_Click* è il metodo che viene eseguito nel momento in cui si preme il button "Eliminare cliente". Nel caso in cui un cliente è stato selezionato viene mostrato a video un messaggio di warning, nel quale viene chiesto se l'utente è sicuro di cancellare il cliente in questione, se si da conferma allora il cliente viene rimosso dalla lista e anche il relativo grafico viene eliminato. Alla rimozione del cliente viene decrementata anche la variabile *index*, in modo che all'inserimento di un nuovo cliente, quest'ultimo venga inserito nella posizione corretta.
- *btnClient_Click* è il metodo che viene eseguito nel momento in cui si preme il button "Pagina cliente". Nel caso in cui un cliente è stato selezionato si crea e si visualizza a schermo la form di tipo *ClientPage* alla quale si passa come parametro il cliente selezionato, altrimenti viene visualizzato un messaggio d'errore.
- *btnBalance_Click* è il metodo che viene eseguito nel momento in cui si preme il button "Inserire saldo". Nel caso in cui un cliente è stato selezionato si crea e si visualizza a schermo la form di tipo *CardBalance* alla quale si passa come parametro il cliente selezionato, una volta che è stato aggiunto il saldo della carta e la relativa data, è presente una rimozione e un reinserimento del cliente per aggiornare la listview.
- *listviewClients_ItemActivate* è il metodo che viene attivato quando un elemento della listview viene selezionato. Ogni qualvolta succeda, il metodo elimina il grafico precedente e ne crea uno basato sul relativo cliente scorrendo la lista di date e saldi. Per avere una giusta visione del grafico si trova il valore minimo tra i saldi della carta e poi quest'ultimo viene settato più o meno come il valore minimo dell'asse Y (valore - 1).
- *LoadDB* è il metodo che viene eseguito all'inizio, richiama il metodo del *dbHandler* per avere la lista dei clienti e caricarli sulla listview.
- *FinalSaveData* è il metodo che viene eseguito nel momento in cui viene chiuso il programma, richiama il metodo del *dbHandler* per salvare la lista dei clienti nel file JSON.

2- NewClient.cs

Le funzione principale di questa form è quella di far inserire all'utente tutti i dati personali del nuovo cliente.

NUOVO CLIENTE

NOME COGNOME DATA DI NASCITA

lunedì 16 agosto 2021

SESSO

☐ M ☐ F

LUOGO DI NASCITA

NUMERO DI TELEFONO NUMERO CARTA IMPORTO INIZIALE €

OK ANNULLA

Attributi:

- Una lista di tipo *Client*, chiamata *clients*, questa lista è public perché viene usata in questa classe e in altre.
- Una stringa privata, chiamata *sex*, utilizzata per il sesso del cliente.

Metodi:

- *btnOk_Click* è il metodo che viene eseguito nel momento in cui si preme il button "OK". All'interno di questo metodo viene effettuato un controllo, richiamando un altro metodo *CheckFields*, sui campi, assicurandosi che tutti siano stati riempiti, se ciò è vero viene richiamato il metodo *CheckRB*, successivamente il metodo *AddClient* e infine tramite il metodo *Close* viene chiusa la form. Se qualche campo non è stato riempito verrà stampato a video un messaggio d'errore.
- *btnCancel_Click* è il metodo che viene eseguito nel momento in cui si preme il button "ANNULLA". Questo metodo si occupa solo della chiusura della form.
- *AddClient* è il metodo che serve a creare un nuovo oggetto di tipo *Client*, al cui interno sono presenti le caratteristiche del cliente inserite dall'utente, e aggiungerlo nella lista dei clienti, inserendo inoltre la data in cui è stato inserito nella lista delle date (in modo tale da sapere in che data il cliente ha iniziato ad utilizzare la carta postepay).

- *CheckFields* è il metodo che serve a controllare se tutti i campi sono stati riempiti, se così è restituisce vero, altrimenti falso.
- *CheckRB* è il metodo che serve a controllare che tipo di radio button è stato selezionato, nel caso sia stato selezionato quello con scritto "M" allora viene settata la variabile sex con "M", altrimenti "F".
- *HandleInput* è il metodo per evitare che venga inserito un input sbagliato da parte dell'utente.

3- ModifyClient.cs

La funzione principale di questa form è quella di far modificare all'utente uno o più dati personali del cliente.

The screenshot shows a window titled "MODIFICA CLIENTE" with a standard Windows title bar (minimize, maximize, close buttons). The form itself is light blue and contains the following elements:

- NOME:** Text input field containing "MARCO".
- COGNOME:** Text input field containing "NERI".
- DATA DI NASCITA:** Date picker showing "venerdì 7 aprile 1989".
- SESSO:** Two radio buttons labeled "M" and "F". The "M" button is selected.
- LUOGO DI NASCITA:** Text input field containing "MILANO".
- NUMERO DI TELEFONO:** Text input field containing "3331020452".
- NUMERO CARTA:** Text input field containing "4157845632510210".
- IMPORTO INIZIALE:** Text input field containing "500", followed by a "€" symbol.
- Buttons:** Two yellow buttons at the bottom labeled "OK" and "ANNULLA".

Attributi:

- Un oggetto di tipo *Client*, chiamato *modifiedClient*, è inizializzato public perché viene utilizzato in questa e in altre classi.

Metodi:

- *btnOk_Click* è il metodo che viene eseguito nel momento in cui si preme il button "OK". Inserisce i nuovi valori inseriti dall'utente nell'oggetto di tipo *Client* chiamato *modifiedClient*.
- *btnCancel_Click* è il metodo che viene eseguito nel momento in cui si preme il button "ANNULLA". Questo metodo si occupa solo della chiusura della form.
- *HandleInput* è il metodo per evitare che venga inserito un input sbagliato da parte dell'utente.
- *SetRB* è il metodo utile per settare il button per la scelta del sesso, con questo metodo all'apertura della form è già settato il sesso che aveva il cliente.

4- ClientPage.cs

La funzione principale di questa form è quella di visualizzare a schermo tutto ciò che riguarda il cliente, compreso il numero della postepay.

The screenshot shows a Windows-style window titled 'PAGINA DEL CLIENTE'. The form has a light blue background and contains the following fields and controls:

- NOME:** Text box containing 'MARCO'.
- COGNOME:** Text box containing 'NERI'.
- DATA DI NASCITA:** Date picker showing 'venerdì 7 aprile 1989'.
- SESSO:** Text box containing 'M'.
- LUOGO DI NASCITA:** Text box containing 'MILANO'.
- NUMERO DI TELEFONO:** Text box containing '3331020452'.
- NUMERO CARTA:** Text box containing '4157845632510210'.
- BACK button:** A yellow button with a left arrow and the text 'BACK'.

Attributi:

- Nessun attributo è stato inizializzato in questa classe.

Metodi:

- *btnCancel_Click* è il metodo che viene eseguito nel momento in cui si preme il button. Questo metodo si occupa solo della chiusura della form.

5- CardBalance.cs

La funzione principale di questa form è quella di far inserire all'utente il nuovo saldo della carta con relativa data del cliente selezionato.

The screenshot shows a Windows-style window titled 'SALDO CARTA'. The form has a light blue background and contains the following fields and controls:

- SALDO IN DATA:** A date picker showing 'LUNEDÌ 6 SETTEMBRE 2021'.
- SALDO DELLA CARTA:** A text box for entering the balance, followed by a Euro symbol (€).
- OK button:** A yellow button labeled 'OK'.
- ANNULLA button:** A yellow button labeled 'ANNULLA'.

Attributi:

- Un oggetto di tipo *Client*, chiamato *client*, è inizializzato public perché viene utilizzato in questa e in altre classi.

Metodi:

- *btnOk_Click* è il metodo che viene eseguito nel momento in cui si preme il button "OK". Aggiunge il nuovo saldo della carta e la relativa data, inseriti dall'utente, nelle rispettive liste del cliente e infine chiude la form.
- *btnCancel_Click* è il metodo che viene eseguito nel momento in cui si preme il button "ANNULLA". Questo metodo si occupa solo della chiusura della form.
- *HandleInput* è il metodo per evitare che venga inserito un input sbagliato da parte dell'utente.

6- Client.cs

Questa classe rappresenta l'oggetto di tipo *Client*, in poche parole descrive il modello del cliente.

Attributi:

- Questa classe presenta diversi attributi, uno per ogni caratteristica del cliente (nome, cognome, data di nascita, sesso, luogo di nascita, numero di telefono, numero della carta).
- Una lista di tipo *DateTime*, chiamata *DateList*, che contiene tutte le date utili a sapere i relativi saldi della carta.
- Una lista di tipo *float*, chiamata *EuroList*, che contiene tutti i saldi della carta inseriti dall'utente.

Metodi:

- *LastDate* è il metodo che restituisce l'ultima data della *DateList*.
- *LastEuro* è il metodo che restituisce l'ultimo saldo della carta della *EuroList*.
- *FindMinimumEuro* è il metodo che permette di restituire il saldo più basso della *EuroList*, implementato appositamente per il grafico nella *MainPage*.

Estensione:

- *ClientExtensions.cs* è un'estensione della classe *Client.cs*, trasforma un oggetto di tipo *Client* in un oggetto di tipo *ListViewItem*, in modo tale che il cliente possa essere inserito nella listview.

7- DBclass.cs

Questa classe è utile per gestire il database. Viene utilizzato il pattern singleton in modo tale da avere una sola istanza di quest'oggetto.

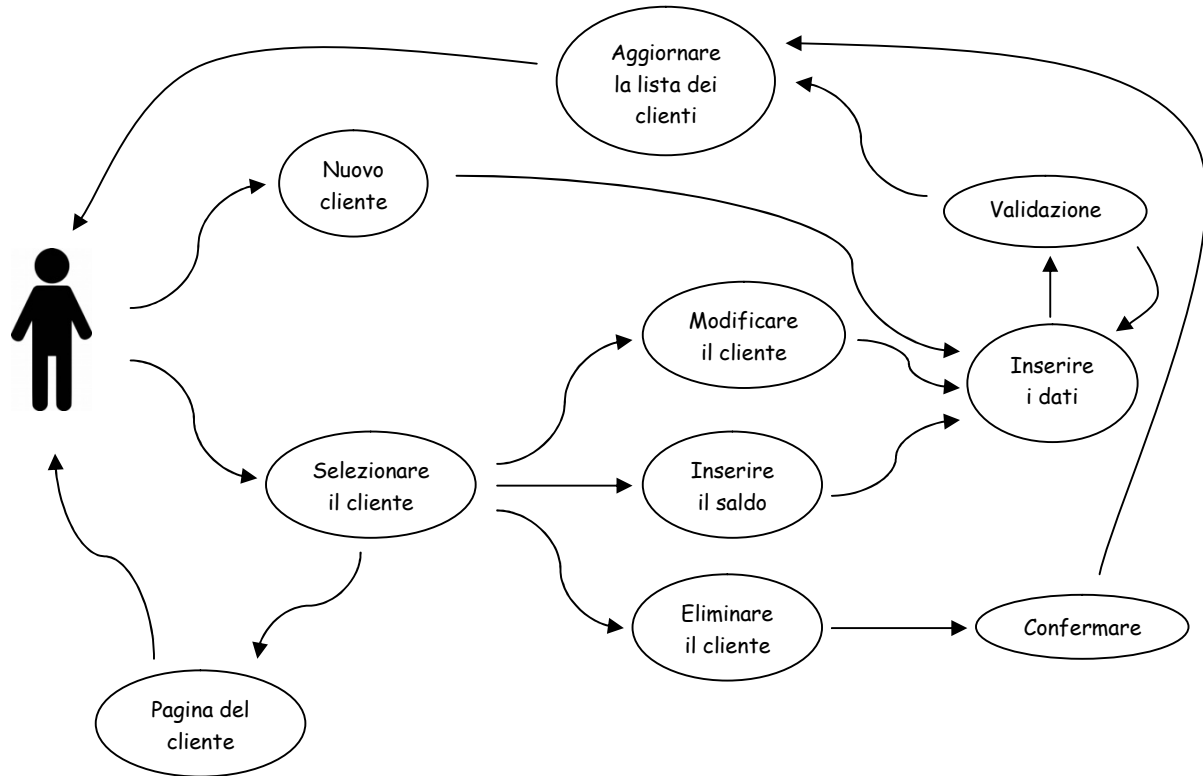
Attributi:

- Un unico campo, chiamato *instance*, utile ad implementare il pattern singleton.

Metodi:

- *GetInstance* è il metodo utilizzato per il pattern singleton; con questo metodo, che verrà richiamato solo quando si vuole creare un oggetto di tipo DBclass, si è sicuri che l'oggetto sarà sempre lo stesso.
- *GetData* è il metodo utilizzato per leggere i dati del cliente dal file JSON e rendere quest'ultimi un oggetto di tipo Client. Per effettuare questa operazione si è dovuto installare il pacchetto Newtonsoft.Json.
- *SaveData* è il metodo utilizzato per salvare i clienti nel file JSON, in questo caso è stato utilizzato il pacchetto System.Text.Json.

USE CASES



Descrizione degli use cases:

Use Case: Nuovo cliente	ID: UC1	Attore: Utente
Precondizione: /////		
Corso degli eventi: L'utente clicca il button "Inserire cliente".		
Postcondizione: UC8		
Alternativa: UC2		

Use Case: Selezionare il cliente	ID: UC2	Attore: Utente
Precondizione: /////		
Corso degli eventi: L'utente seleziona un cliente.		
Postcondizione: UC3 o UC4 o UC5 o UC6		
Alternativa: UC1		

Use Case: Pagina del cliente	ID: UC3	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca il button "Pagina cliente".		
Postcondizione: /////		
Alternativa: UC4 o UC5 o UC6		

Use Case: Modificare il cliente	ID: UC4	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca il button "Modificare cliente".		
Postcondizione: UC8		
Alternativa: UC3 o UC5 o UC6		

Use Case: Inserire il saldo	ID: UC5	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca il button "Inserire saldo".		
Postcondizione: UC8		
Alternativa: UC3 o UC4 o UC6		

Use Case: Eliminare il cliente	ID: UC6	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca il button "Eliminare cliente".		
Postcondizione: UC7		
Alternativa: UC3 o UC4 o UC5		

Use Case: Confermare	ID: UC7	Attore: Utente
Precondizione: UC6		
Corso degli eventi: Compare un messaggio che chiede all'utente di confermare l'operazione.		
Postcondizione: UC10		
Alternativa: /////		

Use Case: Inserire i dati	ID: UC8	Attore: Utente
Precondizione: UC1 o UC4 o UC5		
Corso degli eventi: L'utente inserisce i dati del cliente o della carta.		
Postcondizione: UC9		
Alternativa: /////		

Use Case: Validazione	ID: UC9	Attore: Utente
Precondizione: UC8		
Corso degli eventi: Il programma verifica la correttezza dei dati inseriti.		
Postcondizione: UC8 o UC10		
Alternativa: /////		

Use Case: Aggiornare la lista dei clienti	ID: UC10	Attore: Utente
Precondizione: UC7 o UC9		
Corso degli eventi: La lista dei clienti viene aggiornata in base alle scelte precedenti.		
Postcondizione: /////		
Alternativa: /////		