

RELAZIONE PROGETTO

Programmazione e
modellazione a oggetti



REALIZZATO DA:
Zacchilli Alessandro

a.a. 2020/2021

SPECIFICA DEL SOFTWARE

Il progetto consiste in un software grafico in grado di gestire, per conto di un possibile nutrizionista, la lista dei suoi clienti.

In particolare il programma permette di:

- Aggiungere, modificare e rimuovere un cliente
- Salvare il database dei clienti in un file json
- Visualizzare una pagina contenente le caratteristiche del cliente, con statistiche e grafico sull'andamento del peso
- Inserire nuovi valori di peso nel tempo per ogni cliente

Ogni cliente ha le seguenti caratteristiche:

- Nome e cognome
- Altezza in metri
- Peso in kg
- Sesso
- BMI
- Categoria di peso
- Età

STUDIO DEL PROBLEMA

I punti critici del progetto sono i seguenti:

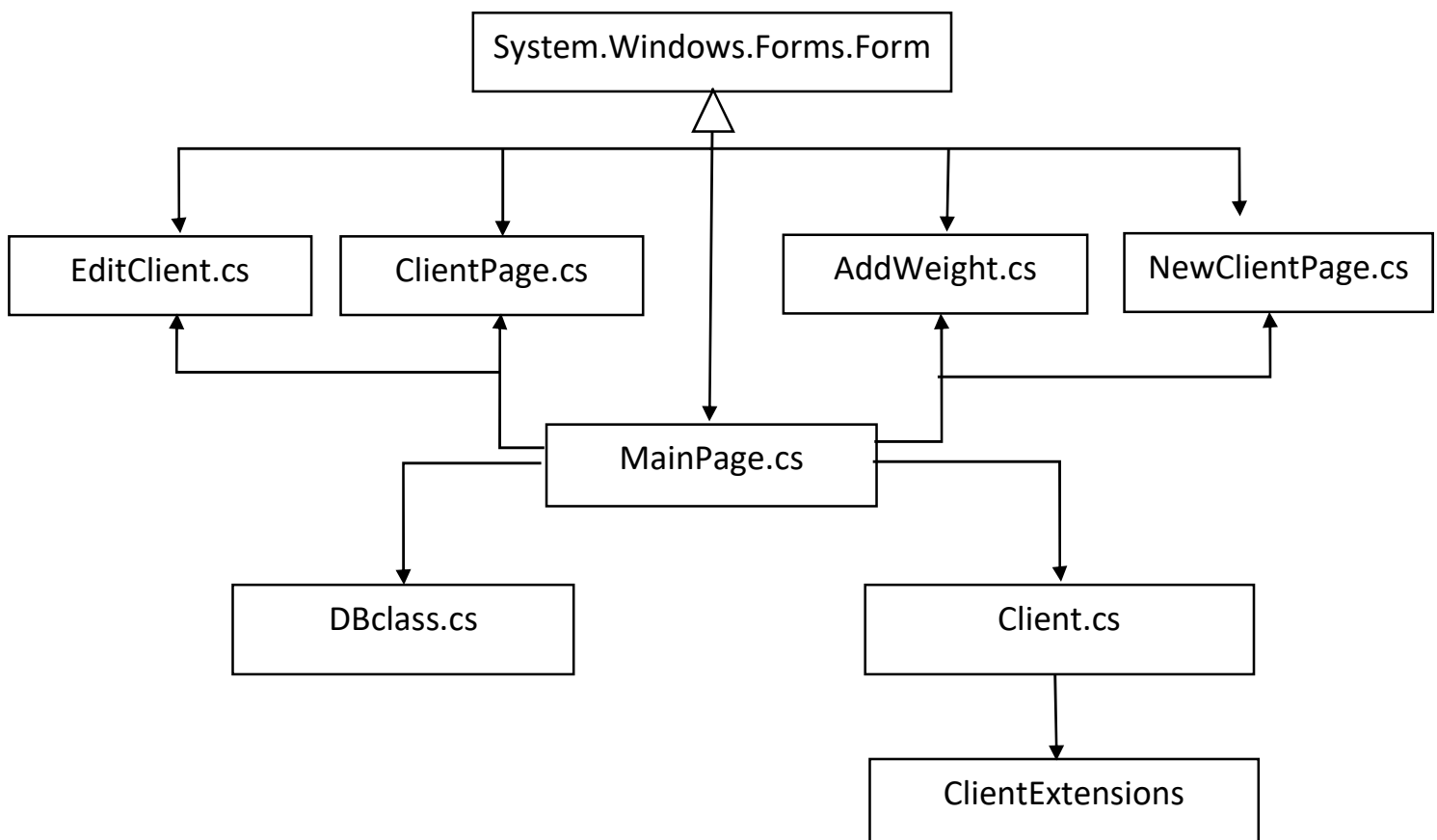
- Aggiungere un nuovo cliente, in questa fase l'utente deve inserire i principali dati relativi alla persona, quindi nome, cognome, data di nascita, sesso, peso e altezza. Una volta inseriti questi dati il programma deve elaborarli e creare un nuovo oggetto (di tipo ListViewItem) da inserire nell'apposita ListView.
- Modificare un cliente, per farlo l'utente deve necessariamente cliccare su un elemento della lista (altrimenti visualizza un messaggio di errore) e sull'apposito tasto. Una volta aperta la schermata l'utente dovrà modificare i campi già riempiti dalle informazioni precedenti e premere ok, il programma provvederà ad eliminare il cliente e aggiungerlo nuovamente modificato nella stessa posizione della lista.
- Aggiungere un nuovo peso ad un cliente già inserito, per fare ciò, come nel caso precedente, l'utente deve selezionare necessariamente un cliente. Una volta aperta la schermata si potrà aggiungere il nuovo peso e la data (necessariamente futura a quelle precedenti), premendo ok il programma

provvederà ad aggiungere il nuovo peso nella “weightlist” e la nuova data nella “datelist” aggiornando così i dati anche nella schermata principale.

- Visualizzare le statistiche del cliente, per fare ciò il software mette a disposizione una schermata con tutte le statistiche e caratteristiche del cliente che, prevedono: nome, cognome, altezza, sesso, età, BMI, categoria, peso iniziale, peso attuale, kg persi o guadagnati, kg persi o guadagnati alla settimana e peso ideale.
- Salvataggio e caricamento del database dei clienti con le relative informazioni, per fare ciò il software utilizza un file in formato json, lo carica all’avvio e lo salva durante la chiusura.

SCELTE ARCHITETTURALI

Il programma ha una classe principale chiamata “MainPage.cs” dove al suo interno ci sono tutti i possibili collegamenti (tramite appositi buttons) alle altre forms, la listview ed il grafico. Come detto in precedenza da questa schermata possiamo aprire le altre forms, che sono: ClientPage.cs, EditClientPage.cs, AddWeightPage.cs e NewClientPage.cs. Le altre classi sono: Client.cs, che contiene tutte le informazioni del cliente; ClientExtensions.cs, utile per creare l’oggetto di tipo listviewitem; DBclass.cs, utile per gestire il database dei clienti.



MainPage.cs

Le principali funzioni di questa form sono:

- Far visualizzare all'utente la lista dei clienti e il relativo grafico di ognuno
- Dare la possibilità di aprire le altre forms

DESIGN:



Come si può notare ci sono 5 pulsanti che aprono le relative form, bisogna sempre selezionare un cliente per aprirle altrimenti c'è un messaggio di errore, tranne per aggiungere un nuovo cliente.

ATTRIBUTI:

- Lista tipizzata con la classe Client (chiamata clients), quindi una lista che contiene tutti i clienti di tipo Client, viene tenuta a private perché utilizzata solo in questa classe o al massimo passata come parametro alle altre forms
- Un oggetto di tipo ListViewItem, chiamato clientItem, utilizzato per aggiungere i clienti nella listview, dato che ogni elemento di quella lista deve essere di quel tipo.
- Un intero chiamato index, per gestire l'inserimento dei clienti nella listview nella giusta posizione
- Un oggetto di tipo DBclass (classe creata) chiamato dbHandler, utile per salvare e caricare i dati dei clienti nel file json

METODI:


- **btnNewClient_Click** è il metodo che viene eseguito alla pressione del pulsante “Nuovo cliente”. Inizialmente crea e visualizza a schermo la nuova form di tipo NewClientPage passandole come parametro la lista dei clienti, poi “trasforma” il cliente che è stato inserito in un oggetto di tipo ListViewItem per inserirlo nella listview. Viene anche utilizzato un controllo sull’attributo di tipo index, se l’utente apre la form ma annulla l’inserimento di un cliente, la lista dei clienti sarà rimasta invariata e quindi il numero dei clienti sarà uguale al numero dell’index, quindi se così è non si inserisce nessun elemento.
- **btnClientPage_Click** è il metodo che viene eseguito quando si clicca sul pulsante “Pagina del cliente”. Se è stato selezionato un cliente crea e visualizza a schermo la nuova form di tipo ClientPage passandole come parametro il cliente selezionato, altrimenti viene visualizzato un messaggio di errore.
- **btnEditClient_Click** è il metodo che viene eseguito quando si preme il pulsante “Modifica cliente”. Se è stato selezionato un cliente crea e visualizza a schermo la form di tipo EditClientPage passandole come parametro il cliente selezionato, una volta che il cliente è stato modificato dall’utente il metodo rimuove il cliente precedente e inserisce quello modificato nella stessa posizione. Se non è stato selezionato nessun cliente viene visualizzato un messaggio di errore
- **btnDelete_Click** è il metodo che viene eseguito quando si preme il pulsante “Elimina”. Se è stato selezionato un cliente viene visualizzato un messaggio di warning, che verifica se l’utente è sicuro di cancellare un cliente, se così è allora si rimuove il cliente e viene cancellato anche il grafico. Quando viene rimosso il cliente viene decrementata anche la variabile index, in modo che se si vorrà inserire un nuovo cliente, sarà inserito nella posizione giusta.
- **btnAddWeight_Click** è il metodo che viene eseguito quando si preme il pulsante “Aggiungi peso”. Se è stato selezionato un cliente crea e visualizza a schermo la form di tipo AddWeightPage passandole come parametro il cliente selezionato, una volta aggiunto il peso e la data c’è una rimozione e reinserimento del cliente per aggiornare la listview.
- **lstVwMain_ItemActivate** è il metodo che viene attivato quando viene selezionato un elemento della listview. Ogni volta che viene selezionato, il metodo pulisce il grafico precedente e crea il nuovo grafico basato sul cliente selezionato scorrendo tutta la lista di date e pesi. Per fare in modo che il grafico abbia la giusta scala, si trova il valore minimo tra i pesi e lo si setta circa (valore - 1) come valore minimo dell’asse delle Y.

- **LoadDB** è il metodo che viene eseguito all'avvio, non fa altro che richiamare il metodo del dbHelper per avere la lista dei clienti e caricarli sulla listView.
- **FinalSaveData** è il metodo che viene eseguito alla chiusura del programma, non fa altro che richiamare il metodo del dbHelper per salvare la lista dei clienti nel file json.

AddWeightPage.cs

Le principali funzioni di questa form sono quelle di far inserire all'utente il nuovo peso da registrare del cliente selezionato, modificando anche la data della pesata (solo nel futuro).

DESIGN:



ATTRIBUTI:

L'unico attributo che c'è è un oggetto di tipo Client, usato per questa classe e anche per altre, infatti è public con le relative proprietà.

METODI:

- **btnCancel_Click** è il metodo che si attiva quando viene premuto il pulsante close, non fa altro che chiudere la form.
- **btnOk_Click** viene attivato quando si clicca sul pulsante OK, aggiunge il peso e la data inseriti dall'utente nelle rispettive liste del cliente e chiude la form.
- **HandleInput**, questo metodo è utile per evitare che l'utente inserisca un input sbagliato.

ClientPage.cs

Le principali funzionalità di questa classe sono quelle di visualizzare a schermo le varie caratteristiche e statistiche del cliente selezionato.

DESIGN:

The screenshot shows a Windows application window titled "Pagina del cliente". It is divided into two main sections: "Cliente" on the left and "Statistiche" on the right.

Cliente Section:

Cliente	
Nome	Francesca
Cognome	Ciancamerla
Altezza	1,73 Metri
Sesso	Femmina
Età	51 Anni
BMI	32,24
Categoria	Obesità di primo grado

Statistiche Section:

lunedì 27 aprile 2020 ▾

Statistiche	
Peso iniziale	107,2 Kg
Peso attuale	96,5 Kg
Kg persi/gudagnati	-10,7 Kg
Perdita/gudagno di peso medio settimanale	-0,51 Kg
Peso ideale	61,7 Kg

lunedì 21 settembre 2020 ▾

ATTRIBUTI:

L'unico attributo di questa classe è un oggetto di tipo Client usato solo all'interno della classe, infatti è private.

METODI:

- **IdealWeightCalculator** è il metodo che calcola il peso ideale del cliente
- **WeightChangedCalculator** è il metodo che calcola il peso totale guadagnato o perso.
- **WeightChangedWeeklyCalculator** calcola il peso guadagnato o perso mediamente a settimana.

EditClientPage.cs

DESIGN:

Modifica cliente

Nome Francesca	Cognome Ciancamerla	Data di nascita lunedì 28 aprile 196 ▾
Sesso <input type="radio"/> Maschio <input checked="" type="radio"/> Femmina	Peso iniziale 107,2 Kg	Altezza 1,73 Metri

OK Annulla

ATTRIBUTI:

L'unico attributo presente è un oggetto di tipo `Client`, che viene usato in questa e in altre classe, infatti è `public` con le relative proprietà.

METODI:

- **btnOk_Click** si attiva quando viene premuto OK, inserisce i nuovi valori inseriti dall'utente nell'oggetto di tipo `Client` (`editedClient`)
- **btnCancel_Click** non fa altro che chiudere la form
- **HandleInput**, questo metodo è utile per evitare che l'utente inserisca un input sbagliato.
- **SetRadioButton** è un metodo utile per settare il bottone per la scelta del sesso, grazie a questo metodo quando si apre la form è già selezionato il sesso che aveva il cliente.

NewClientPage.cs

DESIGN:

The screenshot shows a Windows application window titled "Nuovo cliente". The window contains a form with the following fields and controls:

- Nome**: A text input field.
- Cognome**: A text input field.
- Data di nascita**: A date picker showing "mercoledì 27 gennaio 202" with a dropdown arrow.
- Sesso**: Two radio buttons labeled "Maschio" and "Femmina".
- Peso**: A text input field followed by the unit "Kg".
- Altezza**: A text input field followed by the unit "Metri".
- Buttons**: Two buttons at the bottom right labeled "OK" and "Annulla".

ATTRIBUTI:

- Lista tipizzata di tipo Client, usata in questa classe e in altre, infatti è public con le relative proprietà
- Stringa private usata per il radio button, utile per il sesso del cliente

METODI:

- **btnOk_Click** viene attivato quando si preme ok, prima di tutto controlla, richiamando il metodo **CheckFields**, se tutti i campi sono stati riempiti, se così è allora richiama il metodo **CheckRadioBtn**, il metodo **AddClient** e chiude la form. Se non sono stati riempiti tutti i campi visualizza un messaggio di errore.
- **btnCancel_Click** chiude la form quando si preme annulla.
- **AddClient** è il metodo che si occupa di creare il nuovo oggetto di tipo Client "popolato" con le caratteristiche inserite dall'utente e aggiungerlo nella lista dei clienti, aggiungendo anche la data di inserimento nella lista delle date (per sapere la data iniziale di quando è stato inserito il peso).
- **CheckFields** non fa altro che controllare se sono stati riempiti tutti i campi, se così è restituisce true, altrimenti false.

- **CheckRadioBttn** controlla che tipo di radio button è stato selezionato, se è stato selezionato quello dove c'è scritto "Femmina" allora setta la variabile sex con la stringa "Femmina" altrimenti "Maschio".
- **HandleInput**, questo metodo è utile per evitare che l'utente inserisca un input sbagliato.

Client.cs

Questa è la classe che rappresenta l'oggetto di tipo cliente, ha un attributo per ogni caratteristica, tipo nome, cognome ecc... E non fa altro che rappresentare il modello del cliente.

ATTRIBUTI:

- Come detto in precedenza ha un attributo per ogni caratteristica, inoltre ha 2 liste tipizzate, una di tipo float e una di tipo DateTime.
- La lista tipizzata di tipo float (WeightsList) serve per contenere tutti i possibili pesi che l'utente può inserire per il cliente
- La lista tipizzata di tipo DateTime (DateList) contiene la lista delle date utili per sapere quando il peso è stato registrato.

METODI:

- **AgeCalculator** è il metodo utilizzato per calcolare l'età del cliente, dato che l'utente inserisce solo la data di nascita.
- **BMICalculator** viene utilizzato per calcolare il BMI usando peso e altezza
- **LastWeight** non fa altro che restituire l'ultimo peso della WeightsList
- **CategoryCalculator** calcola la categoria a cui appartiene il cliente, basato sul valore del BMI.
- **LastDate** restituisce l'ultima data della DateList
- **FindMinimumWeight** è un metodo che implementa un algoritmo di ricerca del minimo e restituisce il peso più piccolo della WeightsList, utile per il grafico.

ESTENSIONE:

- **ClientExtension.cs** è una classe estensionale per la classe cliente che non fa altro che "trasformare" un oggetto di tipo Client in un oggetto di tipo ListViewItem, in modo che così può essere inserito nella listview.

DBclass.cs

Questa classe serve per gestire il database, implementa i metodi per salvare i dati in un file json e viceversa. Viene inoltre utilizzato il pattern singleton per questa classe, in modo da assicurarci di avere una sola istanza di quest'oggetto.

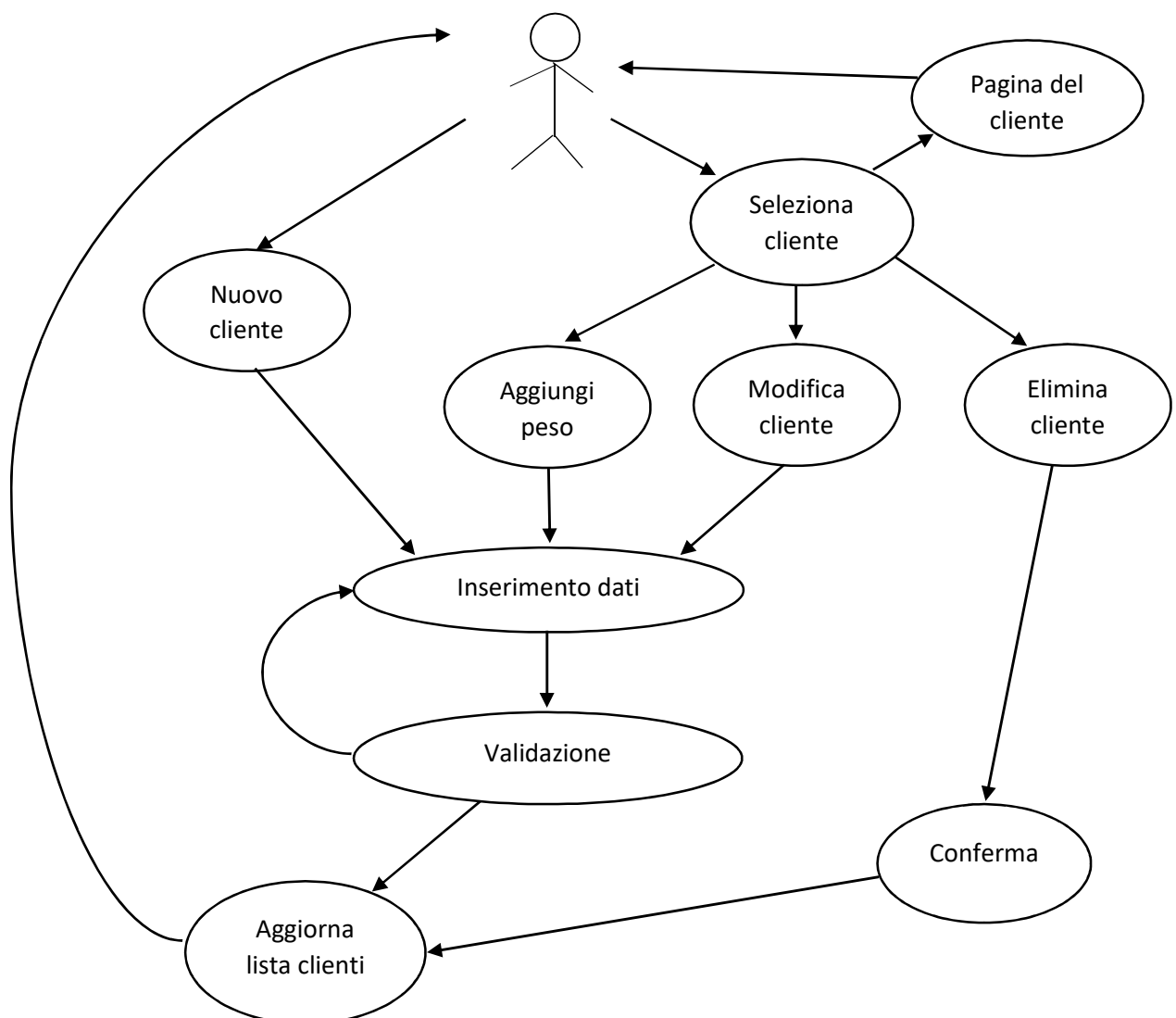
ATTRIBUTI:

- Ha un solo attributo chiamato "instance" utilizzato per implementare il pattern singleton.

METODI:

- **GetInstance** è il metodo del pattern singleton, grazie a questo metodo che verrà richiamato quando vogliamo creare un oggetto di tipo DBclass siamo sicuri che sarà sempre lo stesso.
- **GetData** viene utilizzato per leggere i dati dal file json e renderli un oggetto di tipo Client, per questa operazione si usa il pacchetto Newtonsoft.Json.
- **SaveData** non fa altro che salvare i clienti della lista nel file json utilizzando il pacchetto System.Text.Json.

CASI D'USO:



Descrizione dei casi d'uso:

Use case: Pagina del cliente	ID: UC1	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca nel pulsante chiamato "pagina del cliente"		
Post-condizione: /		
Alternativa: UC4 o UC5 o UC6		

Use Case: Seleziona cliente	ID: UC2	Attore: Utente
Precondizione: /		
Corso degli eventi: L'utente seleziona un cliente		
Post-condizione: UC1 o UC4 o UC5 o UC6		
Alternativa: UC3		

Use Case: Nuovo cliente	ID: UC3	Attore: Utente
Precondizione: /		
Corso degli eventi: L'attore clicca il pulsante chiamato "Nuovo cliente"		
Post-condizione: UC7		
Alternativa: UC2		

Use Case: Aggiungi peso	ID: UC4	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca il pulsante "Aggiungi peso"		
Post-condizione: UC7		
Alternativa: UC1 o UC5 o UC6		

Use Case: Modifica cliente	ID: UC5	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca il pulsante "Modifica cliente"		
Post-condizione: UC7		
Alternativa: UC1 o UC4 o UC6		

Use Case: Elimina cliente	ID: UC6	Attore: Utente
Precondizione: UC2		
Corso degli eventi: L'utente clicca il pulsante "Elimina"		
Post-condizione: UC9		
Alternativa: UC1 o UC4 o UC5		

Use Case: Inserimento dati	ID: UC7	Attore: Utente
Precondizione: UC3 o UC4 o UC5		
Corso degli eventi: L'utente inserisce i dati		
Post-condizione: UC8		
Alternativa: /		

Use Case: Validazione	ID: UC8	Attore: /
Precondizione: UC7		
Corso degli eventi: Il programma verifica se sono stati inseriti i dati correttamente		
Post-condizione: UC10 o UC7		
Alternativa: /		

Use Case: Conferma	ID: UC9	Attore: Utente
Precondizione: UC6		
Corso degli eventi: Viene mostrato un messaggio che chiede all'utente di confermare la scelta		
Post-condizione: UC10		
Alternativa: /		

Use Case: Aggiorna clienti	ID: UC10	Attore: /
Precondizione: UC8 o UC9		
Corso degli eventi: La lista degli utenti viene aggiornata in base alle scelte precedenti		
Post-condizione: /		
Alternativa: /		