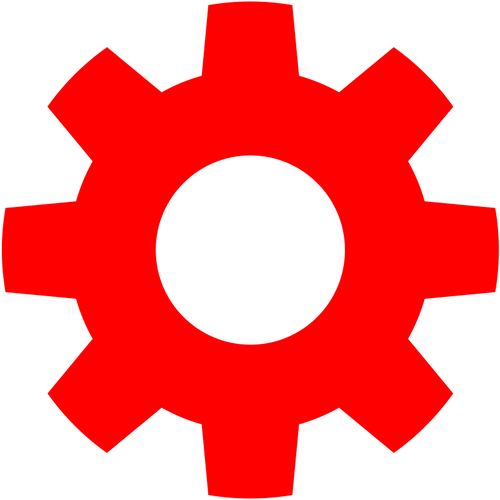
|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**SHOP-AUTORICAMBI  
Requirements Analysis Document - RAD  
Versione 0.2**



**INDICE**

1. **INTRODUCTION**
   1. **Purpose of the system**
   2. **Scope of the system**
   3. **References**
2. **CURRENT SYSTEM**
3. **PROPOSED SYSTEM**
   1. **Overview**
   2. **Functional requirements**
   3. **Nonfunctional requirements**
      1. **Usability**
      2. **Reliability**
      3. **Performance**
      4. **Supportability**
      5. **Implementation**
      6. **Interface**
      7. **Packaging**
      8. **Legal**
   4. **System models**
      1. **Scenarios**
      2. **Use case model**
      3. ***Object model***

***3.4.3.1 Data dictionary***

***3.4.3.2 Class diagram***

* + 1. ***Dynamic model***
    2. **User interface-navigational paths and screen mock-ups**

1. **GLOSSARY**
2. **INTRODUCTION**
   1. **Purpose of the system**

Si vuole realizzare un software, denominato Shop-Autoricambi, per la gestione di un sito e-commerce destinato alla vendita di pezzi di ricambio per automobili. A tal proposito, possiamo notare che un pezzo di ricambio, ossia un prodotto, può essere acquistato da uno e un solo utente e, viceversa, che un utente può acquistare più prodotti. I prodotti sono caratterizzati da un codice, dal modello, dal marchio, dal prezzo, dalla quantità e da una breve descrizione. Inoltre, un prodotto, potrebbe essere messo in offerta ed avere, quindi, un prezzo scontato.

Un utente, che si collega al sito di e-commerce mediante questo software, può registrarsi al sito e diventare un potenziale cliente.

Un utente registrato, avrà uno ed un solo indirizzo ma, viceversa, un indirizzo potrebbe corrispondere ad uno o più utenti registrati. Un indirizzo, in questo caso, è costituito da un CAP, da una via e da una città. Un utente registrato, invece, è caratterizzato da un codice fiscale che lo identifica univocamente, dal nome, dal cognome, dalla data di nascita, da un numero di telefono, da un numero di cellulare, da un’email e da una password. L’utente registrato, però, potrebbe anche essere il gestore di tale sito e svolgere funzioni diverse.

Il software che si vuole realizzare, infatti, dovrà consentire ai gestori di: autenticarsi, visualizzare lo storico degli acquisti di tutti i clienti e caricare dei nuovi prodotti da poter vendere. Il software dovrà consentire agli utenti registrati di: autenticarsi, effettuare acquisti, tenere traccia degli acquisti effettuati, tenere traccia dei prodotti scelti per un eventuale acquisto, eliminare uno o più prodotti dalla lista dei prodotti scelti per un eventuale acquisto e affiliarsi ad officine presso cui montare i prodotti acquistati.

Un utente registrato può, dunque, affiliarsi ad una o più officine o partner e, viceversa, un’officina può avere uno o più utenti affiliati. Un’officina, o partner, è caratterizzata da una partita iva che la identifica univocamente, da un email e da un numero di telefono. Inoltre, un’officina, ha uno ed un solo indirizzo ma, viceversa, ad un indirizzo potrebbero trovarsi una o più officine. Un indirizzo di un’officina è caratterizzato da un CAP, da una via e da una città.

Il software, inoltre, deve consentire agli utenti la ricerca dei pezzi di ricambio tramite il loro codice, la loro marca oppure il loro modello. Ogni prodotto visualizzato dal sito di e-commerce, tramite il supporto del software, avrà anche delle foto ad esso associato. In particolare, un prodotto ha una o più foto che lo descrivono ma, viceversa, una foto ha uno ed un solo prodotto a cui si riferisce.

Il software, come già descritto in precedenza, deve tenere traccia degli acquisti effettuati dai vari clienti, i quali possono avere una o più fatture, viceversa, una fattura è in possesso di uno e un solo cliente, o utente registrato. Una fattura è caratterizzata da un codice fattura che la identifica univocamente, da una data di rilascio della stessa, da un costo totale e dal tipo di pagamento effettuato. Una fattura, di conseguenza, si riferisce ad uno o più prodotti acquistati in un certo momento. Un prodotto acquistato è caratterizzato da un codice di acquisto che lo identifica univocamente e da una descrizione.

* 1. **Scope of the system**

# Le funzionalità del sito che si vuole realizzare sono, principalmente, le seguenti:

# - Registrazione degli utenti

# - Possibilità di acquistare pezzi di ricambio per la propria auto

# - Possibilità di affiliarsi ad officine presso cui montare i pezzi acquistati

# - Informazioni e contatti

# - Visualizzazione delle caratteristiche e della descrizione formale dei pezzi di ricambio

venduti da tale sito.

- Visualizzazione dello storico degli acquisti effettuati da un cliente

* 1. **References**

Riferimenti ad altri siti di e-commerce esistenti

**http://www.mister-auto.it/**

Offre le seguenti possibilità:

- Acquistare pezzi di ricambio in base alla casa costruttrice dell’auto

- Acquistare pezzi di ricambio in base ad una certa marca

- Acquistare pneumatici e accessori vari

- Visualizzare le caratteristiche dei prodotti venduti

**http://www.tuttoautoricambi.it/**

Offre le seguenti possibilità:

- Acquistare pezzi di ricambio in base alla casa costruttrice dell’auto

- Acquistare pezzi di ricambio in base ad una certa marca

- Acquistare pneumatici e accessori vari

- Visualizzare le caratteristiche dei prodotti venduti

- Visualizzare le info e i contatti inerenti al sito

**https://www.shop-ricambiauto.it/**

Offre le seguenti possibilità:

- Acquistare pezzi di ricambio in base ad una certa marca

- Acquistare pneumatici e accessori vari

- Visualizzare le caratteristiche dei prodotti venduti

- Ricercare un pezzo di ricambio in base ad un certo prezzo

1. **CURRENT SYSTEM**

Il sistema che si vuole realizzare è ancora in fase di Analysis. Il sistema non sostituisce nessun altro sistema esistente. Inoltre, essendo ancora in fase di Analysis non è stata sviluppata nessuna funzionalità del sistema.

1. **PROPOSED SYSTEM**
   1. **Overview**

Requirement Analysis Document (**RAD**) dedicato ai:

* Requisiti Funzionali
* Requisiti Non Funzionali
* Scenari
* Use Case Model
* Sequence Diagrams
* State-Chart Diagrams
  1. **Functional requirements**

**RF1:** Il software dovrà permettere ai **visitatori** la visualizzazione dei prodotti commercializzati.

**RF2:** Il software dovrà permettere ai **visitatori** di autenticarsi nel momento in cui vorranno selezionare dei potenziali prodotti da acquistare.

**RF3:** Il software dovrà permettere la registrazione di vari **clienti** compilando un apposito modulo.

**RF4:** Il software dovrà permettere ai **clienti** di acquistare, in diverse quantità, uno o più prodotti.

**RF5:** Il software dovrà permettere ai **clienti** di poter modificare le quantità dei potenziali prodotti da acquistare.

**RF6:** Il software dovrà permettere ai **clienti** di potersi affiliare ad una o più officine presso le quali potranno, in presenza di personale qualificato, installare i prodotti acquistati.

**RF7:** Il software dovrà prevedere un **gestore** offrendo loro la possibilità di caricare vari prodotti.

**RF8:** Il software dovrà essere capace di consigliare i **clienti**, tramite un apposito modulo, la scelta dell’officina presso cui installare i prodotti acquistati.

**RF9:** Il software dovrà permettere al **gestore** di poter gestire i pezzi di ricambio di tutte le marche di auto.

**RF10:** Il software dovrà permettere al **gestore** di visualizzare ed, eventualmente, approvare i moduli, riguardanti le officine, sottomessi dai vari clienti.

**RF11:** Il software dovrà permettere al **gestore** di visualizzare tutte le attività svolte da tutti i clienti.

**RF12:** Il software dovrà permettere l’autenticazione, tramite login e password, dei **clienti** e del **gestore** in maniera tale da dare loro la possibilità di accedere alla loro area utente.

* 1. **Nonfunctional requirements**
     1. **Usability**

L’interfaccia grafica presenterà un menù sul top della pagina dove verranno indicati in maniera molto visibile le varie sezioni che il sito offre dando la possibilità al cliente di avere una visione completa e generale del sito. Nella home la ricerca di un prodotto nel sito verrà facilitata dalla presenza di un form di ricerca.

La schermata principale del sito Shop-Autoricambi è formata da 4 parti:

- che cos’è;

- servizi;

- prodotti;

- contatti.

* + 1. **Reliability**

Il software dovrà essere attivo 24 ore su 24. Inoltre, deve garantire la sicurezza sul tutte le operazioni effettuate sia dai gestori sia dai clienti. Nel caso in cui si verifichino dei comportamenti anomali, da parte del gestore o del cliente, verranno notificati tramite degli avvisi.

* + 1. **Performance**

Il software dovrà rispondere velocemente; le risposte dovranno essere fornite in un periodo pari a circa 1 secondo. Il numero di utenti che potranno collegarsi e acquistare prodotti contemporaneamente sarà dato dalla disponibilità del Web Server utilizzato. La latenza massima di attesa, per una risposta, non dovrà superare i 30 secondi.

* + 1. **Supportability**

Il software consisterà in un sistema client-server in cui il server sarà disponibile su ogni tipo di piattaforma e il client potrà collegarsi a tale server mediante un qualsiasi browser. Il software dovrà essere suddiviso in vari moduli per permettere una più facile modifica e aggiornabilità in futuro.

* + 1. **Implementation**

I dati saranno caricati attraverso appositi form in cui il gestore andrà a specificare il prodotto caricato, attraverso nome e codice ed inoltre andrà a specificare la quantità di questo prodotto a disposizione in magazzino.

* + 1. **Gestione:** il software sarà gestito da un amministratore che sarà anche il gestore

del magazzino.

* + 1. **Packaging**

Il software potrà essere installato, lato server, attraverso una procedura di installazione. Il software, una volta installato dal gestore del magazzino, non sarà legato a vincoli temporali.

* + 1. **Legal**

Il software avrà licenza legata a MySql e il costo sarà legato alle percentuali di guadagno del sistema.

* 1. **System models**
     1. **Scenarios**

**Scenario login del gestore**

**Attori: gestore**

Il gestore si collega, mediante un qualsiasi browser, al software. Il software visualizza la homepage. Il gestore clicca sul tasto login. Il software risponde mostrando la pagina di login con i campi e-mail e password. Il gestore, per autenticarsi, inserisce la sua e-mail e la sua password e clicca sul tasto accedi. A questo punto, il software controlla se l’utente ha compilato i campi correttamente cioè se ha inserito e-mail e password relativi ad un account esistente. Se l’utente ha inserito e-mail e password correttamente, il software visualizzerà nuovamente la homepage, altrimenti mostrerà un messaggio d’errore e la pagina di registrazione

**Scenario registrazione e login di un utente**

**Attori: utente, utente registrato**

L’**utente** si collega al software mediante un qualsiasi browser. Il software visualizza la homepage. L’**utente** clicca sul tasto login. Il software risponde mostrando la pagina di login. L’**utente**, per registrarsi, clicca sulla voce Registrazione. Il software risponde mostrando la pagina di registrazione contenente dei campi da compilare: nome, cognome, codice fiscale, data di nascita, telefono, cellulare, via, CAP, città, e-mail, password e conferma password. L’**utente** inserisce i dati negli appositi campi e clicca sul tasto Registrati. A questo punto, il software controlla che i campi inseriti siano corretti. Se l’**utente** ha compilato tutti i campi correttamente, il software visualizzerà nuovamente la homepage, altrimenti mostrerà uno specifico messaggio d’errore.

**L’utente registrato** si può trovare sulla homepage in due modi: dopo essersi registrato oppure dopo essersi collegato tramite software. L’**utente registrato** clicca sul tasto login. Il software risponde mostrando la pagina di login con i campi e-mail e password. L’**utente registrato**, per autenticarsi, inserisce la sua e-mail e la sua password e clicca sul tasto accedi. A questo punto, il software controlla se l’utente ha compilato i campi correttamente cioè se ha inserito e-mail e password relativi ad un account esistente. Se l’utente ha inserito e-mail e password correttamente, il software visualizzerà nuovamente la homepage, altrimenti mostrerà un messaggio d’errore e la pagina di registrazione.

**Scenario logout del gestore**

**Attori: gestore**

Il gestore che è loggato all’interno del software premendo il tasto logout viene riportato alla pagina principale del software dalla quale tutti possono accedere senza avere un account che sia di gestore che di cliente.

**Scenario logout del cliente**

**Attori: cliente**

Il cliente che è loggato all’interno del software premendo il tasto logout viene riportato alla pagina principale del software, dalla quale tutti possono accedere senza avere un account che sia di gestore che di cliente.

**Scenario carica prodotto**

**Attori: gestore**

Il **gestore** si collega al software mediante un qualsiasi browser. Il software visualizza la homepage. Il **gestore** clicca sul tasto login. Il software risponde mostrando la pagina di login. Il **gestore**, per loggarsi, inserisce     e-mail e password e clicca sulla voce Accedi. Il software risponde mostrando la l’area riservata del **gestore.**

Il **gestore** entra nella pagina prodotti, dove grazie al privilegio **gestore** viene mostrato un form di caricamento del prodotto. Il **gestore** compila i form con le opportune descrizioni del prodotto e clicca su carica.

Il software elabora la richiesta del **gestore.**

Se l’elaborazione va a buon fine sarà mostrato il prodotto appena caricato, altrimenti sarà mostrato un messaggio di errore.

**Scenario ricerca di prodotti**

**Attori: utente**

L’**utente** si collega al software mediante un qualsiasi browser. Il software visualizza la homepage. L’**utente** compila i campi per la ricerca e preme sul pulsante Ricerca. Il software controlla il contenuto dei campi. Se sono stati omessi uno o più campi il software risponde mostrando la pagina di tutti i prodotti trovati non tendendo conto del campo o dei campi omessi, altrimenti il software mostra i prodotti trovati in base al contenuto dei campi compilati dall’utente.

**Scenario carrello cliente**

**Attori: cliente**

Il **cliente** seleziona la voce prodotti dalla homepage del sito. Il software restituisce la lista dei vari prodotti disponibili. Il **cliente** prende visione dei vari prodotti e per accedere ai dettagli di un prodotto clicca sul tasto dettagli. Il software restituisce la pagina di descrizione del prodotto consentendo di aggiungere al carrello il prodotto che si sta visionando. Prima di inserire il prodotto nel carrello deve essere indicata la quantità. Il **cliente** procede all’aggiunta del prodotto nel carrello tramite il pulsante aggiungi al carrello. Il software elabora la richiesta e permette al cliente di continuare la ricerca di altri prodotti. Inserendo una quantità maggiore rispetto alla quantità disponibile del prodotto il software avviserà, tramite messaggio di errore, il **cliente**, ciò accadrà anche nel caso in cui la quantità non fosse specificata o si inserisce una quantità negativa. I prodotti possono essere aggiunti al carrello sia se il **cliente** è loggato sia se non lo è. Il **cliente** una volta selezionati i vari prodotti ne prende visione nella sezione carrello. Per ogni prodotto nel carrello vengono mostrati i campi: descrizione, prezzo, quantità, totale e affiliazione. Il **cliente** può rimuovere un prodotto dal proprio carrello premendoil pulsante elimina. Una volta premuto il pulsante il software risponderà ricaricando la pagina senza mostrare più il prodotto eliminato.

**Scenario acquisto di prodotti**

**Attori: cliente**

Il **cliente** tramite la propria sezione carrello visualizza i vari prodotti precedentemente selezionati.  Il **cliente** ricontrolla i prodotti, verificando il tipo e le quantità selezionate. Il **cliente** prima di procede all’acquisto indica l’officina affiliata per il montaggio del prodotto. Il cliente effettua l’acquisto premendo il pulsante acquista. Il software elabora la richiesta del **cliente.** Se l’elaborazione va a buon fine viene il software risponde visualizzando nuovamente il carrello con le varie informazioni di tutti gli acquisti effettuati del cliente, altrimenti verrà mostrato un messaggio di errore e non si potrà procedere con l’acquisto.

**Scenario registrazione dell’officina affiliata**

**Attori: visitatore**

Il **visitatore** può consigliare il suo servizio affiliandosi al nostro sito. Il **visitatore** entra nel nostro sito, e tramite un link accede alla sezione servizi dove sarà mostrato il form per la registrazione. Il **visitatore** compila il form obbligatorio inserendo i suoi dati. Dopo aver riempito tutti i campi clicca sul pulsante affiliati. Il software elabora la richiesta del **visitatore.** Se l’elaborazione va a buon fine sarà mostrato un messaggio di corretta registrazione, altrimenti sarà mostrato un messaggio di errore.

**Scenario storico del cliente**

**Attori: cliente**

Il **cliente**, se non è già loggato, clicca sul pulsante log-in per entrare nella sua area riservata, altrimenti clicca sul pulsante Area Cliente. Il software mostra l’area utente del **cliente**. Il **cliente** può, a questo punto, visualizzare tutti gli acquisti effettuati in passato.

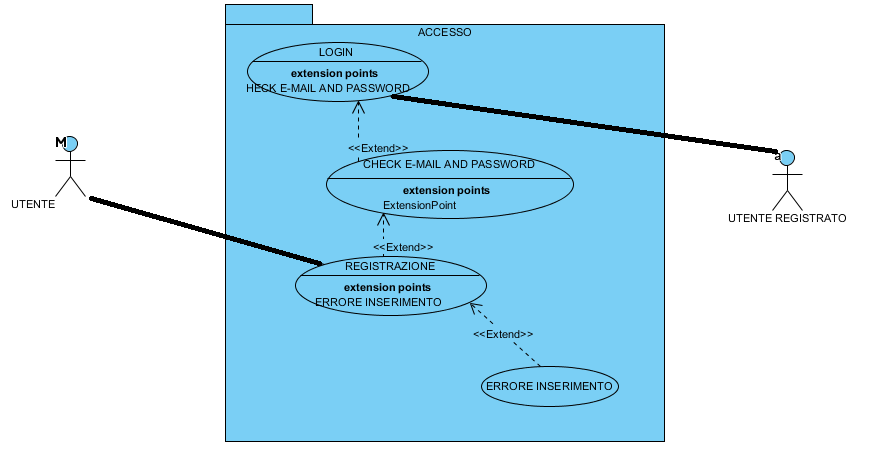
**Scenario storico del gestore**

**Attori: gestore**

Il gestore, se non è già loggato, clicca sul pulsante log-in per poter entrare nella sua area riservata, altrimenti clicca sul pulsante Area Gestore. Il software mostra l’area utente del gestore. Il gestore può, a questo punto, visualizzare lo storico di tutti i prodotti acquistati da tutti i clienti.

* + 1. **Use case model**

**Caso d’uso: registrazione di un utente - login di un utente**



**Caso d’uso: registrazione di un utente**

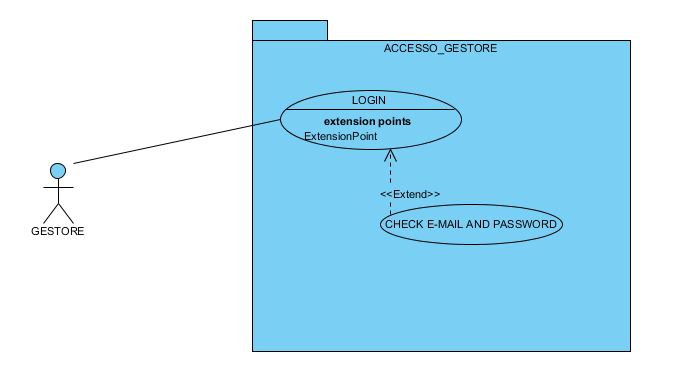
Nome del caso d’uso: **Registrazione**

* Attori partecipanti: **utente**
* Entry Condition:
  + L’utente non possiede un account e si collega al software per registrarsi.
* Flusso Eventi:
  + L’utente invoca la funzione **Registrazione**. Shop-Autoricambi risponde mostrando la pagina di registrazione con alcuni campi da compilare.
  + L’utente inserisce nome, cognome, codice fiscale, data di nascita, telefono, cellulare, via, CAP, città, e-mail, password e conferma password.
* Exit Condition:
  + L’utente clicca sul tasto Registrati
* Exception:
  + Se almeno un campo non è stato compilato in maniera corretta dall’utente si invoca la funzione **Errore Inserimento** la quale mostra un messaggio d’errore.

**Caso d’uso: login di un utente**

* Nome del caso d’uso: **Login**
* Attori partecipanti: **utente registrato**
* Entry Condition:
  + L’utente vuole autenticarsi tramite il software
* Flusso Eventi:
  + L’utente invoca la funzione **Login**. Shop-Autoricambi risponde mostrando la pagina di login con i campi email e password da compilare.
  + L’utente inserisce la sua e-mail e la sua password
* Exit Condition:
  + L’utente clicca sul tasto Accedi.
* Exception:
  + Se l’email e/o la password inseriti dall’utente non sono corretti si invoca la funzione **Check E-mail And Password** la quale mostrerà un messaggio d’errore. Inoltre, Shop-Autoricambi risponderà mostrando la pagina di registrazione.

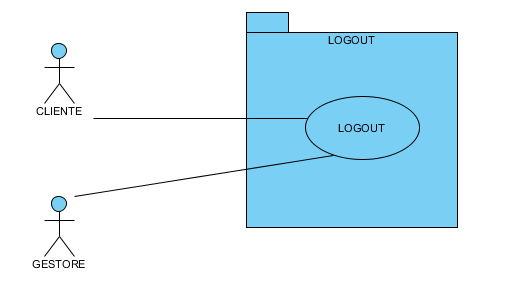
**Caso d’uso: login di un gestore**



* Nome del caso d’uso: **Login**
* Attori partecipanti: **gestore**
* Entry Condition:
  + Il gestore vuole autenticarsi tramite il software
* Flusso Eventi:
  + Il gestore invoca la funzione **Login**. Shop-Autoricambi risponde mostrando la pagina di login con i campi email e password da compilare.
  + Il gestore inserisce la sua e-mail e la sua password
* Exit Condition:
  + Il gestore clicca sul tasto Accedi.
* Exception:

Se l’email e/o la password inseriti dal gestore non sono corretti si invoca la funzione **Check E-mail And Password** la quale mostrerà un messaggio d’errore. Inoltre, Shop-Autoricambi risponderà mostrando la pagina di registrazione.

**Caso d’uso: logout di un gestore - logout di un cliente**



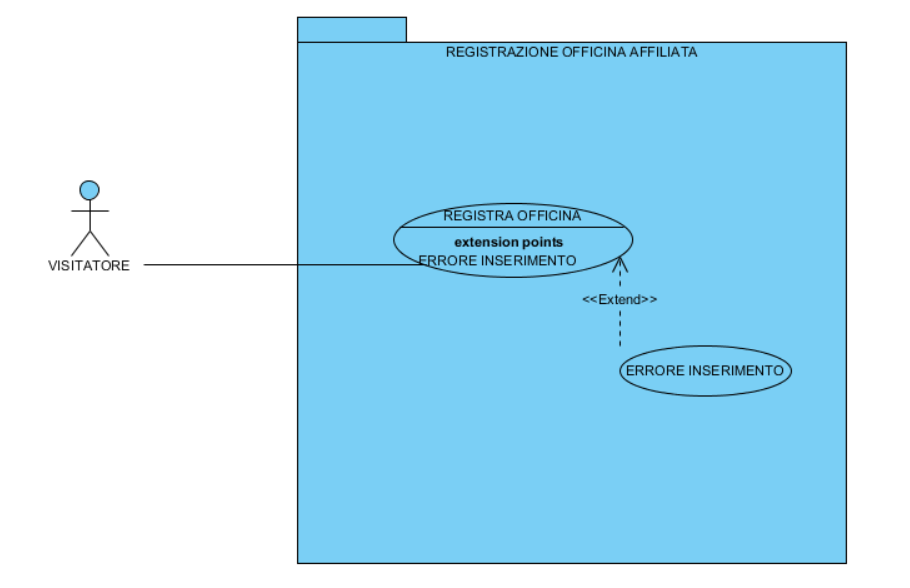
**Caso d’uso: logout di un gestore**

* Nome del caso d’uso: **Logout**
* Attori partecipanti: **gestore**
* Entry Condition:
  + Il gestore è loggato.
* Flusso Eventi:
  + Il gestore pensa di uscire dall’area riservata
  + Preme il tasto LOGOUT
  + Viene riportato alla pagina Home.
* Exit Condition:
  + Il gestore clicca sul tasto LOGOUT.

**Caso d’uso: logout di un cliente**

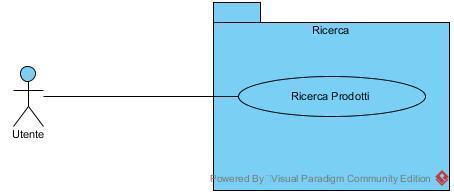
* Nome del caso d’uso: **Logout**
* Attori partecipanti: **cliente**
* Entry Condition:
  + Il cliente è loggato.
* Flusso Eventi:
  + Il cliente pensa di uscire dall’area riservata
  + Preme il tasto LOGOUT
  + Viene riportato alla pagina Home.
* Exit Condition:
  + Il cliente clicca sul tasto LOGOUT.

**Caso d’uso: registrazione dell’officina affiliata**



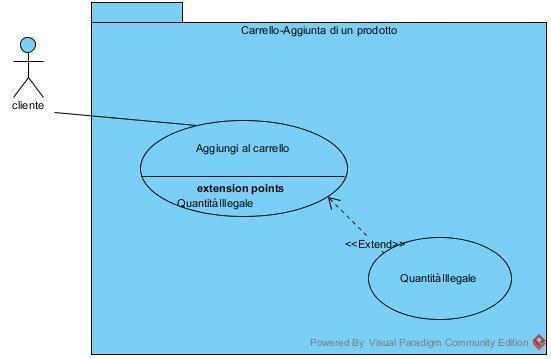
* Nome del caso d’uso: **registrazione dell’officina affiliata.**
* Attori partecipanti: **visitatore.**
* Entry Condition:
* Il visitatore visita il sito.
* Flusso Eventi:
* Il visitatore accede tramite un link all’area servizi.
* Il visitatore compila il form di registrazione.
* Exit Condition:
* Il visitatore clicca sul pulsante affiliati
* Exception:
* Se Il visitatore non ha compilato correttamente il form per la registrazione si invoca la funzione **Errore Di Compilazione**, la quale mostra un messaggio d’errore, vietando così la possibilità di completare l’affiliazione.

**Caso d’uso: ricerca di prodotti**



* Nome del caso d’uso: **Ricerca di prodotti**
* Attori partecipanti: **utente**
* Entry Condition:
  + L’utente si collega alla home page del sito mediante un qualsiasi browser.
* Flusso Eventi:
  + Shop-Autoricambi risponde mostrando la home page in cui è presente un modulo di ricerca contenente alcuni campi da compilare.
* L’utente inserisce codice prodotto, marca e modello.
* Exit Condition:
  + L’utente clicca sul tasto Ricerca. Shop-Autoricambi risponde attivando la funzionalità **Ricerca Prodotti**.
* Exception:
* Se l’utente non compila uno o più campi Shop-Autoricambi mostrerà una pagina contenente tutti i prodotti trovati senza tener conto dei campi omessi.

**Caso d’uso: aggiunta di un prodotto al carrello**

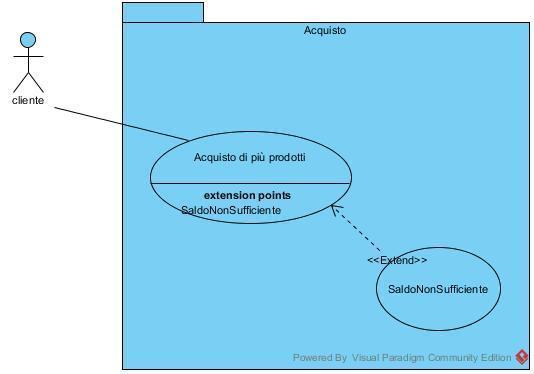


* Nome del caso d’uso: **Aggiungi al carrello**
* Attori partecipanti: **cliente**
* Entry Condition:
* Il cliente può essere loggato o meno.
* Il cliente si trova nella sezione dettagli del prodotto.
* Flusso Eventi:
* Il cliente inserisce la quantità del prodotto.
* Exit Condition:
* Il cliente procede all’aggiunta del prodotto premendo il tasto aggiungi prodotto.
* Exception:
* Se il cliente digita una quantità negativa o nulla si invoca la funzione

**Quantità Illegale**, la quale mostra un messaggio d’errore, vietando

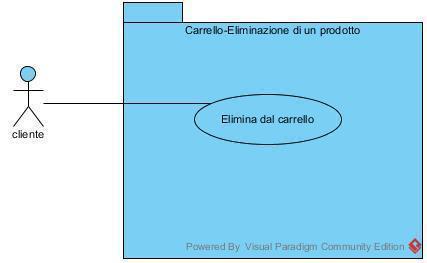
così la possibilità di completare l’aggiunta al carrello.

**Caso d’uso: acquisto di uno o più prodotti**



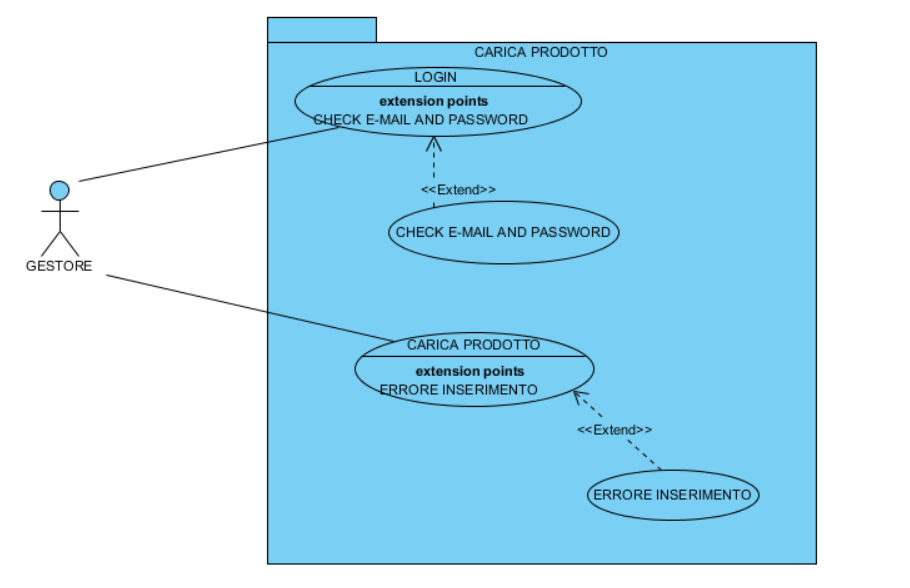
* Nome del caso d’uso: **Acquisto di prodotti**
* Attori partecipanti: **cliente**
* Entry Condition:
* Il cliente si trova nell’area “carrello”.
* Flusso Eventi:
* Il cliente ricontrolla i prodotti nel carrello da acquistare.
* Il cliente indica l’officina affiliata.
* Exit Condition:
* Il cliente procede all’acquisto dei prodotti premendo sul tasto acquista.
* Exception:
  + Se il cliente al momento dell’acquisto il prodotto non è più disponibile, viene lanciata un’eccezione **Prodotto Non Disponibile**, la quale mostra un messaggio d’errore, vietando così la possibilità di completare l’acquisto.

**Caso d’uso: eliminazione di un prodotto**



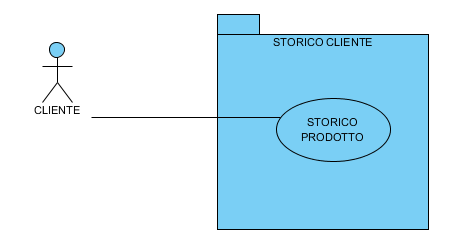
* + - Nome del caso d’uso: **Elimina dal carrello**
    - Attori partecipanti: **cliente**
    - Entry Condition:
      * Il cliente può essere loggato o meno.
      * Il cliente si trova nella sezione carrello
    - Flusso Eventi:
      * Il cliente individua il prodotto da eliminare.
    - Exit Condition:
      * Il cliente procede all’eliminazione del prodotto premendo il tasto elimina.

**Caso d’uso: inserimento dell’officina affiliata**



* Nome del caso d’uso: **carica prodotto.**
* Attori partecipanti: **gestore.**
* Entry Condition:
* Il gestore esegue il login.
* Il gestore accede tramite un link all’area prodotti. Shop-Autoricambi risponde mostrando la pagina per il caricamento di un prodotto.
* Flusso Eventi:
* Il gestore compila il form per caricare il prodotto, inserendo codice del prodotto, marchio, modello, descrizione, prezzo, quantità, lo stato di offerta ed il prezzo inerente alla stato d’offerta.
* Exit Condition:
* Il gestore dell’officina clicca sul pulsante carica.
* Exception:
* Se il gestore non ha compilato correttamente il form per il caricamento del prodotto si invoca la funzione **Errore Di Compilazione**, la quale mostra un messaggio d’errore, vietando così la possibilità di completare il caricamento del prodotto.

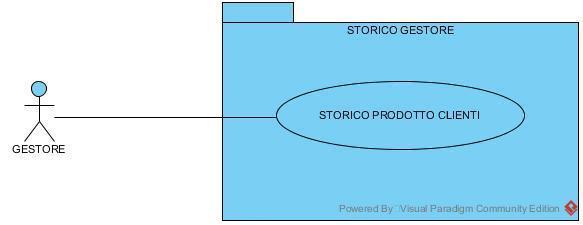
**Caso d’uso: storico di un cliente**



* Nome del caso d’uso: **storico cliente**
* Attori partecipanti: **cliente**
* Entry Condition:
  + Il cliente si trova nella sua pagina riservata.

1. Flusso Eventi:
   * Il cliente pensa di visualizzare lo storico degli acquisti
   * Il cliente invoca la funzione **STORICO PRODOTTO** premendo il tasto Area Cliente
   * Shop-Autoricambi mostra la pagina che visualizza tutti gli acquisti effettuati dal cliente fino a quel momento.
2. Exit Condition:
   * Il cliente esce dalla pagina.

**Caso d’uso: storico gestore**



* Nome del caso d’uso: **storico gestore**
* Attori partecipanti: **gestore**
* Entry Condition:
  + Il gestore si trova nella sua pagina riservata.

1. Flusso Eventi:
   * Il gestore pensa di visualizzare lo storico degli acquisti
   * il gestore invoca la funzione **STORICO PRODOTTO CLIENTI** premendo il tasto Area Gestore
   * Shop-Autoricambi mostra la pagina che visualizza tutti gli acquisti effettuati da tutti i clienti fino a quel momento.
2. Exit Condition:
   * Il gestore esce dalla pagina
     1. ***Object model***
        1. ***Data dictionary***

|  |  |
| --- | --- |
| **Acquisto:** Questa classe modella gli acquisti effettuati ossia, i pezzi di ricambio, o prodotti, acquistati. | |
| **codiceAcquisto** | Indica il codice d’acquisto del prodotto acquistato. |
| **descrizione** | Indica la descrizione del prodotto acquistato. |
| **getCodiceAcquisto()** | Restituisce il codice di acquisto del prodotto acquistato. |
| **setCodiceAcquisto(codiceAcquisto)** | Setta il codice di acquisto del prodotto acquistato. |
| **getDescrizione()** | Restituisce la descrizione del prodotto acquistato. |
| **setDescrizione(descrizione)** | Setta la descrizione del prodotto acquistato. |

|  |  |
| --- | --- |
| **Fattura:** Questa classe modella le fatture che vengono omesse nel momento in cui si effettua un acquisto. | |
| **codiceFattura** | Indica il codice della fattura. |
| **data** | Indica la data di rilascio della fattura. |
| **pagamento** | Indica la modalità di pagamento effettuata. |
| **costoTotale** | Indica l’importo totale omesso nel momento in cui si effettua l’acquisto. |
| **getCodiceFattura()** | Restituisce il codice della fattura omessa. |
| **setCodiceFattura(codiceFattura)** | Setta il codice della fattura verrà omessa. |
| **getData()** | Restituisce la data di rilascio della fattura. |
| **setData(data)** | Setta la data di rilascio della fattura. |
| **getPagamento()** | Restituisce la modalità di pagamento utilizzata. |
| **setPagamento(pagamento)** | Setta la modalità di pagamento utilizzata. |
| **getCostoTotale()** | Restituisce l’importo totale della fattura. |
| **setCostoTotale(costoTotale)** | Setta l’importo totale della fattura. |

|  |  |
| --- | --- |
| **Foto:** Questa classe modella le foto che sono associate ai prodotti venduti dal sito mediante il software. | |
| **numFoto** | Identifica univocamente una foto associata ad un prodotto. |
| **path** | Indica il percorso in cui è possibile recuperare la foto. |
| **getNumFoto()** | Restituisce il numero della foto. |
| **setNumFoto(numFoto)** | Setta il numero della foto. |
| **getPath()** | Restituisce il path della foto. |
| **setPath(path)** | Setta il path della foto. |

|  |  |
| --- | --- |
| **IndirizzoPartner:** Questa classe modella gli indirizzi dei partner. | |
| **CAP** | Indica il CAP dell’indirizzo. |
| **via** | Indica la via in cui ha sede il partner. |
| **citta** | Indica la città in cui ha sede il partner. |
| **getCap()** | Restituisce il CAP dell’indirizzo. |
| **setCap(CAP)** | Setta il CAP dell’indirizzo |
| **getVia()** | Restituisce la via in cui ha sede il partner. |
| **setVia()** | Setta la via in cui ha sede il partner. |
| **getCitta()** | Restituisce la città in cui ha sede il partner. |
| **setCitta()** | Setta la città in cui ha sede il partner. |

|  |  |
| --- | --- |
| **IndirizzoUtente:** Questa classe modella gli indirizzi dei partner. | |
| **CAP** | Indica il CAP dell’indirizzo. |
| **via** | Indica la via in cui l’utente risiede. |
| **citta** | Indica la città in cui l’utente risiede. |
| **getCap()** | Restituisce il CAP dell’indirizzo. |
| **setCap(CAP)** | Setta il CAP dell’indirizzo |
| **getVia()** | Restituisce la via in cui l’utente risiede. |
| **setVia()** | Setta la via in cui l’utente risiede. |
| **getCitta()** | Restituisce la città in cui l’utente risiede. |
| **setCitta()** | Setta la città in cuil’utente risiede. |

|  |  |
| --- | --- |
| **Partner:** Questa classe modella i partner, ossia le officine, che possono essere associati a più clienti. | |
| **pIva** | Indica la partita iva dell’officina. |
| **telefono** | Indica il numero di telefono dell’officina. |
| **email** | Indica l’email dell’officina. |
| **logo** | Indica il path in cui si trova logo dell’officina. |
| **getPiva()** | Restituisce la partita iva dell’officina. |
| **setPiva(pIva)** | Setta la partita iva dell’officina. |
| **getTelefono()** | Restituisce il numero di telefono dell’officina. |
| **setTelefono(telefono)** | Setta il numero di telefono dell’officina. |
| **getEmail()** | Restituisce l’email dell’officina. |
| **setEmail(email)** | Setta l’email dell’officina. |
| **getLogo()** | Restituisce il path del logo dell’officina. |
| **setLogo(logo)** | Setta il path del logo dell’officina. |

|  |  |
| --- | --- |
| **Prodotto:** Questa classe modella i prodotti venduti dal sito mediante il software. Per prodotto si intende il generico pezzo di ricambio. | |
| **codicePezzo** | Indica il codice del pezzo di ricambio. |
| **modello** | Indica il modello del pezzo di ricambio. |
| **marchio** | Indica il marchio di fabbrica del pezzo di ricambio. |
| **prezzo** | Indica il prezzo del pezzo di ricambio. |
| **offerta** | Indica se il prodotto è in offerta. |
| **prezzoScontato** | Indica il prezzo scontato del prodotto se esso è in offerta. |
| **descrizione** | Indica la descrizione del prodotto. |
| **quantita** | Indica la disponibilità attuale del pezzo di ricambio. |
| **getCodicePezzo()** | Restituisce il codice del pezzo di ricambio. |
| **setCodicePezzo(codicePezzo)** | Setta il codice del pezzo di ricambio. |
| **getModello()** | Restituisce il modello del pezzo di ricambio. |
| **setModello(modello)** | Setta il modello del pezzo di ricambio. |
| **getMarchio()** | Restituisce il marchio del pezzo di ricambio. |
| **setMarchio(marchio)** | Setta il marchio del pezzo di ricambio. |
| **getPrezzo()** | Restituisce il prezzo del pezzo di ricambio. |
| **setPrezzo(prezzo)** | Setta il prezzo del pezzo di ricambio. |
| **isOfferta()** | Verifica se il prodotto è in offerta. |
| **setOfferta(offerta)** | Restituisce un valore che permette di discriminare tra un prodotto in offerta ed un prodotto non in offerta. |
| **getPrezzoScontato()** | Restituisce il prezzo scontato del prodotto se esso è in offerta. |
| **setPrezzoScontato(prezzoScontato)** | Setta il prezzo scontato del prodotto se esso è in offerta. |
| **getDescrizione()** | Restituisce la descrizione del prodotto. |
| **setDescrizione(descrizione)** | Setta la descrizione del prodotto. |
| **getQuantita()** | Restituisce la quantità del prodotto. |
| **setQuantita(quantita)** | Setta la quantità del prodotto. |

|  |  |
| --- | --- |
| **Utente:** Questa classe modella le persone che si registrano al sito (clienti) e le persone che assolvono il ruolo di gestore del sito. | |
| **nome** | Indica il nome del cliente o del gestore. |
| **cognome** | Indica il cognome del cliente o del gestore. |
| **dataNascita** | Indica la data di nascita del cliente o del gestore. |
| **codiceFiscale** | Indica il codice fiscale del cliente o del gestore. |
| **telefono** | Indica il numero di telefono del cliente o del gestore. |
| **cellulare** | Indica il numero di cellulare della persona o del gestore. |
| **password** | Indica la password, utilizzata per il login, del cliente o del gestore. |
| **email** | Indica l’email, utilizzata per il login, del cliente o del gestore. |
| **amministratore** | Indica se l’utente è un gestore oppure un cliente. |
| **getNome()** | Restituisce il nome del cliente o del gestore. |
| **setNome(nome)** | Setta il nome del cliente o del gestore. |
| **getCognome()** | Restituisce il cognome del cliente o del gestore. |
| **setCognome(cognome)** | Setta il cognome del cliente o del gestore. |
| **getDataNascita()** | Restituisce la data di nascita del cliente o del gestore. |
| **setDataNascita(dataNascita)** | Setta la data di nascita del cliente o del gestore. |
| **getCodiceFiscale()** | Restituisce il codice fiscale del cliente o del gestore. |
| **setCodiceFiscale(codiceFiscale)** | Setta il codice fiscale del cliente o del gestore. |
| **getTelefono()** | Restituisce il numero di telefono del cliente o del gestore. |
| **setTelefono(telefono)** | Setta il numero di telefono del cliente o del gestore. |
| **getCellulare()** | Restituisce il numero di cellulare del cliente o del gestore. |
| **setCellulare(cellulare)** | Setta il numero di cellulare del cliente o del gestore. |
| **getPassword()** | Restituisce la password del cliente o del gestore. |
| **setPassword(password)** | Setta la password del cliente o del gestore. |
| **getEmail()** | Restituisce l’email del cliente o del gestore. |
| **setEmail()** | Setta l’email del cliente o del gestore. |
| **isAmministratore()** | Restituisce un valore che permette di discriminare tra un utente che sia cliente e un utente che sia gestore. |
| **setAmministratore(amministatore)** | Fa diventare un cliente in un gestore, viceversa, un gestore in un cliente. |

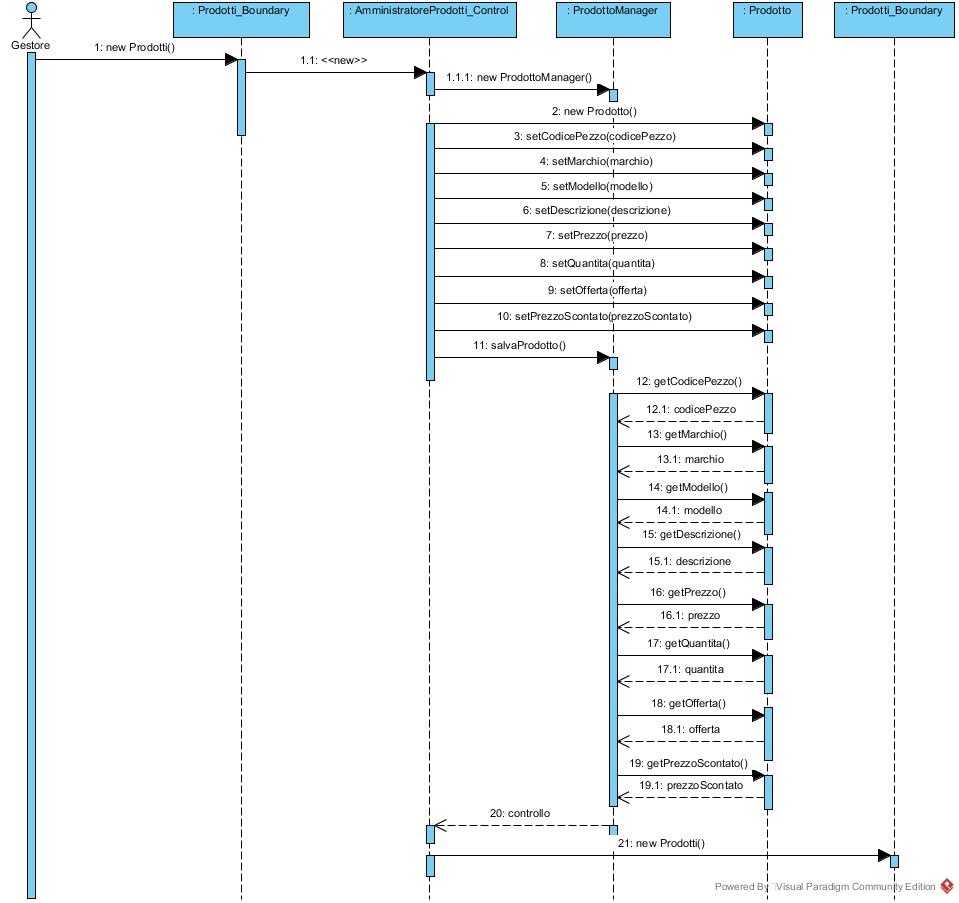
* + - 1. ***Class diagram***



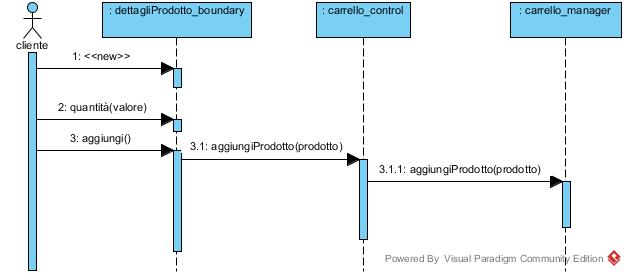
* + 1. ***Dynamic model***

**SEQUENCE DIAGRAMS**

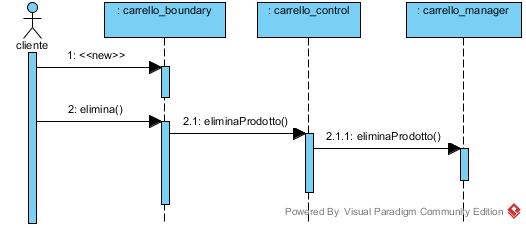
* **Carica prodotto**

****

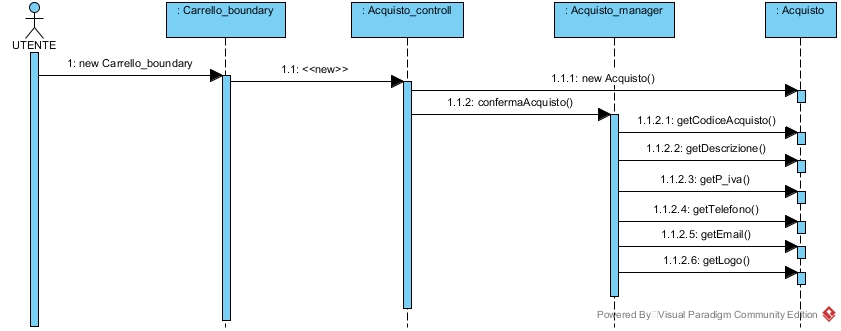
* **Aggiungi prodotto al carrello**

****

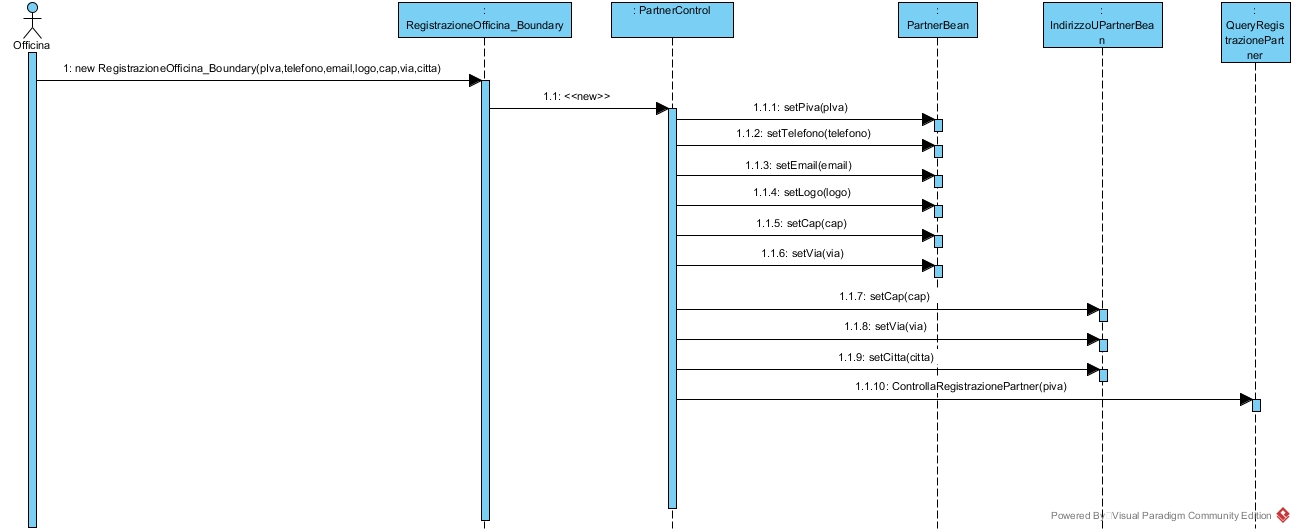
* **Eliminazione di un prodotto dal carrello**

****

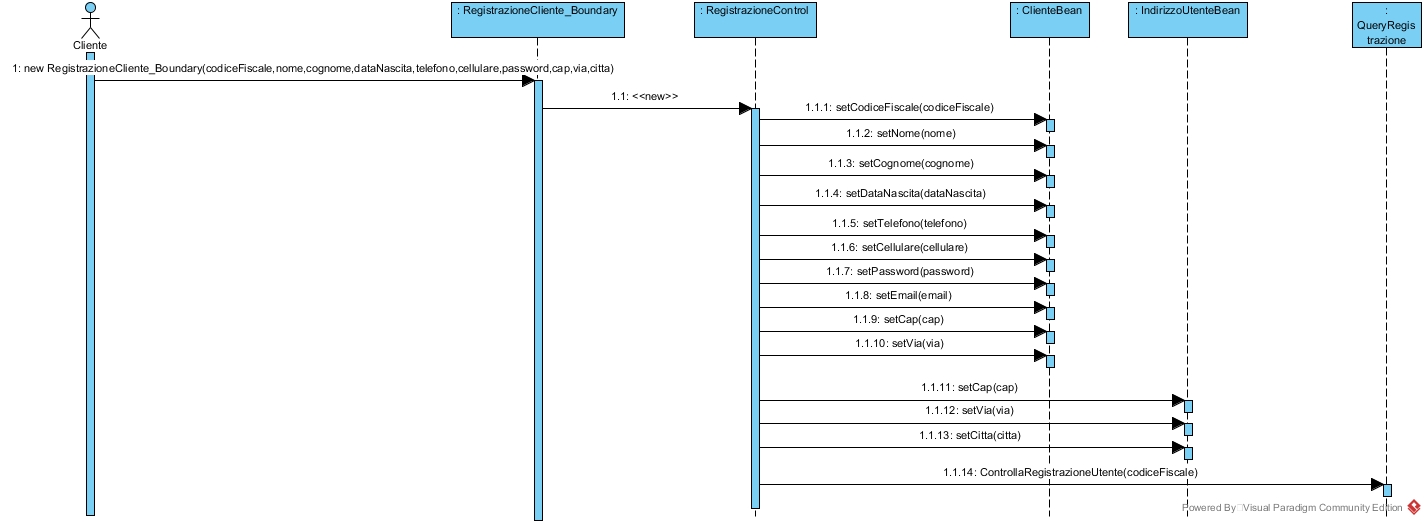
* **Acquisto di un prodotto**

****

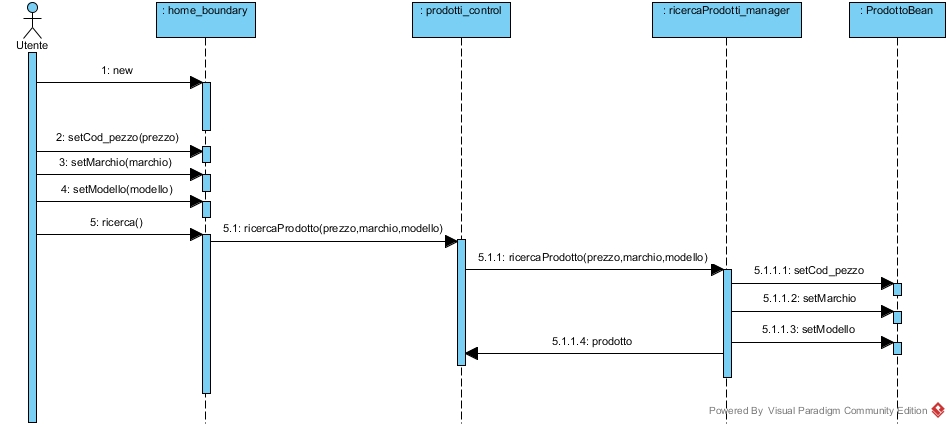
* **Registrazione officina**

****

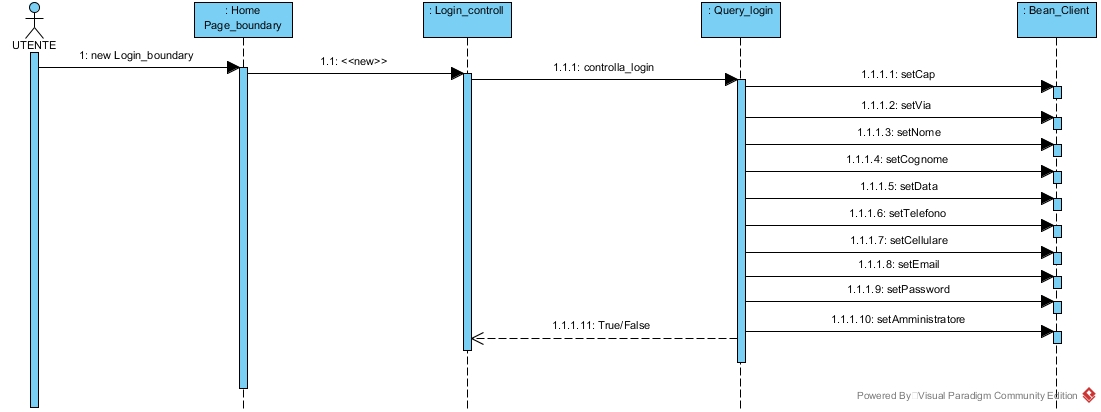
* **Registrazione utente**

****

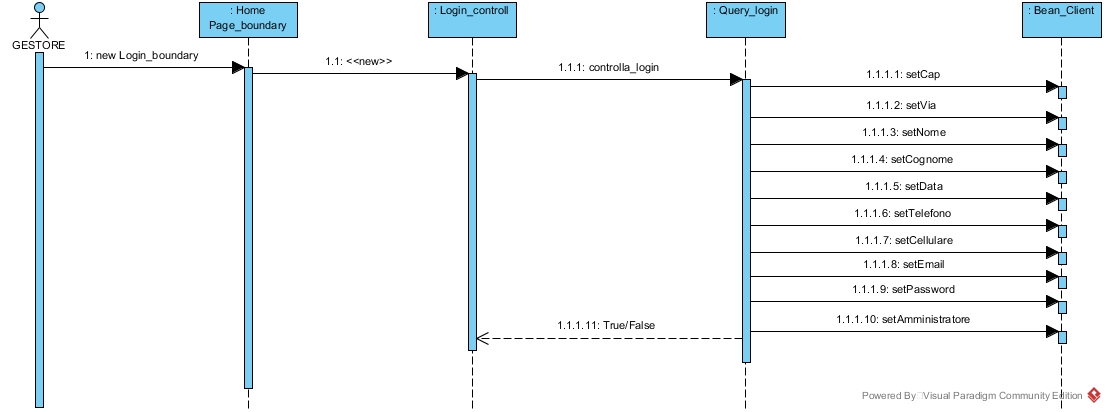
* **Ricerca prodotto**

****

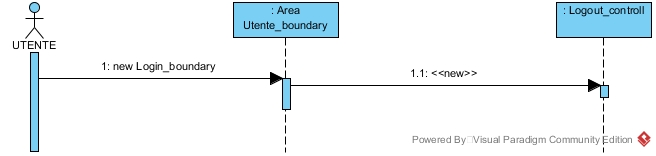
* **Login utente**

****

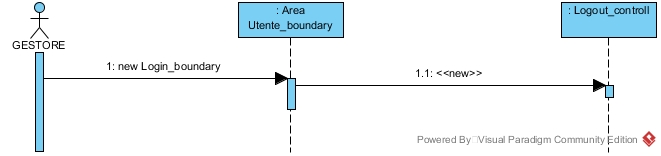
* **Login gestore**

****

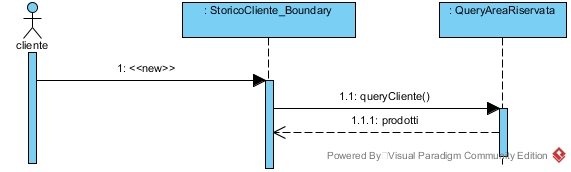
* **Logout utente**

****

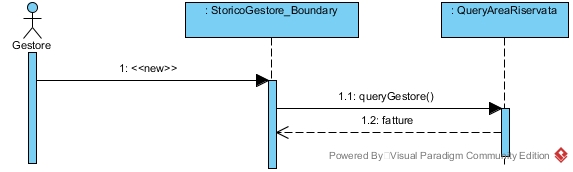
* **Logout gestore**

****

* **Storico cliente**

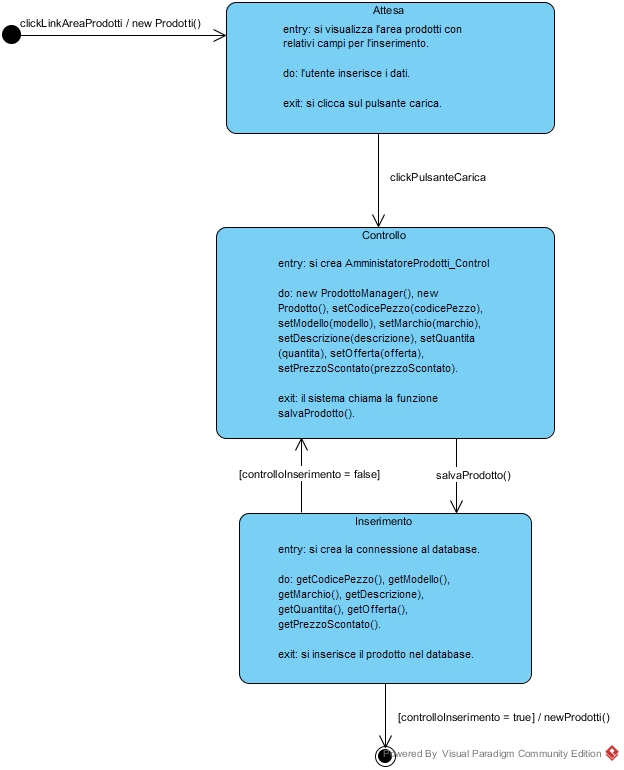
****

* **Storico gestore**

****

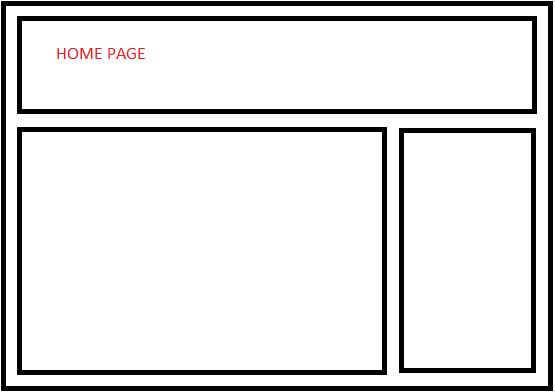
**STATE-CHART DIAGRAMS**

* **Carica Prodotto**

****

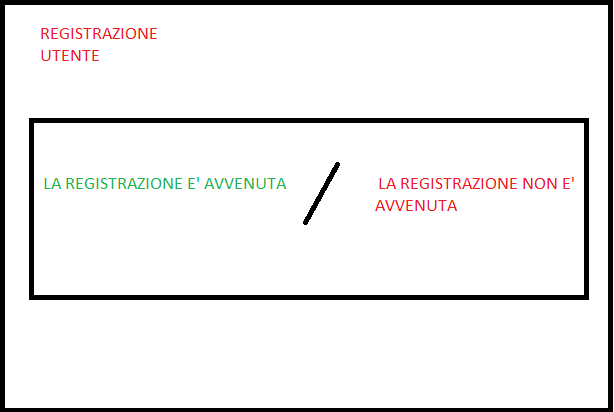
* + 1. **User interface-navigational paths and screen mock-ups**

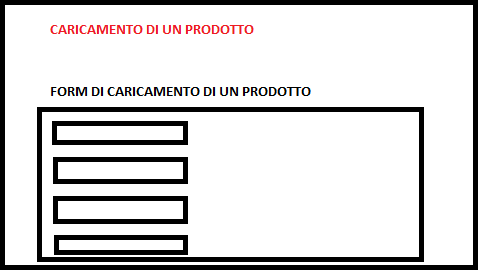
**USER INTERFACE**

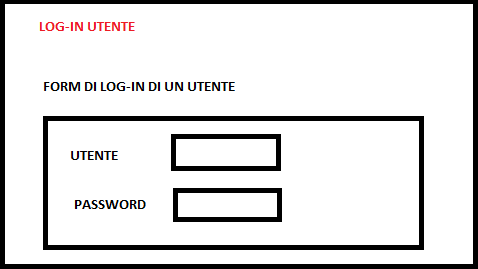




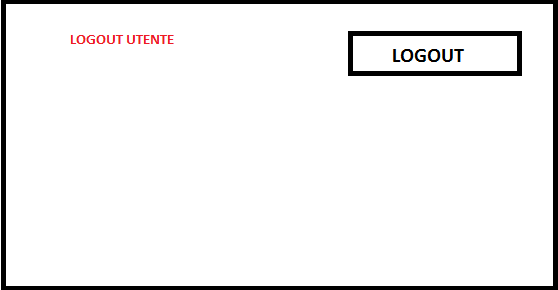








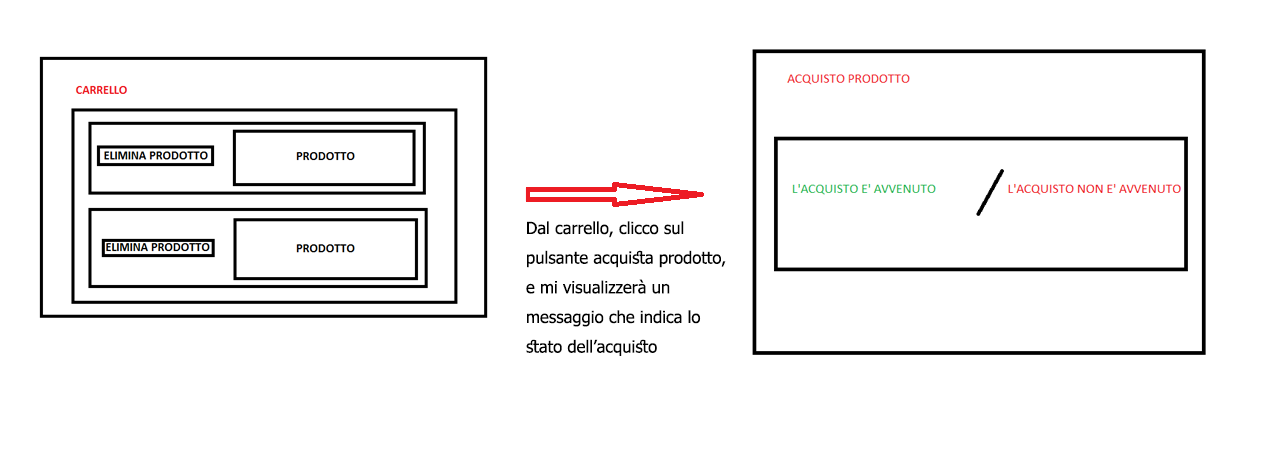


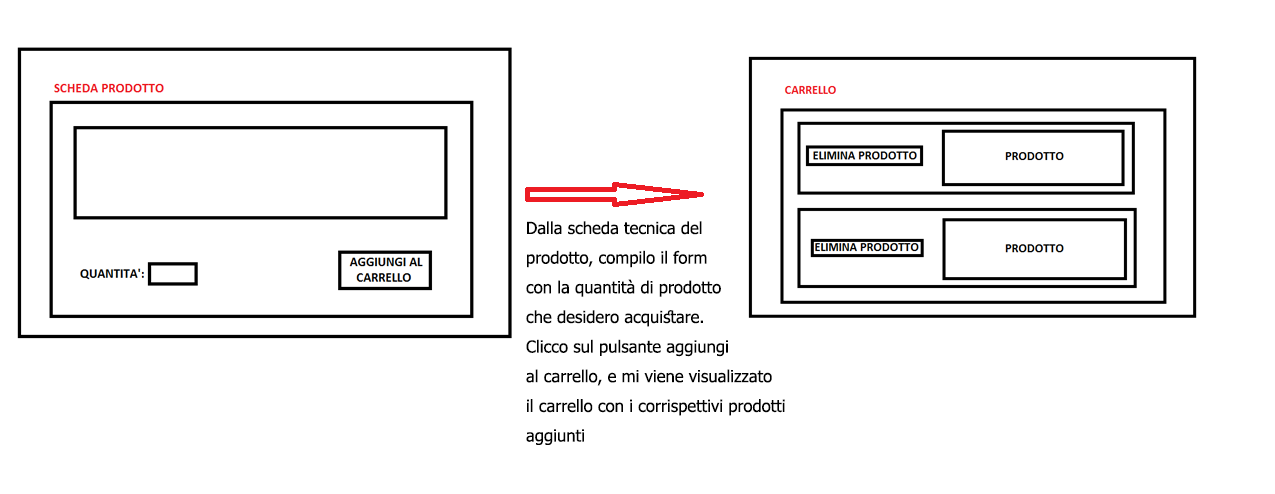


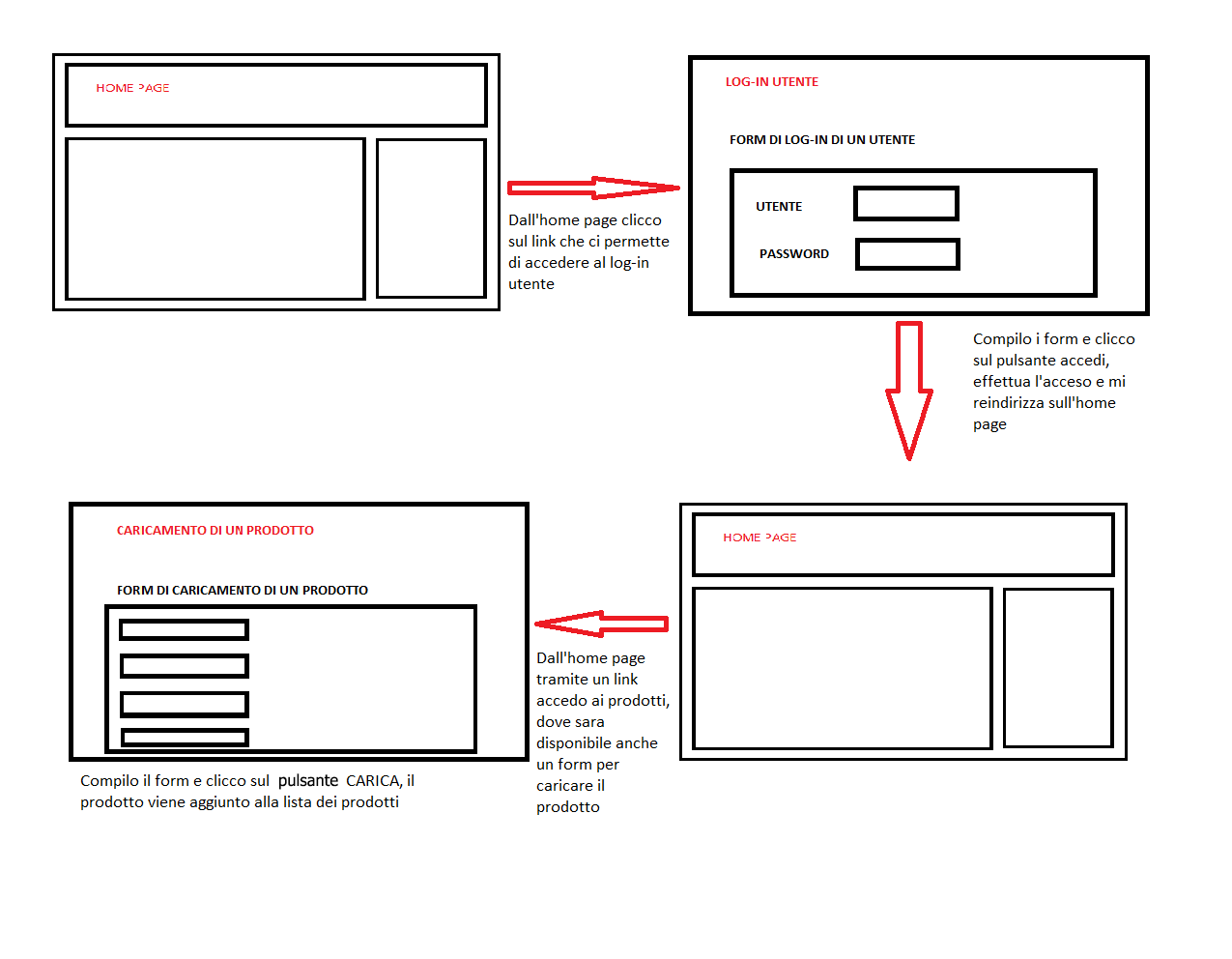


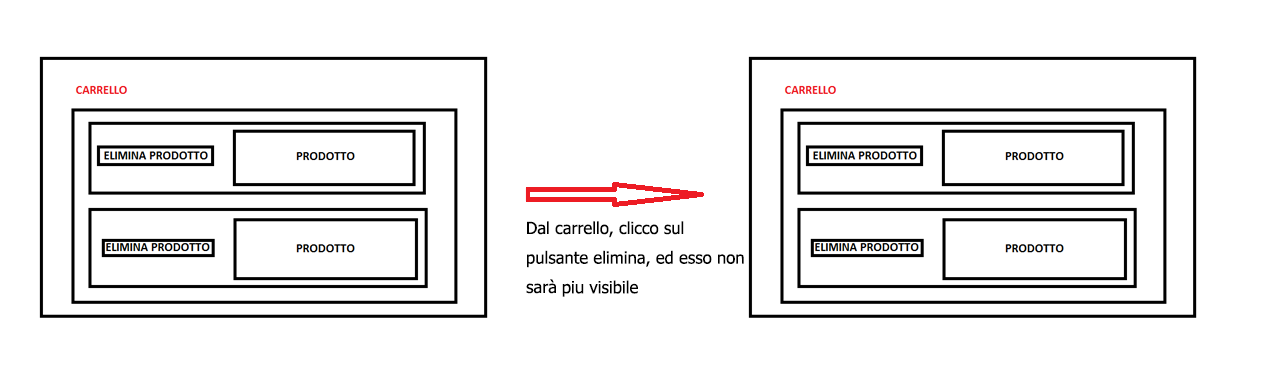


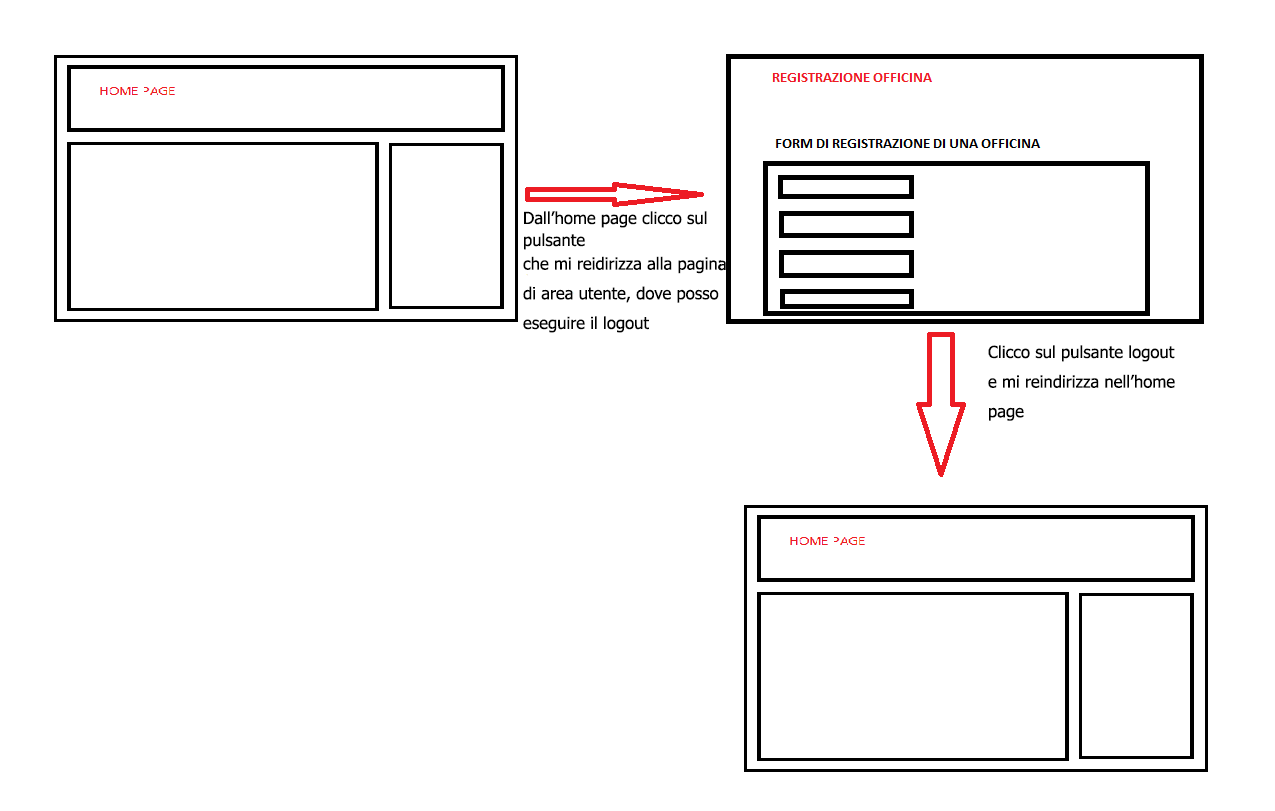
**NAVIGATIONAL PATH**

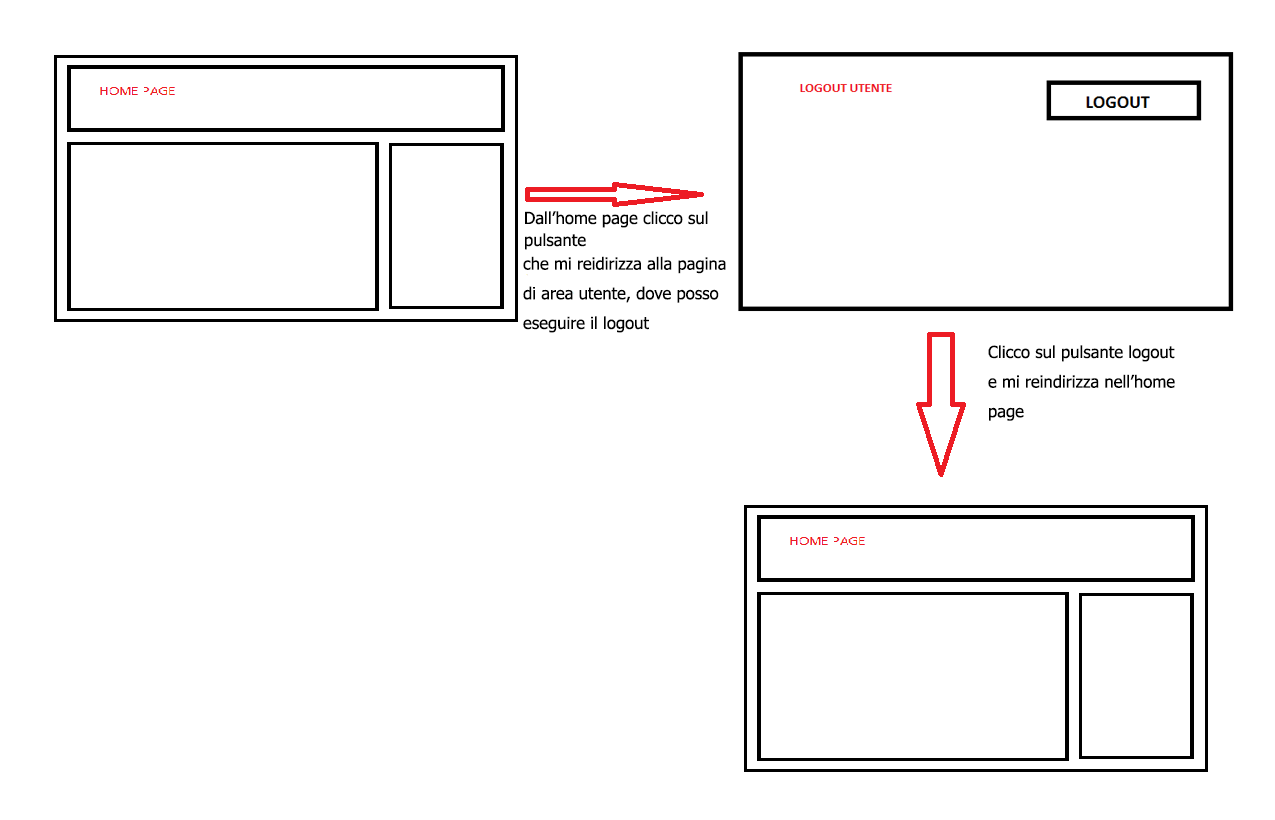


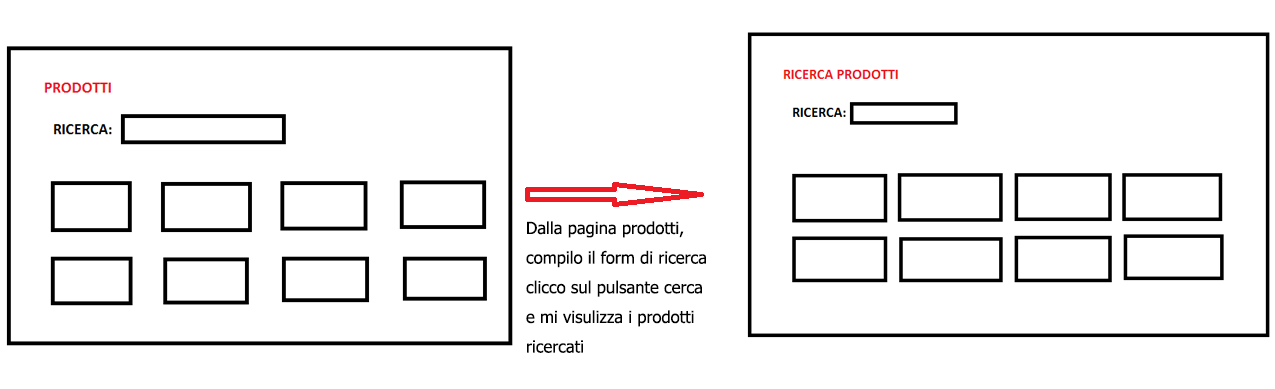
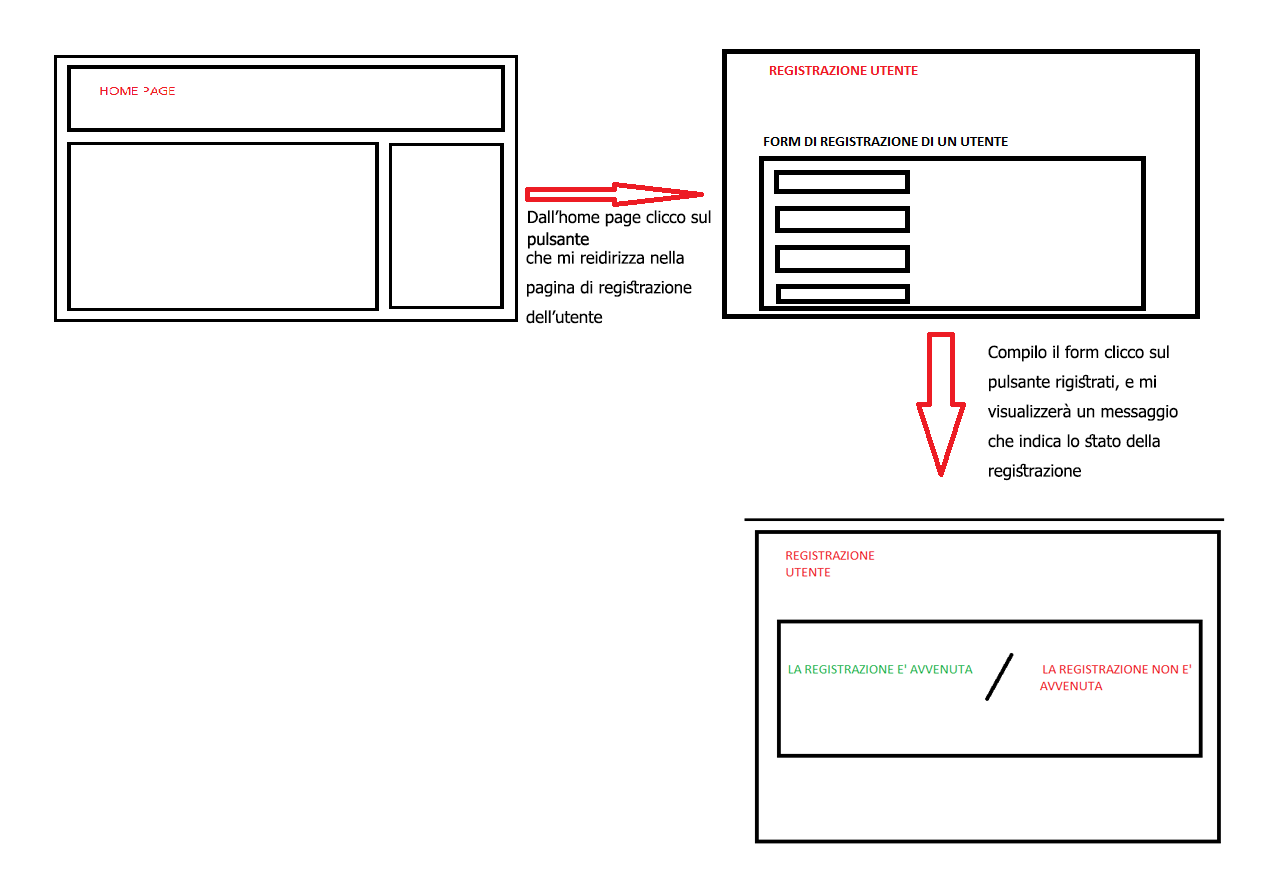
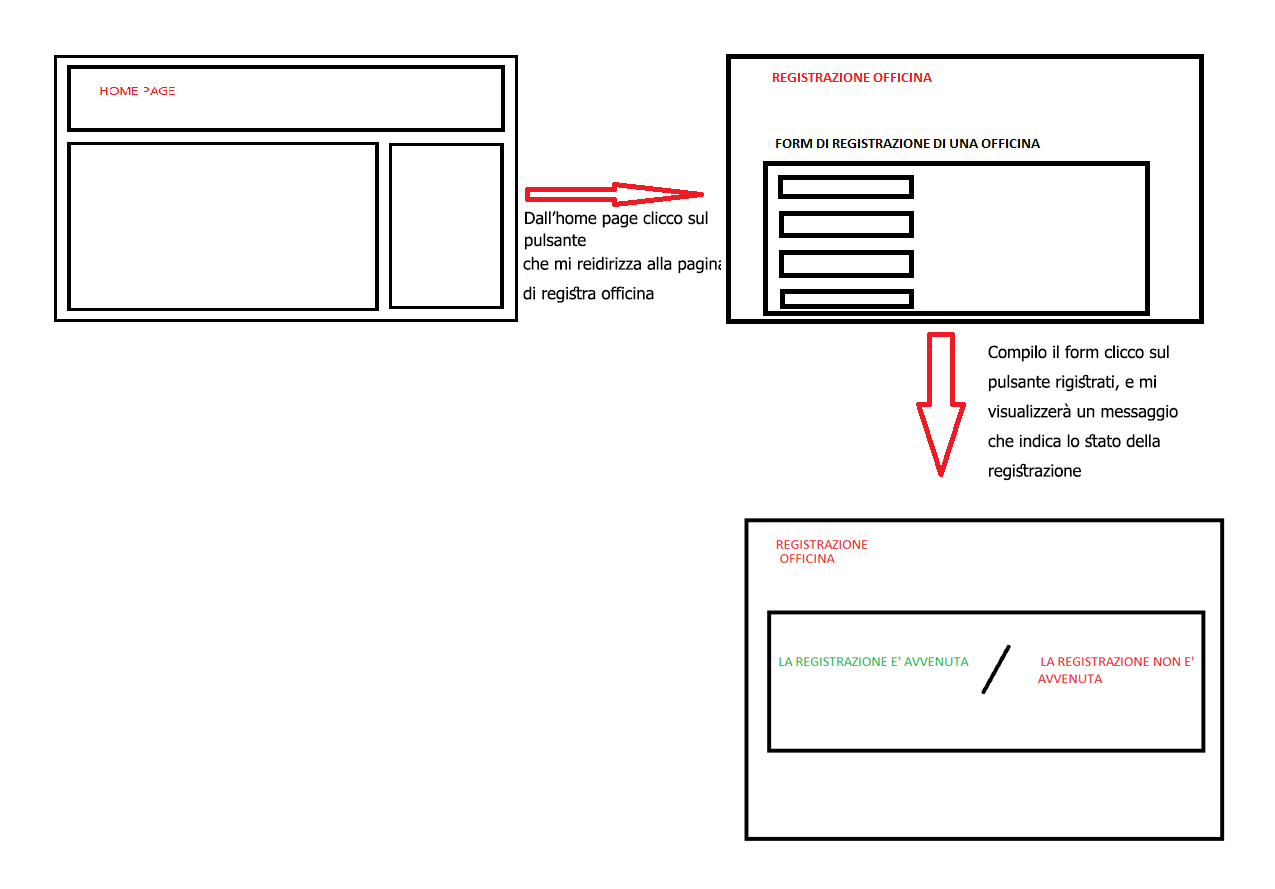


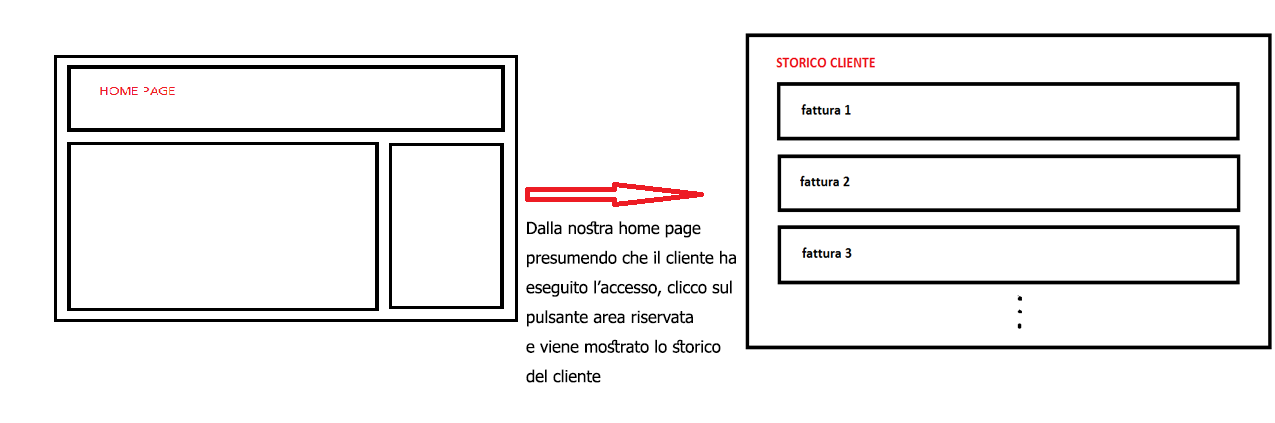


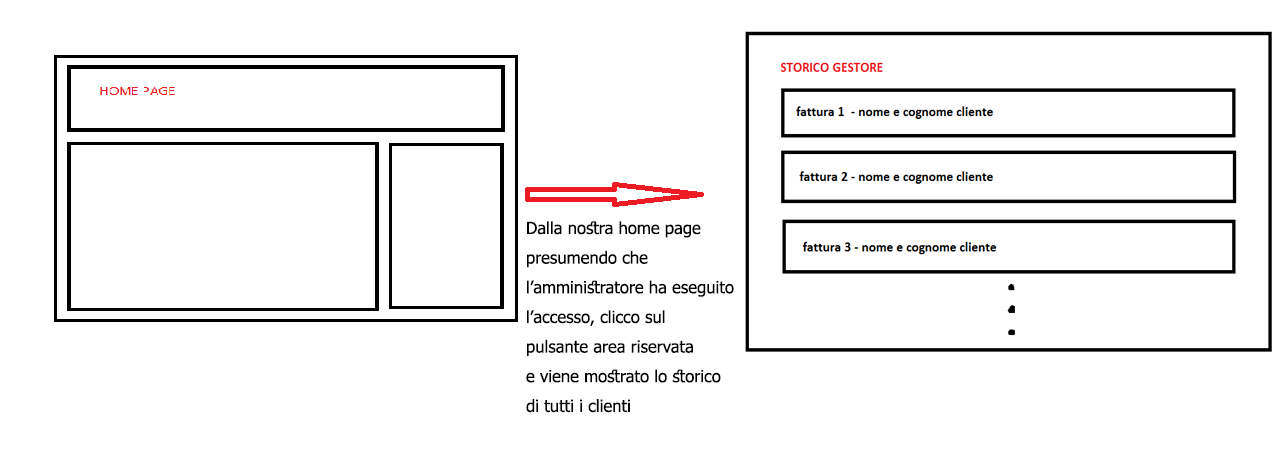












1. **GLOSSARY**

|  |  |  |
| --- | --- | --- |
| **TERMINE** | **DESCRIZIONE** | **SINONIMI** |
| **Utente** | Persona che si registra al sito utilizzato per la vendita dei prodotti di ricambio online.  Persona che assolve il ruolo di Gestore del sito di vendita di prodotti di ricambio online. | Cliente  Gestore |
| **Partner** | Luogo di riferimento per il montaggio del prodotto. | Officina |
| **Prodotto** | Pezzi di ricambio a disposizione per la vendita | ---------- |
| **Foto** | Immagine associata ai determinati prodotti in vendita sul sito. | ---------- |
| **Acquisto** | Rappresenta il prodotto che è stato acquistato dal cliente | ---------- |
| **Fattura** | Documento con le indicazioni della merce fornita, dell'ammontare dell'importo e delle relative condizioni di pagamento, che il venditore trasmette al compratore al momento della spedizione della merce ordinata. | ---------- |