

**Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

**HomeDecore**  
**Requirements Analysis Document**  
**Versione 1.6**



*HomeDecore*

Data: 25/11/2024

Progetto: HomeDecore	Versione: 1.6
Documento: Requirements Analysis	Data: 25/11/2024

### Coordinatore del progetto:

Nome	Matricola

### Partecipanti:

Nome	Matricola
Alfieri Riccardo	0512116533
Cammarota Lucageneroso	0512116941
Fasolino Pietro	0512116473
Marino Michele Graziano	0512117109

Scritto da:	
-------------	--

## Revision History

Data	Versione	Descrizione	Autore
25/10/2024	1.0	Descrizione del sistema in termini di requisiti funzionali e non funzionali. Prima stesura dei diagrammi di casi d'uso.	Full team
10/11/2024	1.1	Sviluppo di: - object model tramite class diagram - dynamic model tramite sequence diagram, activity diagram e state machine diagram - mock-ups	Full team
17/11/2024	1.2	<ul style="list-style-type: none"> <li>Aggiunta UC 1.1, UC 1.2, UC 2.1</li> <li>Aggiunta SD 1.1, SD 1.2, SD 2.1</li> <li>Modifica UC 1 e UC 2</li> </ul>	Lucageneroso Cammarota
18/11/2024	1.3	<ul style="list-style-type: none"> <li>Modifica UC 2</li> <li>Aggiunta UC 2.2</li> <li>Aggiunta UC 1.3</li> <li>Aggiunta SD 2.1</li> <li>Modifica SD 2</li> </ul>	Lucageneroso Cammarota
21/11/2024	1.4	<ul style="list-style-type: none"> <li>Modifica UC 3, 4, 5, 6, 7 e aggiunta UC per flussi alternativi</li> <li>Modifica Class Diagram</li> </ul>	Alfieri Riccardo
22/11/2024	1.5	<ul style="list-style-type: none"> <li>Modifica Class Diagram</li> <li>Modifica dizionario dei dati dynamic model</li> </ul>	Lucageneroso Cammarota
25/11/2024	1.6	<ul style="list-style-type: none"> <li>Consistenza del dizionario dei dati</li> </ul>	Lucageneroso Cammarota

		Ingegneria del Software	Pagina 2 di 14

# Indice

## Requirements Analysis Document

### Sommario

Introduzione .....	4
1.1. Purpose of system.....	4
1.2. Scope of the system.....	4
1.3. Objectives and success criteria of the project .....	5
Current system.....	6
Proposed system .....	6
3.1. Functional requirements.....	6
3.2. Non functional requirements .....	8
3.2.1. Usability .....	9
3.2.2. Reliability .....	9
3.2.3. Performance .....	9
3.2.4. Supportability .....	9
3.2.5. Implementation .....	10
3.2.6. Interface .....	10
3.2.7. Packaging.....	11
3.2.8. Legal.....	11
3.3. System models .....	12
3.3.1. Scenarios.....	12
3.3.2. Use case model.....	16
3.3.3. Object Model .....	35
3.3.4. Dynamic Model.....	38
3.3.5. User interface – navigational paths and screen mock-ups .....	51
4. Glossario.....	53

# Introduzione

## *1.1. Purpose of system*

HomeDecore è un'azienda specializzata nella vendita online di articoli per la casa, tra cui mobili, decorazioni, utensili da cucina e articoli per la pulizia. L'azienda serve una clientela diversificata, composta sia da acquirenti privati che da rivenditori. Con l'obiettivo di espandere il proprio mercato e migliorare l'esperienza utente, HomeDecore ha identificato la necessità di rinnovare la propria piattaforma e-commerce per renderla più flessibile, scalabile e user-friendly.

## *1.2. Scope of the system*

Il sistema **HomeDecor** si propone di sviluppare una piattaforma e-commerce per la vendita di articoli per la casa, con un focus su un'esperienza utente fluida e intuitiva. Le funzionalità principali includono la gestione del catalogo prodotti, l'interazione utente tramite recensioni e valutazioni, e l'acquisto tramite un carrello virtuale. Gli utenti potranno registrarsi, gestire i propri profili, cercare prodotti per categoria e procedere all'acquisto mediante un sistema di checkout integrato con piattaforme di pagamento online. Gli amministratori avranno accesso a strumenti di gestione avanzata per l'aggiornamento e la manutenzione del catalogo prodotti, e la gestione degli utenti registrati.

Il sistema garantirà un'interfaccia responsive compatibile con i principali dispositivi e browser, e implementerà meccanismi di sicurezza per la protezione dei dati degli utenti, come il hashing delle password e la prevenzione contro vulnerabilità comuni (es. SQL injection). Il progetto si baserà su un'architettura web utilizzando Java, JSP/Servlet, MySQL e sarà eseguito su un server Tomcat.

Le esclusioni comprendono il supporto multilingua e una gestione avanzata del magazzino, che verranno considerati per futuri sviluppi. L'obiettivo principale di questa versione è fornire una piattaforma e-commerce funzionante con funzionalità di base, incentrata sulla navigazione dei prodotti e un sistema di recensioni efficace, mirato ad aumentare le vendite e a migliorare la reputazione del brand attraverso il feedback dei clienti.

### 1.3. Objectives and success criteria of the project

#### Objectives:

L'obiettivo principale del progetto **HomeDecor** è quello di creare una piattaforma e-commerce intuitiva e funzionale per la vendita di articoli per la casa, che sia facilmente navigabile e offra agli utenti un'esperienza di acquisto fluida. Gli obiettivi specifici del progetto sono:

1. **Facilità d'uso e navigazione:** Implementare un'interfaccia utente chiara e accessibile, ottimizzata per la ricerca e la visualizzazione dei prodotti attraverso categorie.
2. **Sistema di recensioni e valutazioni:** Consentire agli utenti di lasciare recensioni e valutazioni sui prodotti, creando un meccanismo di feedback che influenzi la fiducia degli acquirenti futuri.
3. **Gestione del carrello e processo di acquisto:** Sviluppare un sistema di gestione del carrello che consenta agli utenti di aggiungere prodotti, modificarli e procedere al checkout in maniera semplice.
4. **Integrazione di metodi di pagamento:** Integrare un sistema di pagamento sicuro che consenta agli utenti di completare gli acquisti utilizzando diversi metodi di pagamento online.
5. **Sicurezza e protezione dei dati:** Implementare misure di sicurezza come il hashing delle password, la crittografia dei dati sensibili e la protezione contro le vulnerabilità più comuni.
6. **Gestione amministrativa:** Consentire agli amministratori di gestire il catalogo prodotti e gli utenti registrati attraverso un'interfaccia dedicata.

#### Success Criteria:

Il successo del progetto sarà misurato attraverso i seguenti criteri:

1. **Usabilità e navigazione intuitiva:** Il sistema sarà considerato di successo se gli utenti troveranno facile navigare tra le categorie di prodotti e completare il processo di acquisto con pochi passaggi.
2. **Funzionamento completo delle funzionalità chiave:** Il sistema sarà ritenuto completo quando tutte le funzionalità principali, come la gestione del carrello, le recensioni e le valutazioni, e il checkout, saranno pienamente operative e testate.
3. **Sicurezza e protezione dati:** Il progetto avrà successo se saranno adottate con successo misure di sicurezza per proteggere i dati degli utenti e prevenire vulnerabilità come SQL injection e attacchi di brute force.
4. **Performance e scalabilità:** Il sistema dovrà garantire buone prestazioni anche con un elevato numero di utenti e prodotti, dimostrando la sua capacità di crescere con la piattaforma.
5. **Feedback positivo degli utenti:** Un altro indicatore di successo sarà il feedback positivo da parte degli utenti finali, che valuteranno l'esperienza d'uso del sito, con particolare attenzione alla navigazione, alla facilità di acquisto e alla gestione del

## Current system

La piattaforma attuale presenta diverse limitazioni che influiscono negativamente sull'efficienza operativa e sull'esperienza d'acquisto dei clienti. Il progetto mira a risolvere queste problematiche, operando sulla gestione del carrello, il processo di acquisto e l'esperienza utente. Inoltre, si intende ottimizzare l'usabilità del sistema su dispositivi mobili e desktop. Attualmente, la comunicazione tra i clienti e l'assistenza è completamente assente. Il sistema non prevede un canale di comunicazione diretto e in tempo reale per risolvere problemi legati agli ordini, ai prodotti o alla gestione. Il sistema attuale mostra solo un riepilogo statico dei prodotti. Il checkout è lungo e frammentato. HomeDecore non consente agli utenti di lasciare recensioni sui prodotti, limitando la possibilità di condivisione di feedback da parte dei clienti. Inoltre, in caso di esaurimento scorte di un determinato prodotto, il gestore del magazzino non può sfruttare la piattaforma per chiedere un rifornimento al fornitore.

## Proposed system

Al fine di migliorare il supporto fornito ai clienti e la comunicazione interna tra i gestori del sistema, verrà introdotta una chat bidirezionale che permetterà agli utenti di comunicare in tempo reale con il supporto clienti o con gli amministratori, migliorando la risoluzione delle richieste e incrementando la soddisfazione dei clienti. Le nuove funzionalità consentiranno una gestione dinamica degli acquisti, offrendo agli utenti la possibilità di aggiungere, rimuovere e modificare i prodotti nel carrello in modo interattivo e in tempo reale. 4 Il nuovo sistema implementerà un processo di acquisto semplificato che riduce il numero di passaggi necessari e permette agli utenti guest di completare l'acquisto con maggiore flessibilità. Infine, il sistema offrirà all'utente la possibilità di recensire la qualità di un prodotto, esprimendo la propria valutazione in termini numerici e sotto forma di commento testuale, migliorando la trasparenza e incrementando la fiducia degli acquirenti.

### ***3.1. Functional requirements***

## **Utente**

- ☐ RF1: Il sistema deve permettere agli utenti di accedere in qualsiasi momento alla sezione "Carrello" per visualizzare gli articoli selezionati.
- ☐ RF2: Il sistema deve mostrare il nome, la quantità e il prezzo unitario di ciascun articolo nel carrello.
- ☐ RF3: Il sistema deve consentire agli utenti di modificare la quantità di un prodotto o rimuoverlo dal carrello.
- ☐ RF4: Il sistema deve aggiornare automaticamente il totale complessivo quando vengono apportate modifiche al carrello.
- ☐ RF5: Il sistema deve permettere agli utenti di salvare il carrello e ritornarci in un secondo momento per completare l'acquisto.
- ☐ RF6: Il sistema deve guidare l'utente nel processo di checkout, richiedendo le informazioni di spedizione e permettendo la selezione di un metodo di spedizione.
- ☐ RF7: Il sistema deve consentire agli utenti di scegliere un metodo di pagamento (es. carta di credito, PayPal, bonifico).
- ☐ RF8: Il sistema deve generare un riepilogo dell'ordine che mostri il costo totale, incluse le spese di spedizione e le tasse.
- ☐ RF9: Il sistema deve notificare all'utente che il suo ordine è stato confermato.
- ☐ RF10: Il sistema deve permettere agli utenti di lasciare una recensione sui prodotti acquistati dopo la consegna.
- ☐ RF11: Il sistema deve consentire agli utenti di assegnare un punteggio al prodotto (es. da 1 a 5 stelle).
- ☐ RF12: Il sistema deve pubblicare la recensione e renderla visibile sulla pagina del prodotto, dopo la verifica.

## **Gestore del magazzino**

- ☐ RF13: Il sistema deve permettere a qualsiasi gestore di visualizzare tutti i prodotti del suo magazzino.
- ☐ RF14: Il sistema deve permettere di modificare nome dei prodotti del magazzino
- ☐ RF15: Il sistema deve permettere di modificare descrizione dei prodotti del magazzino
- ☐ RF16: Il sistema deve permettere di modificare prezzo dei prodotti del magazzino
- ☐ RF17: Il sistema deve permettere di aggiungere nuovi prodotti al magazzino
- ☐ RF18: Il sistema deve permettere ai gestori di richiedere ai fornitori dei prodotti
- ☐ RF19: Il sistema deve permettere di assegnare gli ordini a un gestore
- ☐ RF20: Il sistema deve permettere di modificare le informazioni del proprio magazzino

## **Gestore degli ordini**



- RF21: Il sistema deve permettere a qualsiasi gestore di visualizzare tutti gli ordini a lui assegnati
- RF22: Ogni gestore di ordini ha la capacità di modificare lo stato di un ordine

### **Fornitore**

- RF23: Il sistema deve permettere a qualsiasi fornitore di visualizzare tutte le richieste di prodotti effettuati da gestori dei magazzini
  - RF24: Il sistema deve garantire al fornitore la scelta di soddisfare o meno le richieste di prodotti
- 11
- RF25: Il sistema deve dare la possibilità al fornitore di aggiungere o eliminare prodotti che può fornire
- requisiti comuni a tutti gli utenti
- RF26: Il sistema deve permettere a qualsiasi utente di visualizzare il proprio profilo
  - RF27: Il sistema deve permettere ad ogni utente di modificare le informazioni sul proprio profilo
  - RF28: Il sistema deve offrire la possibilità ad un utente di visitare il profilo di un qualsiasi altro utente, indipendentemente dalla categoria

## ***3.2. Non functional requirements***

### 3.2.1. Usability

- **Guida visiva:** Ogni schermata deve fornire indicazioni e messaggi contestuali per guidare l'utente nelle operazioni (es. completamento di un ordine o scrittura di una recensione).
- **Validazione form:** Tutti i campi di input devono fornire messaggi chiari sugli errori (es. "Inserire un numero valido") e suggerimenti per correggerli.
- **Navigazione strutturata:** Il menu di navigazione deve essere accessibile in ogni pagina e permettere il ritorno rapido alla homepage o al carrello con un massimo di due clic.
- **Accesso rapido:** Gli utenti registrati devono poter accedere direttamente alle sezioni chiave (catalogo, carrello, ordini) tramite scorciatoie visibili nella dashboard personale.

### 3.2.2. Reliability

- **Alta disponibilità:** Il sistema deve essere disponibile per almeno il **99.9% del tempo** durante l'anno.
- **Resilienza:** In caso di guasti, il sistema deve essere in grado di riprendersi automaticamente o notificare l'errore senza perdita di dati importanti.
- **Backup:** Il sistema deve supportare backup regolari dei dati per evitare la perdita di informazioni in caso di disastro.

### 3.2.3. Performance

- **Tempo di risposta:** Tutte le pagine principali (catalogo, carrello, checkout) devono essere caricate in un massimo di 2 secondi per connessioni standard (fibra 100 Mbps).
- **Throughput:** Il sistema deve essere in grado di elaborare almeno 1000 richieste simultanee in condizioni di carico normale.
- **Scalabilità:** Durante i picchi stagionali, il sistema deve mantenere il funzionamento con almeno il 200% del carico normale, senza degradazione delle prestazioni superiore al 10%.

### 3.2.4. Supportability

- **Manutenzione modulare:** Il codice deve essere organizzato in moduli indipendenti per ridurre il tempo di debug a massimo 30 minuti per localizzare un problema.
- **Adattabilita':** Il sistema deve consentire l'integrazione di nuovi metodi di pagamento con una modifica massima di 3 file sorgente.
- **Documentazione tecnica:** Ogni modulo del sistema deve essere accompagnato da una descrizione dettagliata delle sue responsabilita' e della sua struttura.
- **Compatibilita' futura:** L'architettura deve permettere l'aggiornamento delle tecnologie di base (es. passaggio a una nuova versione di Java o Tomcat) senza richiedere modifiche strutturali.
- **Robustezza- Gestione input errati:** Tutti gli input devono essere validati lato server e lato client, con messaggi di errore specifici per ogni problema (es. formato errato, campo obbligatorio mancante).
- **Resilienza agli attacchi:** Il sistema deve essere protetto contro attacchi comuni (es. SQL Injection, Cross-Site Scripting) tramite sanitizzazione degli input e uso di query parametrizzate.

## VINCOLI

### 3.2.5. Implementation

- **Tecnologie:** Il sistema deve essere implementato utilizzando le tecnologie Java EE, MySQL per il database e Tomcat come server applicativo. Devono essere utilizzati **JSP** per le pagine dinamiche e **CDI** per la gestione delle dipendenze.
- **Ambiente di sviluppo:** Gli sviluppatori devono utilizzare IntelliJ IDEA come ambiente di sviluppo integrato (IDE). Il sistema di versionamento scelto è **Git**, con repository ospitato su **GitHub**.

### 3.2.6. Interface

- **Interfaccia utente:** Il sistema deve fornire un'interfaccia utente coerente, basata su standard moderni di web design, con elementi di **UI/UX** pensati per facilitare la navigazione e l'interazione.
- **Interfaccia di integrazione:** Il sistema deve fornire API REST per consentire l'integrazione con applicazioni di terze parti come sistemi di pagamento e servizi di spedizione.
- **Browser compatibility:** L'interfaccia utente deve essere compatibile con i principali browser (Chrome, Firefox, Edge, Safari) e deve supportare le versioni più recenti di questi.

### 3.2.7. Packaging

- **Distribuzione:** Il sistema deve essere distribuito come un **file .war** (Web Application Archive) pronto per il deployment su un server Tomcat. Deve essere incluso un file di configurazione per automatizzare il processo di deployment.
- **Installazione:** Deve essere disponibile una guida dettagliata per il setup e la configurazione del sistema, che includa anche le istruzioni per il database e il server applicativo.

### 3.2.8. Legal

- **Conformità GDPR:** Il sistema deve essere conforme alle normative **GDPR** in materia di protezione dei dati personali. Devono essere implementati meccanismi per il consenso esplicito degli utenti riguardo al trattamento dei loro dati.
- **Proprietà intellettuale:** Deve essere garantito che tutti i contenuti, inclusi immagini e descrizioni, non violino diritti di proprietà intellettuale o copyright di terze parti.
- **Termini di servizio:** Devono essere disponibili i termini di servizio e la privacy policy in conformità con le normative nazionali ed europee.

### 3.3. *System models*

#### 3.3.1. Scenarios

##### **CLIENTE**

##### **Visualizzazione prodotti disponibili**

Luca è alla ricerca di un e-commerce che possa soddisfare i suoi desideri: acquistare un articolo per la sua nuova casa a un prezzo accessibile, e soprattutto in modo facile e veloce. Dopo un'accurata ricerca, decide di accedere alla sezione prodotti di HomeDecore, dove visualizza una vasta gamma di articoli per la casa.

Notando la quantità di opzioni disponibili, Luca decide di affinare la sua ricerca applicando dei filtri. Seleziona la categoria desiderata e restringe ulteriormente i risultati impostando un range di prezzo che rispetti il suo budget.

##### **Visualizzazione singolo prodotto**

Grazie ai filtri applicati, Luca trova subito gli articoli che corrispondono alle sue esigenze. Una volta selezionato il prodotto, il sistema gli mostra informazioni dettagliate e offre la possibilità di aggiungere l'articolo alla lista dei desideri o direttamente al carrello specificando la quantità che desidera.

##### **Visualizzazione del carrello**

Dopo aver trascorso del tempo a navigare tra le categorie di prodotti sul sito **HomeDecore**, Carlo decide di aggiungere alcuni articoli al carrello. Una volta completata la sua selezione, il cliente è pronto per controllare il carrello e rivedere ciò che ha scelto.

Cliccando sull'icona del carrello, Carlo accede a una schermata che riassume tutti gli articoli selezionati, con dettagli quali il nome, la quantità e il prezzo di ciascun prodotto. Qui, il cliente nota che forse uno degli articoli non soddisfa i suoi desideri e decide di rimuoverla. Tramite l'apposita icona "Elimina", l'articolo sparisce dalla lista, e il totale dell'ordine viene aggiornato automaticamente, rendendo immediatamente visibile il nuovo totale complessivo. Dopo aver corretto i dettagli degli articoli nel carrello, Carlo è libero di decidere se continuare a navigare o procedere con l'acquisto. Senza fretta, il sistema salva i dati relativi al carrello e consente a Carlo di tornare più tardi per completare l'acquisto.

## **Acquisto**

Dopo aver trascorso qualche giorno a riflettere, il cliente torna su **HomeDecore**, rivede il carrello che aveva salvato e decide finalmente di procedere all'acquisto. Cliccando su "Procedi all'acquisto", Carlo viene immediatamente reindirizzato alla pagina di checkout, dove gli vengono richieste alcune informazioni necessarie.

Nella prima fase, il cliente inserisce i dettagli di spedizione: il suo indirizzo, il metodo di consegna preferito, e una preferenza per la spedizione rapida. Una volta completati questi passaggi, Carlo sceglie il metodo di pagamento tra quelli disponibili: carta di credito o pagamento alla consegna.

Prima di finalizzare l'acquisto, il sistema gli presenta un riepilogo di tutto l'ordine: il costo dei prodotti, le spese di spedizione, e il totale comprensivo di tasse. Carlo analizza il tutto con attenzione, e se soddisfatto della sua scelta, clicca sul pulsante "Completa ordine".

A Carlo sarà notificato che l'ordine è stato confermato.

## **Recensione del prodotto**

Dopo aver ricevuto i prodotti acquistati su HomeDecore, Carlo è particolarmente colpito dalla qualità dell'articolo che ha comprato. Poco dopo la consegna, decide di tornare sul sito per lasciare una recensione al fine di aiutare altri utenti a effettuare la scelta migliore a seconda delle esigenze. Dopo essersi recato sulla pagina di visualizzazione del prodotto da lui acquistato, il sistema fornisce un semplice modulo per lasciare la sua recensione. Decide di dare al prodotto una valutazione di 5 stelle, e scrive un breve commento mettendo in risalto le caratteristiche dell'articolo.

## **Accesso e registrazione**

Dopo aver esplorato il sito e trovato un articolo di suo interesse, Luca tenta di effettuare un ordine, ma scopre che la funzionalità è riservata agli utenti registrati. Decide quindi di accedere alla sezione di login ed effettuare la registrazione.

Il sistema chiede a Luca di inserire le proprie credenziali, tra cui l'indirizzo email ed il nome. A questo punto, Luca ha accesso alle stesse funzionalità degli utenti registrati e continua la navigazione su HomeDecore.

## **Gestore del magazzino**

### **Modifica dei prodotti del proprio magazzino**

Carlo, gestore di uno dei magazzini di Salerno, ha intenzione di visualizzare i prodotti del proprio magazzino per apportare modifiche. Carlo vede una lista con tutti i prodotti di cui il magazzino è fornito, con rispettive caratteristiche, come il nome, descrizione, prezzo e quantità. Da questa sezione, l'utente ha la possibilità di modificare le caratteristiche del

prodotto o eliminarlo dai prodotti listati nel proprio magazzino, rendendolo così irreperibile per i clienti. Nel caso in cui uno dei prodotti non fosse disponibile, Carlo avrebbe la possibilità di richiederlo al fornitore.

### **Aggiunta prodotti con richiesta al fornitore**

Carlo, vedendo la lista di propri prodotti, si accorge che alcuni di questi non sono reperibili a causa bassa quantità o assenza all'interno del magazzino. Per risolvere il problema, Carlo decide di vedere qual è il fornitore di tale prodotto e contattarlo tramite HomeDecore, chiedendogli una quantità a sua scelta del prodotto di cui desidera rifornirsi. Carlo decide di aggiungere un prodotto completamente nuovo all'interno del suo magazzino, quindi consulta la lista di tutti i filtrando i risultati secondo le sue esigenze. Una volta contattato il fornitore, Carlo può aggiungere il prodotto all'interno della lista degli articoli nel proprio magazzino, stabilendone tutte le caratteristiche.

### **Assegnamento di ordini**

Numerosi clienti, essendo interessati a prodotti per la casa, decidono di acquistare dal magazzino di Carlo, che si ritrova a dover gestire i numerosi ordini effettuati. Carlo ha la possibilità, tramite una tabella, di visualizzare tutti gli acquisti dal più al meno recente e di conoscere quali sono i prodotti acquistati e il totale. Per compiere le spedizioni, Carlo fa riferimento alla lista dei gestori degli ordini presenti sul sistema, scegliendone uno per ogni ordine effettuato. Una volta assegnato l'ordine al gestore, Carlo può controllare lo stato della spedizione e visualizzarlo all'interno della tabella.

## **Fornitore**

### **Visualizzazione richieste dei prodotti**

Mauro, rivenditore di prodotti per la cucina su HomeDecore, visualizza nella pagina apposita del sito l'elenco di richieste da parte dei magazzini dei prodotti mancanti. Per ogni richiesta, Mauro visualizza:

- Nome del magazzino che ha fatto la richiesta;
- Il tipo di prodotto che ha richiesto;
- Lo stato della richiesta che verrà aggiornato in base al click di un bottone "Conferma" alla fine della richiesta;

Mauro, per dare priorità alle richieste in giacenza da più tempo, ordina i risultati in base alla data, in modo da ottimizzare il lavoro.

Una volta concluso, il fornitore ha confermato l'operazione. Le richieste soddisfatte da Mauro, saranno rimosse dall'elenco del fornitore.

## **Gestore dell'ordine**

### **Visualizzazione ordini assegnati**

Antonio, gestore degli ordini, vorrebbe visualizzare gli ordini che i vari magazzini hanno effettuato. Quindi, si reca sull'apposita pagina che gli permette di gestirli. In questa pagina, Antonio visualizza una lista di ordini dei vari magazzini con varie informazioni come id dell'ordine, la data, il nome del magazzino che lo ha effettuato, un elenco di prodotti selezionati e lo stato corrente dell'ordine.

Antonio filtra gli ordini dal meno al più recente, e si mette a lavoro a partire dal primo risultato.

### **Modifica stato dell'ordine**

Antonio, dopo aver prioritizzato gli ordini dal meno recente al più recente, modifica lo stato dell'ordine su cui ha iniziato a lavorare da "in attesa" a "in preparazione". Una volta svolte le sue mansioni, Antonio torna sul sito ed aggiorna lo stato dell'ordine a "spedito". Qualche giorno dopo, il pacco arriva al cliente, ed il gestore dell'ordine si occupa di mutare lo stato dell'ordine a "consegnato".

## **Scenari comuni all'utilizzatore**

### **Visualizzazione profilo**

Ogni utente registrato su HomeDecore ha la possibilità di visualizzare le proprie informazioni e conoscere quelle di un qualsiasi altro utente, indipendentemente dalla categoria di quest'ultimo.

Le informazioni che vengono mostrate per tutti i tipi di utenti sono di carattere generale, come nome, cognome, e-mail, che si trovano nella parte superiore del profilo, poiché sono fondamentali per permettere interazioni. Successivamente è presente un'ulteriore sezione, che mostra informazioni specifiche per le categorie degli utenti.

Se si tratta di un cliente, il sistema, all'interno della sezione dedicata, mostrerà tutti gli ordini effettuati, con i prodotti acquistati.

Nel caso di un gestore di magazzino, saranno visualizzate le informazioni relative al magazzino gestito, inclusi i prodotti in vendita.

Per un fornitore, invece, il sistema mostrerà l'elenco dei prodotti disponibili per il rifornimento dei magazzini.

Infine, se l'utente è un gestore delle spedizioni, saranno elencati gli ordini che ha gestito.

### **Chat bidirezionale**

Un utente qualsiasi (gestore magazzino, gestore ordini, fornitore, admin) vorrebbe comunicare un problema su un determinato prodotto o servizio offerto, lo potrebbe fare tramite l'inserimento di una sezione dedicata presente all'interno del profilo degli utenti che permette di avviare una chat in tempo reale e comunicare con un qualsiasi altro utente.

- Il gestore del magazzino può comunicare direttamente con il fornitore per richiedere informazioni sui tempi di consegna, problemi sulle scorte o altro;
- Il gestore degli ordini può contattare il gestore del magazzino per verificare la disponibilità dei prodotti richiesti nei vari ordini, oppure può coordinarsi con i fornitori per



eventuali ritardi o altro;

- Admin può essere contattato dagli altri utenti per risolvere problemi relativo al proprio profilo o altri bug presenti nelle pagine che non permettono di svolgere operazioni in modo corretto;

### 3.3.2. Use case model

#### UC 1

Nome caso d'uso	Recensione di un prodotto	
<b>Attori</b>	Cliente	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>L'utente si trova sulla pagina di visualizzazione di un prodotto.</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
		<ol style="list-style-type: none"> <li>Clicca sul bottone "Aggiungi Recensione"</li> <li>scrive un breve commento sulle caratteristiche del prodotto.</li> <li>Inserisce una valutazione numerica</li> </ol>
	<ol style="list-style-type: none"> <li>Controlla che l'utente sia autenticato.</li> <li>Controlla che l'utente abbia acquistato il prodotto.</li> <li>mostra un messaggio che informa l'utente dell'esito positivo dell'operazione.</li> </ol>	
<b>Flussi alternativi</b>	<ul style="list-style-type: none"> <li>Se al punto 5 l'utente non ha acquistato il prodotto si esegue il flusso di UC 1.1</li> <li>Se al punto 4 l'utente non è autenticato si esegue il flusso di UC 1.2</li> </ul>	
<b>Exit condition</b>	La recensione è stata aggiunta alla pagina del prodotto.	

#### UC 1.3

Nome caso d'uso	Modifica di una recensione	
<b>Attori</b>	Agente registrato	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>L'utente ha selezionato una recensione nella sezione recensioni di un prodotto.</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
		<ol style="list-style-type: none"> <li>Clicca su "Modifica recensione"</li> </ol>
	<ol style="list-style-type: none"> <li>Controlla che l'utente sia loggato</li> <li>Controlla che l'utente abbia effettivamente scritto quella recensione</li> </ol>	
		<ol style="list-style-type: none"> <li>Aggiorna il campo della recensione.</li> </ol>
<b>Flussi alternativi</b>	<ul style="list-style-type: none"> <li>2.1: se l'utente non è loggato si procede col flusso del UC 1.2</li> <li>3.1 se l'utente non è il proprietario della recensione, il sistema mostra un messaggio di errore e non permette l'inserimento.</li> </ul>	
<b>Exit condition</b>	La recensione del prodotto è stata aggiornata ed è visibile agli utenti.	

## UC 1.1

Nome caso d'uso	L'utente lascia una recensione ma non ha acquistato il prodotto	
<b>Attori</b>	Cliente autenticato	
<b>Entry Condition</b>	L'utente ha lasciato una recensione e una valutazione a un prodotto che non ha acquistato.	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	<ol style="list-style-type: none"> <li>1. solleva un'eccezione e blocca il processo di inserimento della recensione.</li> <li>2. notifica all'utente che non ha acquistato il prodotto e che non può recensirlo</li> </ol>	
<b>Flussi alternativi</b>		
<b>Exit condition</b>	Il sistema non ha inserito la recensione.	

## UC 1.2

Nome caso d'uso	Mancata autenticazione	
<b>Attori</b>	Cliente guest	
<b>Entry Condition</b>	L'utente prova ad effettuare qualche operazione per cui è necessaria l'autenticazione.	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	<ol style="list-style-type: none"> <li>1. Verifica se l'utente è autenticato</li> <li>2. Mostra un messaggio di errore</li> <li>3. Reindirizza alla pagina di login</li> </ol>	
<b>Flussi alternativi</b>		
<b>Exit condition</b>	L'utente è sulla pagina di login.	

## UC 2

Nome caso d'uso	Procedere all'acquisto	
<b>Attori</b>	Cliente registrato	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>L'utente è nella pagina di visualizzazione del carrello.</li> <li>L'utente ha inserito una quantità di prodotti al carrello <math>\geq 1</math>.</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
		1. Clicca su procedi all'acquisto
	2. Controlla che l'utente sia loggato 3. Controlla che la quantità di ciascun prodotto nel carrello sia disponibile. 4. Mostra un form per il pagamento.	
		5. Compila il form di pagamento. 6. Sottomette il form di pagamento.
	7. Convalida i dati di pagamento. 8. Mostra un form per la spedizione.	
		9. Compila il form di spedizione. 10. Sottomette il form di spedizione.
	11. Rimuove gli articoli dal carrello. 12. Mostra il riepilogo dell'acquisto.	
<b>Flussi alternativi</b>	10.1 Se l'utente compila il form di spedizione con parametri non validi, si segue il flusso di UC 3.1 3.1 Se la quantità di un certo prodotto non è disponibile, il sistema chiede all'utente se vuole confermare l'acquisto o rimuovere il prodotto. 7.1 Se l'utente compila il form di pagamento con parametri non validi, si segue il flusso di UC 3.1	
<b>Exit condition</b>	L'ordine risulta nella lista di ordini effettuati dall'utente.	

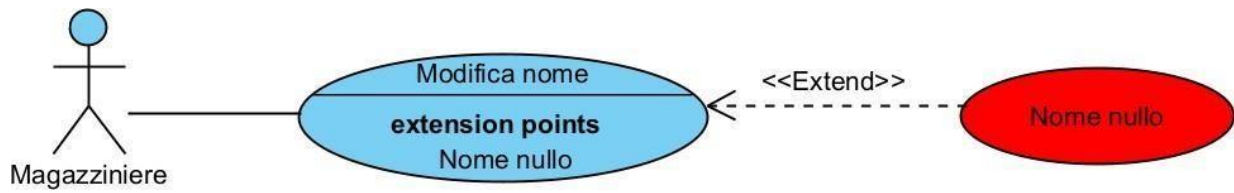
## UC 2.1

Nome caso d'uso	Aggiunta al carrello	
<b>Attori</b>	Cliente	
<b>Entry Condition</b>	L'utente si trova nella pagina di visualizzazione di un prodotto.	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
		1. Clicca sul bottone "Aggiungi al carrello".
	2. Controlla che l'utente sia autenticato. 3. Controlla che il prodotto sia disponibile.	
<b>Flussi alternativi</b>	<ul style="list-style-type: none"> <li>2.1: Se l'utente non è autenticato, si esegue il flusso di UC 1.2</li> <li>3.1 Se il prodotto non è disponibile, si esegue il flusso di 2.2</li> </ul>	
<b>Exit condition</b>		

## UC 2.2

Nome caso d'uso	Aggiunta al carrello di un prodotto non disponibile	
<b>Attori</b>	Cliente registrato	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>L'utente è sulla pagina di visualizzazione del prodotto.</li> <li>L'utente ha selezionato una quantità di prodotto inferiore alle scorte in magazzino.</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	1. Mostra un messaggio di errore.	
<b>Flussi alternativi</b>		
<b>Exit condition</b>	L'utente è sulla pagina di visualizzazione del prodotto, che non è stato aggiunto al carrello.	

### UC 3

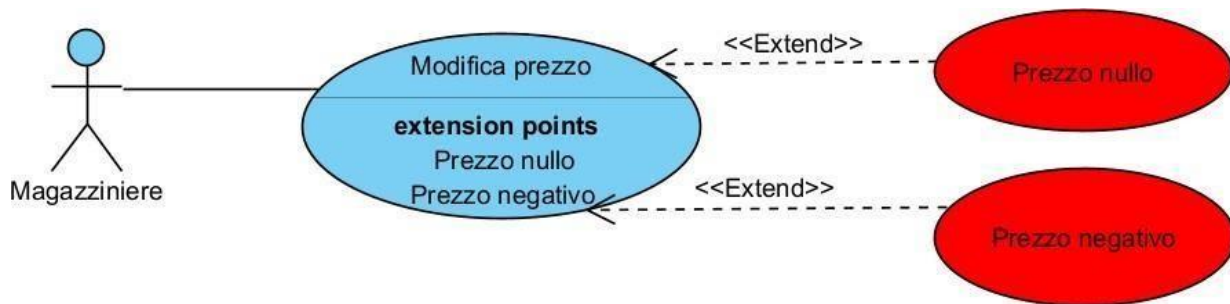


Nome caso d'uso	Modifica nome di un prodotto nel Magazzino	
<b>Attori</b>	Gestore del Magazzino	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere seleziona il prodotto da modificare nella schermata che mostra i prodotti del proprio magazzino</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	1. Il sistema rende modificabile il nome del prodotto selezionato	
		2. Il magazziniere apporta le modifiche scrivendo il nuovo nome
		3. Il magazziniere completa l'azione premendo il tasto di conferma
	4. Il sistema salva le modifiche del prodotto all'interno del database	
<b>Flussi alternativi</b>	Se al punto 2 il magazziniere aggiunge un nome nullo, si esegue il flusso di UC 3.1	
<b>Exit condition</b>	<ul style="list-style-type: none"> <li>Le modifiche relative al nome del prodotto sono apportate al database</li> <li>Tutti gli utenti saranno in grado di vedere il nuovo nome del prodotto</li> </ul>	

### UC 3.1

Nome caso d'uso	Inserimento di un nome nullo nella modifica del nome di un prodotto	
Attori	Magazziniere	
Entry Condition	<ul style="list-style-type: none"> <li>Il magazziniere modifica il nome del prodotto</li> <li>Il magazziniere aggiunge un nome nullo</li> </ul>	
Flusso di eventi	SISTEMA	UTENTE
	1. Mostra un messaggio di errore suggerendo che il nome non deve essere nullo	
Flussi alternativi		
Exit condition	Il magazziniere si ritrova sulla pagina di visualizzazione dei prodotti	

### UC 4



Nome caso d'uso	Modifica prezzo di un prodotto nel Magazzino	
Attori	Gestore del Magazzino	
Entry Condition	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere seleziona il prodotto da modificare nella schermata che mostra i prodotti del proprio magazzino</li> </ul>	
Flusso di eventi	SISTEMA	UTENTE
	1. Il sistema rende modificabile il nome del prodotto selezionato	
		2. Il magazziniere apporta le modifiche scrivendo il nuovo prezzo
		3. Il magazziniere completa l'azione premendo il tasto di conferma
	4. Il sistema salva le modifiche del prodotto all'interno del database	
Flussi alternativi	Se al punto 2 il magazziniere aggiunge un prezzo nullo (ovvero pari a 0), si esegue il flusso di UC 4.1 Se al punto 2 il magazziniere aggiunge un prezzo negativo, si esegue il flusso di UC 4.2	
Exit condition	<ul style="list-style-type: none"> <li>Le modifiche relative al prezzo del prodotto sono apportate al database</li> <li>Tutti gli utenti saranno in grado di vedere il nuovo nome del prodotto</li> </ul>	

### UC 4.1

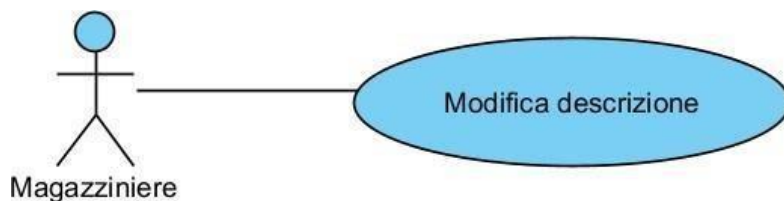
Nome caso d'uso	Inserimento di un prezzo nullo nella modifica del nome di un prodotto
-----------------	---

<b>Attori</b>	Magazziniere	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere modifica il prezzo del prodotto</li> <li>Il magazziniere aggiunge un prezzo pari a 0</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	1. Mostra un messaggio di errore suggerendo che il prezzo non può essere nullo	
<b>Flussi alternativi</b>		
<b>Exit condition</b>	Il magazziniere si ritrova sulla pagina di visualizzazione dei prodotti	

## UC 4.2

<b>Nome caso d'uso</b>	Inserimento di un prezzo negativo nella modifica del nome di un prodotto	
<b>Attori</b>	Magazziniere	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere modifica il prezzo del prodotto</li> <li>Il magazziniere aggiunge un prezzo negativo</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	1. Mostra un messaggio di errore suggerendo che il prezzo deve essere di un valore positivo	
<b>Flussi alternativi</b>		
<b>Exit condition</b>	Il magazziniere si ritrova sulla pagina di visualizzazione dei prodotti	

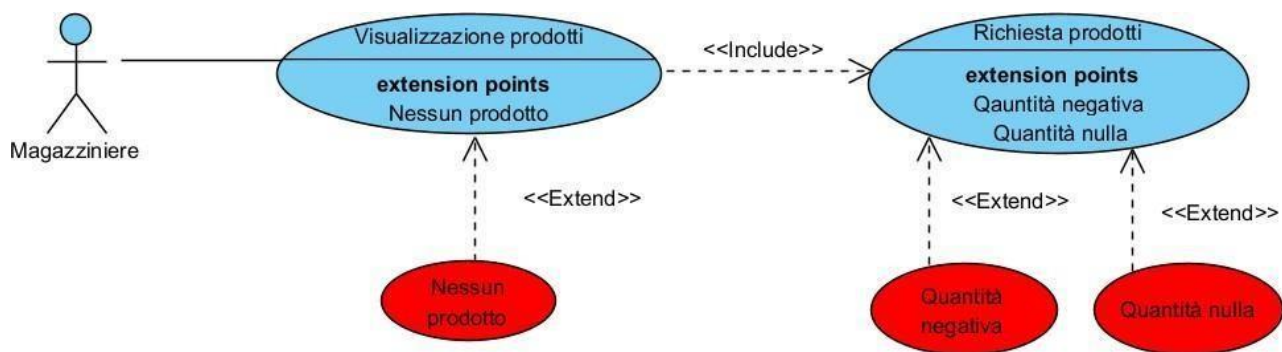
## UC 5



<b>Nome caso d'uso</b>	Modifica descrizione di un prodotto nel Magazzino	
<b>Attori</b>	Gestore del Magazzino	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere seleziona il prodotto da modificare nella schermata che mostra i prodotti del proprio magazzino</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	1. Il sistema rende modificabile il nome del prodotto selezionato	
		2. Il magazziniere apporta le modifiche scrivendo la nuova descrizione
		3. Il magazziniere completa l'azione premendo il tasto di conferma
	4. Il sistema salva le modifiche del prodotto all'interno del database	

<b>Flussi alternativi</b>	
<b>Exit condition</b>	<ul style="list-style-type: none"> <li>Le modifiche relative alla descrizione del prodotto sono apportate al database</li> <li>Tutti gli utenti saranno in grado di vedere il nuovo nome del prodotto</li> </ul>

## UC 6



Nome caso d'uso	Richiesta di un prodotto	
<b>Attori</b>	Gestore del Magazzino	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere seleziona il prodotto da modificare nella schermata che mostra i prodotti del proprio magazzino</li> </ul>	
<b>Flusso di eventi</b>	<b>SISTEMA</b>	<b>UTENTE</b>
	1. Il sistema mostra la lista di tutti i prodotti del proprio magazzino	
		2. Il magazziniere seleziona il prodotto da rifornire
	3. Il sistema chiederà al magazziniere informazioni per rifornire i prodotti	
		4. Il magazziniere sceglie le quantità del prodotto da rifornire
		5. Il magazziniere conferma la richiesta
	6. Il sistema informa il fornitore della richiesta	
<b>Flussi alternativi</b>	Se al punto 1 non sono presenti prodotti in magazzino, si esegue il flusso di UC 6.1 Se al punto 4 il magazziniere seleziona una quantità nulla, si esegue il flusso di UC 6.2 Se al punto 4 il magazziniere seleziona una quantità negativa, si esegue il flusso di UC 6.3	
<b>Exit condition</b>	<ul style="list-style-type: none"> <li>La richiesta di rifornimento dei prodotti sarà notificata al fornitore selezionato</li> <li>Il magazziniere si trova sulla schermata di visualizzazione dei prodotti</li> </ul>	



### UC 6.1

Nome caso d'uso	Nessun Prodotto in magazzino	
Attori	Gestore del Magazzino	
Entry Condition	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere si trova sulla pagina che mostra tutti i prodotti del magazzino</li> </ul>	
Flusso di eventi	SISTEMA	UTENTE
	1. Il sistema mostra l'informazione relativa all'assenza di prodotti	
Flussi alternativi		
Exit condition	<ul style="list-style-type: none"> <li>Il magazziniere si trova sulla pagina che mostra tutti i prodotti del magazzino</li> </ul>	

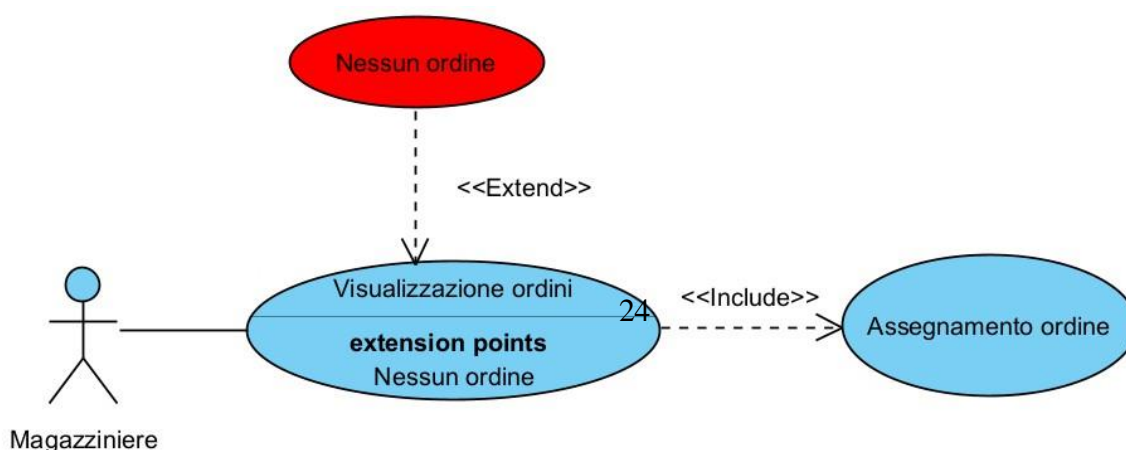
### UC 6.2

Nome caso d'uso	Selezione quantità nulla	
Attori	Gestore del Magazzino	
Entry Condition	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere seleziona una quantità nulla di prodotti da rifornire</li> </ul>	
Flusso di eventi	SISTEMA	UTENTE
	1. Mostra un messaggio di errore suggerendo che la quantità non deve essere nulla	
Flussi alternativi		
Exit condition	<ul style="list-style-type: none"> <li>Il magazziniere si trova sulla pagina che mostra tutti i prodotti del magazzino</li> </ul>	

### UC 6.3

Nome caso d'uso	Selezione quantità negativa	
Attori	Gestore del Magazzino	
Entry Condition	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere seleziona una quantità negativa di prodotti da rifornire</li> </ul>	
Flusso di eventi	SISTEMA	UTENTE
	1. Mostra un messaggio di errore suggerendo che la quantità deve essere positiva	
Flussi alternativi		
Exit condition	<ul style="list-style-type: none"> <li>Il magazziniere si trova sulla pagina che mostra tutti i prodotti del magazzino</li> </ul>	

### UC 7



Nome caso d'uso	Assegnamento dell'ordine	
<b>Attori</b>	Gestore del Magazzino	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere si deve trovare sulla schermata degli ordini effettuati dai clienti</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	1. Il sistema mostra gli ordini effettuati dai clienti dal più recente al meno recente	
		2. Il magazziniere seleziona l'ordine da gestire
		3. Il magazziniere sceglie a quale gestore affidare l'ordine
	4. Il sistema informa il gestore dell'ordine della richiesta	
<b>Flussi alternativi</b>	Se al punto 1 non sono presenti prodotti in magazzino, si esegue il flusso di UC 7.1	
<b>Exit condition</b>	<ul style="list-style-type: none"> <li>La richiesta di rifornimento dei prodotti sarà notificata al fornitore selezionato</li> <li>Il magazziniere si trova sulla schermata di visualizzazione degli ordini</li> </ul>	

### UC 7.1

Nome caso d'uso	Nessun ordine effettuato	
<b>Attori</b>	Gestore del Magazzino	
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere deve essere autenticato.</li> <li>Il magazziniere si deve trovare sulla schermata degli ordini effettuati dai clienti, con nessun ordine effettuato</li> </ul>	
<b>Flusso di eventi</b>	SISTEMA	UTENTE
	1. Mostra un messaggio informando che non ci sono ordini eseguiti	
<b>Flussi alternativi</b>		
<b>Exit condition</b>	<ul style="list-style-type: none"> <li>Il magazziniere si trova sulla pagina che mostra tutti i prodotti del magazzino</li> </ul>	

### UC 8 Modifica stato richiesta



Attori: gestore di magazzino e fornitore;

Pre condizioni:

- 58. Il fornitore deve essere autenticato;
- 59. Il fornitore deve visualizzare l'elenco delle richieste dei prodotti mancanti inviate dal gestore di magazzino;

Flow event:

- 60. Il sistema mostra le richieste dalla meno recente alla più recente;
- 61. Il sistema permette l'aggiornamento dello stato della richiesta;
- 62. Il fornitore dopo aver soddisfatto una richiesta, preme sul bottone "Conferma", visibile alla fine di ogni richiesta, che aggiorna anche lo stato della richiesta;

Flussi alternativi:

- 63. Se il fornitore non ha richieste da soddisfare, comparirà il messaggio "nessuna richiesta presente";

Post condizioni:

- a. Il gestore del magazzino riceverà una notifica dal fornitore;
- b. La quantità di quel determinato prodotto richiesto verrà aggiornata automaticamente;

## UC 9 Modifica stato ordine



Attori: gestore degli ordini;

Pre condizioni:

- 64. Il gestore degli ordini deve essere autenticato;
- 65. Il gestore degli ordini deve visualizzare una lista con tutti gli ordini assegnati dal gestore di magazzino;

Flow event:

- 66. Il sistema dovrà mostrare una lista di ordini dal meno recente al più recente;
- 67. Il sistema deve permette l'aggiornamento dello stato di un ordine in base alla fase di sviluppo in cui si trova;
- 68. Appena arriva l'ordine assume lo stato di "attesa";
- 69. Il gestore che sta provvedendo a soddisfarlo muta lo stato da "attesa" a "preparazione";
- 70. Dopo aver eseguito un ordine, il gestore cambia lo stato da "preparazione" a "spedito";
- 71. Una volta che il pacco è arrivato al cliente, il gestore modifica lo stato dell'ordine in "consegnato";

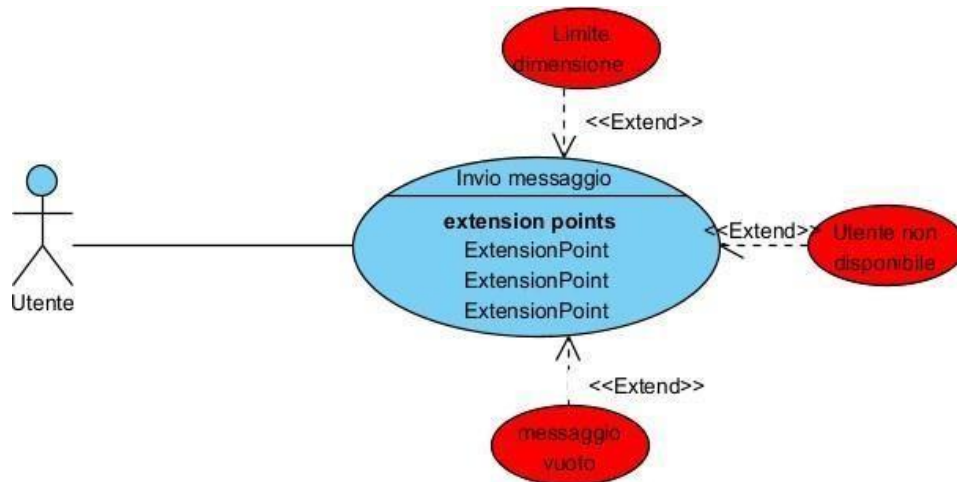
Flussi alternativi:

- 72. Se non è stato assegnato alcun ordine, il sistema mostrerà il messaggio "nessun ordine";

Post condizioni:

- 73. Il sistema viene notificato dell'arrivo dell'ordine;

## UC 10 Invio messaggio



Attori: Utente (fornitore, gestore di ordini, gestore di magazzini) e un altro utente che può essere uno di quelli citati precedentemente;

Pre condizioni:

- 74. L'utente deve essere autenticato;
- 75. L'utente deve visualizzare una lista di utenti con cui può conversare;

Flow event:

- 76. L'utente seleziona l'utente con cui vuole messaggiare dalla lista a disposizione;
- 77. Il sistema mostra un campo di testo da riempire per il messaggio;
- 78. L'utente preme sul campo di testo per poter digitare ciò che vuole mandare;
- 79. L'utente dopo aver scritto il messaggio, può premere il bottone "invio", presente affianco al campo di testo, che permette che il messaggio arrivi al destinatario;

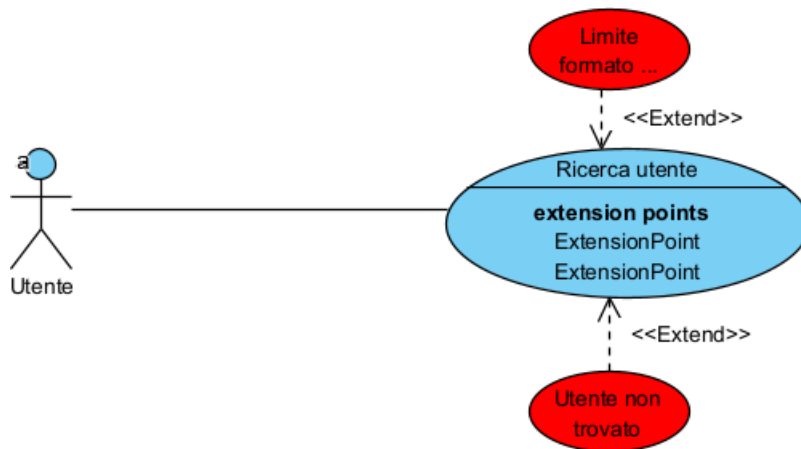
Flussi alternativi:

- 80. Il messaggio non verrà inviato se sarà superata la dimensione limite e verrà mostrato un messaggio di errore ("limite superato").
- 81. Se il messaggio è vuoto non potrà essere inviato con relativo messaggio di errore ("inserire un messaggio");
- 82. Se l'altro utente non è disponibile, sarà mostrato un messaggio di errore ("utente non disponibile");

Post condizioni:

- 83. Il messaggio è stato inviato correttamente;

## UC 11 Ricerca utente



Attori: Utente (fornitore, gestore di ordini, gestore di magazzino);

Pre condizioni:

- 84. L'utente deve essere autenticato;
- 85. Il sistema dovrà mostrare all'utente una lista di utenti;

Flow event:

- 86. Il sistema mostra all'utente una piccola barra di ricerca;
- 87. L'utente preme su questa barra di ricerca e digita il nome dell'utente con cui vuole aprire una conversazione;
- 88. Dopo aver digitato il nome preme il bottone "cerca";

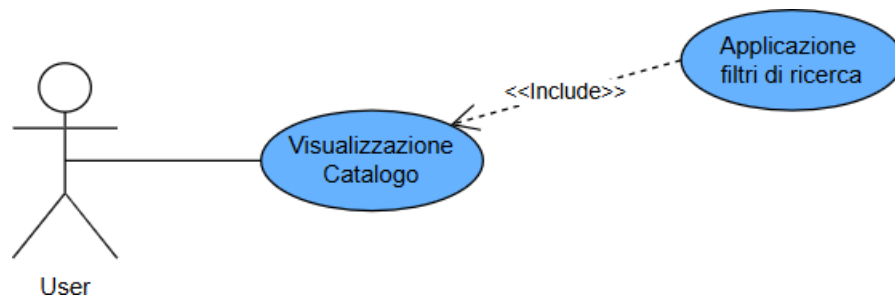
Flussi alternativi:

- 89. L'utente ricercato non è stato trovato;
- 90. Il limite del formato da ricercare è stato superato;

Post condizioni:

- 91. L'utente è stato trovato;

## UC 12 Visualizzazione Catalogo



Attori: User

Precondizioni:

1. È visualizzato il comando per visualizzare il catalogo.

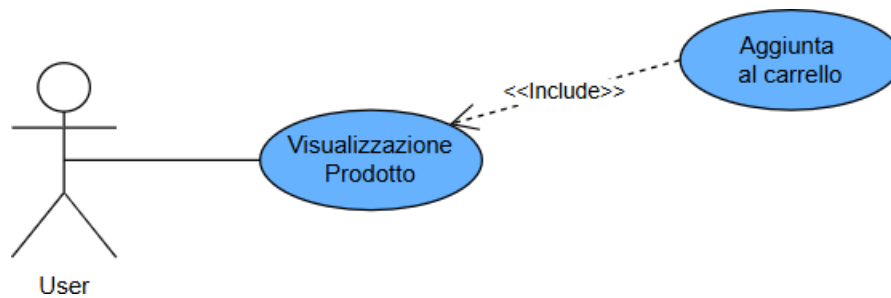
Flow event:

1. User: Richiede di poter visualizzare il catalogo presso il sito attraverso il comando apposito.
2. Sistema: Interroga il database per ottenere tutti i prodotti disponibili nel catalogo, visualizzando l'immagine, il prezzo e il nome di ogni singolo prodotto. Inoltre, fornisce un comando che permette di applicare filtri di ricerca.

Post-condizioni:

1. Visualizzazione Catalogo e filtro di ricerca.

## UC 13 Visualizzazione Prodotto



Attori: User

Precondizioni:

1. È visualizzato il comando(img) per visualizzare il prodotto.

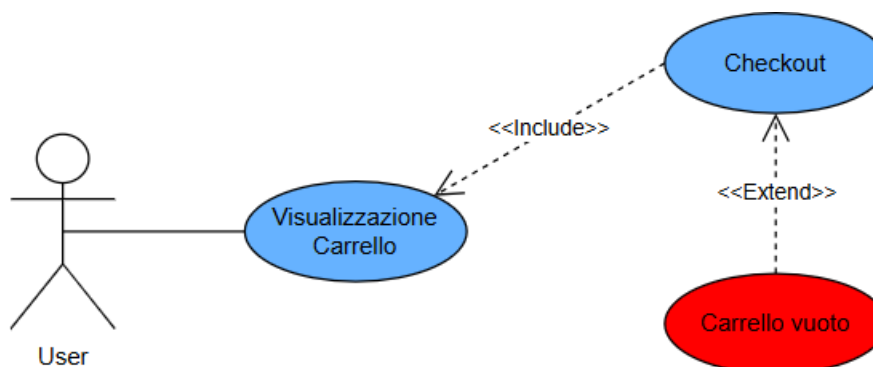
Flow event:

1. User: Seleziona un prodotto.
2. Sistema: Mostra una schermata con l'immagine, il prezzo, la descrizione del prodotto selezionato. Inoltre, fornisce due comandi per aggiungere il prodotto al carrello o alla lista dei desideri, insieme a un menu a discesa per specificare la quantità del prodotto.
3. User: Sceglie l'opzione dal menu a discesa e sottomette uno dei due comandi.
4. Sistema: Aggiunge il prodotto nel carrello.

Post-condizioni:

1. Prodotto aggiunto al carrello con le informazioni relative alla quantità.

## UC 14 Visualizzazione Carrello, Checkout



Attori: User

Precondizioni:



1. È visualizzato il comando per visualizzare il carrello.

Flow event:

1. User: Richiede di poter visualizzare il carrello presso il sito attraverso il comando apposito.
2. Sistema: visualizza l'immagine, il prezzo, la quantità e il nome di ogni singolo prodotto presente nel carrello. Inoltre, fornisce un comando per effettuare il Checkout.
3. User: Sottomette il comando Checkout.
4. Sistema: Verifica che il carrello non sia vuoto.
5. Sistema: Mostra una schermata che reindirizza l'utente alla pagina di registrazione.

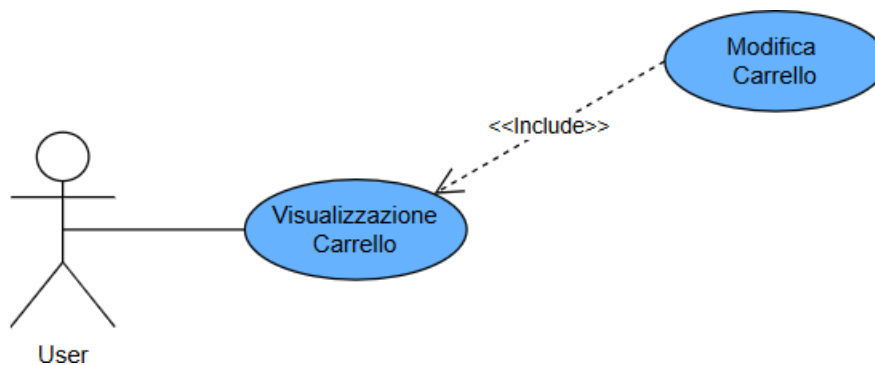
Flussi alternativi: Modifica Carrello

1. Sistema: Visualizza un messaggio di errore che segnala all'utente che il carrello è vuoto.
2. Sistema: Resta in attesa di una nuova sottomissione.

Post-condizioni:

1. Reindirizzamento alla pagina di registrazione.

## UC 15 Modifica Carrello



Attori: User

Precondizioni:

1. È visualizzato il comando per visualizzare il carrello.

Flussi degli eventi

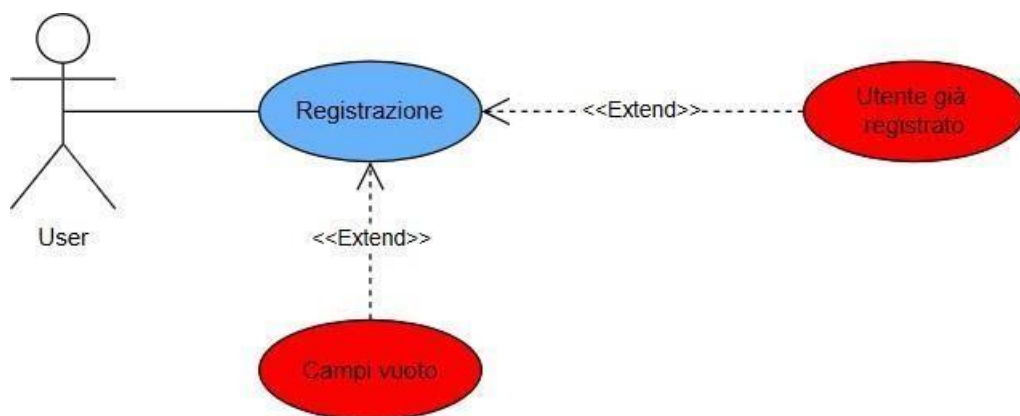
1. User: Richiede di poter visualizzare il carrello presso il sito attraverso il comando apposito.

2. Sistema: visualizza l'immagine, il prezzo, la quantità e il nome di ogni singolo prodotto presente nel carrello. Inoltre, fornisce un comando per effettuare la Modifica al carrello.
3. User: Sottomette il comando Modifica carrello.
4. Sistema: visualizza l'immagine, il prezzo, la quantità e il nome di ogni singolo prodotto presente nel carrello. Inoltre, fornisce, per ogni prodotto, due comandi:
  - Elimina prodotto
  - Modifica quantità: campo di testo.
5. User: Sottomette le modifiche.
6. Sistema: Modifica il carrello con le nuove informazioni.

Post-condizioni:

1. L'utente visualizza il carrello aggiornato con le nuove modifiche.

## UC 16 Registrazione



Attori: User

Precondizioni:

1. È visualizzato il comando per effettuare la registrazione.

Flusso degli eventi:

1. Cliente: Richiede di potersi registrare presso il sito attraverso il comando apposito.
2. Sistema: Visualizza una form che richiede l'inserimento di:
  - Nome: Stringa di caratteri alfabetici.
  - Cognome: Stringa di caratteri alfabetici.
  - Data di nascita: Sono presenti tre menù che consentono di specificare, giorno mese ed anno di nascita.
  - Indirizzo e-mail: Stringa.
  - Password: Stringa.
  - p. IVA: Stringa di caratteri numerici.
  - IBAN: Stringa di caratteri numerici.

Tutti i campi suddetti sono obbligatori.

3. Cliente: Riempie tutti i campi obbligatori e sottomette la form compilata.
4. Sistema: Verifica che:
  - Tutti i campi obbligatori siano stati compilati.
  - Il campo p.IVA sia valido.
  - Il campo IBAN sia valido.
  - L'email non sia già in uso.
  - Il cliente non sia già presente nel sistema.
5. Sistema: Salva i dati del cliente.
6. Sistema: Mostra una Schermata che informa il cliente della registrazione avvenuta con successo.

Flusso di eventi Alternativo: Qualche campo non è stato compilato:

1. Sistema: Visualizza un messaggio di errore che segnali all'utente che non ha inserito tutti i dati.
2. Sistema: Resta in attesa di una nuova sottomissione della form.

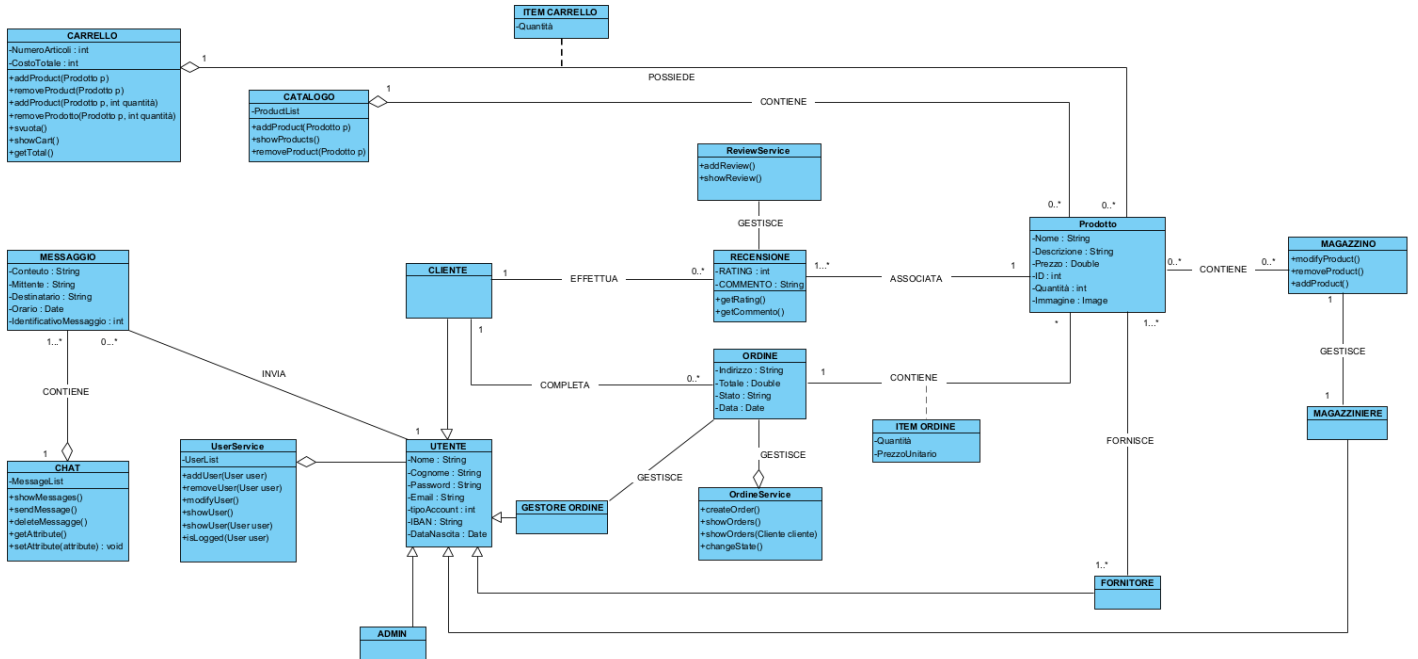
Flusso di eventi Alternativo: l'e-mail inserita dall'utente è già in uso:

1. Sistema: Visualizza un messaggio di errore che segnali all'utente che l'e-mail inserita è già in uso.
2. Sistema: Resta in attesa di una nuova sottomissione della form.

Post-condizioni:

1. L'account è registrato ed attivo. Il cliente potrà effettuare il login (con successo).

### 3.3.3. Object Model



### 3.3.4. Dynamic Model

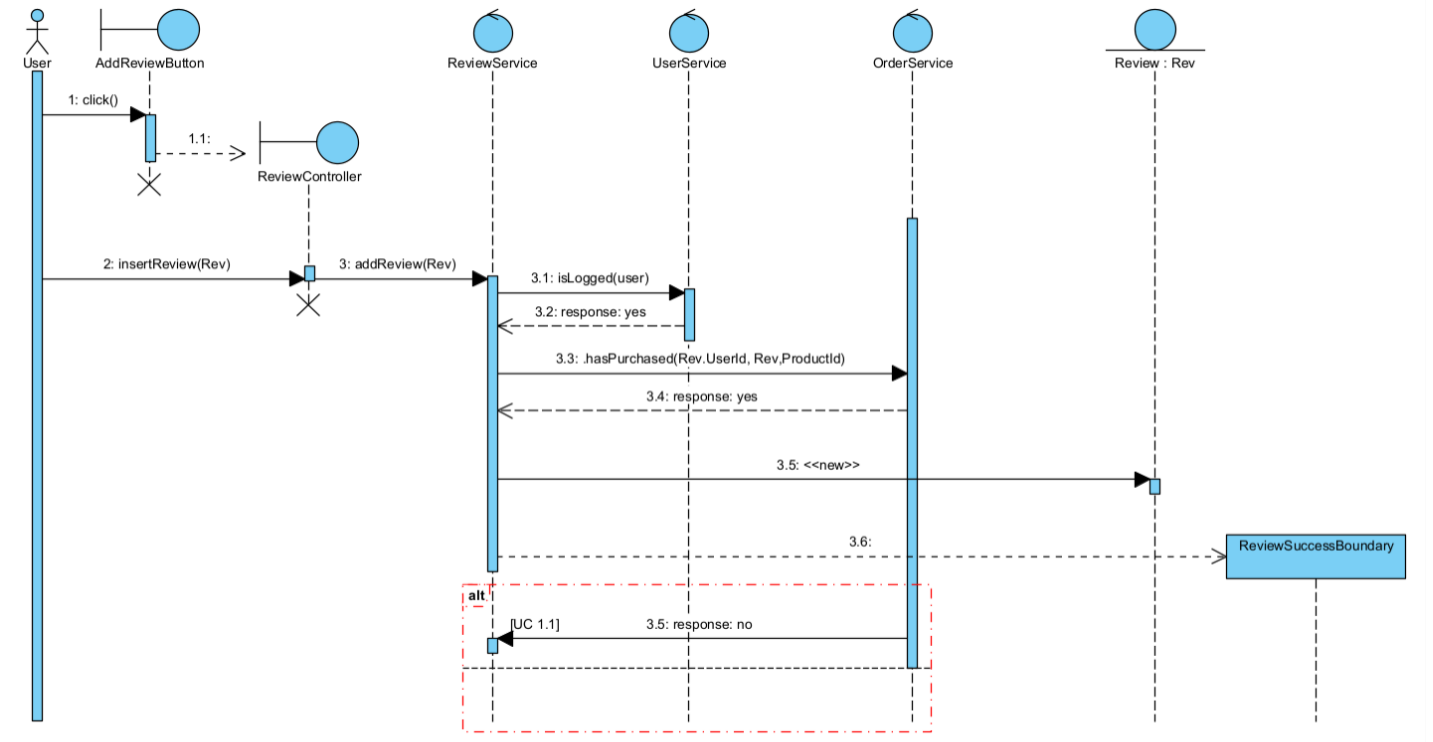
#### DIZIONARIO DEI DATI

Tipo	Nome oggetto	Descrizione
Entity	`User`	Modello che rappresenta un utente del sistema, include dati personali e informazioni di autenticazione.
Entity	`Product`	Gestisce le informazioni sui prodotti, come nome, descrizione, prezzo e quantità disponibile.
Entity	`Order`	Contiene i dati relativi agli ordini, incluso lo stato, i prodotti acquistati e i dettagli di spedizione.
Entity	`Review`	Rappresenta le recensioni lasciate dagli utenti sui prodotti, con valutazione e commenti.
Entity	`Cart`	Raccoglie i prodotti selezionati dall'utente per l'acquisto.
Entity	`Message`	Contiene le specifiche relative all'invio del messaggio, compreso il body-
Control	`UserService`	Coordina operazioni legate agli utenti, come login, registrazione e modifica profilo.
Control	`CartService`	Supervisiona l'aggiunta, la modifica e la rimozione di prodotti dal carrello.
Control	`OrderService`	Gestisce il processo di checkout e l'aggiornamento dello stato degli ordini.
Control	`ReviewService`	Coordina l'inserimento e la pubblicazione delle recensioni sui prodotti.
Control	`CatalogService`	Gestisce le richieste relative ai prodotti, inclusa la visualizzazione e la ricerca nel catalogo.

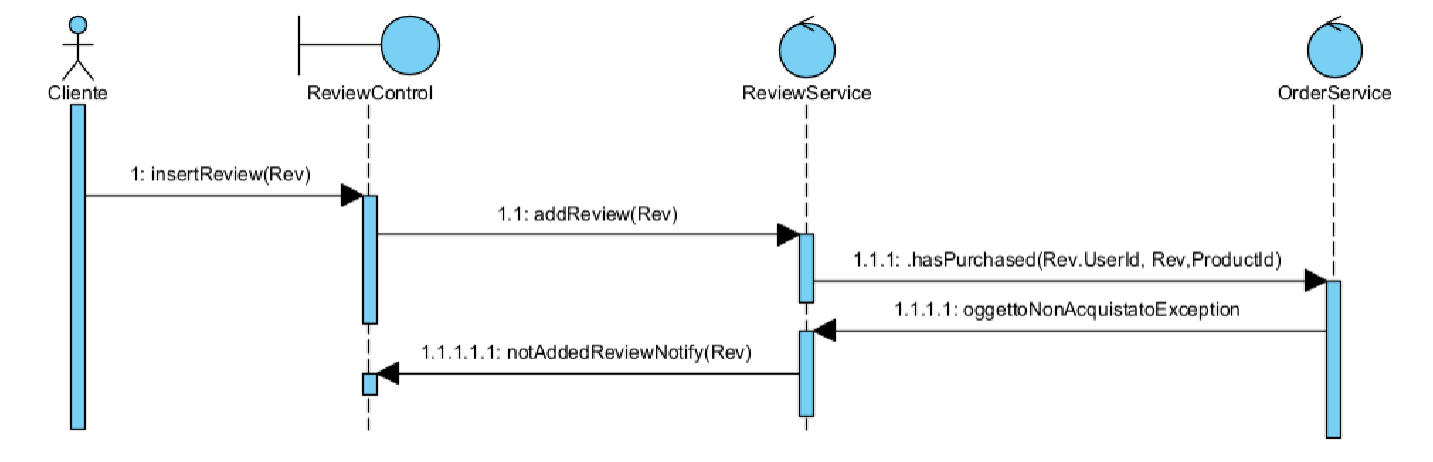
Control	'MagazzinoService'	Gestisce le operazioni relative alle scorte nel magazzino.
Control	'ChatService'	Gestisce l'invio e la ricezione del messaggio
Boundary	'PurchaseController'	Permette l'interazione nelle operazioni di acquisto.
Boundary	'CartController'	Permette l'interazione nelle operazioni sul carrello.
Boundary	'AuthenticationController'	Permette l'interazione nelle operazioni di autenticazione.
Boundary	'ReviewController'	Permette l'interazione nelle operazioni di recensione.
Boundary	'ChatController'	Permette l'interazione nelle operazioni di invio e ricezione del messaggio.
Boundary	'MagazzinoController'	Permette l'interazione nelle operazioni di aggiornamento delle scorte del magazzino.

## Sequence diagrams

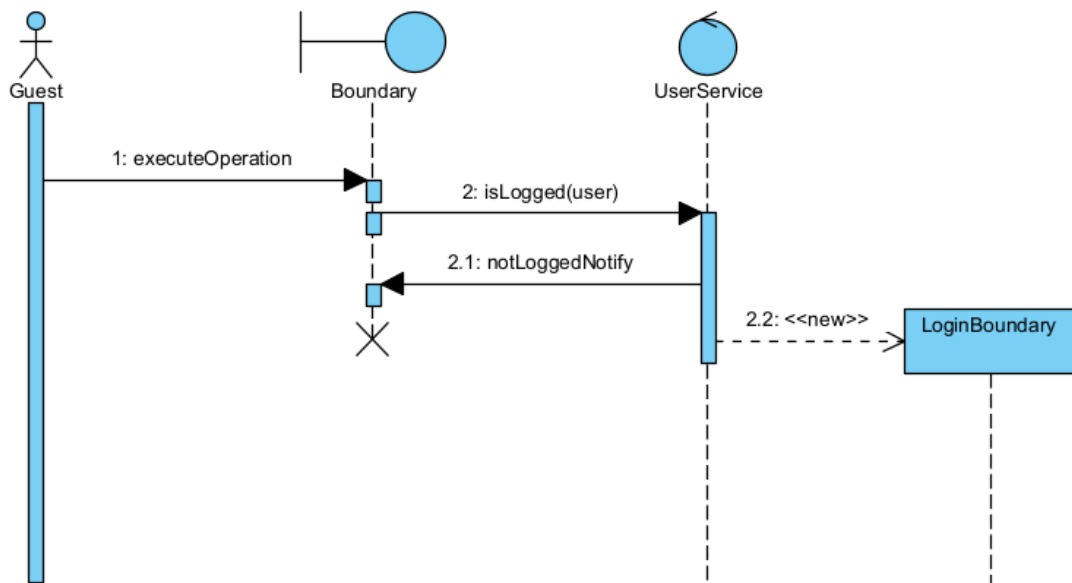
### SD1: Recensione di un prodotto – (UC 1: Recensione di un prodotto)



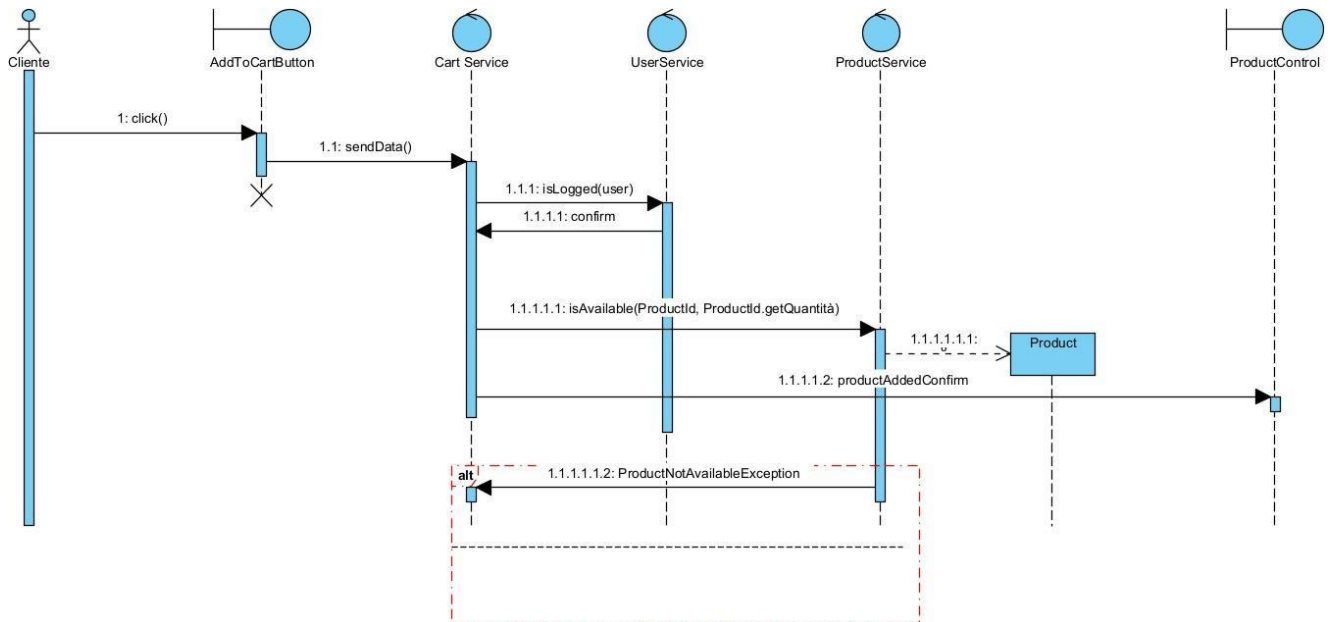
### SD 1.1 : Recensione di un prodotto non acquistato – (UC 1.1)



## SD 1.2: Mancata autenticazione – (UC 1.2 – Mancata autenticazione)



## SD 2.1 – UC 2.1 (Aggiunta al carrello)

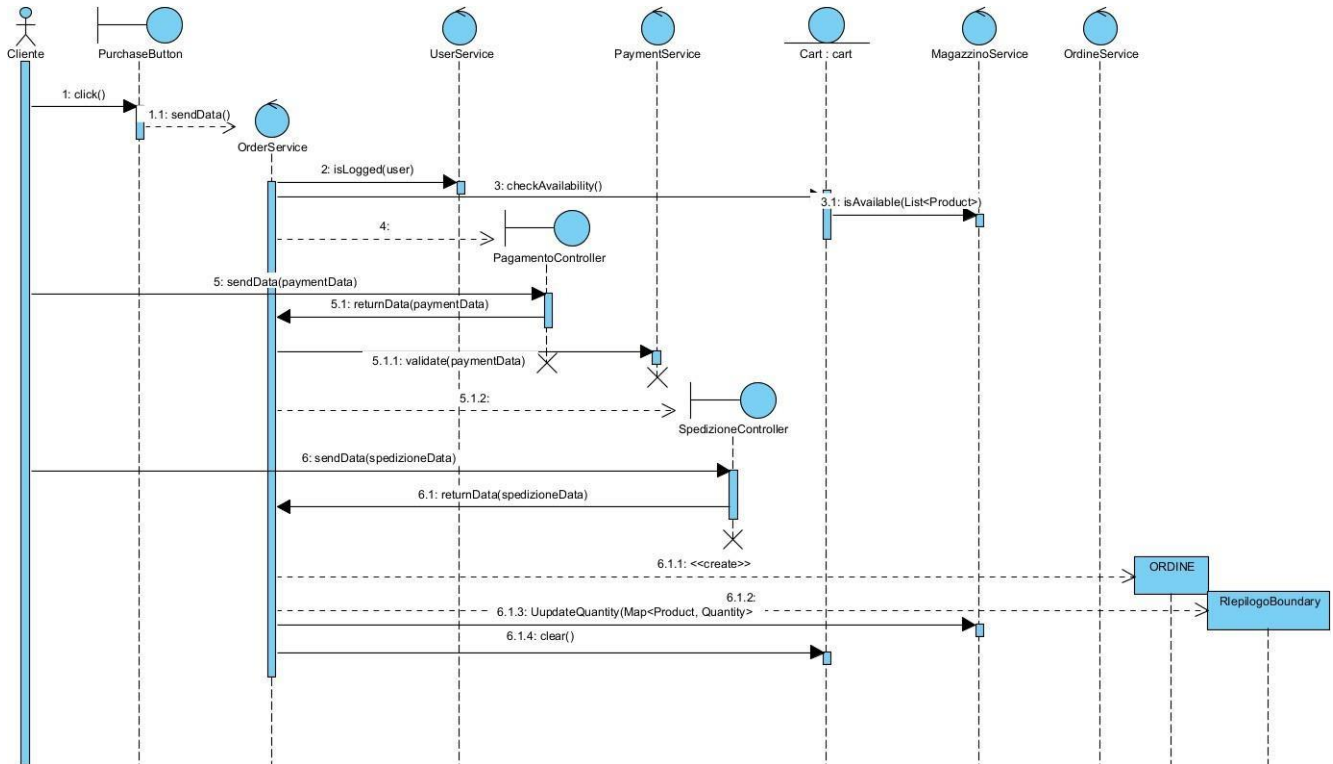




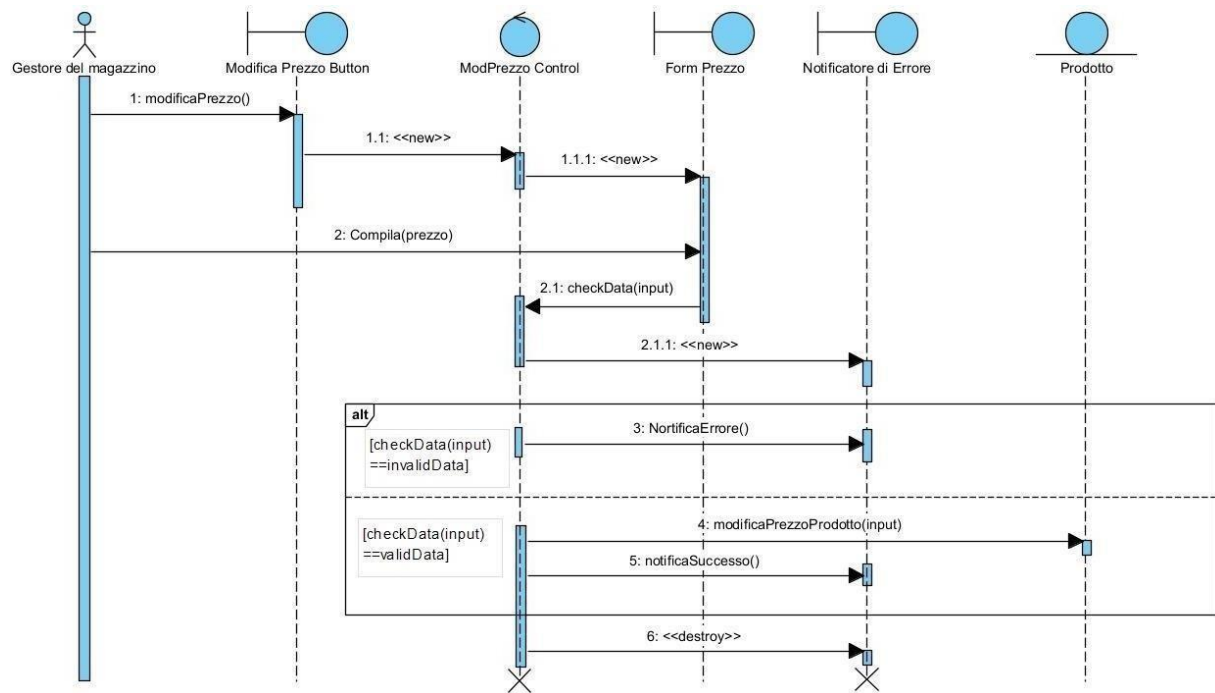
## SD 2: Acquisto di un prodotto – (UC 2: Procedere all’acquisto)

Nota: per motivi di leggibilità si assumono le seguenti omissioni:

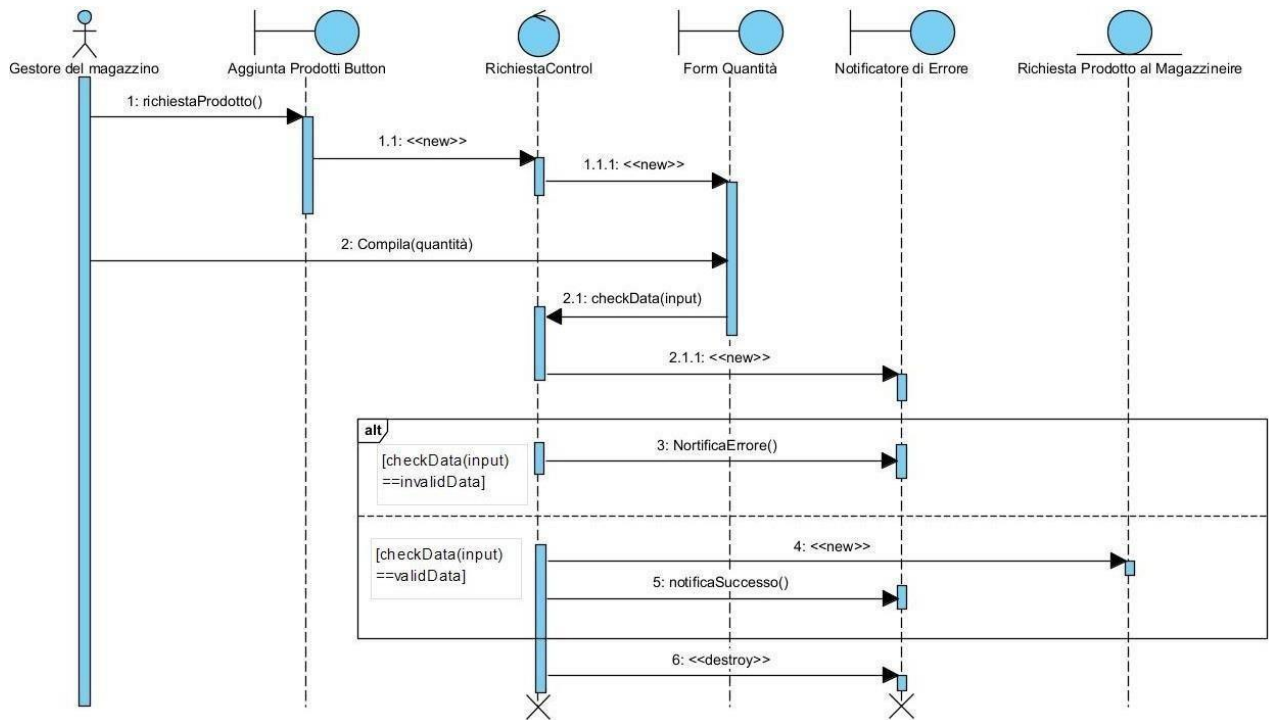
- Il valore di ritorno dei controlli isLoggedIn, checkAvailability, validate(paymentData)



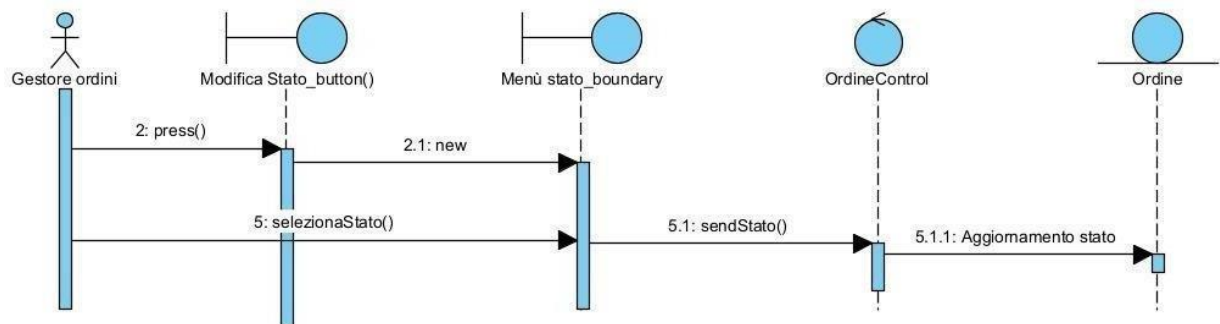
### SD 3: Modifica prezzo di un prodotto



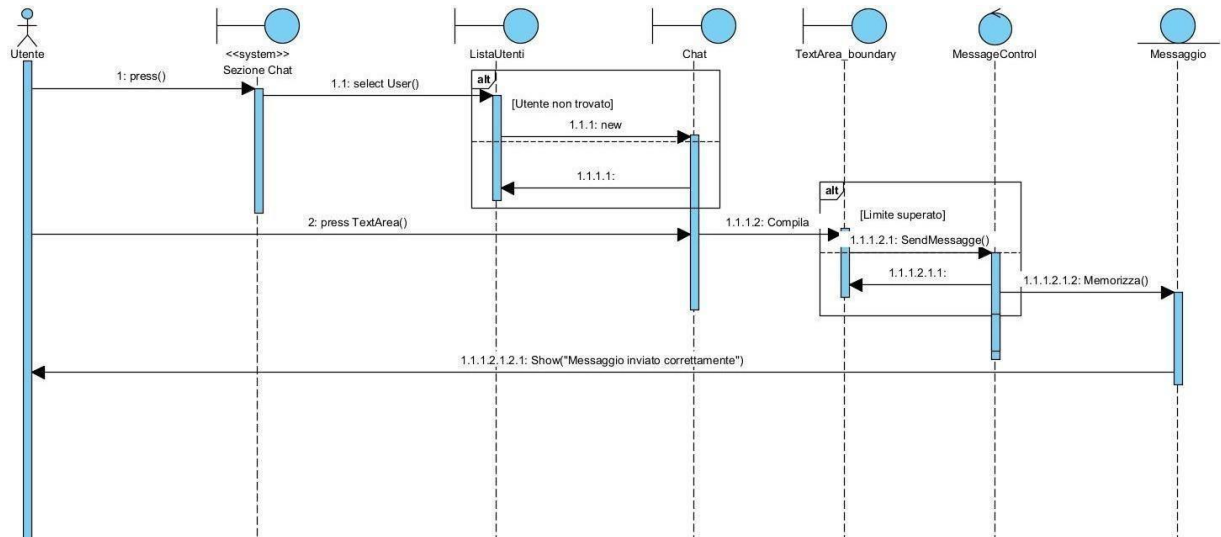
### SD 4: Richiesta di un prodotto



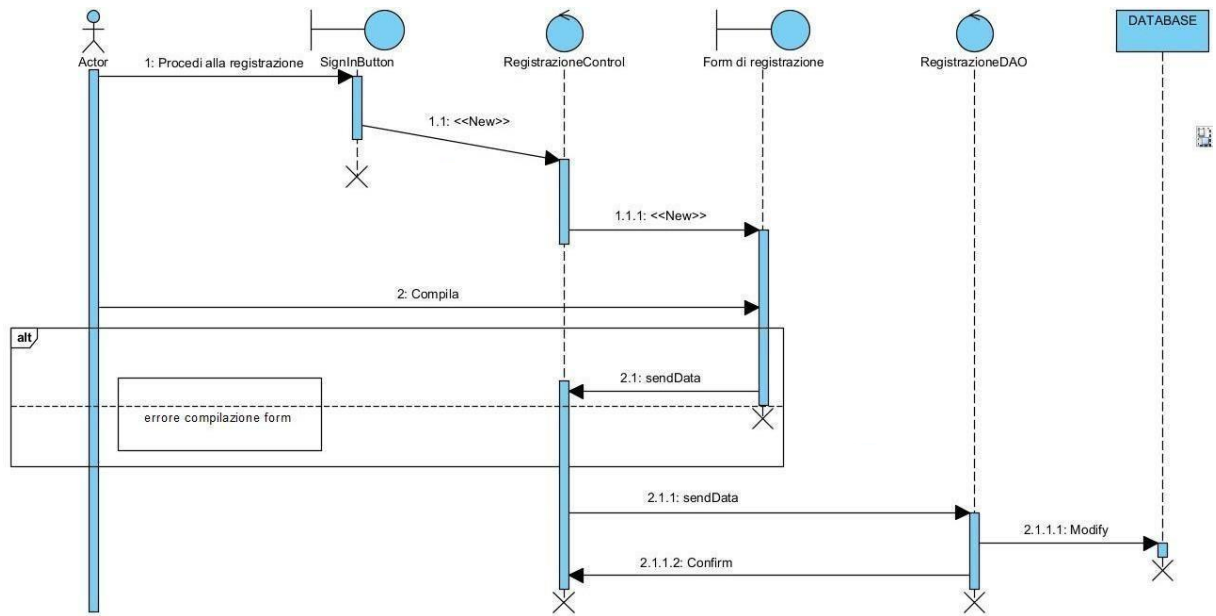
## SD 5: Modifica stato ordine



## SD 6: Invio messaggio nella chat

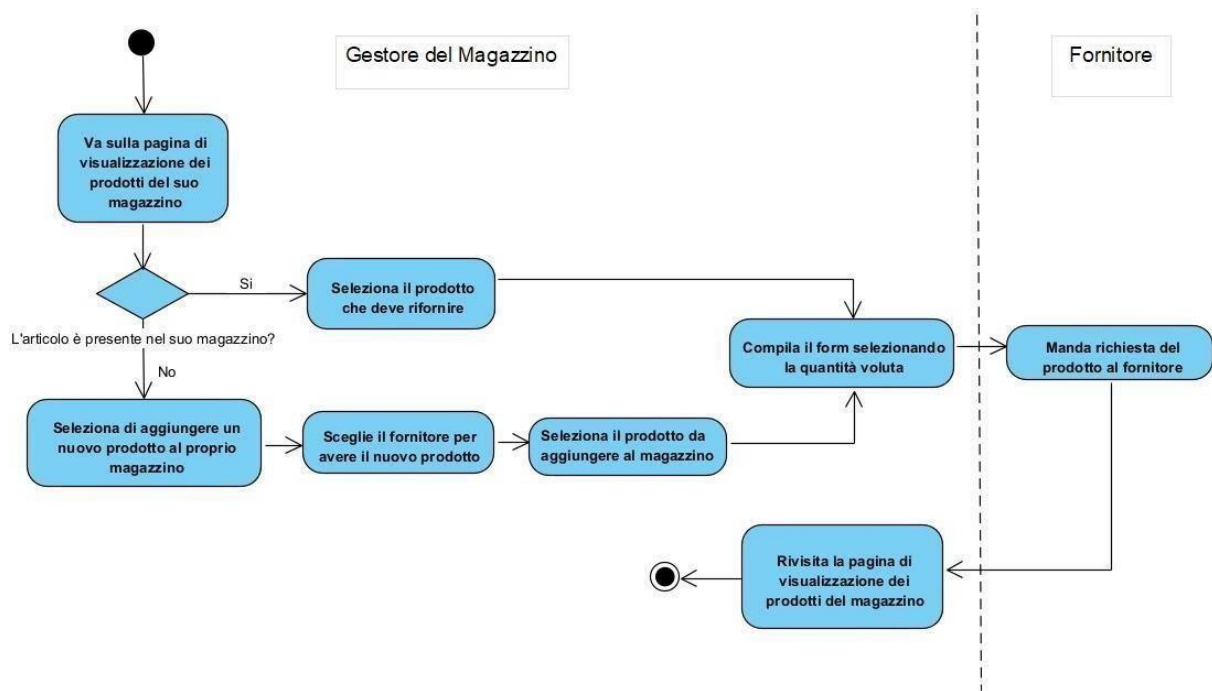


## SD 7: Registrazione utente

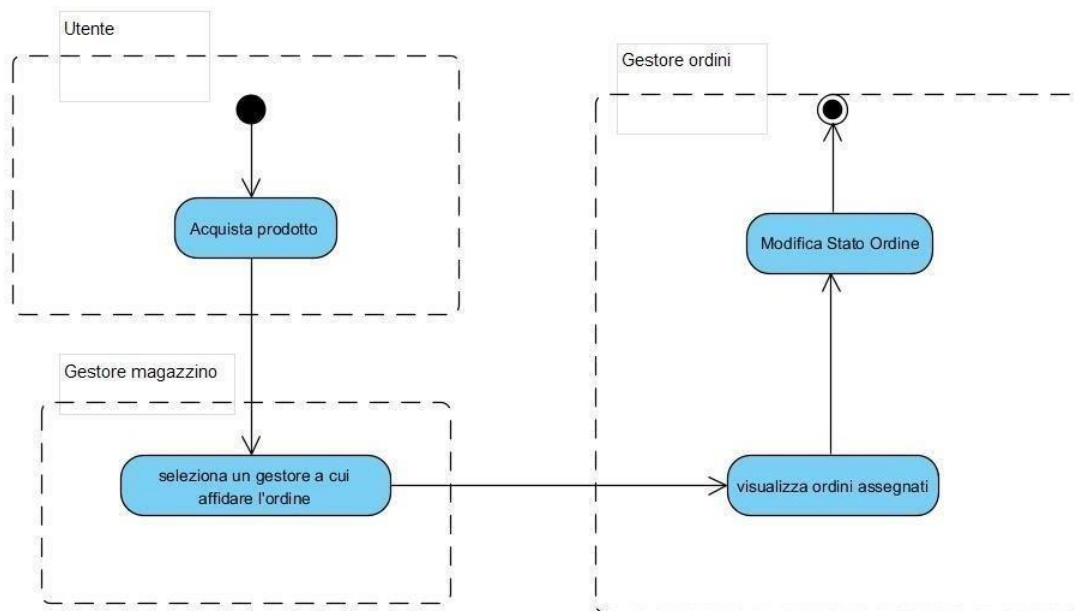


## Activity diagrams

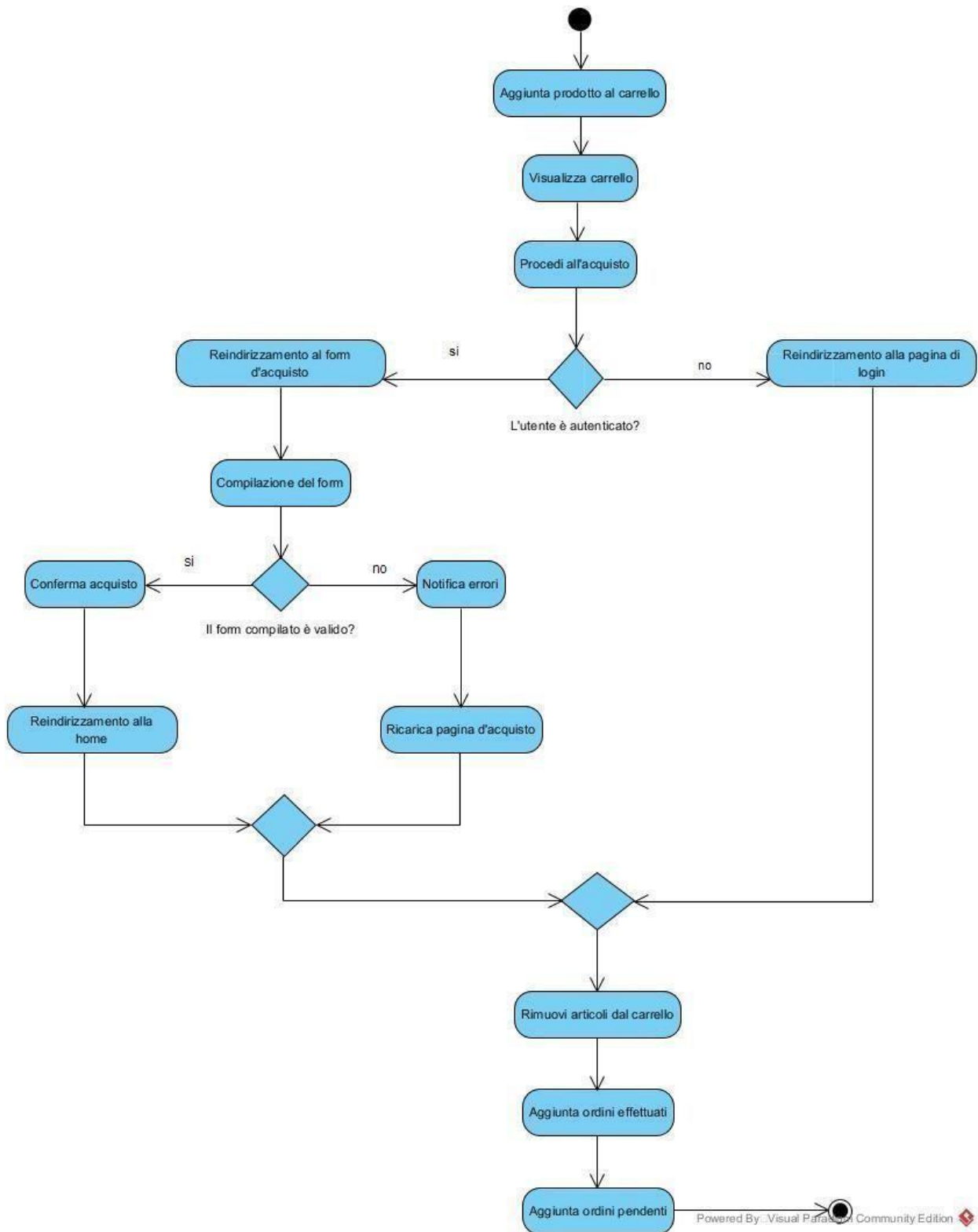
### AD 1: Richiesta prodotto al fornitore



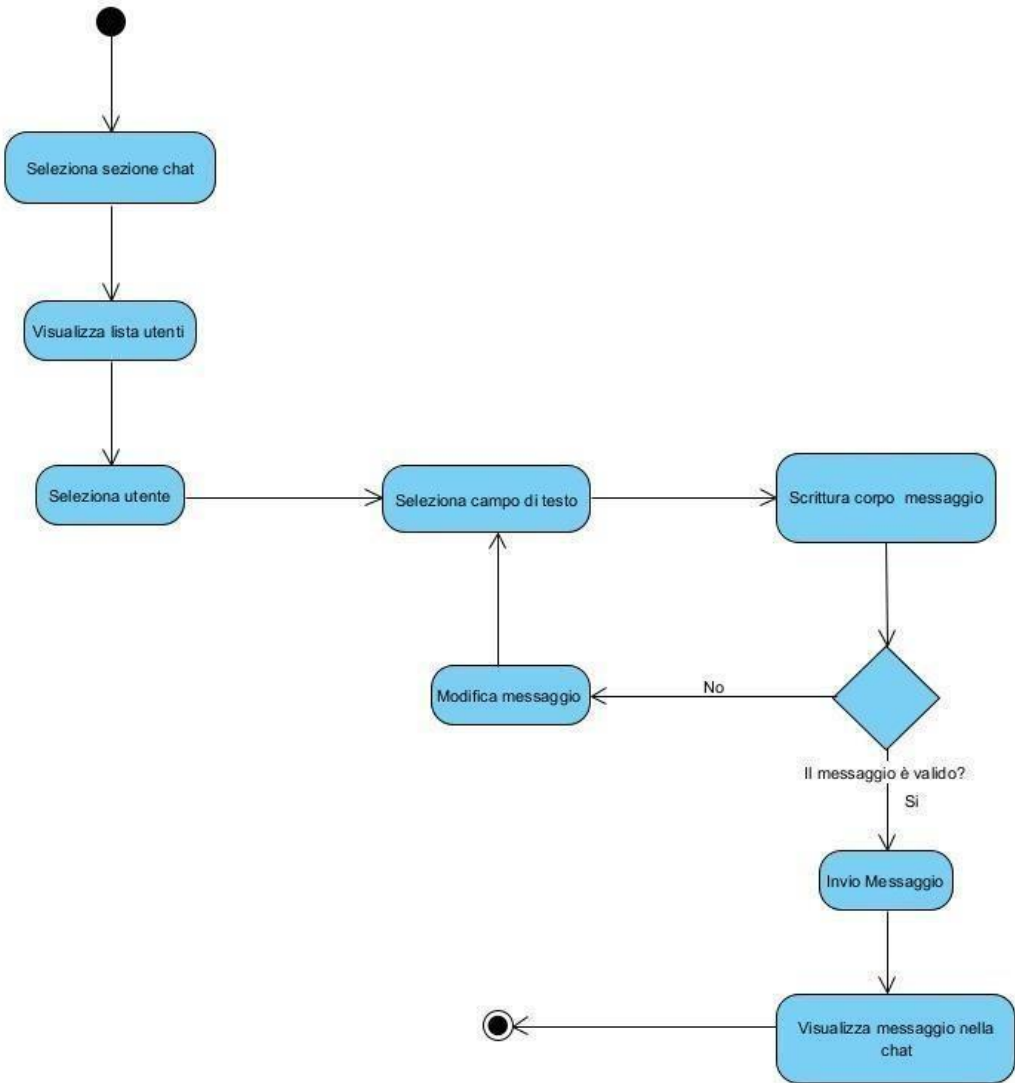
## AD 2: Gestione dell'ordine



### AD 3: Acquisto di un prodotto



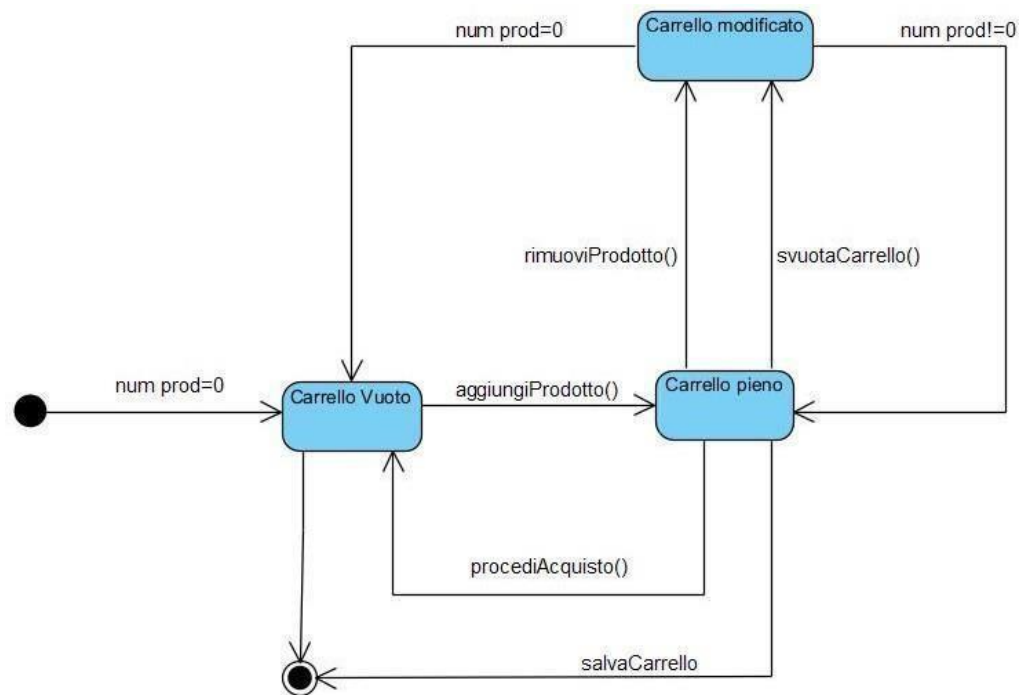
AD 4: Invio di un messaggio



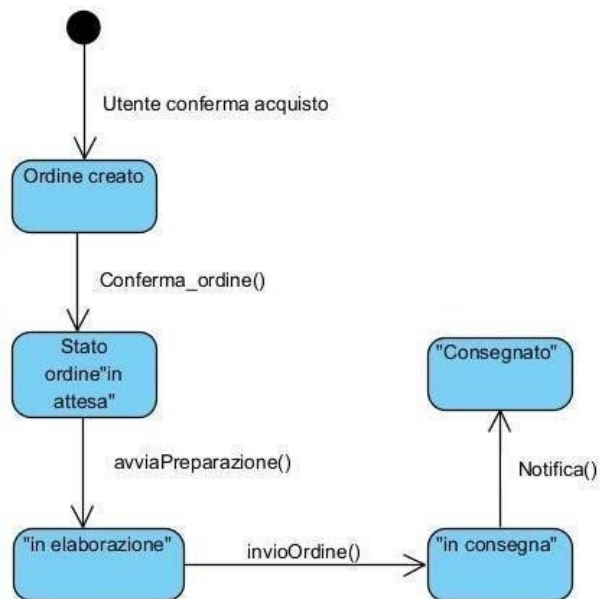


## State machine diagrams

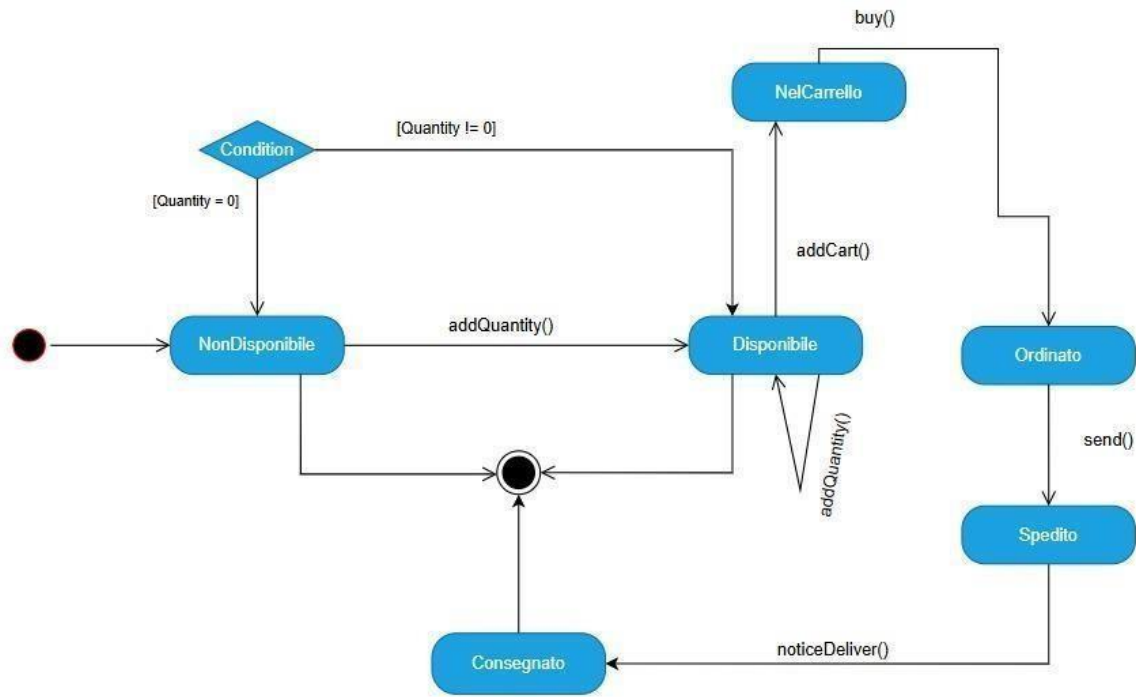
### SM 1: Carrello



### SM 2: Ordine

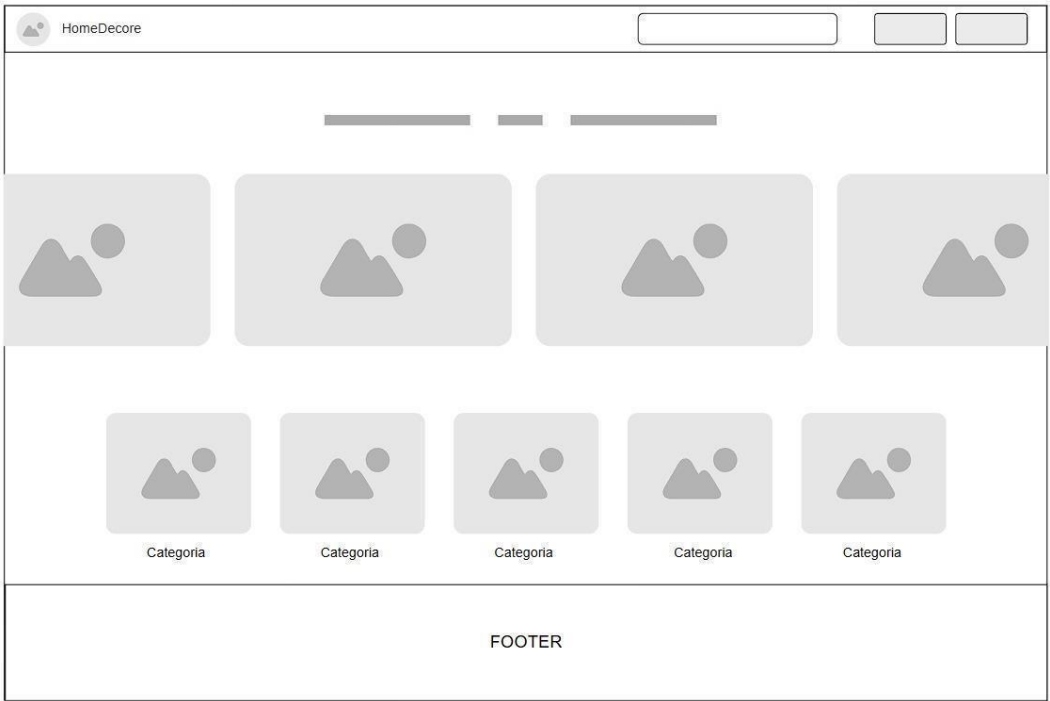


**SM 3: Prodotto**

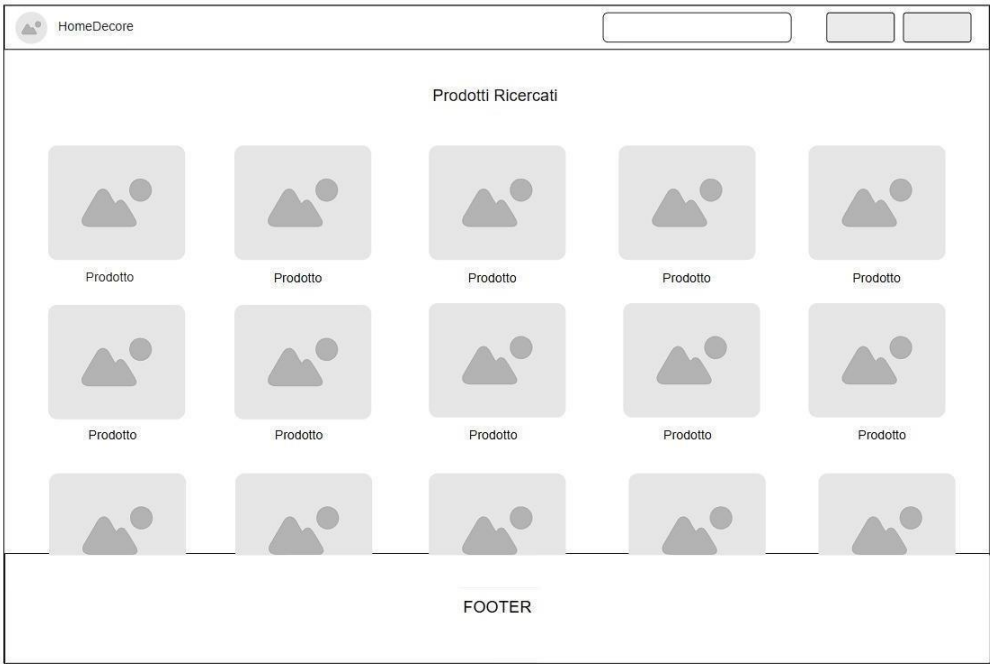


3.3.5. User interface – navigational paths and screen mock-ups

Homepage



Visualizzazione Prodotti



Login

 HomeDecore

Placeholder

Email

Placeholder

Password

Login

FOOTER

## 4. Glossario

Requisito Funzionale: Funzionalità offerte dal sistema.

Requisito non funzionale: Descrivono caratteristiche del sistema non legate al suo comportamento

Use case diagram: diagramma utilizzato per comprendere le funzionalità principali degli utenti. Sono utilizzate per capire quali sono i requisiti del sistema.

Sequence Diagram: è un diagramma previsto dall'UML utilizzato per descrivere uno scenario

Statechart Diagram: è un diagramma UML per descrivere il comportamento del valore delle entità a seconda di eventi, come interazioni dell'utente. Vengono tracciate le transizioni: cambio dei valori.

Mock up: realizzazione puramente a scopo illustrativo per far comprendere quale sarà l'ambiente in cui si trova l'utente.

Form: un'interfaccia di un programma che permette ad un utente di inserire e inviare dati.