

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

**Future Forge Gear  
Requirements Analysis Document  
Versione 1.1**



Data: 11/11/2025

Progetto: Nome Progetto	Versione: X.Y
Documento: Titolo Documento	Data: GG/MM/AAAAA

### Coordinatore del progetto:

Nome	Matricola

### Partecipanti:

Nome	Matricola
Francesco Mascolo	0512117352
Francesco Lamanna	0512117166
Luigi Mazza	0512118612

Scritto da:	Francesco Mascolo
-------------	-------------------

### Revision History

Data	Versione	Descrizione	Autore
20/10/2025	1.0	Analisi del sistema attraverso requisiti funzionali e non funzionali, accompagnata da una bozza iniziale dei diagrammi dei casi d'uso. Aggiunta di Mock-up	Full Team
10/11/2025	1.1	Sviluppo di: - object model tramite class diagram - dynamic model tramite sequence diagram, activitydiagram e state machine diagram	Full Team

# Indice

1 Introduzione .....	4
1.1. Purpose Of System.....	4
1.2. Scope Of System .....	4
1.3. Objectives And Success Criteria Of The Project .....	5
2 Current System .....	6
3 Proposed System .....	6
3.1. Requisiti Funzionali .....	7
3.2. Non Functional Requirements.....	9
3.2.1. Usability .....	9
3.2.2. Reliability .....	9
3.2.3. Performance.....	10
3.2.4. Supportability .....	10
3.2.5. Implementation .....	10
3.2.6. Interface .....	10
3.2.7. Packaging .....	11
3.2.8. Legal .....	11
3.3. System Model .....	11
3.3.1. Scenari .....	11
3.3.2. Use Case Model.....	15
3.3.3 Object Model.....	25
3.3.4 Dynamic Model.....	28
3.3.5. User Interface-Navigational Paths and Screen Mock-Ups.....	36
4. Glossario .....	38

# 1 INTRODUZIONE

## 1.1. Purpose of system

Future Forge Gear intende realizzare una nuova piattaforma e-commerce avanzata per ottimizzare la vendita online di prodotti informatici, con particolare attenzione ai PC fissi preassemblati per il gaming e l'ambito professionale. Il sistema sarà progettato per offrire un'esperienza utente fluida e personalizzata, supportando una gestione efficiente del catalogo, del magazzino, delle spedizioni e delle relazioni con i fornitori. L'obiettivo è garantire flessibilità, scalabilità e semplicità d'uso, così da soddisfare le esigenze di una clientela variegata composta da privati, professionisti e rivenditori, migliorando al contempo la competitività e la capacità di espansione sul mercato.

## 1.2 Scope of System

Il sistema Future Forge Gear è pensato per la realizzazione di una piattaforma e-commerce dedicata alla vendita di prodotti informatici, con particolare enfasi sui PC fissi preassemblati per il gaming e l'ambito professionale. L'obiettivo è offrire un'interfaccia utente semplice e reattiva, capace di garantire un'esperienza d'acquisto fluida e personalizzata. Le funzionalità principali includeranno: La gestione dinamica del catalogo prodotti, suddiviso per categorie come computer, componenti hardware, periferiche e accessori. La possibilità per gli utenti di registrarsi, modificare il proprio profilo, consultare le schede tecniche, lasciare recensioni e valutazioni. Un sistema di carrello virtuale con checkout integrato, compatibile con i principali servizi di pagamento online. Strumenti di amministrazione per la gestione del catalogo, delle promozioni, dei profili utente e delle richieste di assistenza. Il sistema sarà progettato per essere compatibile con dispositivi mobili e desktop, supportando i browser più diffusi. Verranno implementate misure di sicurezza per la protezione dei dati sensibili, come la cifratura delle credenziali e la mitigazione di vulnerabilità note (es. SQL injection). La piattaforma sarà sviluppata su architettura web, utilizzando tecnologie come Java, JSP/Servlet, MySQL e sarà distribuita tramite server Tomcat. Le funzionalità escluse da questa prima versione comprendono il supporto multilingua e una gestione avanzata del magazzino, che potranno essere introdotte in fasi successive. Il focus iniziale sarà sulla navigazione del catalogo, sull'ottimizzazione del processo di acquisto, con l'obiettivo di incrementare le vendite e rafforzare la fiducia del cliente nel brand.

## 1.3 Objectives and success criteria of the project

### Objectives

L'obiettivo principale del progetto Future Forge Gear è la realizzazione di una piattaforma e-commerce moderna e funzionale per la vendita online di prodotti informatici, con particolare attenzione ai PC fissi preassemblati per il gaming e l'ambito professionale. Il sistema dovrà garantire un'esperienza utente fluida, intuitiva e personalizzata. Gli obiettivi specifici includono:

1. **Usabilità e navigazione semplificata** Progettare un'interfaccia utente chiara e reattiva, che consenta una facile esplorazione del catalogo suddiviso per categorie e specifiche tecniche.
2. **Gestione del carrello e acquisto semplificato** Implementare un sistema di carrello virtuale che consenta l'aggiunta, la modifica e la rimozione dei prodotti, con un processo di checkout rapido e intuitivo.
3. **Integrazione con sistemi di pagamento online** Supportare metodi di pagamento sicuri e diversificati, compatibili con le principali piattaforme digitali.
4. **Sicurezza e protezione delle informazioni** Adottare misure di sicurezza avanzate come la cifratura delle password, la protezione dei dati sensibili e la prevenzione contro attacchi comuni (es. SQL injection).
5. **Gestione amministrativa del sistema** Fornire agli amministratori strumenti per aggiornare il catalogo, monitorare gli ordini, gestire gli utenti registrati e controllare le richieste di assistenza.

### Success Criteria

Il successo del progetto Future Forge Gear sarà valutato in base al raggiungimento dei seguenti obiettivi:

1. **Esperienza utente semplificata e navigazione efficace** La piattaforma sarà considerata efficace se gli utenti riusciranno a esplorare facilmente il catalogo, selezionare i prodotti desiderati e completare l'acquisto in pochi passaggi, senza difficoltà.
2. **Operatività delle funzionalità principali** Il sistema sarà ritenuto completo quando tutte le funzioni essenziali — come la gestione del carrello e il processo di checkout — saranno pienamente implementate, testate e funzionanti.
3. **Protezione dei dati e sicurezza informatica** Il progetto sarà considerato sicuro se verranno integrate misure di protezione efficaci, tra cui la cifratura delle credenziali, la tutela dei dati sensibili e la difesa contro attacchi informatici comuni come SQL injection e brute force.
4. **Efficienza e capacità di crescita** Il sistema dovrà dimostrare buone prestazioni anche in presenza di un elevato numero di utenti e prodotti, confermando la sua scalabilità e la capacità di supportare l'espansione dell'attività.

## **2 Current System**

La piattaforma e-commerce di Future Forge Gear è attualmente in fase di sviluppo preliminare, ma sono già emerse diverse problematiche che necessitano di interventi mirati per garantire un'esperienza d'acquisto moderna ed efficiente. Le analisi iniziali hanno evidenziato criticità nel processo di acquisto, in particolare nella gestione del carrello e nella fluidità del checkout, che risulta ancora troppo articolato. L'interfaccia utente necessita di un miglioramento sostanziale per risultare più intuitiva e coerente, soprattutto in ottica di compatibilità con dispositivi mobili e desktop. La sezione dedicata ai prodotti è attualmente statica e non offre funzionalità interattive come filtri avanzati, suggerimenti personalizzati o strumenti di comparazione, limitando la capacità dell'utente di trovare rapidamente ciò che cerca. Inoltre, manca un sistema di comunicazione diretta tra cliente e assistenza, rendendo difficile la risoluzione tempestiva di problematiche legate agli ordini o ai prodotti. Un'altra area ancora in fase di definizione riguarda l'introduzione di un configuratore per PC personalizzati, che permetta agli utenti di selezionare e combinare componenti in base alle proprie esigenze. Questo strumento dovrà garantire la compatibilità tra i vari elementi, aggiornare dinamicamente il prezzo e integrarsi con il sistema di magazzino per mostrare la disponibilità in tempo reale. L'obiettivo è quindi quello di superare le limitazioni attuali, implementando una piattaforma più interattiva, accessibile e orientata alla personalizzazione, capace di migliorare sia l'esperienza utente che l'efficienza operativa.

## **3 Proposed System**

Il carrello diventa uno strumento più fluido e interattivo. Gli utenti potranno gestire i propri prodotti – aggiungendo, eliminando o modificando le quantità – e vedere le modifiche applicarsi all'istante, senza necessità di ricaricare la pagina.

L'obiettivo è rendere il processo di pagamento il più lineare possibile, eliminando gli intoppi. Il flusso sarà ottimizzato per richiedere il minor numero di passi, garantendo una conferma dell'ordine rapida e semplice. Grande attenzione sarà data agli utenti non registrati, che potranno completare l'acquisto in massima libertà e con la stessa efficienza.

### **3.1. REQUISITI FUNZIONALI**

#### **Utente**

##### **RF1**

Il sistema deve generare un riepilogo dell'ordine che includa i dettagli dei prodotti, il costo totale, le spese di spedizione e le tasse applicate.

##### **RF2**

Il sistema deve offrire la possibilità di salvare il contenuto del carrello, così che l'utente possa riprendere l'acquisto in un momento successivo.

##### **RF3**

Il sistema deve permettere all'utente di selezionare una modalità di pagamento tra quelle disponibili (ad esempio carta di credito, PayPal o bonifico bancario).

##### **RF4**

Il sistema deve consentire agli utenti di configurare un computer personalizzato, selezionando manualmente i componenti hardware desiderati tra quelli disponibili nel catalogo.

##### **RF5**

Il sistema deve consentire agli utenti di accedere in qualsiasi momento alla sezione "Carrello" per consultare gli articoli selezionati.

##### **RF6**

Il sistema deve inviare una notifica all'utente per confermare l'avvenuta ricezione dell'ordine.

##### **RF7**

Il sistema deve guidare l'utente durante la procedura di checkout, richiedendo le informazioni di spedizione e consentendo la scelta della modalità di consegna.

##### **RF8**

Il sistema deve mostrare per ciascun prodotto nel carrello informazioni come nome, quantità e prezzo unitario.

##### **RF9**

Il sistema deve permettere all'utente di aggiornare la quantità di un articolo o di rimuoverlo completamente dal carrello.

##### **RF10**

Il sistema deve ricalcolare automaticamente il totale complessivo ogni volta che viene effettuata una modifica nel carrello

## **Gestore di magazzino**

### **RF11**

Il sistema deve offrire la possibilità di modificare le informazioni generali del magazzino, come nome, sede o dettagli di contatto.

### **RF12**

Il sistema deve consentire a ciascun gestore di visualizzare l'intero elenco dei prodotti presenti nel proprio magazzino.

### **RF13**

Il sistema deve fornire la possibilità di aggiungere nuovi articoli al magazzino, definendone tutte le caratteristiche principali.

### **RF14**

Il sistema deve permettere al gestore di modificare il prezzo dei prodotti presenti in catalogo.

### **RF15**

Il sistema deve consentire ai gestori di inoltrare richieste di rifornimento ai fornitori per i prodotti esauriti o con disponibilità limitata.

### **RF16**

Il sistema deve consentire l'aggiornamento della descrizione associata ai prodotti del magazzino.

### **RF17**

Il sistema deve permettere al gestore di assegnare gli ordini a un responsabile delle spedizioni per la gestione delle consegne.

### **RF18**

Il sistema deve permettere al gestore di modificare il nome dei prodotti già presenti nell'inventario.

## **Gestore degli ordini**

### **RF19**

Il sistema deve permettere al gestore di aggiornare lo stato di ciascun ordine, variandolo in base all'avanzamento del processo (es. "in preparazione", "spedito", "consegnato").

### **RF20**

Il sistema deve consentire a ciascun gestore degli ordini di visualizzare la lista completa degli ordini assegnati al proprio profilo.

## **Fornitore**

### **RF21**

Il sistema deve consentire al fornitore di gestire il proprio catalogo prodotti, permettendo l'aggiunta di nuovi articoli o la rimozione di quelli non più disponibili.

### **RF22**

Il sistema deve offrire al fornitore la possibilità di accettare o rifiutare le richieste di rifornimento inviate dai gestori dei magazzini.

### **RF23**

Il sistema deve consentire a ogni fornitore di visualizzare tutte le richieste di approvvigionamento provenienti dai magazzini registrati.

## **3.2. Non functional requirements**

### **3.2.1. Usability**

**Facilità d'Uso:** Il portale deve offrire un'esperienza semplice e comprensibile anche per utenti con competenze tecniche limitate, facilitando la navigazione e l'interazione con il catalogo.

**Design Responsive:** Il layout e le funzionalità devono essere ottimizzati per funzionare correttamente su qualsiasi dispositivo, inclusi smartphone, tablet e computer, garantendo una visualizzazione coerente e reattiva.

**Documentazione e supporto:** Il sistema dovrà includere una sezione di supporto con istruzioni dettagliate sull'utilizzo delle principali funzionalità, così da accompagnare l'utente durante l'intero processo d'acquisto.

### **3.2.2. Reliability**

**Operatività continua:** La piattaforma deve garantire una disponibilità costante, con un uptime minimo del 99,9% su base annuale, per assicurare continuità nel servizio.

**Gestione dei malfunzionamenti:** In caso di errori o interruzioni, il sistema dovrà essere in grado di recuperare autonomamente o segnalare il problema, evitando la perdita di dati rilevanti.

**Salvataggio dei dati:** È necessario implementare un sistema di backup periodico per proteggere le informazioni da eventuali guasti o eventi critici, assicurando la possibilità di ripristino.

### **3.2.3. Performance**

**Tempo di risposta:** Le pagine del sito devono essere visualizzabili entro un massimo di 2 secondi per utenti con connessioni standard, garantendo un'esperienza fluida e reattiva. Il backend dovrà essere ottimizzato per gestire almeno 1000 richieste al secondo in condizioni operative normali.

**Scalabilità:** L'infrastruttura del sistema deve essere progettata per adattarsi a un aumento progressivo del numero di utenti e della mole di dati, mantenendo stabilità e reattività anche durante periodi di traffico intenso come campagne promozionali o eventi stagionali.

### **3.2.4. Supportability**

**Facilità di manutenzione** Il codice sorgente deve essere organizzato e documentato in modo da agevolare interventi futuri di aggiornamento o correzione. Ogni componente software dovrà rispettare i principi di progettazione SOLID e seguire un'architettura modulare per favorire la riusabilità e la scalabilità.

**Assistenza tecnica** Il sistema dovrà prevedere un servizio di supporto tecnico attivo durante l'orario lavorativo (9:00–18:00), con tempi di risposta garantiti entro 24 ore dalla segnalazione, per assicurare continuità operativa e risoluzione tempestiva delle problematiche.

### **3.2.5. Implementation**

**Tecnologie:** Il sistema deve essere implementato utilizzando le tecnologie Java EE, del database MySQL e del server applicativo Apache Tomcat. La generazione delle pagine web dinamiche sarà affidata a JSP (JavaServer Pages), mentre per l'Inversion of Control e l'integrazione dei componenti sarà utilizzato CDI (Contexts and Dependency Injection).

**Ambiente di sviluppo:** Per lo sviluppo del codice, il team dovrà utilizzare in modo uniforme l'ambiente di

sviluppo integrato (IDE) IntelliJ IDEA. Il controllo di versione del codice sorgente sarà gestito tramite Git, con il repository remoto ospitato sulla piattaforma GitHub.

### 3.2.6. Interface

**Interfaccia utente:** La piattaforma dovrà disporre di un'interfaccia utente (UI) uniforme, progettata seguendo gli standard del web design moderno. Gli elementi grafici e di interazione dovranno essere realizzati per rendere l'esperienza di navigazione e di acquisto semplice e immediata per l'utente finale.

**Interfaccia di integrazione:** Il sistema dovrà essere predisposto per l'interconnessione con servizi esterni, in particolare con piattaforme di pagamento online e con i sistemi logistici dei corrieri per la gestione delle spedizioni.

**Browser compatibility:** L'interfaccia utente dovrà garantire un corretto funzionamento e una visualizzazione ottimale sui browser web più diffusi (quali Chrome, Firefox, Edge e Safari), con supporto privilegiato per le loro ultime versioni.

### 3.2.7. Packaging

**Distribuzione:** L'applicazione finale dovrà essere assemblata in un unico file .war (Web Application Archive), idoneo per la distribuzione e il deployment su un server Tomcat. All'interno del pacchetto dovrà essere presente un file di configurazione per agevolare e automatizzare le operazioni di installazione sul server.

**Installazione:** Dovrà essere disponibile una documentazione tecnica dettagliata che illustri le procedure di installazione e configurazione del sistema. La guida dovrà includere, tra le altre cose, i passaggi per la creazione e il popolamento del database e per la messa a punto del server applicativo.

### 3.2.8. Legal

**Conformità GDPR:** La piattaforma dovrà essere progettata e sviluppata in piena conformità con il Regolamento Generale sulla Protezione dei Dati (GDPR). Dovranno essere implementati appositi meccanismi per raccogliere e registrare il consenso esplicito e informato degli utenti al trattamento dei loro dati personali.

**Proprietà intellettuale:** È necessario adottare tutte le misure ragionevoli per assicurare che i materiali pubblicati sulla piattaforma, come fotografie, testi descrittivi e loghi, non infrangano diritti d'autore, marchi registrati o altri diritti di proprietà intellettuale di terzi.

**Termini di servizio:** Il sito dovrà rendere facilmente accessibili all'utente i Termini e Condizioni di utilizzo del servizio e l'Informativa sulla Privacy, redatti in conformità con la normativa italiana ed europea vigente.

## **3.3 System Model**

### **3.3.1 Scenari**

#### **CLIENTE**

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

Luigi è alla ricerca di un sito affidabile dove poter acquistare componenti e periferiche per aggiornare il proprio computer da gaming, cercando prodotti di qualità a un prezzo competitivo e con un'esperienza d'acquisto semplice e veloce.

Dopo alcune ricerche, decide di visitare la piattaforma Future Force Gear, dove trova un ampio catalogo di PC preassemblati, accessori e parti hardware.

Per rendere la ricerca più mirata, Luigi utilizza i filtri di navigazione messi a disposizione dal sistema: seleziona la categoria di interesse (come "schede video" o "PC da gaming") e imposta un intervallo di prezzo in base al proprio budget, visualizzando così solo i prodotti più adatti alle sue esigenze.

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

Dopo aver applicato i filtri di ricerca, Luigi individua rapidamente i prodotti che rispondono alle sue necessità. Selezionandone uno, il sistema apre la pagina dedicata dove vengono presentate tutte le informazioni tecniche e descrittive del componente scelto, incluse le specifiche, le immagini e la disponibilità in magazzino.

Da questa sezione, Luigi può decidere se aggiungere l'articolo direttamente nel carrello, scegliendo la quantità desiderata prima di procedere con l'ordine.

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

Dopo aver esplorato a lungo le varie sezioni del sito Future Force Gear, Luigi decide di aggiungere al carrello alcuni componenti hardware che ha trovato interessanti. Una volta terminata la selezione, accede alla pagina del carrello per verificare i prodotti scelti prima di procedere all'acquisto. Cliccando sull'apposita icona, il sistema mostra un riepilogo dettagliato degli articoli aggiunti, includendo nome, quantità, prezzo unitario e totale parziale. Mentre controlla la lista, Luigi si accorge che uno dei prodotti non gli piace e decide di rimuoverlo. Con un semplice clic sull'icona "Elimina", l'articolo viene eliminato e il totale dell'ordine si aggiorna automaticamente.

Dopo aver sistemato il contenuto del carrello, Luigi può scegliere se continuare a esplorare il catalogo per aggiungere nuovi prodotti oppure procedere con il pagamento. Il sistema conserva automaticamente lo stato del carrello, permettendogli di tornare in un secondo momento per completare l'ordine senza perdere i dati precedentemente inseriti

## **Acquisto**

Dopo alcuni giorni di riflessione, il cliente decide di tornare sul sito Future Force Gear per completare l'acquisto dei prodotti precedentemente salvati nel carrello. Una volta effettuato l'accesso, verifica che tutti gli articoli siano ancora disponibili e, soddisfatto della selezione, clicca sul pulsante "Procedi all'acquisto" per iniziare la fase di checkout. Il sistema lo indirizza a una pagina dedicata in cui deve inserire le informazioni di spedizione, specificando l'indirizzo, il tipo di consegna desiderato e, se necessario, l'opzione di spedizione veloce. Successivamente, Luigi sceglie la modalità di pagamento tra quelle proposte dal sito, come carta di credito, PayPal o pagamento alla consegna. Prima di confermare l'ordine, la piattaforma mostra un riepilogo completo dell'acquisto, comprendente il dettaglio dei prodotti, il costo della spedizione e il totale finale con le tasse incluse. Dopo aver controllato con attenzione i dati, Luigi conferma la sua decisione cliccando su "Conferma ordine". Il sistema visualizza infine una notifica di conferma, informando Luigi che l'acquisto è andato a buon fine e che riceverà aggiornamenti sullo stato della spedizione.

## **Accesso e Registrazione**

Durante la navigazione sul sito Future Force Gear, Luigi trova un prodotto che desidera acquistare e tenta di completare l'ordine. Tuttavia, il sistema gli segnala che la procedura di acquisto è riservata agli utenti registrati.

Per proseguire, Luigi accede alla sezione dedicata al login e alla registrazione e decide di creare un nuovo account. Il sistema gli richiede di compilare un modulo di registrazione, inserendo le informazioni di base come nome, indirizzo email e una password di accesso.

Una volta completata la registrazione, Luigi effettua il login e ottiene pieno accesso alle funzionalità riservate agli utenti autenticati, potendo così continuare la navigazione, gestire il proprio profilo e procedere con l'acquisto del prodotto selezionato.

## **Gestore del Magazzino**

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

Roberto, responsabile del magazzino di Salerno, accede alla piattaforma con l'obiettivo di consultare l'elenco dei prodotti attualmente presenti nel proprio inventario. Il sistema mostra una lista dettagliata di tutti gli articoli disponibili, completa di informazioni come nome, descrizione, prezzo e quantità in stock. Da questa sezione, Roberto può modificare le informazioni relative a ciascun prodotto, come aggiornare il prezzo o la disponibilità, oppure rimuovere un articolo dal catalogo rendendolo temporaneamente non acquistabile dai clienti. Nel caso in cui uno dei prodotti risulti esaurito, il sistema consente a Roberto di inoltrare una richiesta di rifornimento direttamente al fornitore, garantendo così una gestione più efficiente del magazzino e una migliore continuità di approvvigionamento.

### **Aggiunta Prodotti con richiesta al Fornitore**

Durante la consultazione dell'elenco dei prodotti del proprio magazzino, Roberto nota che alcuni articoli risultano quasi esauriti o completamente non disponibili. Per risolvere la situazione, decide di

individuare il fornitore associato a ciascun prodotto e contattarlo direttamente tramite la piattaforma Future Force Gear, specificando la quantità di cui necessita per il rifornimento.

Inoltre, Roberto intende ampliare il catalogo del suo magazzino introducendo un nuovo prodotto. A tal fine, accede alla sezione dedicata ai fornitori e consulta l'elenco completo degli articoli disponibili, applicando filtri di ricerca per individuare quelli più adatti alle esigenze del punto vendita.

Dopo aver contattato il fornitore e concordato la fornitura, Roberto può procedere con l'aggiunta del nuovo prodotto all'interno dell'inventario, definendone tutte le caratteristiche, come nome, descrizione, prezzo e quantità iniziale.

## **Assegnamento di ordini**

A seguito dell'aumento delle vendite, numerosi clienti effettuano acquisti dal magazzino gestito da Carlo, che si trova quindi a dover coordinare un elevato numero di ordini. Tramite l'apposita sezione gestionale, il sistema gli presenta una tabella riepilogativa contenente tutti gli ordini effettuati, ordinati cronologicamente dal più recente al meno recente.

Per ciascun ordine, Carlo può visualizzare i dettagli completi dell'acquisto, tra cui i prodotti richiesti, le quantità e l'importo totale. Una volta verificati i dati, procede all'assegnazione delle spedizioni, selezionando per ogni ordine uno dei gestori presenti nella piattaforma.

Dopo aver completato l'assegnazione, Carlo ha la possibilità di monitorare lo stato di avanzamento di ogni spedizione direttamente dalla tabella, visualizzando in tempo reale l'evoluzione delle consegne.

## **Fornitori**

### **Visualizzazione richieste dei prodotti**

Vincenzo, fornitore di componenti informatici registrato sulla piattaforma Future Force Gear, accede alla sezione dedicata alle richieste di approvvigionamento provenienti dai vari magazzini. In questa pagina, il sistema mostra un elenco dettagliato di tutte le richieste ricevute, indicando per ciascuna:

il nome del magazzino che ha inoltrato la richiesta, il tipo di prodotto richiesto e lo stato della richiesta, che può essere aggiornato tramite l'apposito pulsante "Conferma" una volta completata l'operazione.

Per gestire le richieste in modo più efficiente, Vincenzo ordina l'elenco in base alla data di invio, così da dare priorità a quelle in sospeso da più tempo. Dopo aver verificato e soddisfatto una richiesta, il sistema la rimuove automaticamente dall'elenco, mantenendo visibili solo quelle ancora da evadere.

## **Gestore dell'ordine**

### **Visualizzazione ordini assegnati**

Giacomo, gestore delle spedizioni, accede alla sezione della piattaforma dedicata alla gestione degli ordini provenienti dai magazzini. Da qui può consultare una lista completa degli ordini assegnati, accompagnata da tutte le informazioni principali, come l'ID dell'ordine, la data di creazione, il nome del magazzino che lo ha generato, l'elenco dei prodotti inclusi e lo stato attuale di ciascun ordine.

Per organizzare meglio il lavoro, Giacomo utilizza i filtri di ordinamento, visualizzando gli ordini in ordine cronologico dal meno recente al più recente, così da gestire prima quelli rimasti in sospeso da più tempo.

## **Aggiornamento dello stato dell'ordine**

Dopo aver individuato gli ordini prioritari, Giacomo inizia a lavorare sul primo disponibile. Accedendo ai dettagli dell'ordine, aggiorna lo stato da “In attesa” a “In preparazione”, indicando che il processo di spedizione è stato avviato.

Completate le operazioni di imballaggio e spedizione, Giacomo torna nella sezione di gestione e imposta lo stato su “Spedito”. Una volta confermata la consegna del pacco al cliente finale, il gestore aggiorna nuovamente l'ordine segnalandolo come “Consegnato”.

In questo modo, il sistema mantiene una tracciabilità completa di ogni ordine, consentendo di monitorare in tempo reale l'avanzamento di tutte le spedizioni.

## **Scenario comune all'utilizzatore**

### **Visualizzazione profilo**

Ogni utente registrato sulla piattaforma Future Force Gear può accedere alla propria area personale per consultare le informazioni del profilo e visualizzare i dati di altri utenti presenti nel sistema, indipendentemente dal loro ruolo.

Nella parte superiore della pagina sono riportate le informazioni generali, comuni a tutte le categorie di utenti, come nome, cognome ed indirizzo email — elementi essenziali per consentire l'interazione tra i diversi profili.

A seguire, il sistema mostra una sezione dedicata alle informazioni specifiche in base alla tipologia di utente:

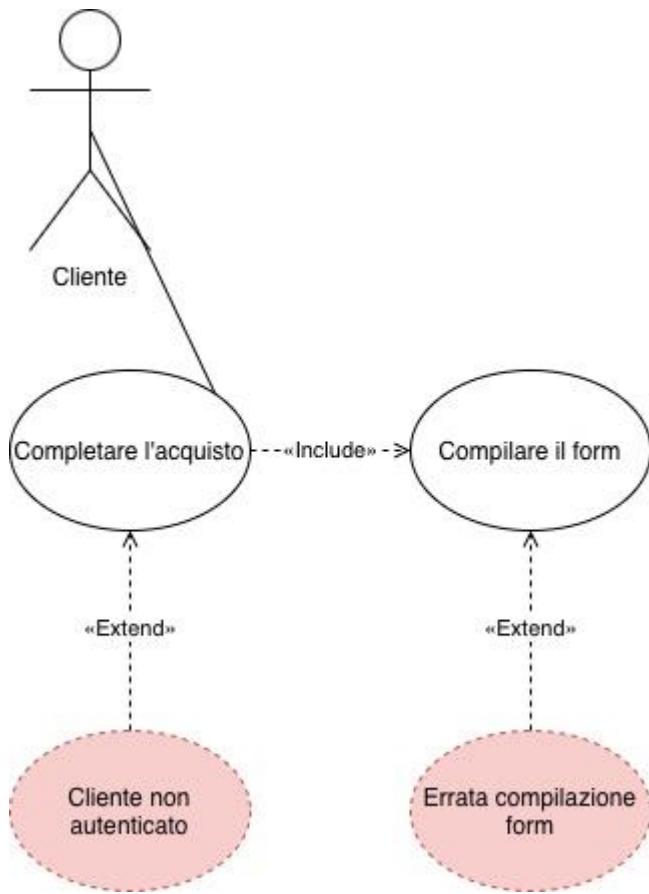
Se l'utente è un cliente, vengono visualizzati tutti gli ordini effettuati, con l'elenco dei prodotti acquistati. Per un gestore di magazzino, sono mostrati i dettagli del magazzino di competenza, inclusa la lista dei prodotti in vendita e le relative quantità.

Nel caso di un fornitore, la pagina presenta l'elenco dei prodotti disponibili per il rifornimento dei magazzini.

Infine, se l'utente è un gestore delle spedizioni, vengono elencati gli ordini gestiti, insieme al loro stato aggiornato.

### 3.3.2. Use case model

#### 3.3.2.1 Acquisto di un prodotto



#### Pre-condizioni:

- L'utente ha aggiunto alcuni prodotti nel proprio carrello della spesa.
- L'utente si trova nella pagina in cui può vedere il contenuto del carrello.

#### Flusso di eventi:

1. L'utente preme il pulsante "Procedi all'acquisto" e viene portato alla pagina per completare l'ordine.
2. L'utente riempie tutti i campi del modulo con le informazioni richieste per la spedizione e il pagamento.
3. L'utente clicca per confermare e finalizzare l'acquisto.
4. Il sistema mostra una pagina di riepilogo che conferma che l'ordine è andato a buon fine e offre un link per tornare alla pagina principale.

#### Flussi alternativi:

- Se l'utente non ha fatto l'accesso: Quando clicca

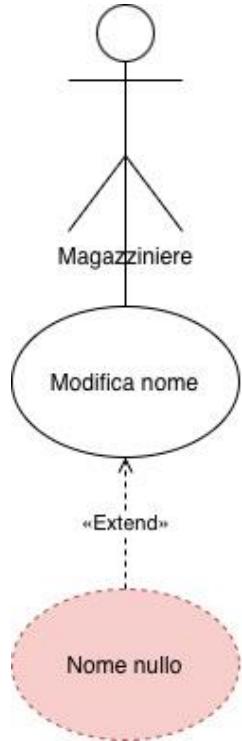
"Procedi all'acquisto", viene invece reindirizzato alla schermata di Login per autenticarsi.

- Se ci sono errori nel modulo: Se l'utente inserisce dei dati in un formato non valido o lascia campi obbligatori vuoti, il sistema blocca l'invio. Ricarica la stessa pagina di acquisto, mostrando un messaggio di errore vicino ai campi che devono essere corretti.

#### Post condizioni:

- Il sistema svuota il carrello dell'utente.
- Il sistema aggiunge l'elenco dei prodotti acquistati nella sezione "I miei ordini" del profilo utente.
- Il sistema registra il nuovo ordine nella lista degli "Ordini in attesa".

### 3.3.2.2 Modifica nome di un Prodotto



**Attori:** Magazziniere

**Pre-condizioni:**

- Il magazziniere ha effettuato l'accesso al sistema.
- Il magazziniere ha scelto il prodotto da modificare dalla schermata di gestione del catalogo.

**Flusso di eventi:**

1. Il sistema mostra il nome del prodotto selezionato in un campo che può essere modificato.
2. Il magazziniere cambia il nome del prodotto digitando quello nuovo.
3. Il magazziniere conferma la modifica premendo il pulsante "Salva".
4. Il sistema aggiorna il nome del prodotto all'interno del database.

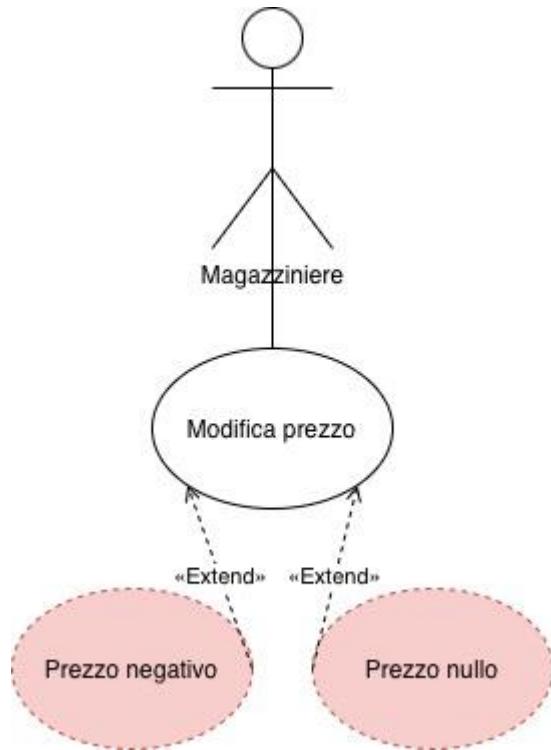
**Flusso Alternativo**

- **Se il nome non è valido:** Se il magazziniere lascia il campo del nome vuoto, il sistema non salva le modifiche e mostra un avviso che richiede di inserire un nome valido per il prodotto.

**Post condizioni:**

- Il nuovo nome del prodotto viene memorizzato nel database.
- Tutti i clienti vedranno il nome aggiornato del prodotto quando visualizzeranno il catalogo.

### 3.3.2.3 Modifica Prezzo di un Prodotto



**Attori:** Magazziniere

**Pre-condizioni:**

- Il magazziniere ha effettuato l'accesso al sistema.
- Il magazziniere ha scelto il prodotto da modificare dalla schermata di gestione del catalogo.

**Flusso di eventi:**

1. Il sistema mostra il prezzo del prodotto selezionato in un campo modificabile.
2. Il magazziniere inserisce il nuovo importo per il prodotto.
3. Il magazziniere completa l'operazione premendo il pulsante di conferma.
4. Il sistema aggiorna il prezzo del prodotto all'interno del database.

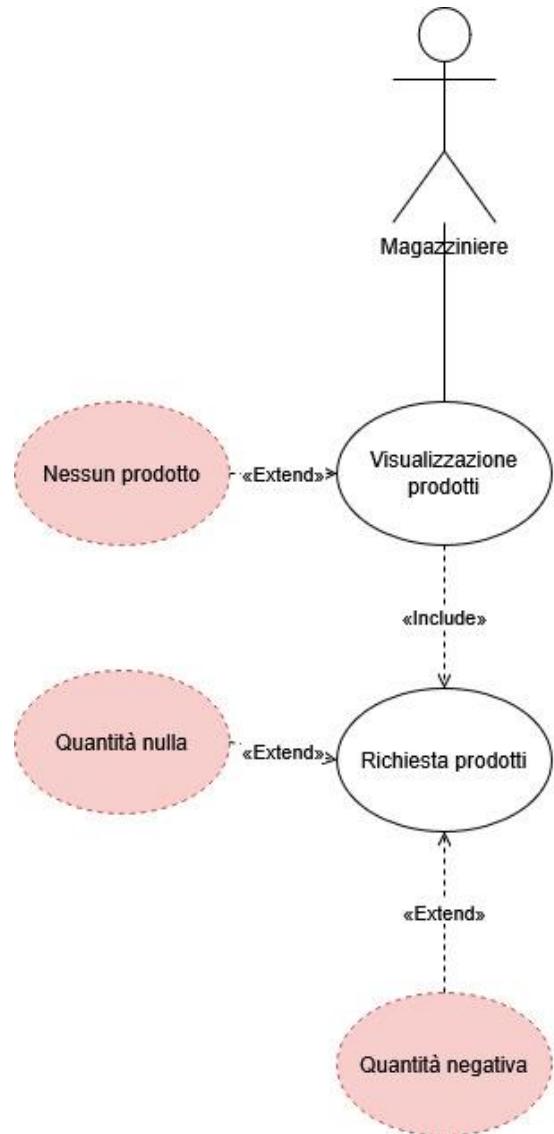
**Flussi alternativi:**

- Se il magazziniere imposta un prezzo con un valore negativo, il sistema blocca l'operazione e chiede di inserire un prezzo valido (maggiore o uguale a zero).
- Se il magazziniere imposta un prezzo pari a zero, il sistema non accetta il valore e richiede di inserire un importo superiore a zero.

**Post Condizioni:**

- La modifica del prezzo del prodotto viene registrata nel database.
- Tutti gli utenti vedranno il nuovo prezzo aggiornato quando visualizzeranno quel prodotto.

### 3.3.2.4 Richiesta di un prodotto



**Attori:** Magazziniere

**Pre-condizioni:**

- Il magazziniere ha effettuato l'accesso al sistema.
- Il magazziniere si trova nella pagina di gestione del catalogo prodotti.

**Flusso di eventi:**

1. Il sistema visualizza l'elenco completo dei prodotti presenti a magazzino.
2. Il magazziniere seleziona il prodotto di cui vuole ordinare nuove scorte.
3. Il sistema mostra un modulo per inserire i dettagli del nuovo ordine al fornitore.
4. Il magazziniere specifica la quantità di unità da ordinare.

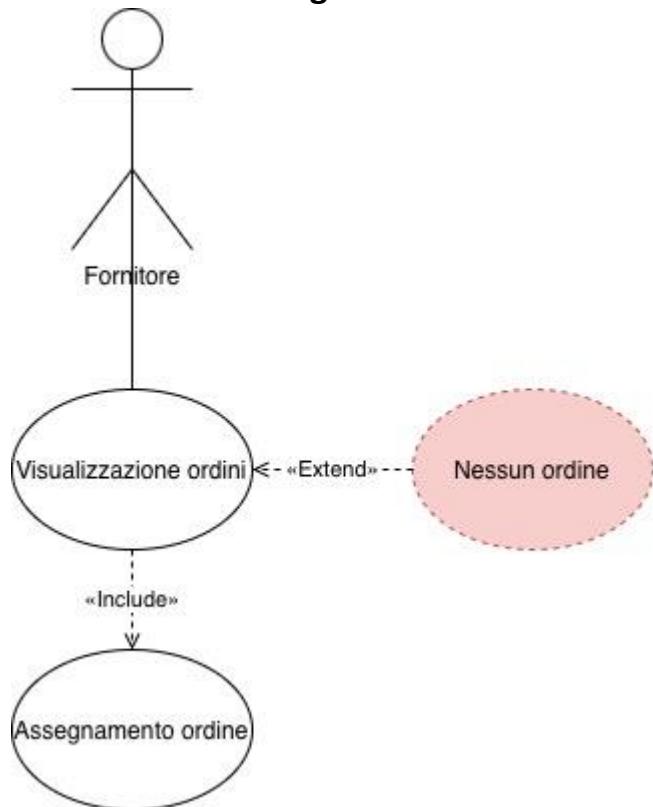
**Flussi alternativi:**

- Se non ci sono prodotti nel catalogo, il sistema mostra un messaggio che informa che l'elenco è vuoto.
- Se il magazziniere inserisce una quantità con un valore negativo, il sistema segnala un errore e richiede di inserire un numero positivo.
- Se il magazziniere inserisce zero come quantità, il sistema non procede e chiede di specificare una quantità maggiore di zero.

**Post Condizioni:**

- La richiesta di rifornimento viene registrata nel database del sistema.
- Il fornitore potrà vedere la nuova richiesta di prodotti nel proprio pannello di controllo.

### 3.3.2.5 Gestione degli ordini



**Attori:** Fornitore

**Pre-condizioni:**

- Il fornitore ha effettuato l'accesso al sistema.
- Il fornitore si trova nella schermata di visualizzazione degli ordini di rifornimento.

**Flusso di eventi:**

1. Il sistema mostra gli ordini di rifornimento ricevuti, dal più recente al meno recente.
2. Il fornitore seleziona l'ordine da gestire.
3. Il fornitore assegna l'ordine a un proprio incaricato per la preparazione.

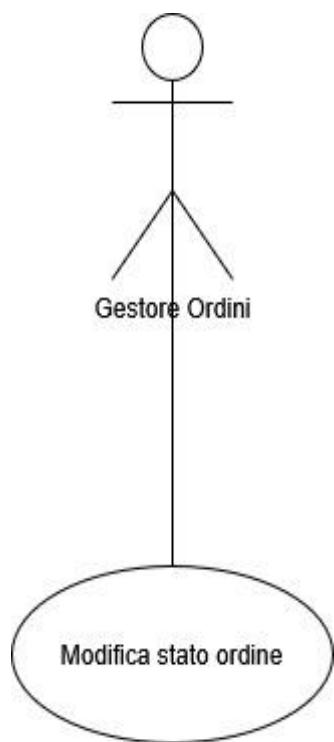
**Flussi alternativi:**

- Se non sono presenti ordini di rifornimento, il sistema mostra un messaggio che indica l'assenza di ordini.

**Post condizioni:**

- L'assegnazione dell'ordine viene registrata nel database.
- L'incaricato sarà informato di dover gestire l'ordine di rifornimento.

### 3.3.2.6 Modifica stato ordine



**Attori:** Gestore degli ordini

**Pre-condizioni:**

- Il gestore degli ordini ha effettuato l'accesso al sistema.
- Il gestore degli ordini visualizza l'elenco degli ordini che gli sono stati assegnati dal magazziniere.

**Flusso di eventi:**

1. Il sistema mostra una lista degli ordini assegnati, dal più vecchio al più recente.
2. Il sistema permette di aggiornare lo stato di ogni ordine in base alla fase di lavorazione.
3. All'arrivo dell'ordine, il suo stato iniziale è "In attesa".
4. Quando il gestore inizia a occuparsi dell'ordine, ne cambia lo stato da "In attesa" a "In preparazione".
5. Dopo aver completato la preparazione, il gestore cambia lo stato da "In preparazione" a "Spedito".
6. Una volta che la merce è stata recapitata al cliente, il gestore imposta lo stato dell'ordine su "Consegnato".

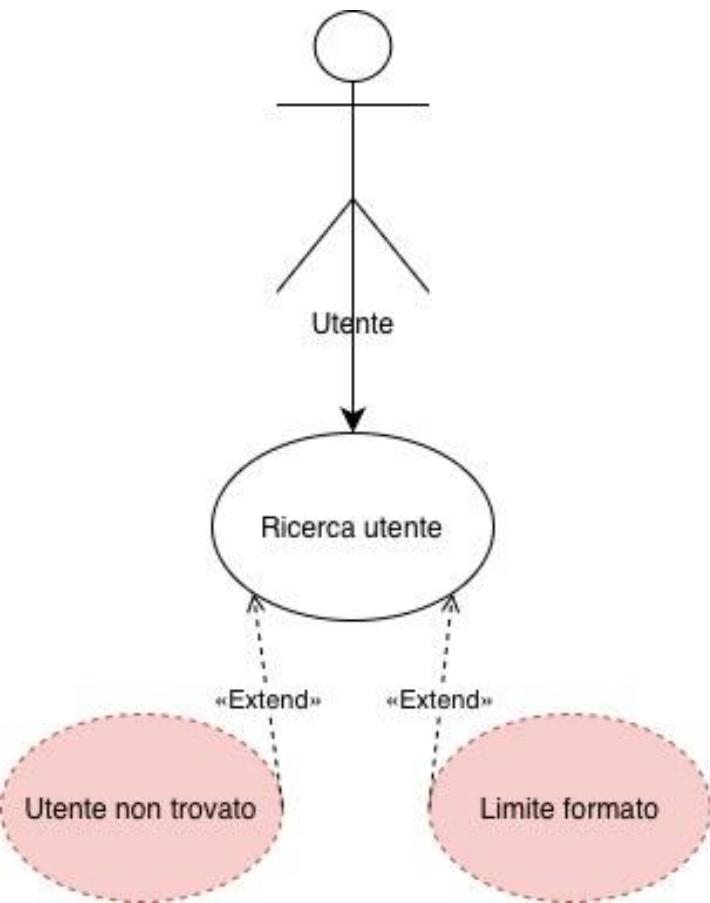
**Flussi alternativi:**

- Se non sono stati assegnati ordini, il sistema visualizza un messaggio che dice "Nessun ordine".

**Post condizioni:**

- Il sistema riceve e registra l'aggiornamento di stato per ogni ordine.

### 3.3.2.7 Ricerca utente



**Attori:** Utente (fornitore, gestore ordini, gestore magazzino)

#### Pre-condizioni:

- L'utente ha effettuato l'accesso al sistema.
- Il sistema sta mostrando all'utente l'elenco degli utenti registrati.

#### Flusso di eventi:

1. Il sistema visualizza una barra di ricerca compatta.
2. L'utente clicca sulla barra e digita il nome dell'utente con cui desidera comunicare.
3. Dopo aver inserito il nome, l'utente preme il pulsante "Cerca".

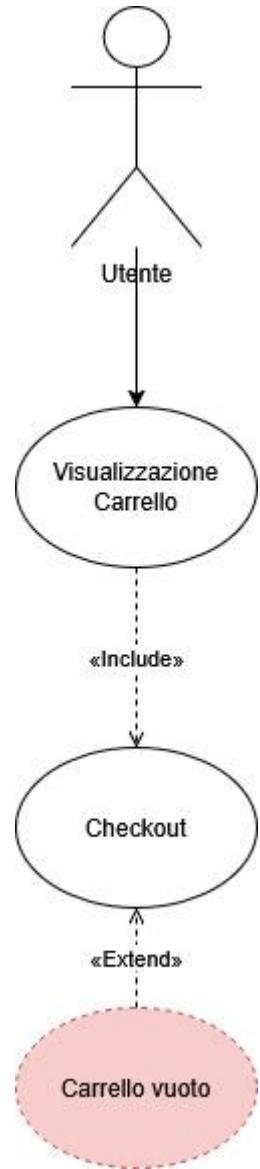
#### Flussi alternativi:

- Se l'utente ricercato non esiste nel sistema, la ricerca non produce risultati.
- Se il testo inserito nella ricerca supera la lunghezza massima consentita, il sistema segnala l'errore.

#### Post condizioni:

- L'utente ricercato viene trovato e visualizzato nel sistema.

### 3.3.2.8 Visualizzazione Carrello, Checkout



**Attori:** Utente

**Pre-condizioni:**

1. È visibile il pulsante per visualizzare il carrello.

**Flusso di eventi:**

1. L'utente clicca sul pulsante per vedere il contenuto del carrello.
2. Il sistema mostra per ogni prodotto nel carrello: l'immagine, il prezzo, la quantità e il nome. Inoltre, mostra un pulsante per "Procedi all'acquisto" (Checkout).
3. L'utente preme il pulsante "Procedi all'acquisto".
4. Il sistema controlla che ci sia almeno un prodotto nel carrello.
5. Il sistema reindirizza l'utente alla pagina per completare la registrazione o il login.

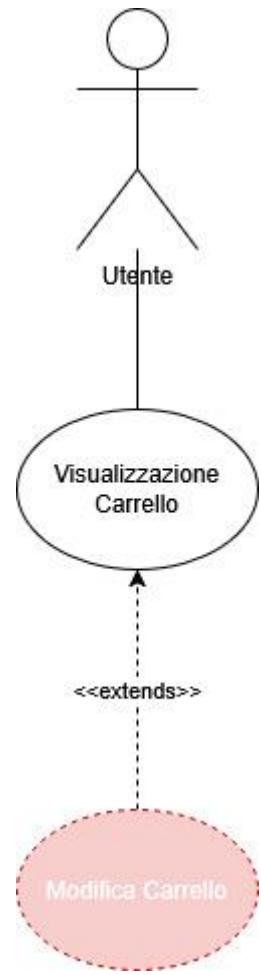
**Flussi alternativi: Carrello Vuoto**

1. Se il carrello è vuoto, il sistema mostra un messaggio di errore che informa l'utente della situazione.
2. Il sistema rimane nella pagina del carrello, in attesa di una nuova azione da parte dell'utente.

**Post-condizioni:**

1. L'utente viene reindirizzato alla pagina di registrazione o accesso.

### 3.3.2.9 Modifica Carrello



**Attori:** Utente

**Pre-condizioni:**

1. È visibile il pulsante per accedere al carrello.

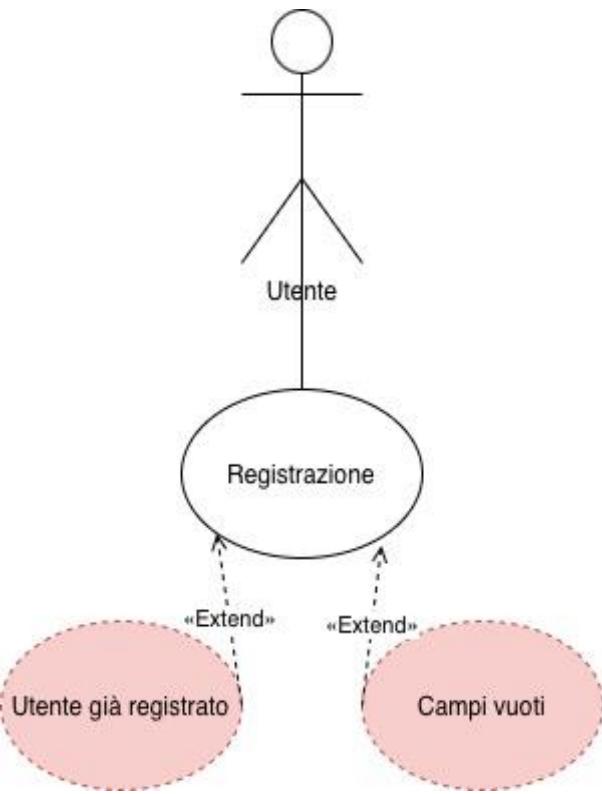
**Flusso degli eventi:**

1. L'utente clicca sul comando per visualizzare il carrello.
2. Il sistema mostra l'elenco dei prodotti nel carrello con: immagine, prezzo, quantità e nome per ciascuno. Inoltre, viene fornita un'opzione per "Modifica Carrello".
3. L'utente seleziona l'opzione "Modifica Carrello".
4. Il sistema visualizza nuovamente i prodotti, ma ora affianco a ognuno compaiono due controlli: "Rimuovi" per eliminare il prodotto e un campo di testo per modificare la quantità desiderata
5. L'utente apporta le modifiche e conferma inviandole.
6. Il sistema aggiorna il carrello con le nuove informazioni inserite.

**Post-condizioni:**

1. L'utente visualizza la pagina del carrello aggiornata, che riflette tutte le modifiche apportate.

### 3.3.2.10 Registrazione (vedi mock-up Login)



**Attori:** Utente

**Pre-condizioni:**

1. È visibile il pulsante per registrarsi al sito.

**Flusso degli eventi:**

1. Il cliente clicca sul comando per effettuare la registrazione.
2. Il sistema mostra un modulo da compilare con i seguenti campi obbligatori:
  - Nome: Testo (solo lettere)
  - Cognome: Testo (solo lettere)
  - Data di nascita: Selezionabile da tre menu a discesa (giorno, mese, anno)
  - Indirizzo email: Testo
  - Password: Testo
  - Partita IVA: Serie di numeri
  - IBAN: Serie di numeri
3. Il cliente compila tutti i campi richiesti e invia il modulo.
4. Il sistema verifica che:
  - Tutti i campi obbligatori siano compilati
  - La Partita IVA sia valida
  - L'IBAN sia valido
  - L'email non sia già registrata da un altro account
  - Il cliente non sia già presente nel sistema
5. Il sistema salva i dati del nuovo cliente nel database.
6. Il sistema mostra una schermata di conferma che comunica l'avvenuta registrazione.

**Flusso alternativo: Dati mancanti**

1. Se alcuni campi obbligatori non sono compilati, il sistema mostra un messaggio di errore che avvisa l'utente.
2. Il sistema rimane in attesa che il modulo venga inviato nuovamente, correttamente.

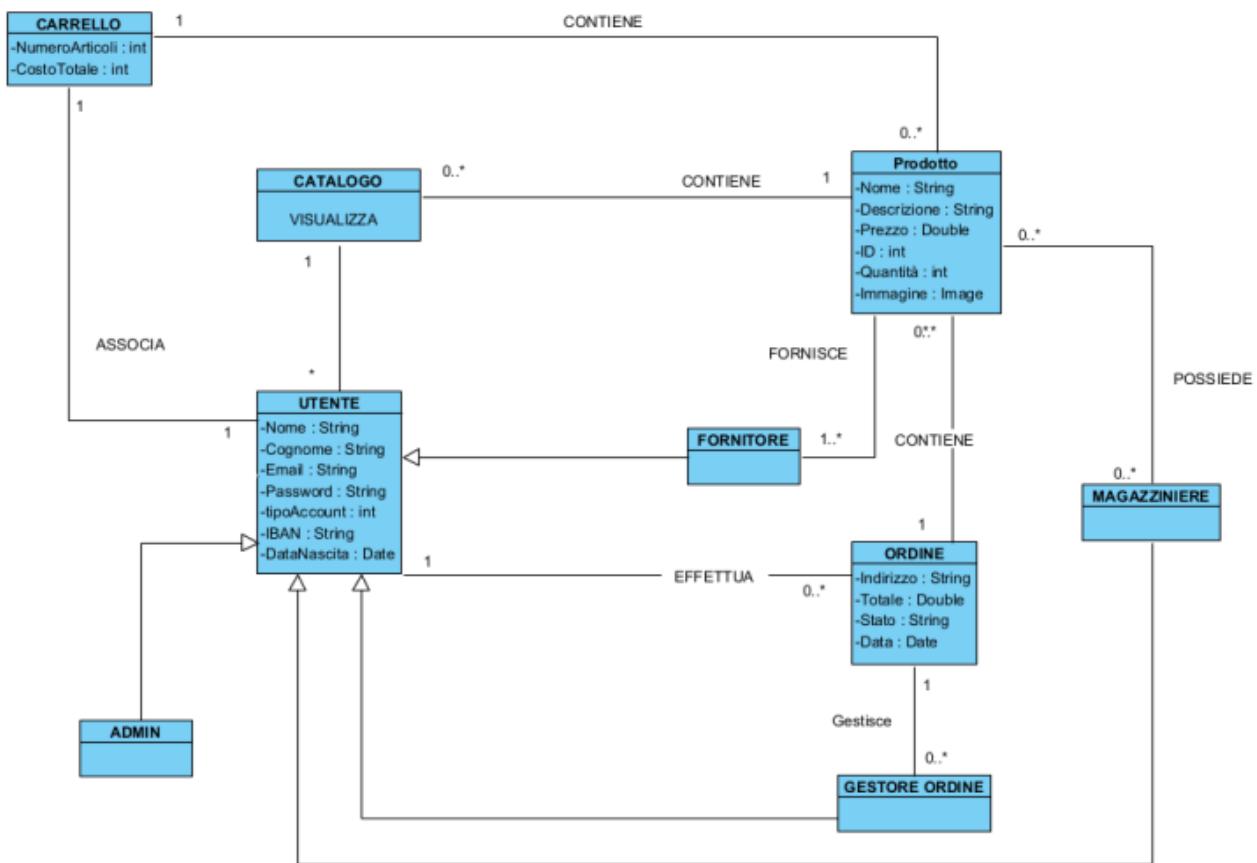
**Flusso alternativo: Email già esistente**

1. Se l'email inserita è già in uso, il sistema mostra un messaggio di errore che lo segnala.
2. Il sistema rimane in attesa di una nuova invio del modulo.

**Post-condizioni:**

1. L'account è stato creato ed è attivo. Il cliente può ora accedere al sito effettuando il login.

### 3.3.3. Object Model



### 3.3.4. Dynamic Model

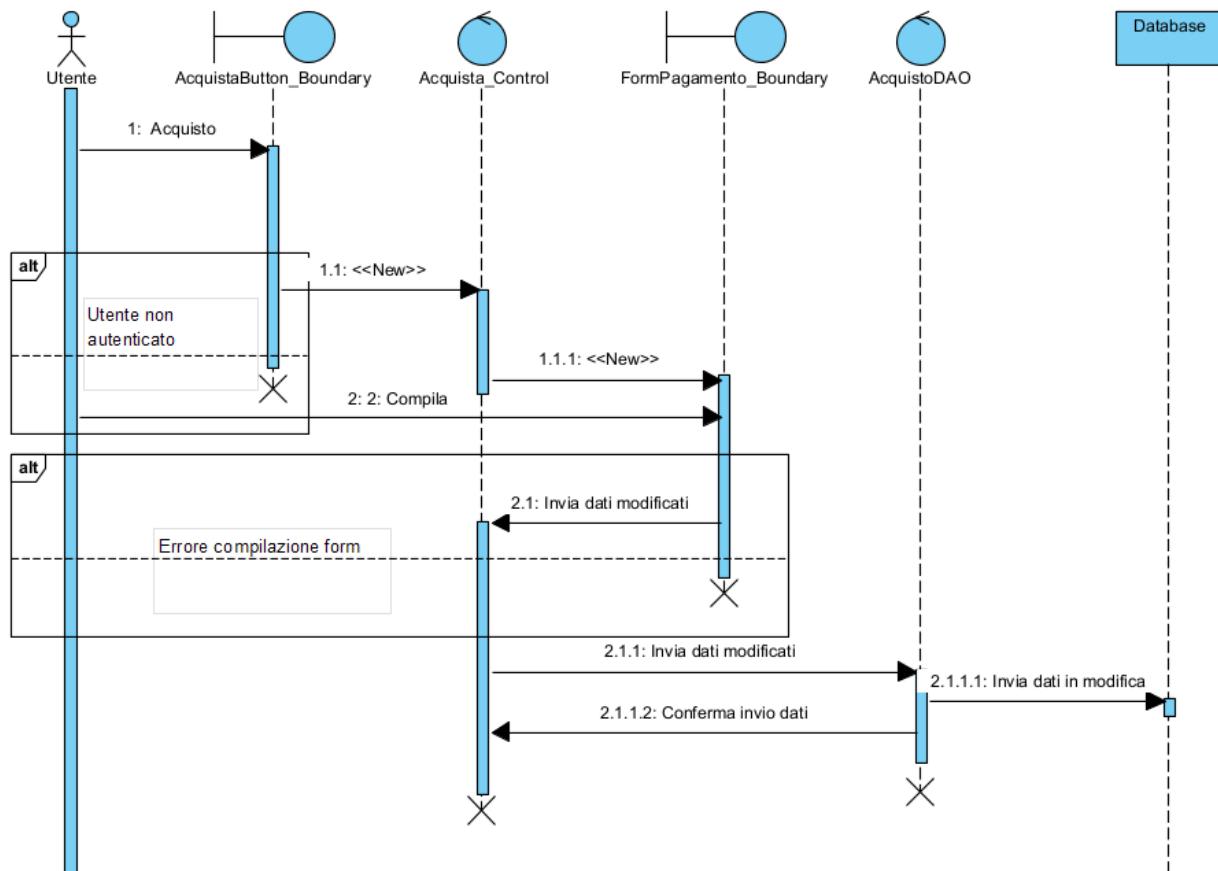
#### DIZIONARIO DEI DATI

Tipo	Nome oggetto	Descrizione
Entity	`Utente`	Modello che rappresenta un utente del sistema, include dati personali e informazioni di autenticazione.
Entity	`Prodotto`	Gestisce le informazioni sui prodotti, come nome, descrizione, prezzo e quantità disponibile.
Entity	`Ordine`	Contiene i dati relativi agli ordini, incluso lo stato, i prodotti acquistati.
Entity	`Carrello`	Raccoglie i prodotti selezionati dall'utente per l'acquisto.
Entity	'Richiesta del prodotto al magazziniere'	Registra e gestisce le richieste di approvvigionamento effettuate dal gestore verso il magazzino.
Control	`AcquistaControl`	Gestisce le operazioni per il completamento dell'acquisto.
Control	'ModificaPrezzoProdotto_Control'	Gestisce le operazioni di modifica del prezzo di un prodotto.
Control	`AcquistoDAO`	Gestisce le operazioni di salvataggio e recupero dei dati relativi agli acquisti.
Control	`RichiestaProdotto_Control`	Gestisce le operazioni di richiesta di nuovi prodotti al magazzino.
Control	`OrdineControl`	Gestisce le operazioni di aggiornamento e monitoraggio dello stato degli ordini.

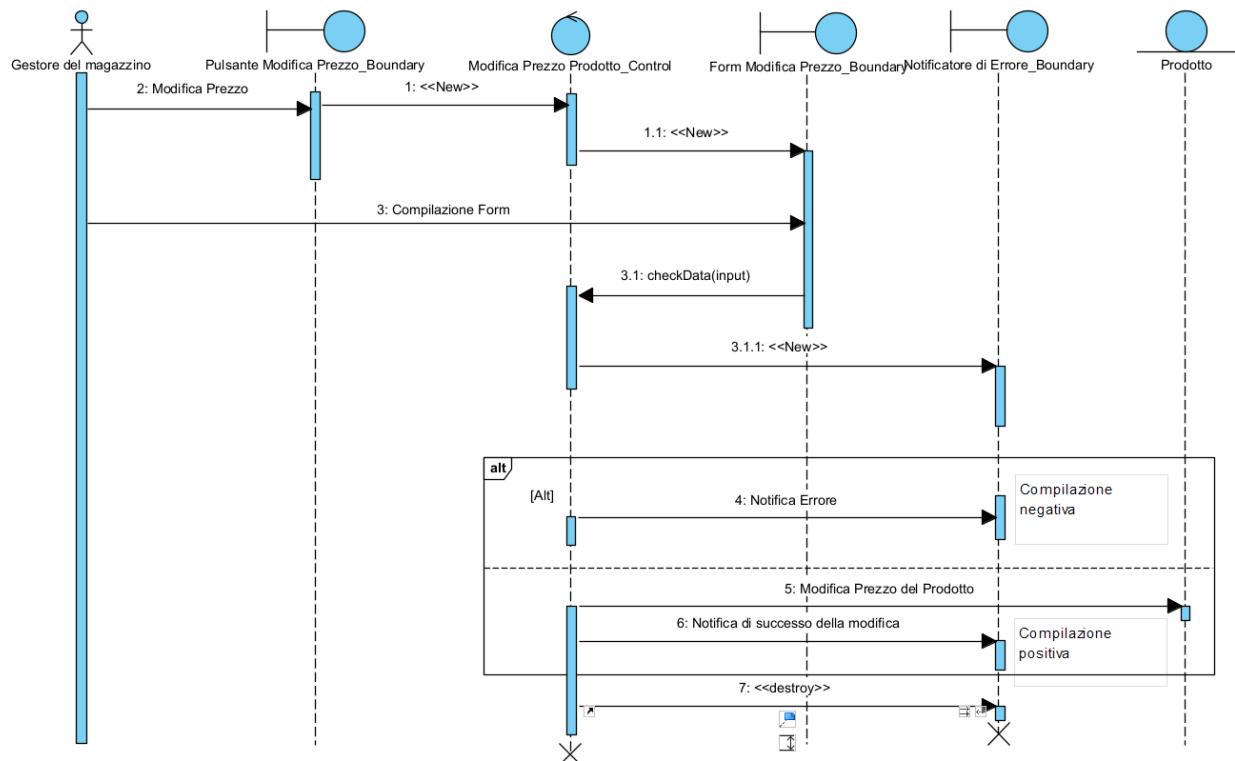
Control	'UserRegistrationController'	Gestisce le operazioni di registrazione di un nuovo utente.
Control	'RegistrazioniDAO'	Gestisce le operazioni di memorizzazione e recupero dei dati di registrazione.
Boundary	'AcquistaButton'	Permette l'interazione nelle operazioni di acquisto.
Boundary	'FormPagamento'	Permette l'inserimento delle informazioni di pagamento durante l'acquisto.
Boundary	'PulsanteModificaPrezzo'	Permette l'interazione per l'avvio della modifica del prezzo di un prodotto.
Boundary	'FormModificaPrezzo'	Permette l'inserimento dei dati per la modifica del prezzo del prodotto.
Boundary	'Notificatore Di Errore'	Permette la visualizzazione di messaggi di errore o di notifica all'utente.
Boundary	'AggiuntaProdotti'	Permette l'interazione nelle operazioni di aggiunta di nuovi prodotti.
Boundary	'FormRichiestaProdotto'	Permette l'inserimento delle informazioni relative alla richiesta di prodotti al magazzino.
Boundary	'ModificaPulsanteStato'	Permette l'interazione per la modifica dello stato di un ordine.
Boundary	'Menustato'	Permette la selezione e la gestione dello stato di un ordine.
Boundary	'BottoneLogin'	Permette l'interazione nelle operazioni di autenticazione.
Boundary	'SignupForm'	Permette l'inserimento dei dati necessari alla registrazione di un nuovo utente.

## Sequence diagrams

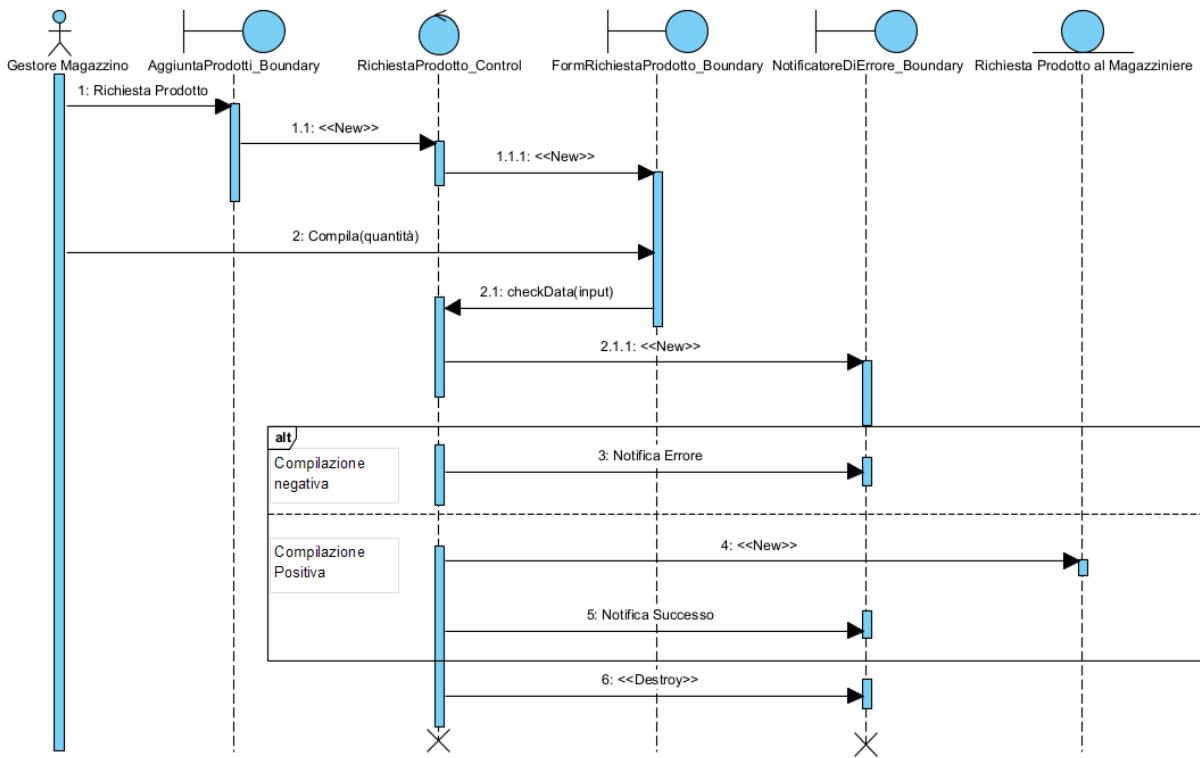
### SD 2: Acquisto di un prodotto



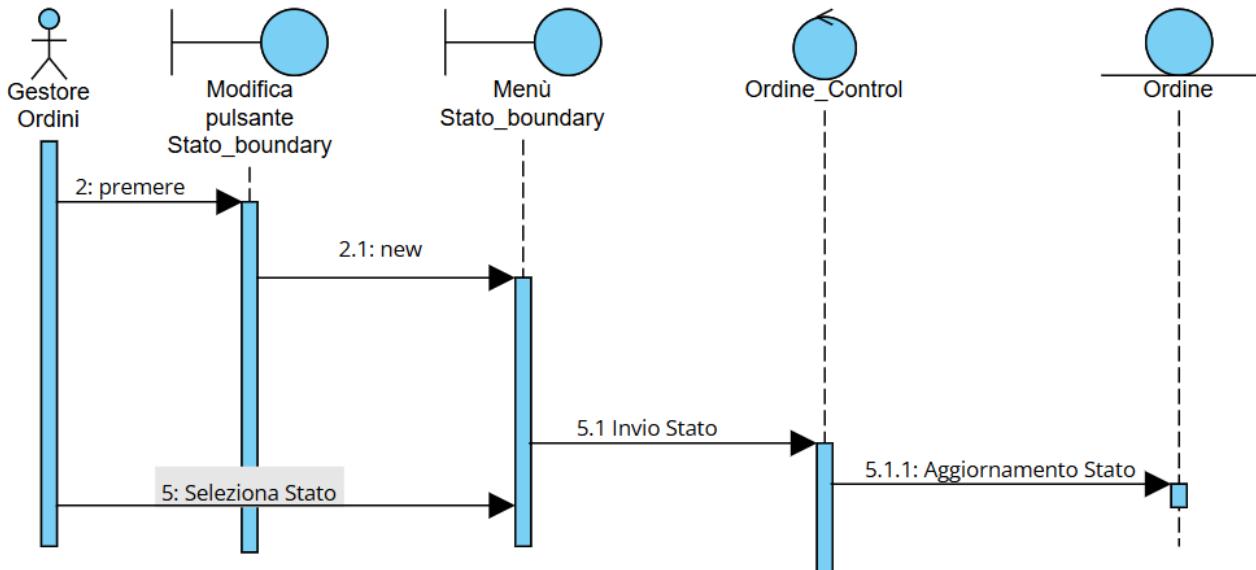
### SD 3: Modifica prezzo di un prodotto



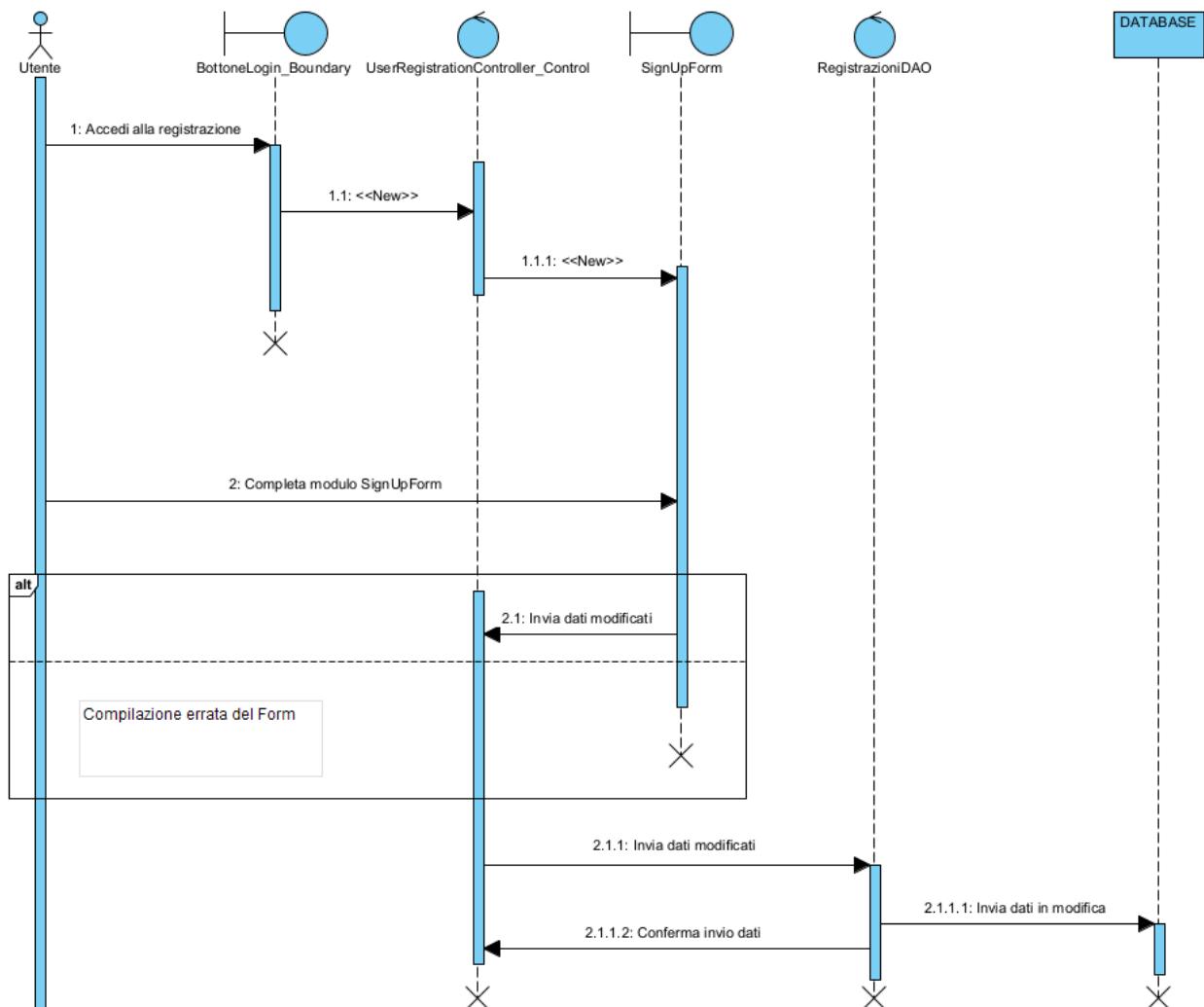
## SD 4: Richiesta di un prodotto



## SD 5: Modifica stato ordine

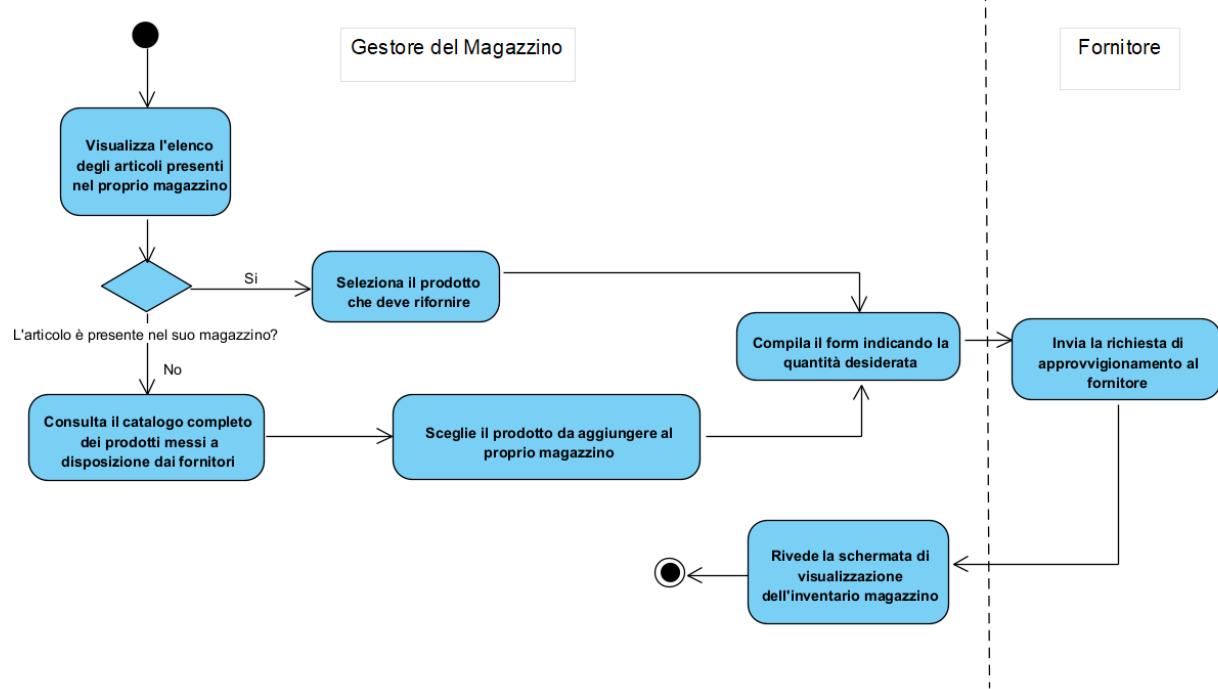


## SD 7: Registrazione utente

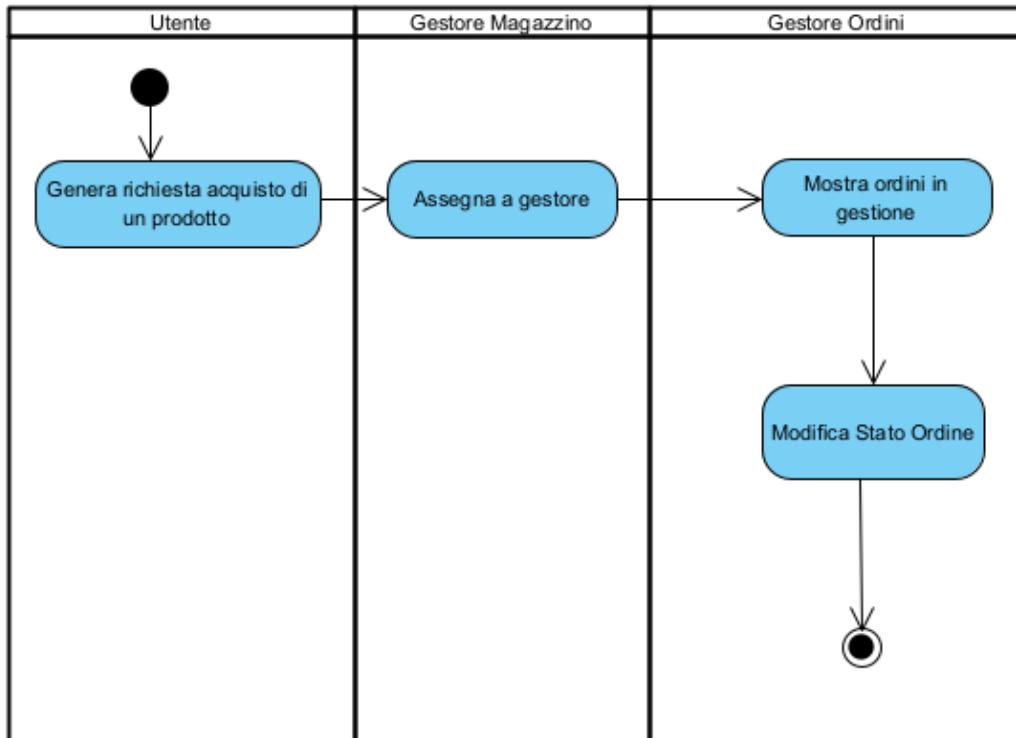


## Activity diagrams

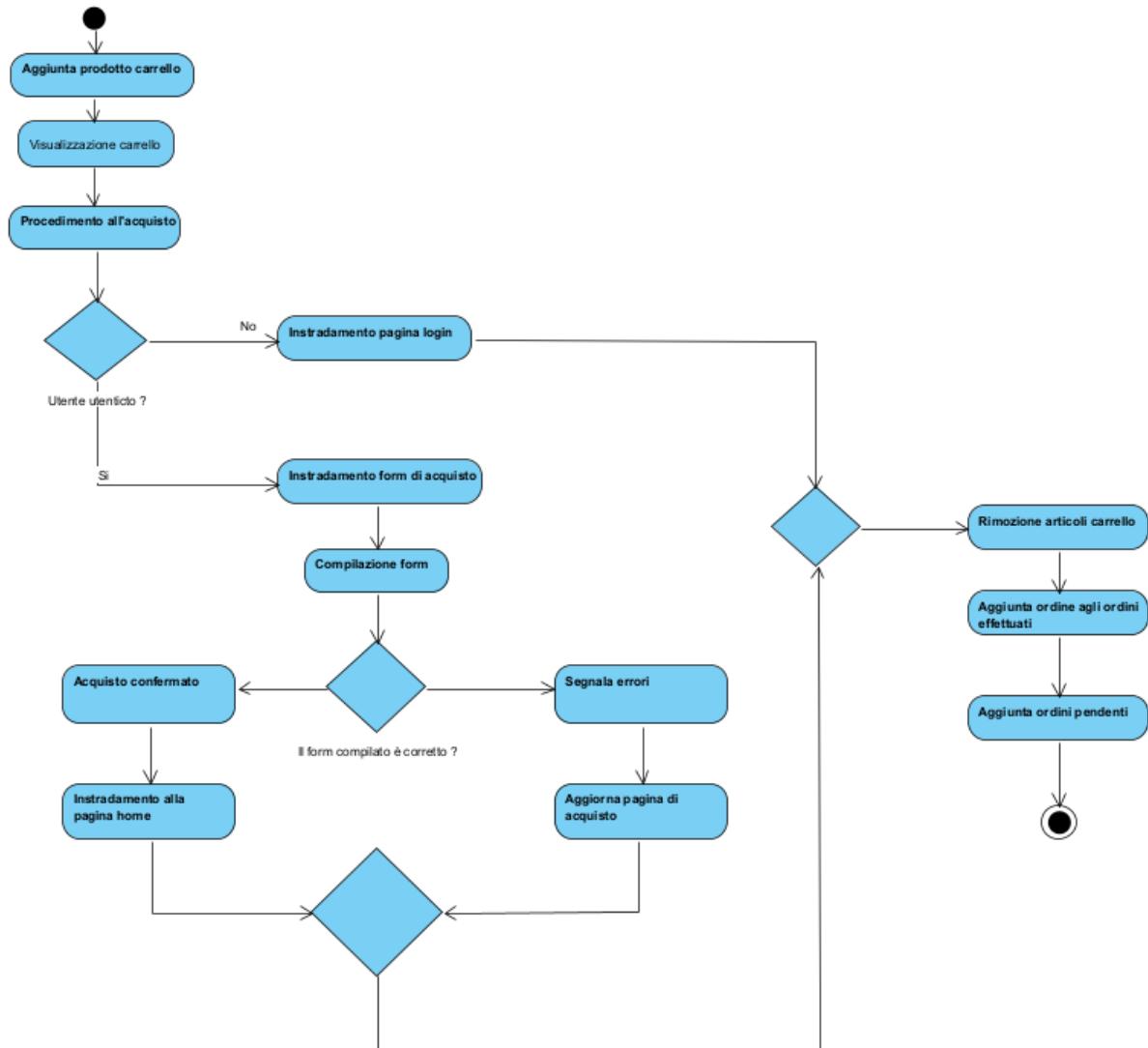
### AD 1: Richiesta prodotto al fornitore



### AD 2: Gestione dell'ordine

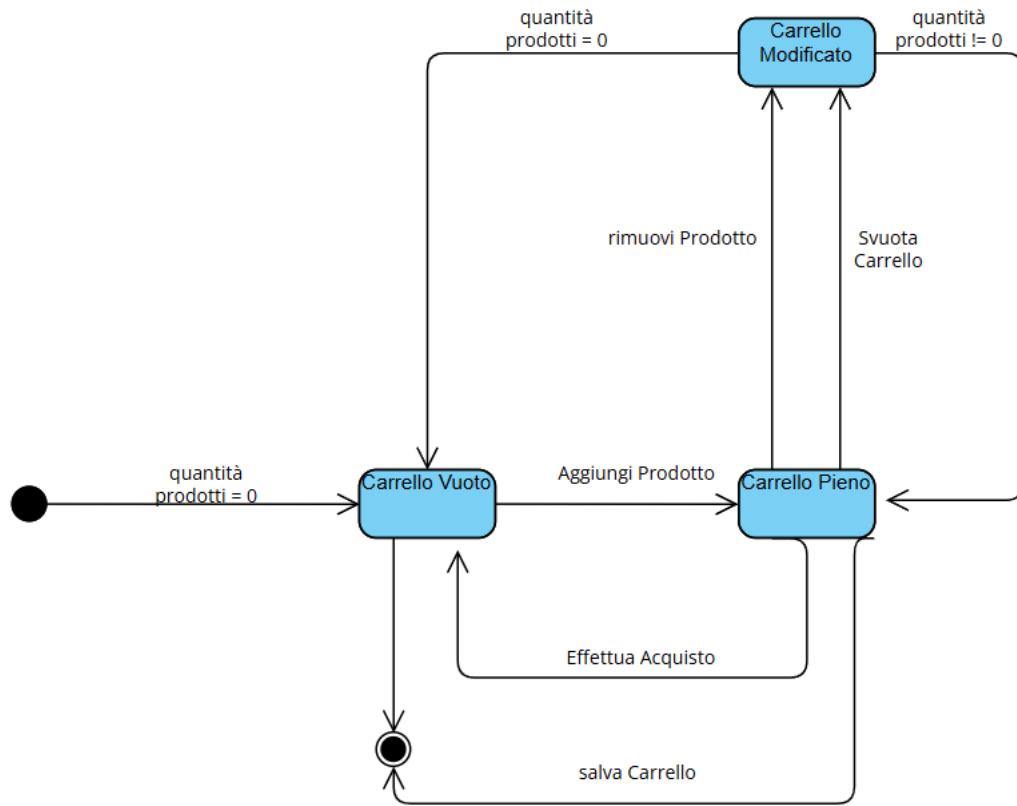


### AD 3: Acquisto di un prodotto

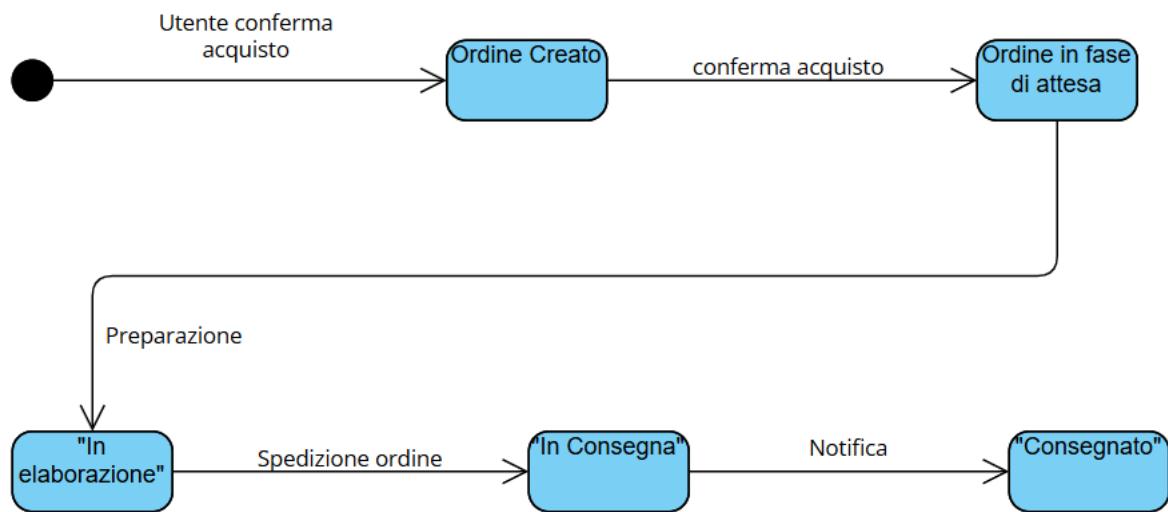


## State machine diagrams

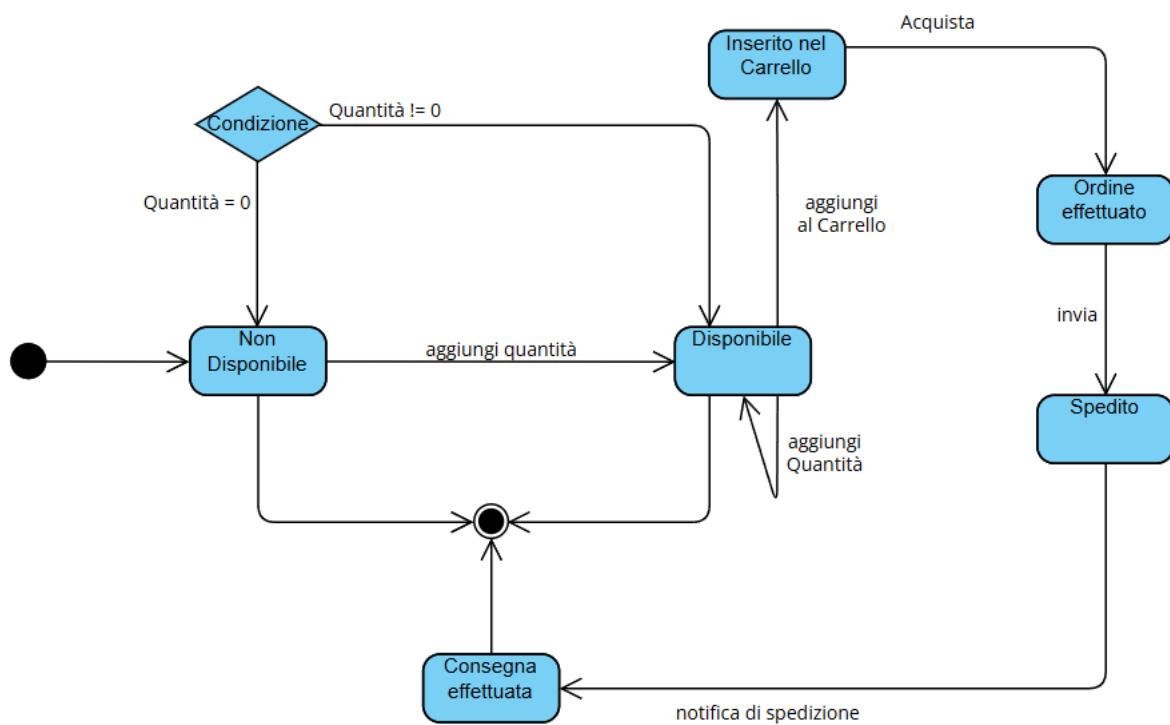
### SM 1: Carrello



### SM 2: Ordine

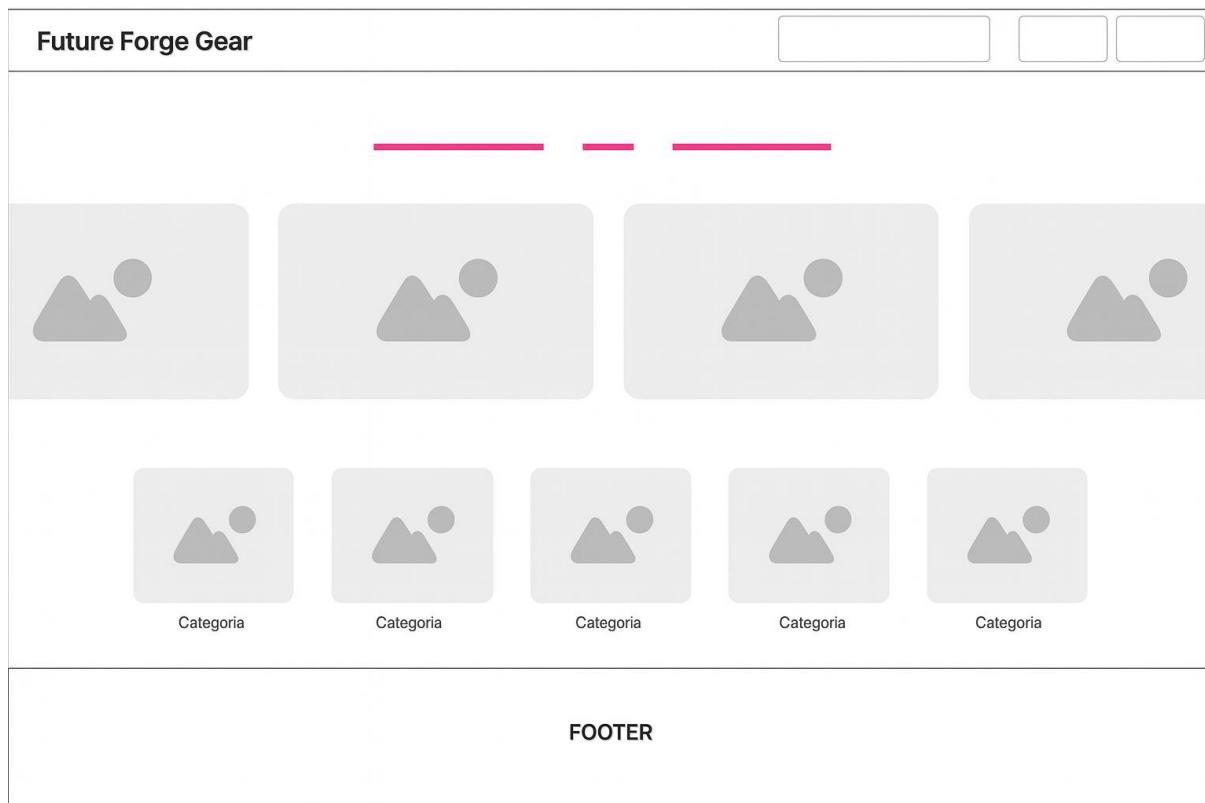


### SM 3: Prodotto

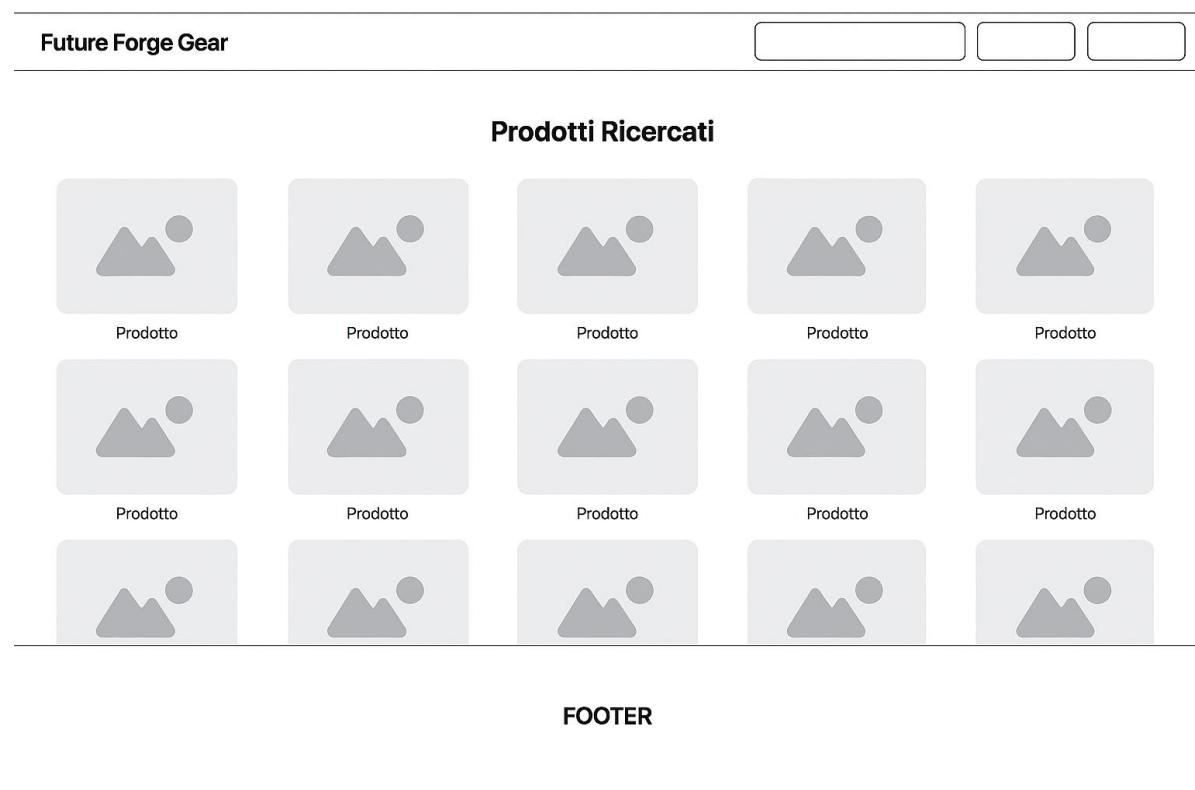


### 3.3.5. User interface-navigational paths and screen mock-ups

#### Homepage



#### Visualizzazione Prodotti



## Login (Vedi caso d'uso Registrazione)

Future Forge Gear

The wireframe shows a top navigation bar with the text "Future Forge Gear" and three empty rectangular boxes. Below this is a large central form area enclosed in a rounded rectangle. Inside the form, there are two input fields: one labeled "Placeholder" and "Email" to its right, and another labeled "Placeholder" and "Password" to its right. At the bottom of the form is a single button labeled "Login". Below the main form is a footer section labeled "FOOTER".

Placeholder Email

Placeholder Password

Login

FOOTER

## 4 GLOSSARIO

**Requisito Funzionale:** Funzionalità specifiche che il sistema deve fornire.

**Requisito Non Funzionale:** Caratteristiche qualitative del sistema (performance, sicurezza, usabilità).

**Use Case Diagram:** Diagramma UML che descrive le funzionalità principali del sistema dal punto di vista degli attori.

**Sequence Diagram:** Diagramma UML che modella le interazioni temporali tra oggetti in uno scenario specifico.

**Statechart Diagram:** Diagramma UML che rappresenta gli stati di un oggetto e le transizioni in risposta a eventi.

**Mockup:** Prototipo statico dell'interfaccia utente, realizzato a scopo dimostrativo.

**Form:** Componente d'interfaccia per l'inserimento e l'invio di dati da parte dell'utente.