



UNIVERSITÀ DEGLI STUDI DI SALERNO
CORSO DI LAUREA TRIENNALE IN INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE
ANNO ACCADEMICO 2021/2022

FOODART



SYSTEM DESIGN
DOCUMENT:
versione 1.1

TOP MANAGER

PROFESSORE
Prof. De Lucia Andrea
TUTOR
Iannone Emanuele

PARTECIPANTI

NOME E COGNOME	MATRICOLA
Davide Amitrano	0512106034
Donato Miranda	0512106148
Alfonso Zappia	0512106076

STORICO DELLE VERSIONI

DATA	VERSIONE	DESCRIZIONE	AUTORI
13/10/2021	1.0	Creazione del documento con completamento degli obiettivi posti nella panoramica	Tutti
29/10/2021	1.1	Revisione finale e ultime correzioni	Amitrano-Zappia

INDICE

1. Introduzione	4
1.1 Scopo del sistema	
1.2 Obiettivi del sistema	
1.3 Definizioni, acronimi e abbreviazioni	
1.4 Riferimenti	
1.5 Panoramica	
2. Architettura Software Attuale	6
3. Architettura Software Proposta	9
3.1 Panoramica	
3.2 Decomposizione in sottosistemi	
3.3 Mappatura hardware/software	
3.4 Gestione dei dati persistenti	
3.5 Controllo degli accessi e sicurezza	
3.6 Controllo globale del software	
3.7 Condizioni limite	
4. Servizi dei sottosistemi	12

1. INTRODUZIONE

1.1 SCOPO DEL SISTEMA

La nostra proposta si basa sulla realizzazione di una web application, chiamata *FoodArt*. Il nome gioca sulle parole inglesi *food*, cibo, e *artisan*, che significa artigianale. FoodArt si propone di superare i limiti che sono stati descritti precedentemente, mettendo a disposizione un sito web in cui tutti i piccoli rivenditori possono mettere in vendita i propri prodotti e guadagnarci senza nessuna commissione. Inoltre, il sistema permetterà a qualsiasi utente di poter fare acquisti liberamente rispettando le specifiche dei fornitori.

1.2 OBIETTIVI DEL SISTEMA

L'applicazione permetterà:

- ad un rivenditore di registrarsi e poter inserire liberamente i suoi prodotti, impostando alcune specifiche, come il prezzo e la quantità minima di acquisto.
- ad un cliente di poter registrarsi e navigare nel sito, aggiungendo al carrello i prodotti che più gli interessano, prima di poterli acquistare.
- ad un rivenditore di poter anch'egli acquistare senza impedimenti i prodotti presenti all'interno del sito.
- ad un amministratore di poter accedere alla propria dashboard, dove potrà controllare il sistema, visionando alcuni dati, sia riguardo i prodotti che gli utenti.
- ad un amministratore di approvare, sospendere o respingere le varie richieste che gli si presenteranno all'interno del sito, come l'approvazione di un rivenditore, l'approvazione di un prodotto o di un commento.

1.3 DEFINIZIONI, ACRONIMI E ABBREVIAZIONI

DENOMINAZIONE	DESCRIZIONE
FoodArt	Nome del sistema.
Utente generico	Dicitura che viene utilizzata per definire diverse tipologie di utenti.
Utente ospite	Un qualsiasi utente che utilizza il sistema, senza essere però loggato.
Cliente	Un utente loggato che ha la possibilità di acquistare prodotti all'interno del sito
Prodotto	L'insieme dei dati relativi ad un prodotto alimentare

	artigianale che viene inserito all'interno del sito.
Rivenditore	Un utente loggato che ha la possibilità di vendere i propri prodotti all'interno del sito.
Amministratore	Un utente loggato che ha la responsabilità di gestire le approvazioni e di controllare l'andamento del sito.
DBMS	<i>Database Management System</i> , un sistema software per la gestione di database.
MySQL	Un DBMS per la gestione di database relazionali.
HTML	<i>HyperText Markup Language</i> , linguaggio di markup per realizzare pagine web.
JSP	<i>JavaServer Pages</i> , insieme di tag all'interno di una pagina HTML in cui può essere integrato del codice Java.
JS	<i>JavaScript</i> , linguaggio di programmazione utilizzato per la creazione di effetti dinamici e interattivi all'interno delle pagine web.
CSS	<i>Cascading Style Sheets</i> , insieme di direttive che permettono di definire la formattazione di pagine web.
HTTPS	<i>HyperText Transfer Protocol over Secure Socket Layer</i> , protocollo per la comunicazione sicura attraverso la rete.
MVC	<i>Model-view-controller</i> , pattern architetturale in grado di separare la logica di presentazione dei dati dalla logica di business.
Apache TomCat	Un server web che fornisce una piattaforma per

	l'esecuzione di applicazioni web in Java.
RAD	<i>Requirements Analysis Document</i> , documento redatto per la descrizione dei requisiti di un progetto.
PS	<i>Problem Statement</i> , documento redatto per delineare punti negativi della situazione attuale e fornire una prima panoramica del sistema proposto.
Servlet	Oggetti scritti in linguaggio Java che operano all'interno di un server web.
API	<i>Application Programming Interface</i> , cioè un insieme di librerie software di un determinato linguaggio.
JDBC	<i>Java DataBase Connectivity</i> , un connettore per database che consente l'accesso ai dati persistenti da qualsiasi programma scritto in Java.
Failure	Fallimento, che può essere legato a problemi hardware o software.

1.4 RIFERIMENTI

- PS
- RAD
- Object-Oriented Software Engineering Using UML, Patterns, and Java by Bernd Bruegge and Allen H. Dutoit.
- Corso di Ingegneria del Software → <http://elearning.informatica.unisa.it/>

1.5 PANORAMICA

In questo documento verranno mostrati i risultati del processo di progettazione del sistema. Quest'ultimo verrà descritto a livello di architettura:

- decomposizione in sottosistemi, con le loro responsabilità.
- mappatura hardware/software, descrive come i sottosistemi sono assegnati all'hardware e ai componenti di serie.
- gestione dei dati persistenti, descrivendo come sono stati memorizzati all'interno del sistema, attraverso schemi.
- controllo degli accessi e sicurezza, descrivendo l'interfaccia utente in termini di accesso, elencando i problemi di sicurezza, l'uso della crittografia e la gestione delle chiavi.
- controllo globale del software, descrive come vengono avviate le richieste e come i sottosistemi vengono sincronizzati.
- condizioni limite, descrivono l'avvio, l'arresto e il comportamento in caso di errore

Infine, è presente una sezione riguardo la descrizione dettagliata dei servizi forniti da ciascun sottosistema.

2. ARCHITETTURA SOFTWARE ATTUALE

A livello architetturale, non sono state adoperate nuove tecnologie per la realizzazione del nostro sistema.

3. ARCHITETTURA SOFTWARE PROPOSTA

3.1 PANORAMICA

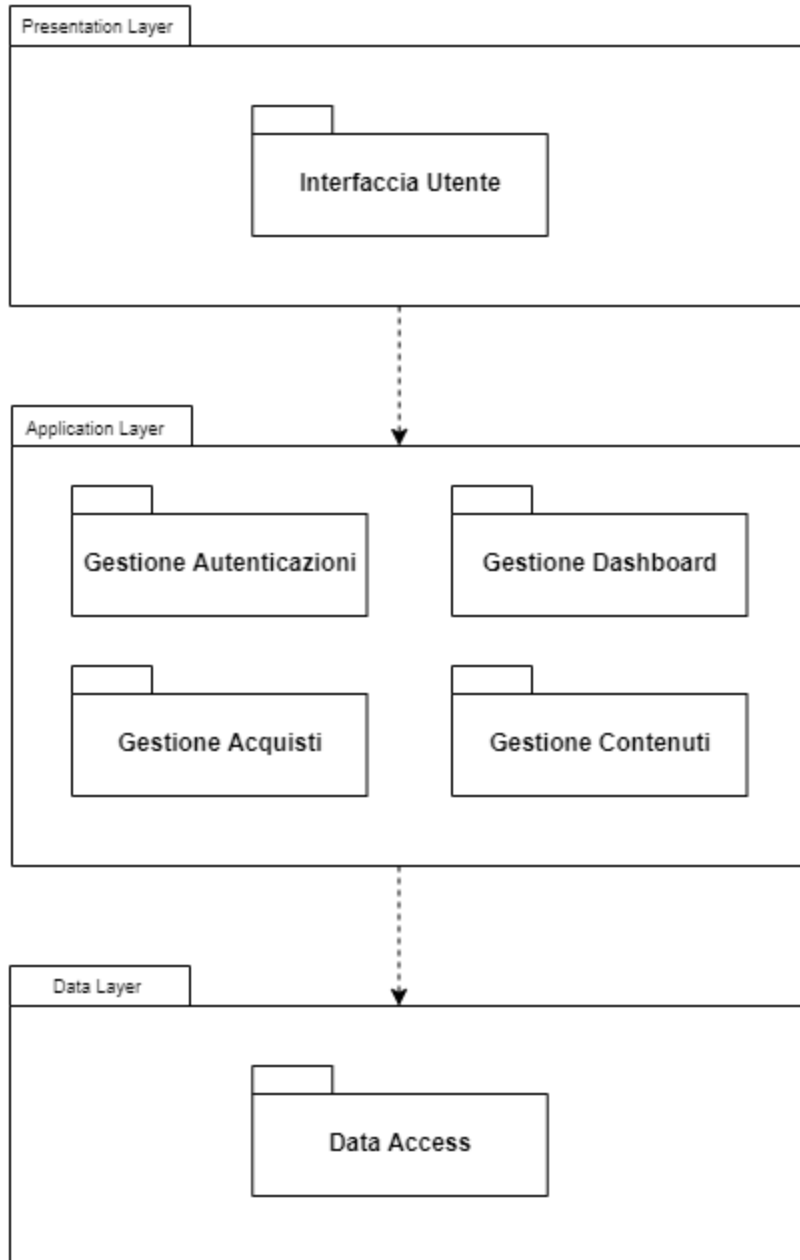
Il sistema è decomposto in sottosistemi, secondo lo stile architetturale dell'MVC:

- **Model:** rappresenta tutti i sottosistemi responsabili della logica applicativa e dell'accesso ai dati.
- **View:** rappresenta tutti i sottosistemi responsabili delle informazioni visualizzate dagli utenti, comprese le loro interazioni con i livelli sottostanti.
- **Controller:** rappresenta tutti i sottosistemi responsabili della ricezione dei comandi utente (attraverso il View) ed esegue le operazioni del Model, che portano generalmente ad un cambio di stato delle View.

3.2 DECOMPOSIZIONE IN SOTTOSISTEMI

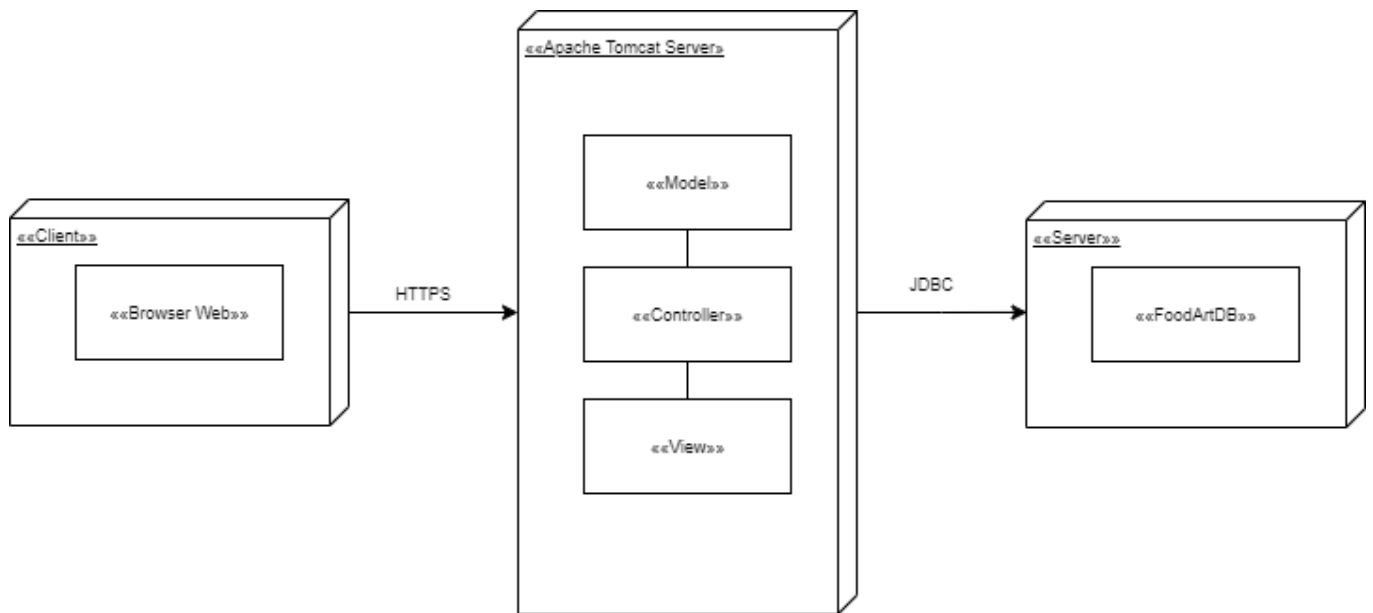
Il sistema è una web-application e viene divisa secondo un'architettura three-tier, del tipo client-server, in cui interfaccia utente, processi aziendali e archiviazione dei dati sono sviluppati e mantenuti come moduli indipendenti. I livelli sono tre e sono:

- **Presentation Layer:** si tratta del livello più alto dell'applicazione. Il client si interfaccia con questo livello per interagire col sistema. Il PL accede al livello sottostante per avere i servizi che il client richiede.
- **Application Layer:** si tratta del livello intermedio dell'applicazione. Si occupa di gestire la logica applicativa, di fornire i servizi richiesti al PL, oltre che comunicare con il livello più basso.
- **Data Layer:** si tratta del livello più basso dell'applicazione. Si occupa della gestione dei dati persistenti e comunica con l'AL fornendo i servizi richiesti.



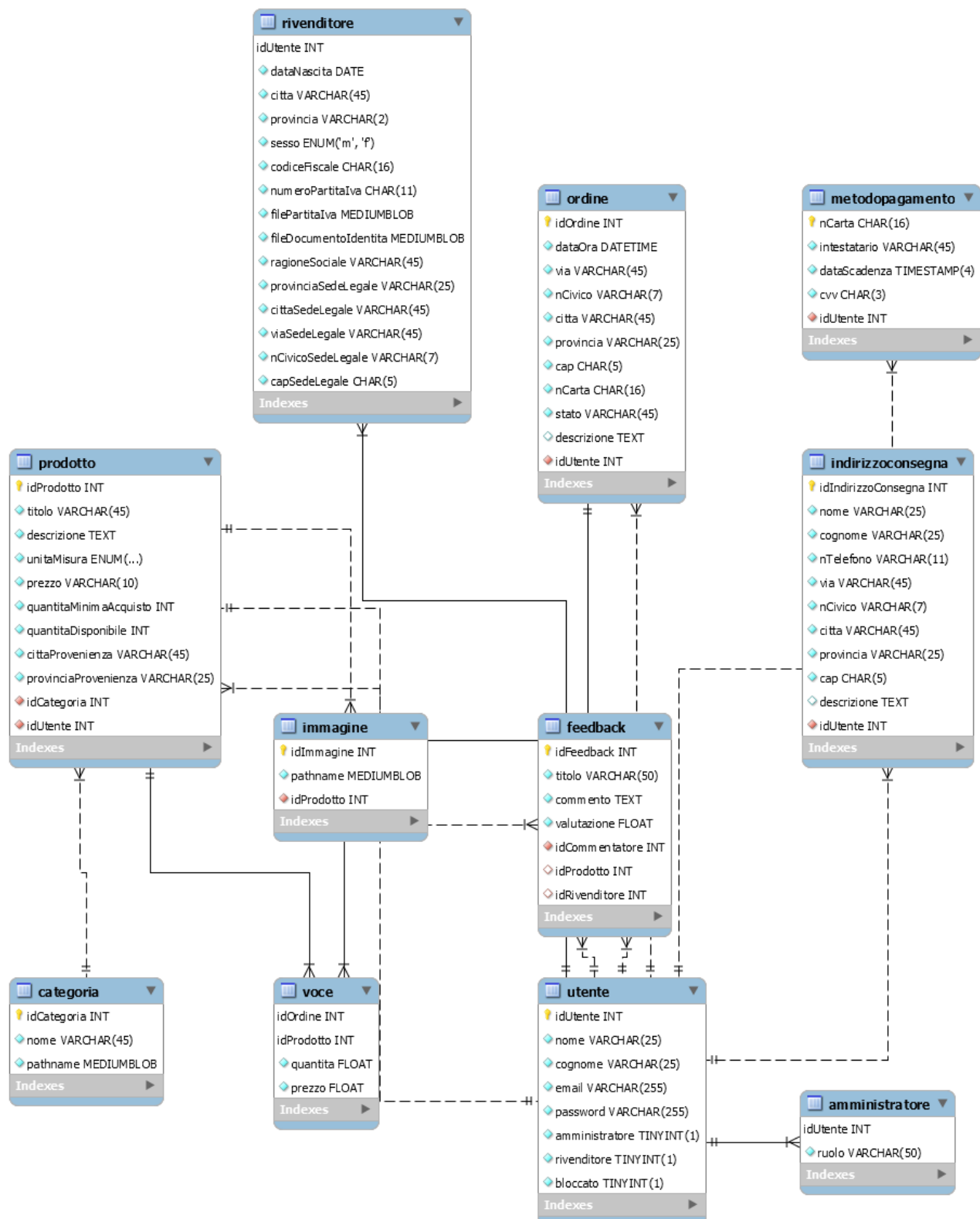
3.3 MAPPATURA HARDWARE/SOFTWARE

Il sistema che andremo a realizzare è una web-application; quindi, risulterà accessibile da qualsiasi client abbia un browser web. Il nostro server-web di riferimento è Apache Tomcat. Al suo interno, le pagine in HTML o JSP, CSS e JS si occuperanno di realizzare l'interfaccia utente, mentre la logica applicativa verrà gestita dalle Servlet. Il client comunica con il web-server attraverso il protocollo di rete HTTPS. Il server può comunicare con il database, attraverso il connettore JDBC. Il web-server comunica con un DBMS, nel nostro caso quello per database relazionali, MySQL, usando le API del JDBC.



3.4 GESTIONE DEI DATI PERSISTENTI

La gestione dei dati persistenti avviene tramite il DBMS relazionale MySQL. Individuando le entità all'interno del diagramma delle classi, le abbiamo rese persistenti all'interno del database.



3.5 CONTROLLO DEGLI ACCESSI E SICUREZZA

La comunicazione tra client e server avviene attraverso il protocollo di rete HTTPS, per garantire che non ci siano intercettazioni.




Ogni informazione sensibile viene trattata con estrema sicurezza, con l'accesso al sistema che può essere effettuato tramite e-mail e password. La password è stata cifrata attraverso la tecnica di *Base64*. Si tratta di un sistema di codifica che consente la traduzione di dati binari in stringhe di testo ASCII, rappresentandoli sulla base di 64 caratteri ASCII diversi.

Per controllare gli accessi, abbiamo deciso di suddividere le azioni consentite in base alle tipologie utenti, qui sottoelencate.

Legenda:

- Utente Ospite ●
- Cliente ●
- Rivenditore ●
- Amministratore ●

InvioRichiestaRegistrazioneRivenditore()	●
InvioRichiestaRegistrazioneCliente()	●
VerificaAccount()	●
Login()	● ● ●
Logout()	● ● ●
VerificaAccountRivenditore()	●
SospendiAccount()	●
EliminaAccount()	●
VerificaRecensioni()	●
VerificaProdotto()	●
ModificaDatiPersonali()	● ●
InserisciIndirizzo()	● ●
InserisciCarta()	● ●
InserisciProdotto()	● ●
ModificaProdotto()	● ●
RicercaProdottoPerCategoria()	● ● ● ●
RicercaProdottoPerParolaChiave()	● ● ● ●
VisualizzaPaginaSingoloProdotto()	● ● ● ●
RicercaRivenditore()	● ● ● ●
VisualizzaPaginaInformativaRivenditore()	● ● ● ●

AggiuntaAlCarrello()	
Acquisto()	
InviaRichiestaInserimentoRecensione()	

3.6 CONTROLLO GLOBALE DEL SOFTWARE

Il nostro controllo globale è di tipo event-driven-control. Il ciclo principale attende un evento esterno e, quando esso si verifica, viene spedito all'oggetto appropriato. Il controllo viene gestito da un dispatcher, che invoca i servizi di un sottosistema. Il web-server gestisce le richieste del client, smistandole alle servlet, che gestiscono la richiesta, interagiscono con i Model e aggiornano la View per il reindirizzamento dell'interfaccia utente.

3.7 CONDIZIONI LIMITE

In caso di failure dovuto all'hardware o al software, gli sviluppatori cercheranno di ripristinare il sistema allo stato precedente all'errore. Invece, in caso di failure del DBMS, MySQL fornisce meccanismi di ripristino attraverso dei backup.

4. SERVIZI DEI SOTTOSISTEMI

4.1 INTERFACCIA UTENTE

Gestisce le pagine che vengono presentate all'utente attraverso l'uso di HTML/JSP, JS e CSS.

4.2 GESTIONE AUTENTICAZIONE

- Login: consente di effettuare l'accesso al sistema.
- Logout: consente di uscire dal sistema.
- Invio Richiesta Registrazione Cliente: consente di inviare una richiesta di registrazione come cliente.
- Invio Richiesta Registrazione Rivenditore: consente di inviare una richiesta di registrazione come rivenditore.
- Verifica Account: consente di verificare un cliente appena registrato.

4.3 GESTIONE DASHBOARD

- Inserimento Indirizzo: consente ad un utente di inserire un nuovo indirizzo.
- Inserimento Carta: consente ad un utente di inserire una nuova carta.
- Inserisci Prodotto: consente ad un rivenditore di inviare una richiesta di inserimento prodotto.
- Modifica Prodotto: consente ad un rivenditore di inviare una richiesta di modifica prodotto.
- Verifica Account Rivenditore: consente ad un amministratore di approvare un rivenditore.
- Sospendi Account: consente ad un amministratore di sospendere un account a tempo indeterminato.
- Elimina Account: consente ad un amministratore di eliminare totalmente un account.
- Verifica Recensioni: consente ad un amministratore di verificare una recensione.
- Verifica Prodotto: consente ad un amministratore di verificare l'inserimento o la modifica di un prodotto.

4.3 GESTIONE CONTENUTI

- Ricerca Prodotto per Categoria: consente ad un utente di cercare un prodotto in base alla sua categoria.
- Ricerca Prodotto per Parola Chiave: consente ad un utente di cercare un prodotto in base ad una parola chiave nel titolo.

- Visualizza Pagina Singolo Prodotto: consente ad un utente di visualizzare la pagina singola di un prodotto.
- Ricerca Rivenditore: consente ad un utente di cercare un rivenditore in base ad una parola chiave nella sua ragione sociale.
- Visualizza Pagina Informativa Rivenditore: consente ad un utente di visualizzare la pagina singola di un rivenditore.

4.4 GESTIONE ACQUISTI

- Aggiunta al Carrello: consente ad un utente di aggiungere un prodotto al carrello.
- Acquisto: consente ad un utente di acquistare un prodotto.

4.5 STORAGE

Gestisce i servizi per la memorizzazione dei dati persistenti, relativi ai sottosistemi sopra indicati