



2024/25

SDD

System Design Document
Digitronics



System Design Document

CHIERCHIA PAOLO GIOVANNI, DI MICCO VINCENZO, ZOCCOLA DOMENICO

Revision History.....	2
Team Members	3
1. Introduzione.....	3
1.1. Scopo del sistema	3
1.2. Obiettivi di design (Design Goals)	3
Design Goals	4
Trade-off	6
1.3 Definizioni, acronimi e abbreviazioni.....	6
1.4 Riferimenti.....	6
1.5 Organizzazione del documento	7
2. Architettura del sistema corrente.....	7
3. Architettura del sistema proposto	7
3.1. Panoramica della sezione.....	7
3.2. Decomposizione del sistema	8
Diagramma Architetture	10
Sottosistema Gestione Ordine.....	10
Sottosistema Recensione	12
Sottosistema Modifica Prodotto	13
3.3 Mapping hardware/software.....	13
3.4 Gestione dei dati persistenti	14
Entity Class Diagram ristrutturato	15
3.5 Controllo degli accessi e sicurezza	16
3.6 Controllo globale del software	16
3.7 Condizioni limite	17
Avvio del sistema.....	17
Spegnimento del sistema.....	17
Fallimento del sistema.....	18
4.Servizi dei sottosistemi	19
4.1 Registrazione.....	19
4.2 Autenticazione	19
4.3 Info Prodotto	20
4.4 Gestione Prodotto	20
4.5 Gestione Ordine	21
4.6 Gestione Carrello	21
4.7 Gestione Recensione	21
5.Glossario.....	22

Revision History

Data	Versione	Descrizione	Autori
23/11/2024	0.1	Stesura capitolo 1 e 2	Tutto il team
30/11/2024	0.2	Stesura capitolo 3.1, 3.2, 3.3	Tutto il team
30/11/2024	0.3	Capitolo 3.4: ristrutturazione class diagram	GPC
30/11/2024	0.4	Capitolo 3.5: Controllo degli accessi e sicurezza	VDM
30/11/2024	0.5	Stesura capitolo 3.6	Tutto il team
07/12/2024	0.6	Aggiunta UCBC_1	VDM
07/12/2024	0.7	Aggiunta UCBC_2	DZ
07/12/2024	0.8	Aggiunta UCBC_3	GPC
07/12/2024	0.9	Capitolo 4: Servizi dei sottosistemi	DZ
07/12/2024	0.10	Aggiunta glossario	VDM
16/01/2025	1.0	Revisione finale	Tutto il team

Team Members

Nome	Ruolo nel Progetto	Acronimo	Informazioni di contatto
Giovanni Paolo Chierchia	Team Member	GPC	g.chierchia8@studenti.unisa.it
Vincenzo Di Micco	Team Member	VDM	v.dimicco4@studenti.unisa.it
Domenico Zoccola	Team Member	DZ	d.zoccola3@studenti.unisa.it

1. Introduzione

1.1. Scopo del sistema

Digitronics è un e-commerce che si occupa di vendere principalmente prodotti di elettronica di consumo quindi tablet, smartphone, smartwatch ed accessori relativi. Il sito offre la possibilità agli utenti di effettuare recensioni ed acquistare i prodotti presenti nel catalogo del sito.

Il sistema è gestito da uno o più admin, permette la registrazione a chiunque voglia fare un ordine sul sito o fare recensioni dei prodotti presenti sul sito.

1.2. Obiettivi di design (Design Goals)

In questa sezione vengono presentati i Design Goal, ovvero le caratteristiche su cui si deve concentrare il nostro sistema. I Design Goals sono suddivisi in categorie e ogni obiettivo è prioritizzato per guidare efficacemente lo sviluppo del sistema.

Seguendo le linee guida del libro Bernd Bruegge – Object Oriented Software Engineering i design goal sono stati suddivisi nelle seguenti categorie:

- **Performance:** Include i requisiti di velocità e utilizzo delle risorse (spazio, memoria).
- **Dependability:** Riguarda la capacità del sistema di ridurre al minimo i fallimenti, come crash e vulnerabilità di sicurezza.
- **Maintenance:** indicano la quantità di sforzo necessaria per modificare il sistema dopo il suo rilascio.
- **End User:** comprendono le qualità desiderabili dal punto di vista dell'utente, che non sono coperte da Performance e Dependability.

Ogni Design Goal è descritto da:

- **Rank:** un valore che indica la priorità dell'obiettivo, compreso tra 1 (massima priorità) e 16 (minima priorità).
- **ID Design Goal:** un identificatore univoco accompagnato da un nome descrittivo.
- **Descrizione:** una spiegazione dettagliata dell'obiettivo di design.

- **Categoria:** la categoria a cui appartiene ciascun Design Goal.
- **RNF di origine:** ovvero il requisito non funzionale che lo ha generato.

Design Goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
9	DG_1 Tempi di risposta	Il tempo di caricamento di ogni pagina non deve superare i 5 secondi	Performance	RNF_P_5
8	DG_2 Memoria	Il sistema deve garantire la capacità di memorizzare e gestire almeno: 1000 prodotti, 2000 account utente, 10000 ordini e 10000 recensioni con i relativi dati associati	Performance	RNF_P_3
10	DG_3 Navigazione concorrente	Il sistema deve essere progettato per gestire un massimo di 150 utenti concorrenti che accedono simultaneamente al sito.	Performance	RNF_P_2
2	DG_4 Affidabilità	Deve funzionare in modo continuo con il 95% di tolleranza agli errori	Dependability	RNF_A_1
11	DG_5 Disponibilità	Disponibilità garantita 24/7 con esclusione dei 5 minuti di manutenzione giornaliera dalle 2:00 alle 2:05	Dependability	RNF_P_4
3	DG_6 Fault Tolerance	In caso di errore notifica l'utente entro 5 secondi specificando un	Dependability	RNF_A_3

		messaggio che spiega l'errore avvenuto.		
1	DG_7 Sicurezza	Il sistema deve garantire la restrizione degli accessi in modo programmatico per garantire la separazione tra utente ed admin. Deve essere garantita la memorizzazione della password crittografata tramite SHA-1. I dati devono viaggiare in maniera sicura su internet utilizzando il protocollo HTTPS.	Dependability	RNF_A_2 RNF_A_4
7	DG_8 Portabilità	Il sistema deve essere sviluppato in modo da essere disponibile e funzionale sui browser Chrome, Firefox e Safari ed Edge. In particolare, deve essere capace di adattarsi ai principali dispositivi presenti sul web: smartphone, tablet, desktop e portatili, deve adattarsi a schermi che vanno dai 4 ai 30 pollici e con rapporto 16:9 e 19.5:9	Maintenance	RNF_S_2 RNF_I_1 RNF_P_1 RNF_PA_1
4	DG_9 Manutenibilità	Deve essere possibile risolvere i problemi con tempo medio inferiore alle 4 ore per il 90% degli interventi	Maintenance	RNF_S_1

5	DG_10 Usabilità	Garantire che il 95% degli utenti, inclusi i meno esperti, completino operazioni principali senza assistenza in massimo 6 click	End User	RNF_U_1 RNF_U_3
6	DG_11 Utilità	Interfaccia intuitiva che rappresenti tutte le funzionalità principali per l'utente registrato e l'admin tramite elementi visivi riconoscibili.	End User	RNF_U_2 RNF_O_1

Trade-off

Trade-off	Descrizione
Tempi di risposta vs Sicurezza	Per migliorare la sicurezza del sito, si prevede di implementare sistemi che privilegino la protezione, anche se ciò potrebbe comportare un aumento dei tempi di elaborazione, fino a un massimo di 5 secondi.
Tempo di distribuzione vs Funzionalità	Per garantire il rispetto della consegna, in caso di ritardi, possono essere sviluppate meno funzionalità di quelle specificate rientrando nelle tempistiche concordate

1.3 Definizioni, acronimi e abbreviazioni

Vengono riportate di seguito alcune definizioni presenti nel documento corrente:

- **SDD**: System Design Document
- **RAD**: Requirements Analysis Document
- **UCBC**: Use Case Boundary Condition
- **DG**: Design Goal
- **UC**: Use Case

1.4 Riferimenti

- Slides del corso presenti sulla piattaforma e-learning;
- Libro di testo "Object Oriented Software Engineering Using UML Patterns and Java Prentice Hall 2010 Bernd Bruegge Allen H.Dutoit".

- RAD.

1.5 Organizzazione del documento

- **Introduzione:** Fornisce una descrizione generale dello scopo del sistema e degli obiettivi di design che si intendono raggiungere con il suo sviluppo.
- **Architettura software attuale:** Illustra la situazione corrente dell'architettura del software già esistente.
- **Architettura software proposta:** Descrive in che modo il sistema verrà strutturato e suddiviso in sottosistemi, con dettagli sul mapping tra hardware e software, oltre alla gestione dei dati persistenti. Vengono inoltre illustrate la configurazione dei sottosistemi reputati più importanti e le condizioni limite che riguardano l'intero sistema.
- **Glossario:** Include un elenco dei termini utilizzati nel documento, accompagnati dalla loro relativa definizione.

2. Architettura del sistema corrente

Esistono software simili al nostro, ad esempio MediaWorld, Expert o Euronics. Le architetture di questo e-commerce di vendita di prodotti di elettronica solitamente utilizzano un'architettura three-tier o MVC. Il nostro obiettivo è quello di prendere ispirazione da questi software e implementare un software usabile e sicuro facendo attenzione anche alla portabilità.

3. Architettura del sistema proposto

3.1. Panoramica della sezione

Il sistema proposto adotta lo stile architetturale **Three-Tier**, una scelta che consente di separare logicamente e fisicamente il sistema in tre livelli distinti: **Interface Layer**, **Application Logic Layer** e **Storage Layer**. Questa struttura non solo organizza il sistema in modo chiaro, ma offre anche una serie di benefici che migliorano la qualità complessiva del software.

In primo luogo, questa architettura favorisce una maggiore **leggibilità** del codice. La suddivisione in livelli con responsabilità ben definite rende il sistema più semplice da comprendere e navigare. Ad esempio, la logica applicativa e le regole di business risiedono esclusivamente nel livello applicativo, evitando di mescolarsi con il codice che gestisce l'interfaccia utente o l'accesso ai dati.

In secondo luogo, il modello facilita la **manutenzione**. Ogni livello è progettato per essere indipendente dagli altri, il che significa che le modifiche apportate in un livello – come un aggiornamento al database o un miglioramento dell'interfaccia utente – non richiedono necessariamente cambiamenti negli altri. Questo isolamento riduce il rischio di introdurre errori e velocizza gli aggiornamenti del sistema.

Un altro aspetto fondamentale è il **riuso dei componenti**. I moduli sviluppati in uno dei livelli, ad esempio quelli per la logica di business, possono essere facilmente riutilizzati in altri progetti con requisiti simili. Ciò è particolarmente vantaggioso quando si sviluppano applicazioni che condividono funzionalità comuni ma differiscono per l'interfaccia o il contesto di utilizzo.

Infine, il modello **Three-Tier** è stato scelto poiché si adatta allo sviluppo di web application e i vari membri del gruppo hanno avuto già esperienza con l'architettura, ciò permette di velocizzare lo sviluppo del sistema e di facilitare la manutenzione dei vari componenti.

Per la gestione del database relazionale verrà utilizzato **MySQL**.

Per il collegamento al database sarà utilizzato **JDBC**.

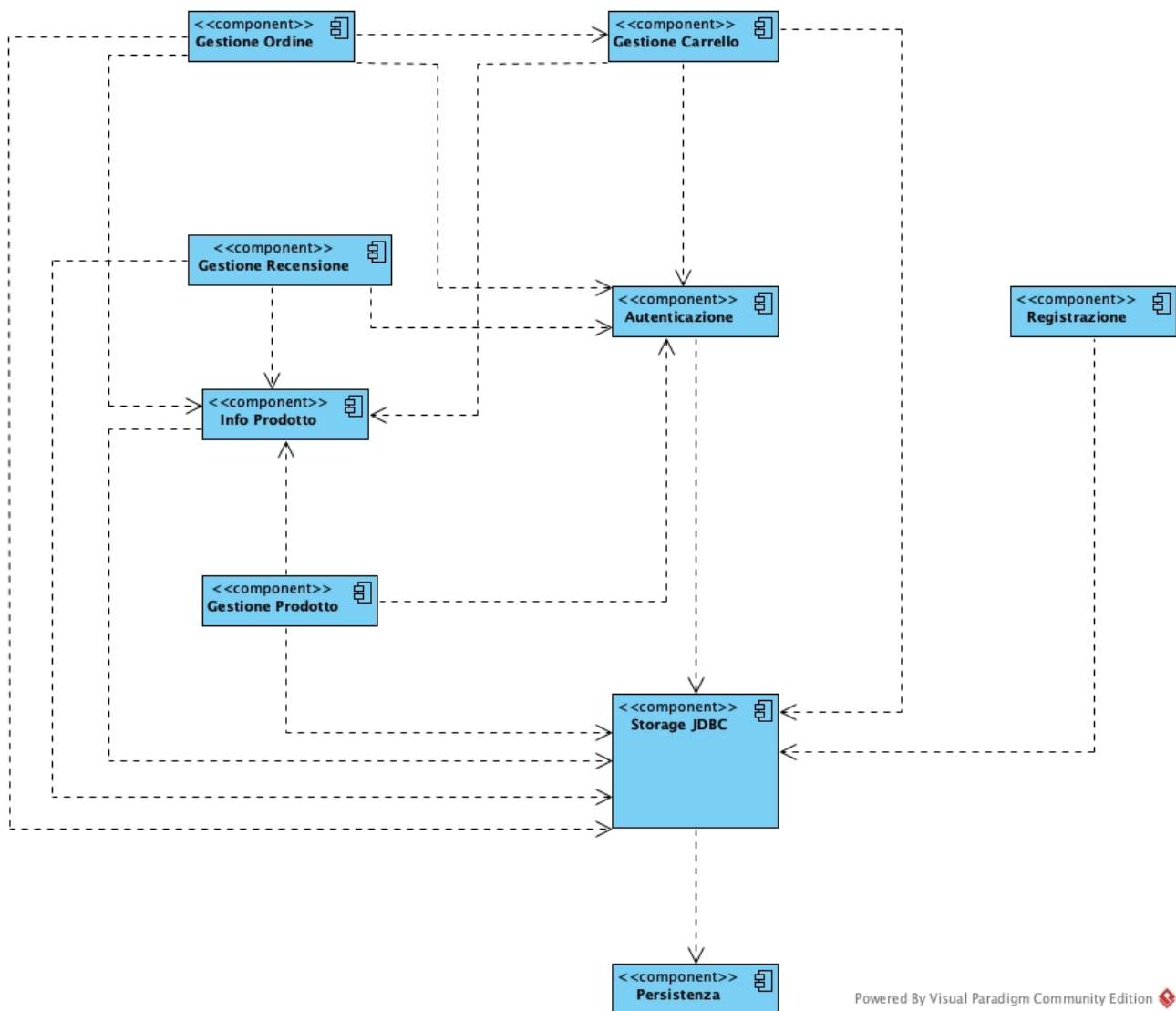
Le interfacce utente e il front-end saranno realizzati con **JSP**, integrando **HTML5** e **CSS** per la definizione delle view.

3.2. Decomposizione del sistema

I sottosistemi individuati sono:

- **Registrazione**: include l'operazione di registrazione per un utente non registrato.
- **Autenticazione**: include operazioni di accesso e di logout per l'utente registrato e l'admin, di visualizzazione dei dati profilo esclusivamente per l'utente registrato e di aggiunta admin esclusivamente per l'admin.
- **Info Prodotto**: include le operazioni che consentono la ricerca e la visualizzazione dei dettagli di un prodotto.
- **Gestione Prodotto**: include le operazioni che permettono all'admin di aggiungere, modificare ed eliminare un prodotto.
- **Gestione Ordine**: include le operazioni che permettono all'utente registrato effettuare un ordine e di visualizzare lo storico dei propri ordini e i dettagli relativi a quegli ordini. Permettono all'admin di visualizzare lo storico ordini di tutti gli utenti registrati e anche i dettagli relativi a quegli ordini.
- **Gestione Carrello**: include operazioni che consentono all'utente registrato di aggiungere un prodotto al carrello, rimuoverlo e modificare le quantità del prodotto.
- **Gestione Recensione**: include operazioni che consentono all'utente registrato di fare una recensione e ad un admin di effettuare recensioni e di rimuovere quelle inopportune.
- **Persistenza**: si occupa di gestire la persistenza dei dati con un database MySQL.
- **Storage JDBC**: sottosistema che permette la comunicazione tra i vari sottosistemi e il sottosistema Persistenza.

Sono mostrate di seguito le dipendenze tra i sottosistemi attraverso un component diagram UML:



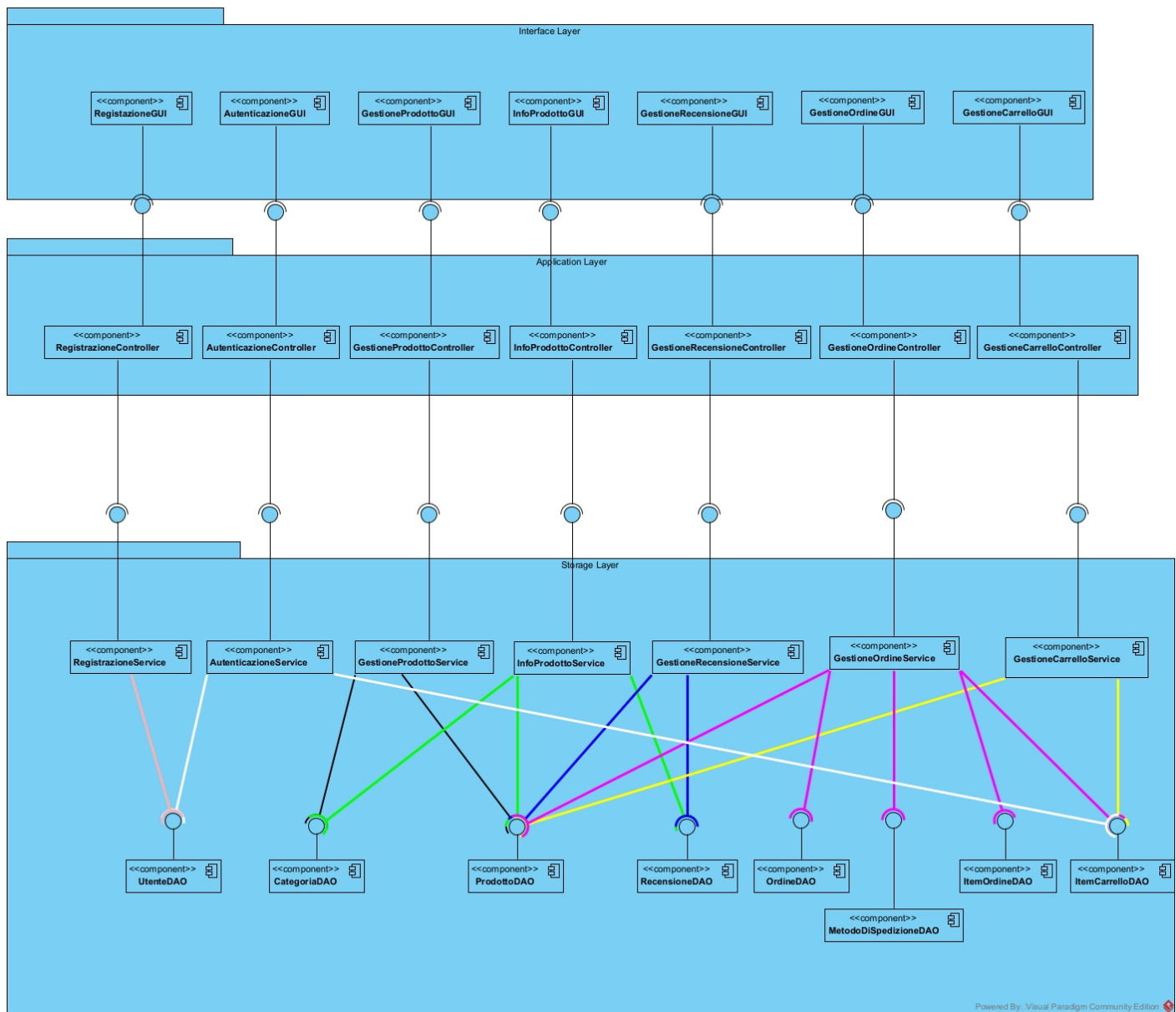
Due sottosistemi sono gestiti da componenti COTS (Commercial off the shelf):

- **JDBC**: utilizzato per gestire il sottosistema **Storage JDBC**
- **MySQL**: DBMS relazionale usato per gestire il sottosistema **Persistenza**

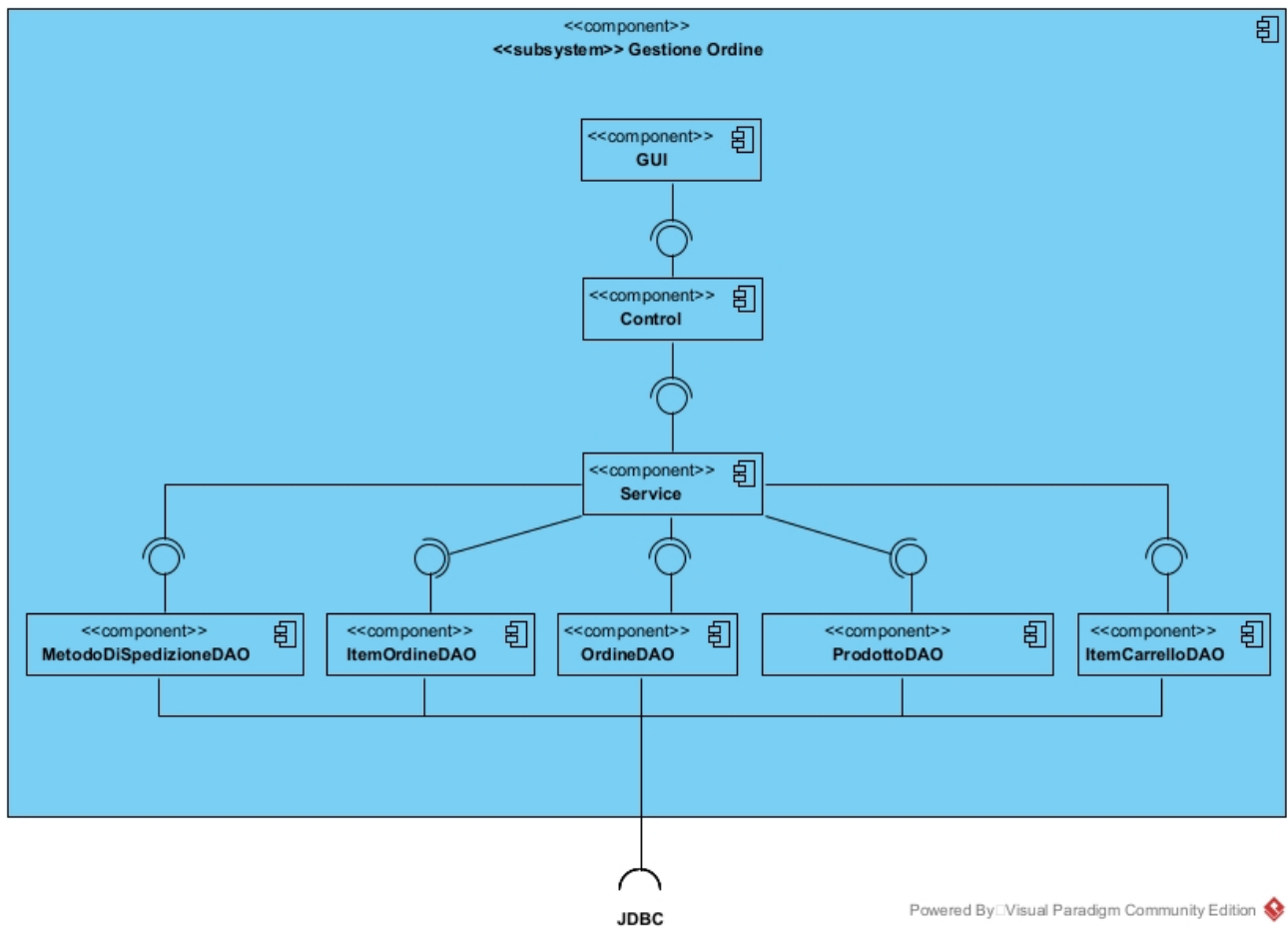
Di seguito viene fornita una descrizione dettagliata dei principali sottosistemi e delle loro componenti:

- **Interfaccia Grafica (GUI)**: Comprende le diverse visualizzazioni (view) che verranno generate e utilizzate per costruire le pagine web destinate agli utenti finali.
- **Controller**: Gestisce la logica di controllo del sistema, coordinando le interazioni tra l'interfaccia utente e i servizi sottostanti.
- **Servizio (Service)**: Si occupa dell'elaborazione della logica di business.
- **Oggetto di Accesso ai Dati (DAO)**: Fornisce i metodi necessari per interagire con i dati persistenti, consentendo operazioni come la lettura e la scrittura su database o altre fonti di dati.

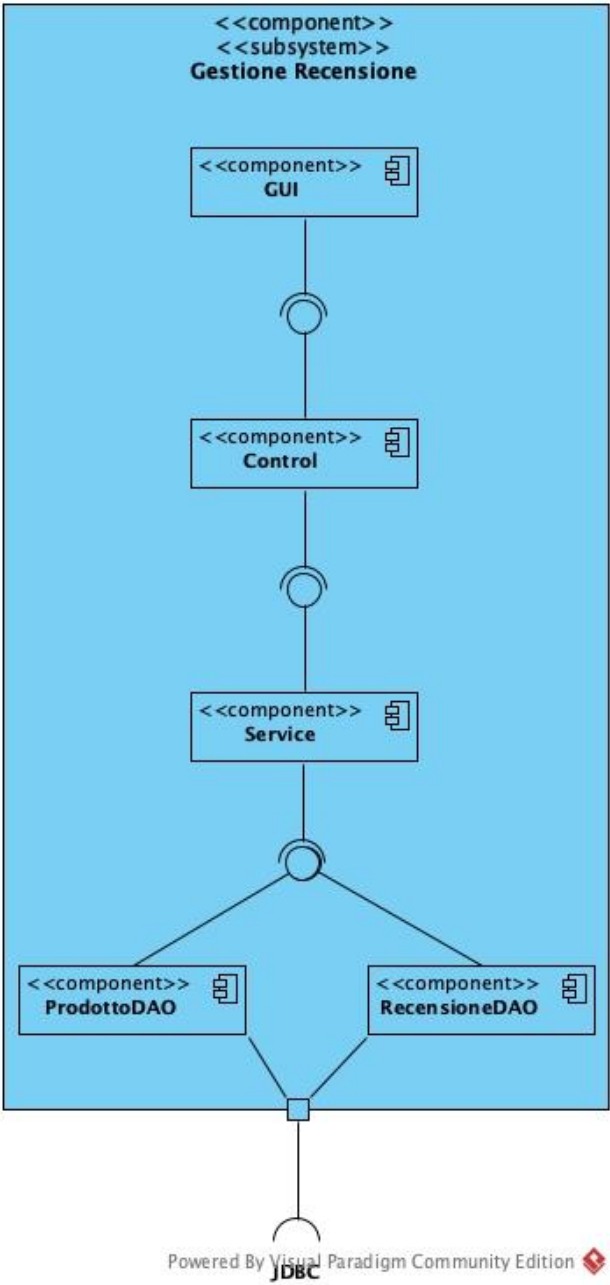
Diagramma Architeturale



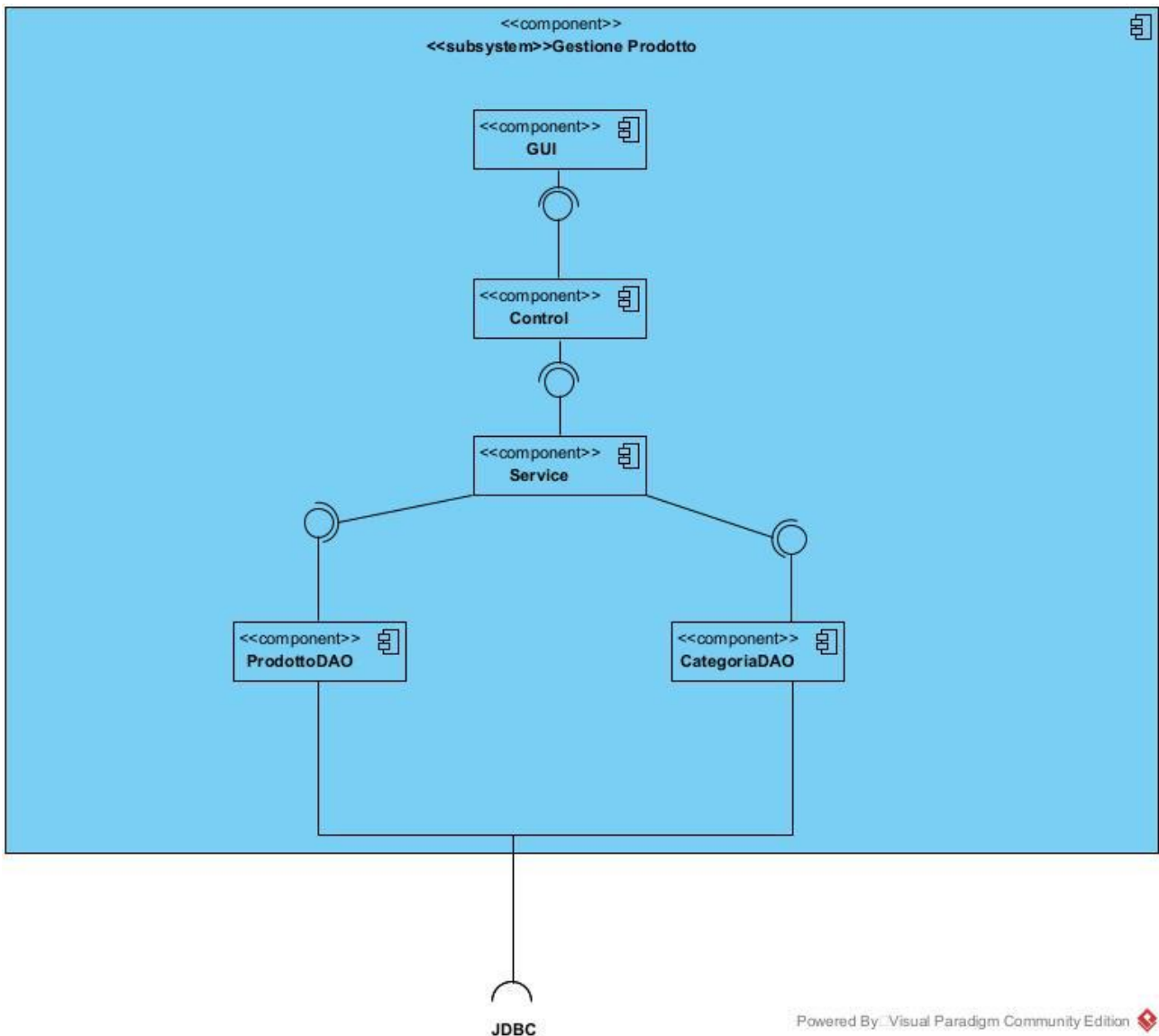
Sottosistema Gestione Ordine



Sottosistema Recensione



Sottosistema Modifica Prodotto

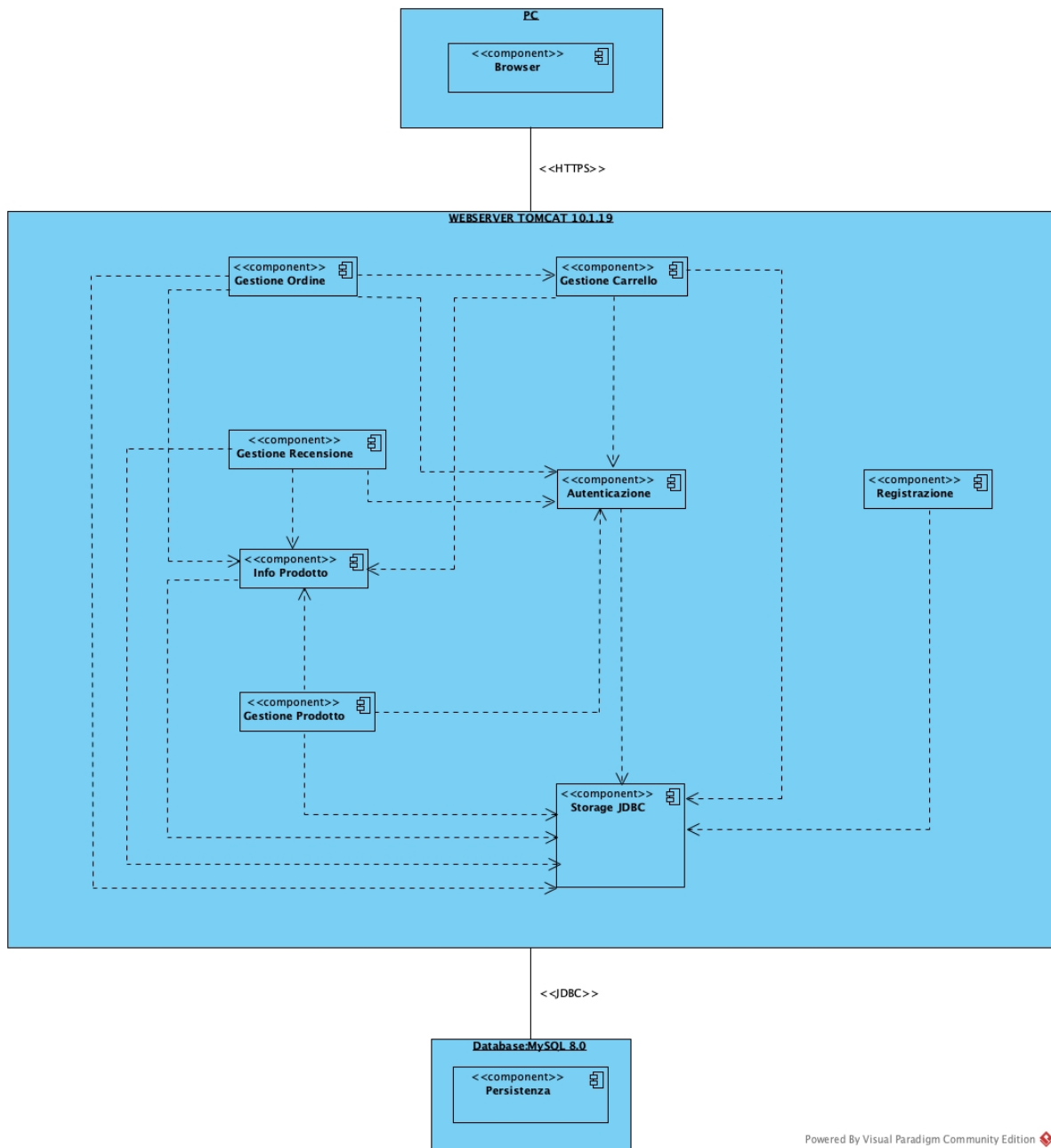


3.3 Mapping hardware/software

L'applicazione web in sviluppo sarà implementata su una piattaforma hardware composta da un server, il quale gestisce le richieste provenienti dai client attraverso un browser e una connessione a Internet.

Poiché si tratta di una web application ospitata su un web server e basata su un'architettura non distribuita, l'intero sistema risiede su un unico nodo.

Di seguito viene presentato un **UML deployment diagram** che illustra il mapping tra hardware e software.



Powered By Visual Paradigm Community Edition

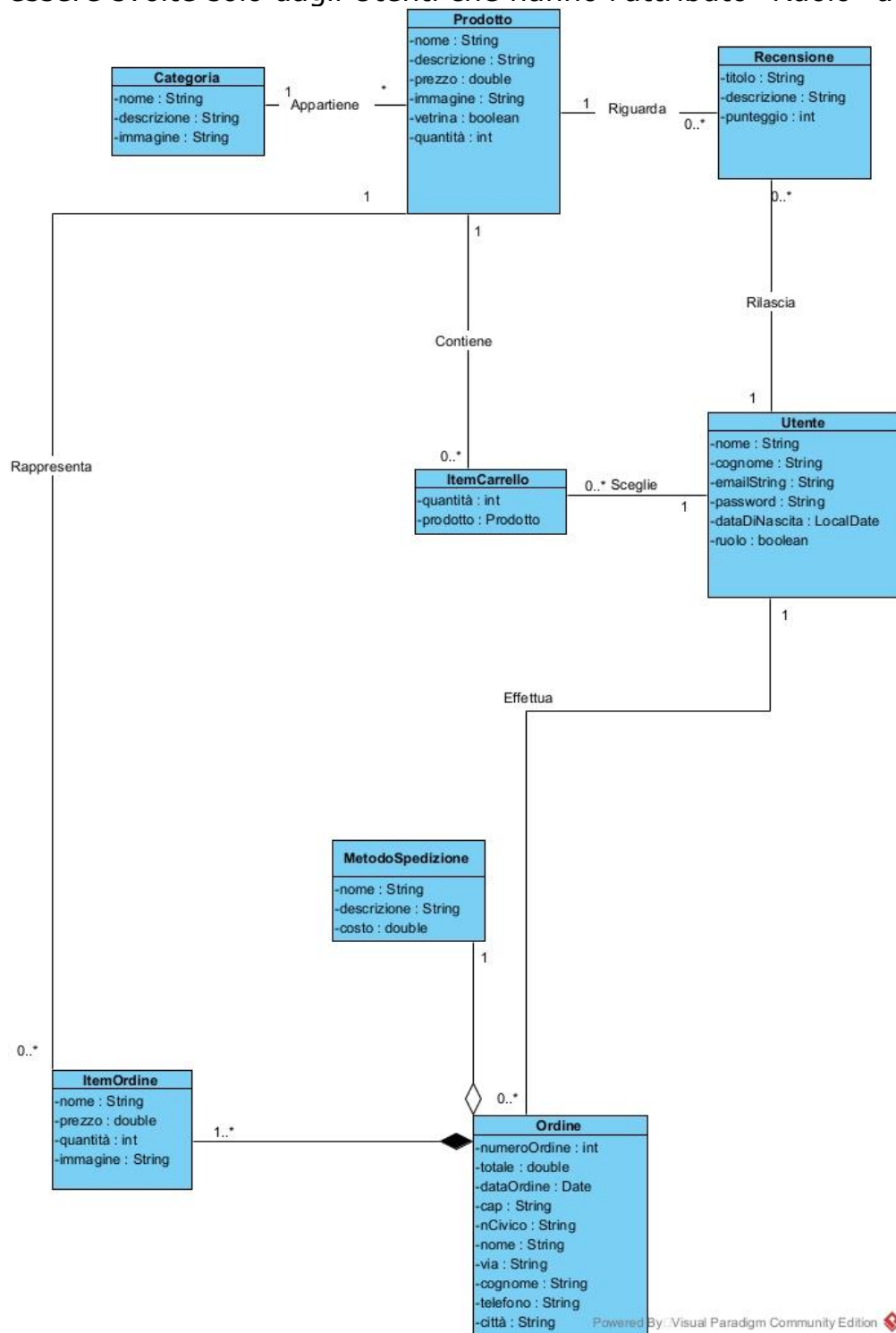
3.4 Gestione dei dati persistenti

Per la gestione dei dati persistenti si è deciso di usare un database relazionale in quanto permette di gestire in modo facile e sicuro i dati e facilitare la gestione dell'accesso da parte di utenti multipli. La gestione dei dati avviene tramite il DBMS MySQL che ci permette di rispettare al meglio i design goal, dal momento che offre la possibilità di utilizzare:

- Vincoli di integrità sui dati
- Protezione dei dati da accessi multipli

Entity Class Diagram ristrutturato

Durante l'analisi dei requisiti si era deciso di creare un'entità Carrello con un proprio ID, nella ristrutturazione si è deciso di rimuoverla in quanto si ha un carrello per ogni utente e rimuoverla permette di semplificare il diagramma senza togliere informazioni importanti. Si è anche deciso di accorpare Utente Registrato ed Admin in un'unica entità chiamata Utente ed identificare l'Admin tramite un attributo aggiuntivo chiamato "Ruolo". Le attività di Admin possono essere svolte solo dagli Utenti che hanno l'attributo "Ruolo" uguale a true.



3.5 Controllo degli accessi e sicurezza

Di seguito viene mostrata la matrice degli accessi per poter tenere traccia di quali attori possono accedere a quali servizi offerti dal sistema.

Oggetti \ Attori	Admin	Utente registrato
Registrazione	(Possibile solo quando non è stata fatta la registrazione)	(Possibile solo quando non è stata fatta la registrazione)
Autenticazione	Login Logout Aggiunta Admin	Login Logout Visualizza profilo
Gestione prodotto	Aggiunta prodotto Rimozione prodotto Modifica prodotto	
Info prodotto	Visualizza dettagli prodotti Ricerca prodotto tramite barra di ricerca Visualizza lista prodotti raggruppati in categorie	Visualizza dettagli prodotti Ricerca prodotto tramite barra di ricerca Visualizza lista prodotti raggruppati in categorie
Gestione carrello		Aggiunta prodotto al carrello Rimozione prodotto dal carrello Modifica quantità prodotto nel carrello
Gestione ordine	Visualizza tutti gli ordini Visualizza dettagli ordine	Checkout prodotti Visualizza storico ordine Visualizza dettagli ordine
Gestione recensione	Aggiunta recensione Rimozione recensione	Aggiunta recensione

3.6 Controllo globale del software

Il sistema Digitronics è un'applicazione interattiva in cui ogni funzionalità viene attivata in risposta a un comando inviato dall'utente attraverso un'interfaccia grafica. Quando l'utente desidera accedere e utilizzare una specifica funzionalità del sistema, può farlo tramite l'interfaccia grafica, che individua il controllo associato alla funzionalità richiesta.

Questa azione genera un evento che viene gestito da un gestore dedicato. Il gestore a sua volta, dirige il flusso di controllo verso il sottosistema responsabile della logica di controllo, il quale si interfaccia con i servizi per eseguire la logica applicativa.

Di conseguenza, il sistema adotta un meccanismo di gestione del flusso basato sugli eventi (**event-driven**), caratteristico di una web application.

3.7 Condizioni limite

In questo paragrafo verranno illustrate le condizioni limite relative all'avvio del sistema, alla sua chiusura, ai possibili guasti del sistema e agli errori nell'accesso ai dati persistenti.

Avvio del sistema

Identificativo <i>UCBC_1</i>	Avvio del sistema	Data	04/12/24
		Versione	1.00.000
		Autore	Vincenzo Di Micco
Descrizione	Lo UC permette l'avvio del sistema		
Attore Principale	Admin		
Attori Secondari	NA		
Entry Condition	L'admin accede al Server		
Exit Condition On success	Il sistema viene avviato correttamente		
Exit Condition On failure	Il sistema non viene avviato		
Rilevanza/User Priority	Alta		
Frequenza Stimata	1-2/settimana		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Admin:	Esegue il comando che avvia il sistema	
2	Sistema:	Verifica la connessione ai dati persistenti, se sani rende disponibili i suoi servizi e le disponibilità agli utenti	
Scenario/Flusso di eventi Alternativo: Non è possibile reperire i dati persistenti			
2.a1	Sistema:	Notifica l'admin di problemi con i dati persistenti e non effettua l'avvio	
2.a2	Admin	Corregge i dati persistenti	
2.a3	Admin	Esegue il punto 1	

Spegnimento del sistema

Identificativo <i>UCBC_2</i>	Spegnimento del sistema	Data	04/12/24
		Versione	1.00.000
		Autore	Domenico Zoccola
Descrizione	Lo UC permette lo spegnimento del sistema		
Attore Principale	Admin È interessato ad effettuare lo spegnimento del sistema		

Attori Secondari	NA	
Entry Condition	L'admin accede al sistema AND Il sistema è stato in precedenza avviato AND Il sistema non è stato ancora spento	
Exit Condition On success	Il sistema viene spento correttamente ed impedisce ad ogni client di collegarsi al sistema	
Exit Condition On failure	NA	
Rilevanza/User Priority	Alta	
Frequenza Stimata	1-2 usi/settimana	
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Utente:	Esegue il comando che stoppa il sistema
2	Sistema:	Procedi con l'arresto del sistema

Fallimento del sistema

Identificativo <i>UCBC_3</i>	Fallimento del sistema	Data	04/12/24
		Versione	1.00.000
		Autore	Giovanni Paolo Chierchia
Descrizione	Lo UC descrive il comportamento del sistema in caso di fallimento che possono avvenire principalmente per mancanza di elettricità, guasti all’hardware o perdita di connessione con database		
Attore Principale	Admin È interessato a riavviare il sistema		
Attori Secondari	NA		
Entry Condition	Il sistema viene terminato in modo inaspettato		
Exit Condition On success	Il sistema viene riavviato correttamente		
Exit Condition On failure	Il sistema non viene riavviato		
Rilevanza/User Priority	Alta		
Frequenza Stimata	1-2 usi/mese		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Admin:	Include UCBC_1	
Identificativo <i>UCBC_3</i>	Fallimento del sistema	Data	04/12/24
		Versione	1.00.000
		Autore	Giovanni Paolo Chierchia

Descrizione	Lo UC descrive il comportamento del sistema in caso di fallimento
Attore Principale	Admin È interessato a riavviare il sistema
Attori Secondari	NA
Entry Condition	Il sistema viene terminato in modo inaspettato
Exit Condition On success	Il sistema viene riavviato correttamente
Exit Condition On failure	Il sistema non viene riavviato
Rilevanza/User Priority	Alta
Frequenza Stimata	1-2 usi/mese
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO	
1	Admin: Include UCBC_1

4. Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

4.1 Registrazione

Sottosistema	Registrazione
Descrizione	Sottosistema che comprende la funzionalità di registrazione per gli utenti non registrati.
Servizi Offerti	
Servizio	Descrizione
Registrazione	Funzione che permette agli utenti non registrati di registrarsi.

4.2 Autenticazione

Sottosistema	Autenticazione
Descrizione	Sottosistema che gestisce le funzionalità per l'autenticazione e di visualizzazione dei dati dell'account: login e logout per gli utenti registrati e admin, visualizzazione dei dati profilo per l'utente registrato e admin.
Servizi Offerti	
Servizio	Descrizione
Login	Funzione che permette all'utente di accedere al sistema tramite le proprie

	credenziali (e-mail e password) per sfruttare tutte le funzionalità offerte dal sistema.
Logout	Funzione che permette all'utente di disconnettersi dal sistema.
Visualizza Profilo	Funzione che permette all'utente registrato di visualizzare i dati relativi al proprio profilo.
Aggiunta Admin	Funzione che consente all'admin l'aggiunta di un utente come ordine

4.3 Info Prodotto

Sottosistema	Info Prodotto
Descrizione	Sottosistema che comprende la funzionalità inerenti alla ricerca di un prodotto e di visualizzazione dei dettagli di un prodotto.
Servizi Offerti	
Servizio	Descrizione
Ricerca Prodotto tramite barra di ricerca	Funzione che permette all'utente di cercare un prodotto inserendo il nome nella barra di ricerca.
Visualizza lista prodotti raggruppati per categorie	Funzione che permette all'utente di visualizzare una lista di prodotti basati su una certa categoria.
Visualizza dettagli prodotti	Funzione che permette all'utente di visualizzare tutti i dettagli inerenti a un determinato prodotto dopo averlo selezionato.

4.4 Gestione Prodotto

Sottosistema	Gestione Prodotto
Descrizione	Sottosistema che comprende funzionalità per la gestione dei prodotti da parte dell'admin.
Servizi Offerti	
Servizio	Descrizione
Aggiunta Prodotto	Funzione che permette all'admin di aggiungere un prodotto.
Modifica Prodotto	Funzione che permette all'admin di modificare un prodotto.
Rimozione Prodotto	Funzione che permette all'admin di rimuovere un prodotto.

4.5 Gestione Ordine

Sottosistema	Gestione Ordine
Descrizione	Sottosistema che gestisce le funzionalità che gestisce la visualizzazione degli ordini.
Servizi Offerti	
Servizio	Descrizione
Checkout Prodotti	Funzione che permette all'utente registrato di acquistare i prodotti presenti nel carrello.
Visualizza tutti gli ordini	Funzione che permette all'admin di visualizzare tutti gli ordini effettuati dagli utenti registrati.
Visualizza storico ordine	Funzione che permette all'utente registrato di visualizzare lo storico dei propri ordini.
Visualizza dettagli ordine	Funzione che permette all'utente di visualizzare i dettagli di un ordine.

4.6 Gestione Carrello

Sottosistema	Gestione carrello
Descrizione	Sottosistema che gestisce le funzionalità relative al carrello.
Servizi Offerti	
Servizio	Descrizione
Aggiunta prodotto al carrello	Funzione che permette all'utente registrato di aggiungere un prodotto al carrello, specificando anche la quantità.
Rimozione prodotto dal carrello	Funzione che permette all'utente registrato di rimuovere un prodotto dal carrello.
Modifica quantità prodotto nel carrello	Funzione che permette all'utente registrato di modificare la quantità di un prodotto nel carrello.

4.7 Gestione Recensione

Sottosistema	Gestione recensione
Descrizione	Sottosistema che gestisce le funzionalità relative alle recensioni.
Servizi Offerti	
Servizio	Descrizione

Aggiunta recensione	Funzione che permette all'utente di aggiungere una recensione.
Rimozione recensione	Funzione che permette all'admin di rimuovere una recensione.

5. Glossario

Nella Seguevole sezione sono raccolti le sigle o i termini del documento che necessitano di una definizione.

- **Piattaforma:** Base software o hardware su cui sono sviluppate o eseguite applicazioni.
- **Utente:** generalizzazione di utente registrato ed admin
- **Utente registrato:** Utente che può inserire prodotti nel carrello ed effettuare acquisti e recensioni.
- **Admin:** Amministratore del sistema, può gestire in modo semplice i prodotti e visualizzare gli ordini effettuati dai clienti.
- **SHA-1:** Funzione di hash che genera un valore di 160 bit, usata per verificare l'integrità dei dati.
- **HTTPS:** Protocollo sicuro per la trasmissione di dati tra un browser e un server, crittografato tramite SSL/TLS.
- **Browser:** Software che permette di navigare su internet visualizzando pagine web.
- **Tomcat:** Server web open-source per l'esecuzione di applicazioni web basate su Java, in particolare quelle che utilizzano servlets e JSP (JavaServer Pages).
- **MySQL:** Sistema di gestione di database relazionali che utilizza SQL per l'interazione con i dati.
- **Trade-off:** sono compromessi in cui si rinuncia a un aspetto per favorirne un altro, bilanciando vantaggi e svantaggi in base alle priorità del progetto o del contesto.
- **DBMS:** (Database Management System) è un software che consente di creare, gestire e interagire con database, fornendo strumenti per l'archiviazione, l'accesso, la manipolazione e la sicurezza dei dati in modo organizzato ed efficiente.
- **Mapping Hardware-Software:** è il processo di assegnazione delle funzioni del software alle risorse hardware disponibili, definendo come il software interagirà con l'hardware per garantire prestazioni ottimali e compatibilità.
- **Sottosistema:** è una componente autonoma di un sistema più grande, progettata per svolgere una specifica funzione o insieme di funzioni, contribuendo al funzionamento complessivo del sistema principale.
- **UCBC:** Boundary condition use case