System Design Document

GoldenOnlineStore

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 18/11/2023 | 0.1 | Prima stesura | Antonio Spatuzzi  Vittorio Giordano |
| 20/11/2023 | 0.1.2 | Aggiunta Subsystem decomposition e Mapping | Tutti i membri |
| 26/11/2023 | 0.1.3 | Persistent Data Management | Tutti i membri |
| 30/11/2023 | 0.1.4 | Aggiunta Access Control and Security | Tutti i membri |
| 4/12/2023 | 0.1.5 | Aggiunta Boundary Condition | Tutti i membri |
| 10/12/2023 | 0.2 | Revisione | Tutti i membri |
| 08/01/2024 | 0.3 | Correzione errori | Tutti i membri |

[1. Introduction 4](#_Toc156400149)

[1.1 Purpose of the system 4](#_Toc156400150)

[1.2 Design Goals 4](#_Toc156400151)

[1.2.1 Performance Criteria 4](#_Toc156400152)

[1.2.2 Dependability Criteria 4](#_Toc156400153)

[1.2.3 Cost Criteria 5](#_Toc156400154)

[1.2.4 Maintenance Criteria 5](#_Toc156400155)

[1.2.5 End User Criteria 6](#_Toc156400156)

[1.3 Definitions, acronyms, and abbreviations 8](#_Toc156400157)

[1.4 References 8](#_Toc156400158)

[1.5 Overview 8](#_Toc156400159)

[2 Current Software Architecture 9](#_Toc156400160)

[3 Proposed Software Architecture 9](#_Toc156400161)

[3.1 Overview 9](#_Toc156400162)

[3.2 Subsystem Decomposition 9](#_Toc156400163)

[3.3 Mapping Hardware/Software 12](#_Toc156400164)

[3.4 Persistent Data Management 13](#_Toc156400165)

[3.5 Access control and Security 15](#_Toc156400166)

[3.6 Global Software Control 16](#_Toc156400167)

[3.7 Boundary conditions 16](#_Toc156400168)

[3.7.1 Avvio Sistema 16](#_Toc156400169)

[3.7.2 Terminazione Sistema 17](#_Toc156400170)

[3.7.3 Fallimento Sistema 17](#_Toc156400171)

[4. Subsystem services 17](#_Toc156400172)

# Introduction

GOS si presenta come uno strumento per una fumetteria di media/piccola taglia. che punta a migliorare le proprie vendite in loco ed online. GoldenOnlineStore intende anche semplificare l’acquisto di prodotti da parte di possibili utenti interessati.

## Purpose of the system

GOS è più di una semplice piattaforma di vendita online. È l'anello di congiunzione tra una fumetteria di dimensioni medio/piccole e un vasto mondo di appassionati di fumetti, giochi e collezionismo. Questa iniziativa ambiziosa non si limita a migliorare le vendite della fumetteria, ma cerca di trasformare completamente l'esperienza di acquisto, sia per i clienti abituali che per i nuovi interessati. Il cuore pulsante di GoldenOnlineStore è la sua dedizione a semplificare il processo di acquisto per gli utenti. Non importa se sei un veterano nell'universo dei fumetti o un principiante che si avvicina a questo affascinante mondo per la prima volta, GoldenOnlineStore ti accoglie con un'interfaccia intuitiva e accessibile. È un'opportunità di migliorare la tua esperienza di acquisto, di rimanere aggiornato con le ultime novità. Unisce il meglio di entrambi i mondi, offrendoti il meglio delle vendite locali e delle esperienze online in un unico, eccezionale pacchetto dedicato ai fumetti.

## Design Goals

Gli obiettivi di design identificati sono i seguenti:

### 1.2.1 Performance Criteria

|  |  |
| --- | --- |
| **Design criterion** | **Definition** |
| **Tempo di Risposta** | Il sistema deve garantire tempi di risposta agli input dell’utente non superiori a cinque secondi. |

### Dependability Criteria

|  |  |
| --- | --- |
| **Design criterion** | **Definition** |
| **Robustezza** | Gli Input immessi dai clienti ritenuti non validi dal sistema verranno segnalati attraverso messaggi di errore. |
| **Sicurezza** | Il sistema garantisce:  Il protocollo HTTPS per il trasferimento sicuro dei dati in rete;  Crittografia dati all’interno del database con l’algoritmo SHA-512  L’assegnazione dei permessi in più tipi di amministratori. |
| **Disponibilità** | Garantisce il sistema sempre attivo salvo manutenzione. |

### 1.2.3 Cost Criteria

|  |  |
| --- | --- |
| **Design criterion** | **Definition** |
| **Manutenzione** | Costo richiesto per la correzione degli errori e per l’evoluzione del sistema |
| **Amministrazione** | Costo richiesto degli amministratori delegati nel sistema. |

### 1.2.4 Maintenance Criteria

|  |  |
| --- | --- |
| **Design criterion** | **Definition** |
| **Estendibilità** | È consentito aggiungere in futuro nuove funzionalità al sistema oppure creare nuove classi, con l’estensione di quelle già esistenti. |
| **Adattabilità** | Il sistema può essere facilmente portato su diversi domini di applicazione (multipiattaforma). |
| **Portabilità** | Tutto il software relativamente associato al sistema sarà scritto usando il linguaggio di programmazione Java, HTML, CSS, nessun vincolo è imposto dalla piattaforma hardware. Quindi, in virtù della portabilità del linguaggio, l’intero sistema è portabile su differenti piattaforme. |
| **Leggibilità** | Mediante la lettura del codice il sistema è comprensibile. Infatti semplici istruzioni consentono di comprendere le funzionalità del sistema e con l’ausilio di commenti e documentazione il lavoro di comprensione è ulteriormente semplificato. |

### 1.2.5 End User Criteria

|  |  |
| --- | --- |
| **Design criterion** | **Definition** |
| **Usabilità** | Gli utenti del sistema sono agevolati nella navigazione tramite l’utilizzo dell’interfaccia user-friendly grazie agli interaction design pattern (design pattern d’interazione) |
| **Utilità** | Il lavoro degli amministratori verrà supportato nel miglior modo possibile dal sistema. Infatti l’amministratore compierà le operazioni consentitegli, senza aver il minimo problema. Il sistema lo seguirà passo passo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | ID | Descrizione | Categoria | NFR di origine |
| 4 | **Tdr000** | Il sistema deve garantire tempi di risposta agli input dell’utente non superiori a cinque secondi. | Performance Criteria | NFR\_7 |
| 11 | **Rob000** | Gli Input immessi dai clienti ritenuti non validi dal sistema verranno segnalati attraverso messaggi di errore. | Dependability Criteria | NFR\_2 |
| 5 | **Sec000** | Il sistema garantisce:   * Il protocollo HTTPS per il trasferimento sicuro dei dati in rete; * Crittografia dati all’interno del database con l’algoritmo SHA-512 * L’assegnazione dei permessi in più tipi di amministratori. | Dependability Criteria | NFR\_1 |
| 1 | **Disp000** | Garantisce il sistema sempre attivo salvo manutenzione. | Dependability Criteria | NFR\_3 |
| 10 | **Man000** | Costo richiesto per la correzione degli errori e per l’evoluzione del sistema | Cost Criteria | NFR\_6 |
| 8 | **Adm000** | Costo richiesto degli amministratori delegati nel sistema. | Cost Criteria | NFR\_1 |
| 12 | **Ext000** | È consentito aggiungere in futuro nuove funzionalità al sistema oppure creare nuove classi, con l’estensione di quelle già esistenti. | Maintenance Criteria | NFR\_5 |
| 3 | **Adp000** | Il sistema può essere facilmente portato su diversi domini di applicazione (multipiattaforma). | Maintenance Criteria | NFR\_1 |
| 2 | **Por000** | Tutto il software relativamente associato al sistema sarà scritto usando il linguaggio di programmazione Java,HTML,CSS, nessun vincolo è imposto dalla piattaforma hardware. Quindi, in virtù della portabilità del linguaggio, l’intero sistema è portabile su differenti piattaforme. | Maintenance Criteria | NFR\_8 |
| 9 | **Leg000** | Mediante la lettura del codice il sistema è comprensibile. Infatti semplici istruzioni consentono di comprendere le funzionalità del sistema e con l’ausilio di commenti e documentazione il lavoro di comprensione è ulteriormente semplificato. | Maintenance Criteria | NFR\_4 |
| 7 | **Usa000** | Gli utenti del sistema sono agevolati nella navigazione tramite l’utilizzo dell’interfaccia user-friendly grazie agli interaction design pattern (design pattern d’interazione) | End User Criteria | NFR\_4 |
| 6 | **Uti000** | Il lavoro degli amministratori verrà supportato nel miglior modo possibile dal sistema. Infatti l’amministratore compierà le operazioni consentitegli, senza aver il minimo problema. Il sistema lo seguirà passo passo. | End User Criteria | NFR\_4 |

#### Design Trade-Offs

|  |  |
| --- | --- |
| **Trade Off** | **Razionale** |
| Funzionalità vs. Usabilità | Il sistema mira ad essere intuitivo da utilizzare, consentendo la navigazione agevole e permettendo l’utilizzo delle funzionalità offerte anche senza consultare la documentazione. |
| Costi vs. Robustezza | Il sistema mira ad essere robusto e tollerante ai guasti, in modo da garantire la correttezza delle informazioni e delle funzionalità fornite. |
| Efficienza vs. Portabilità | Il sistema mira ad essere portabile, infatti l’utilizzo di GOS avviene mediante un web browser senza interazione con il sistema sottostante, c’è quindi indipendenza dal sistema operativo. |
| Sviluppo rapido vs. Funzionalità | Il sistema mira ad offrire all’utente un’esperienza ottimale, consentendo l’utilizzo di funzionalità complete e sviluppate adeguatamente |

## Definitions, acronyms, and abbreviations

* GOS: Golden Online Store
* TCG: Trading Card Game
* FR: Functional Requirements
* NFR: Non Functional Requirements
* SC: Scenario
* UC: Use Case
* UCD: Use Case Diagram
* CD: Class Diagram
* SD: Sequence Diagram
* SCD: State Chart Diagram
* UI: Mock-up
* UML: Unified Modeling Language
* JSP: Java Servlet Page
* HTML: HyperText Markup Language
* XML: eXtensible Markup Language
* CSS: Cascading Style Sheets

## References

• Bernd Bruegge e Allen H. Dutoit - Object-Oriented Software Engineering (using UML, Patterns

and JavaTM) – Prentice Hall.

• Documentazione generata dalle interviste con i futuri stakeholder del sistema software.

## Overview

Al secondo punto del documento verrà presentato il sistema corrente.

Al terzo punto sarà presentata l’architettura del sistema proposto:

* La decomposizione in sottosistemi;
* Il mapping hardware/software;
* I dati persistenti;
* Il controllo degli accessi;
* l controllo del sistema globale;
* Le condizioni limite.

Al quarto punto saranno presentati i servizi del sottosistema

# 2 Current Software Architecture

Il sistema prende spunto da vari siti e-commerce già presenti sul web adattato per le richieste del negozio.

# 3 Proposed Software Architecture

## 3.1 Overview

L’architettura del sistema GoldenOnlineStore è di tipo client/server basata sul modello Three Tiers.

In questo modo il server riceve le richieste dal client rispondendo e fornendo l’affidabilità del sistema, la robustezza e la sicurezza.

## 3.2 Subsystem Decomposition

È stata presa la decisione di adottare un'architettura Three-tier, suddivisa nei livelli di presentazione, applicazione e storage. Questa scelta è basata su una serie di fondamentali ragioni che comportano notevoli vantaggi in termini di manutenibilità, flessibilità e scalabilità.

Motivazioni per l'utilizzo dell'architettura Three-tier:

* Separazione delle Preoccupazioni: La suddivisione in tre livelli consente una netta separazione delle preoccupazioni tra la presentazione, la logica di business e la gestione dei dati. Questa chiarezza nella separazione rende il sistema più facilmente manutenibile e agevola la gestione di modifiche e aggiornamenti.
* Riusabilità del Codice: Ogni livello è concepito in modo che possa essere sviluppato e gestito in modo indipendente, promuovendo così la riusabilità del codice. Ad esempio, le funzionalità di business possono essere riusate in diverse interfacce utente senza necessità di ridisegnare l'intero sistema.
* Scalabilità: La separazione dei livelli facilita la scalabilità sia orizzontale che verticale. In caso di incremento del carico sulla presentazione, è possibile distribuire più server di presentazione. Analogamente, la scalabilità della logica di business o del livello dei dati può essere gestita in modo indipendente.
* Facilità di Manutenzione: L'architettura three-tier semplifica notevolmente le attività di manutenzione. Le modifiche apportate alla logica di business possono essere effettuate senza impattare sulla presentazione o sui dati, e viceversa. Questo riduce il rischio di effetti collaterali indesiderati durante gli aggiornamenti.

L'adozione di questa architettura, pertanto, riflette una scelta strategica orientata a massimizzare l'efficienza, la gestibilità e la resilienza del sistema di e-commerce.

La logica del sistema è divisa in tre layer principali:

* **Presentation Layer**

Composto dall’insieme delle interfacce grafiche per ogni tipo di utente del sistema.

* **Application Layer**

Composto da oggetti con lo scopo di interagire col database e di notificare i cambiamenti al Presentation Layer.

* **Storage Layer**

Ha lo scopo di memorizzare i dati e di rispondere alle interrogazioni.

All’interno del Presentation Layer vi sono 2 sottosistemi:

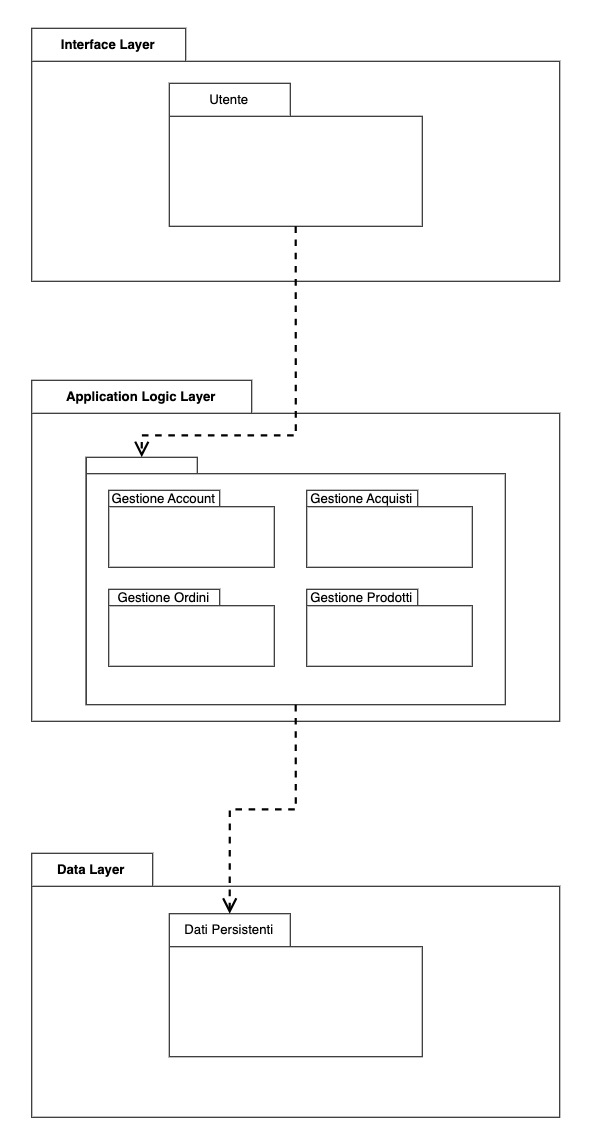
* **Interfaccia Utente**

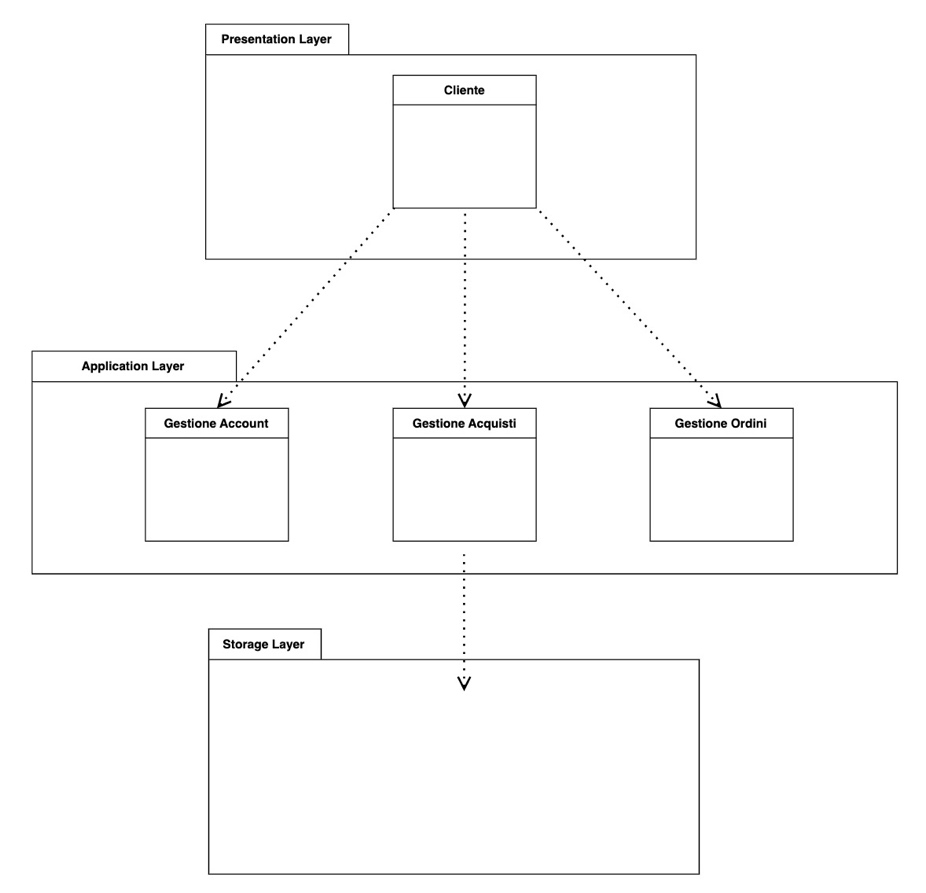
Interagisce con i sottosistemi

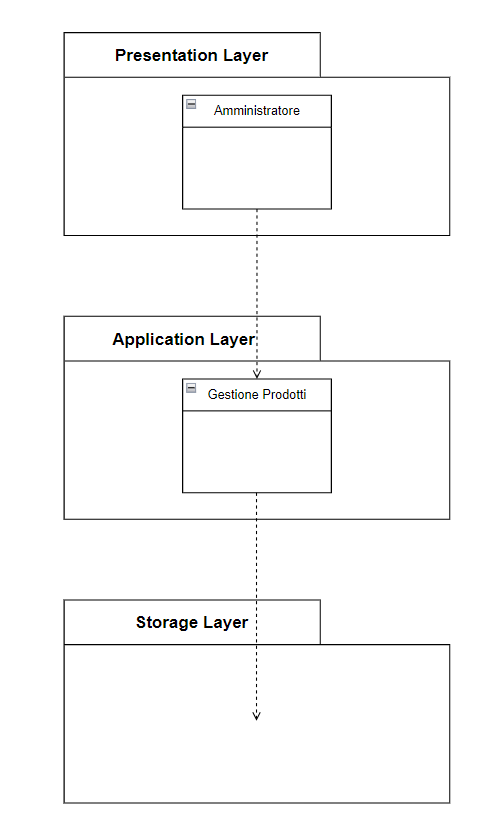
* + **Gestione Account**
  + **Gestione Acquisti**
  + **Gestione Ordini**
  + **Gestione Prodotti**
* **Interfaccia Amministratore**

Interagisce con i sottosistemi

* + **Gestione Account**
  + **Gestione Prodotti**







## 3.3 Mapping Hardware/Software

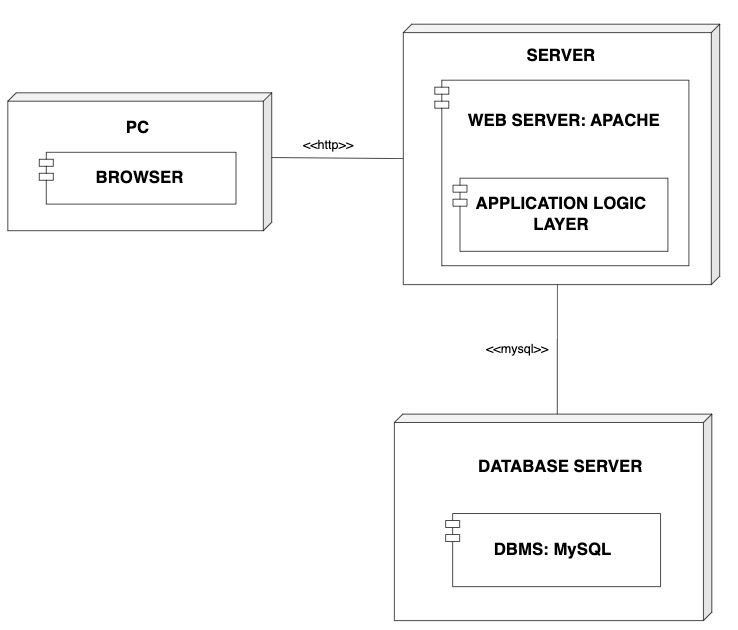
La decisione di adottare un database come sistema di gestione dei dati per un'applicazione è motivata da diverse considerazioni che mirano a garantire l'efficienza, la scalabilità e la robustezza del sistema. La capacità dei database di gestire dati complessi, le relazioni tra tabelle e di supportare query avanzate li rende particolarmente idonei per applicazioni che richiedono una struttura dati sofisticata. Inoltre, i database sono progettati con l'obiettivo di gestire operazioni di lettura e scrittura multiple in modo efficiente e sicuro. L'implementazione di meccanismi di gestione transazionale assicura l'integrità dei dati, prevenendo corruzioni in situazioni di concorrenza. Tale affidabilità è fondamentale per applicazioni in cui più utenti possono accedere simultaneamente ai dati o quando sono richieste operazioni atomiche su più risorse.

Un ulteriore vantaggio dei database è rappresentato dalla loro facilità di portabilità. Essi consentono una transizione agevole dell'applicazione su diverse piattaforme senza la necessità di riscrivere il codice di accesso ai dati. Questa caratteristica semplifica notevolmente la manutenzione e l'aggiornamento dell'applicazione, riducendo i costi associati ai cambiamenti di ambienti di sviluppo o di hosting.

Infine, la sicurezza dei dati emerge come una priorità essenziale quando si tratta di gestire informazioni sensibili. I database forniscono strumenti avanzati per regolare l'accesso ai dati, permettendo la definizione di ruoli e privilegi che garantiscono che solo utenti autorizzati possano accedere o modificare informazioni sensibili.

Il sistema che si vuole sviluppare sarà installato su un solo computer e utilizzerà un DBMS MySQl installato sullo stesso. Il sistema quindi sarà formato solo dal server che in futuro potrà essere installato su un’altra macchina. Il sistema utilizzerà un DBMS per la gestione di dati persistenti di nostro interesse.

* Protocollo richiesto: HTTP ;
* Memorizzazione dati: DBMS MySQL ;
* WebServer: Apache Tomcat ;
* Linguaggi di programmazione utilizzati: Java, Html, CSS, Javascript;



## 3.4 Persistent Data Management

Il sistema utilizzerà MySQL come DBMS relazionale per la memorizzazione dei dati. L'approccio relazionale è stato scelto per la sua affidabilità, flessibilità e supporto per le transazioni atomiche.

Immagine che contiene testo, diagramma, Piano, Disegno tecnico

Descrizione generata automaticamente

Amministratore

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Vincoli | Key |
| Id | int | AUTO\_INCREMENT  NOT NULL | PRIMARY KEY |
| Username | varchar(30) | NOT NULL | UNIQUE KEY |
| Email | varchar(50) | NOT NULL | UNIQUE KEY |
| Password | varchar(30) | NOT NULL |  |
| Nome | varchar(30) | NOT NULL |  |
| Cognome | varchar(30) | NOT NULL |  |
| CodiceDiAccesso | int | NOT NULL | UNIQUE KEY |

Cliente

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Vincoli | Key |
| Id | int | AUTO\_INCREMENT  NOT NULL | PRIMARY KEY |
| Username | varchar(30) | NOT NULL | UNIQUE KEY |
| Email | varchar(50) | NOT NULL | UNIQUE KEY |
| Password | varchar(30) | NOT NULL |  |
| Nome | varchar(30) | NOT NULL |  |
| Cognome | varchar(30) | NOT NULL |  |

Categoria

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Vincoli | Key |
| Id | int | AUTO\_INCREMENT  NOT NULL | PRIMARY KEY |
| Nome | varchar(30) | NOT NULL |  |
| Descrizione | varchar(150) | DEFAULT NULL |  |
| Macrocategoria | varchar(30) | NOT NULL |  |

Prodotto

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Vincoli | Key |
| Id | int | AUTO\_INCREMENT  NOT NULL | PRIMARY KEY |
| Nome | varchar(30) | NOT NULL |  |
| Descrizione | varchar(50) | NOT NULL |  |
| Marca | varchar(30) | NOT NULL |  |
| Prezzo | float | NOT NULL |  |
| Categoria | int | DEFAULT NULL | FOREIGN KEY |

Ordine

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Vincoli | Key |
| Id | int | AUTO\_INCREMENT  NOT NULL | PRIMARY KEY |
| Cliente | varchar(30) | NOT NULL | FOREIGN KEY |
| DataOrdine | date | NOT NULL |  |
| DataConsegna | date | NOT NULL |  |
| Totale | float | NOT NULL |  |

Indirizzo Spedizione

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Vincoli | Key |
| Ordine | int | NOT NULL | PRIMARY KEY  FOREIGN KEY |
| Via | varchar(30) | NOT NULL |  |
| Cap | varchar(5) | NOT NULL |  |
| Città | varchar(30) | NOT NULL |  |

Linea Ordine

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Vincoli | Key |
| Ordine | int | NOT NULL | PRIMARY KEY  FOREIGN KEY |
| Prodotto | int | NOT NULL | PRIMARY KEY |
| Quantità | int | NOT NULL |  |
| PrezzoUnitario | float | NOT NULL |  |

## 3.5 Access control and Security

Il controllo degli accessi è garantito tramite l’utilizzo di username e password, che verranno richieste per ogni singolo accesso.

Ogni tipo di utente potrà accedere al sito.

Il sistema prevede 3 figure di utenti: Amministratore, Cliente e Utente non registrato.

Ogni attore può accedere alle diverse funzionalità del sistema ma con diritti di accesso regolamentati sulla base delle differenti tipologie di utenza. Per documentare i diritti di accesso e per tenere traccia all’interno del sistema usufruiamo di una tabella di controllo di accesso che descrive le varie operazioni permesse agli attori sui diversi oggetti. Le colonne della matrice rappresentano gli attori del sistema mentre le righe rappresentano gli oggetti su cui sono regolamentati gli accessi.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actors** | | |
| **Operation** | Cliente | Utente Non Registrato | Amministratore |
| Login | X |  | X |
| Logout | X |  | X |
| Registrazione |  | X |  |
| Modifica Profilo | X |  |  |
| Visualizza Profilo | X |  |  |
| Cancellazione Profilo | X |  |  |
| Visualizza Carrello | X | X |  |
| Aggiunta Prodotto al Carrello | X | X |  |
| Modifica quantità | X | X |  |
| Cancellazione Prodotto dal Carrello | X | X |  |
| Svuota Carrello | X | X |  |
| Acquista Prodotto | X |  |  |
| Visualizza Dettagli Prodotto | X | X | X |
| Aggiunta Prodotto |  |  | X |
| Modifica Prodotto |  |  | X |
| Elimina Prodotto |  |  | X |
| Visualizza Lista Ordini | X |  |  |
| Visualizza Dettagli Ordine | X |  |  |
| Annulla Ordine | X |  |  |

## 3.6 Global Software Control.

Nel sistema progettato, l'approccio di controllo del flusso adottato è di tipo Procedure-driven.

Le operazioni e le attività sono strutturate all'interno di procedure specifiche, con un'attesa esplicita di input dell'utente quando necessario. Esse vengono eseguite in modo sequenziale, seguendo un flusso predefinito. Ogni passo è dipendente dal completamento del passo precedente.

## 3.7 Boundary conditions

### 3.7.1 Avvio Sistema

Per lo start-up del sistema è necessario avviare il web server ed il database ad esso associato. In seguito, tramite l'interfaccia di Login, sarà possibile autenticarsi tramite le opportune credenziali (username e password) come Amministratore e Cliente.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_BC\_1* | | | *Avvio Sistema* | *Data* | | *23/11/2023* |
| *Vers.* | *0.1* | |
| *Autore* | *Antonio Spatuzzi* | |
| **Descrizione** | | | *Lo UC descrive la funzionalità di avvio del sistema* | | | |
| **Attore Principale** | | | **Amministratore**  È interessato ad avviare il sistema. | | | |
| **Entry Condition** | | | L’amministratore si trova nella Console del sistema | | | |
| **Exit Condition**  On Success | | | Il sistema viene avviato | | | |
| **Flusso di Eventi Principale/Main Scenario** | | | | | | |
| 1 | Amministratore: | L’amministratore preme il pulsante Avvia. | | | | |
| 2 | Sistema: | Il sistema viene avviato rendendo disponibili tutti i servizi tramite server. | | | | |

### 3.7.2 Terminazione Sistema

Per terminare il server è necessario l’intervento dell’amministratore.

Questo dovrà effettuare la procedura di terminazione dopo la quale non sarà possibile per nessun client connettersi al sistema.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_BC\_2* | | | *Terminazione Sistema* | *Data* | | *23/11/2023* |
| *Vers.* | *0.1* | |
| *Autore* | *Vittorio Giordano* | |
| **Descrizione** | | | *Lo UC descrive la funzionalità di terminazione del sistema* | | | |
| **Attore Principale** | | | **Amministratore**  È interessato a terminare il sistema. | | | |
| **Entry Condition** | | | L’amministratore si trova nella Console del sistema | | | |
| **Exit Condition**  On Success | | | Il sistema viene terminato | | | |
| **Flusso di Eventi Principale/Main Scenario** | | | | | | |
| 1 | Amministratore: | L’amministratore preme il pulsante Arresta. | | | | |
| 2 | Sistema: | Il sistema si arresta non rendendo più disponibili i servizi. | | | | |

### 3.7.3 Fallimento Sistema

I servizi del sistema possono non essere reperibili nel caso in cui si verifichino le seguenti condizioni:

* Interruzione dell’alimentazione dell’hardware che mantiene il server;
* Eventuali errori di sistema causati dall’inserimento di nuove funzionalità;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UC\_BC\_3* | | | *Fallimento Sistema* | *Data* | | *23/11/2023* |
| *Vers.* | *0.2* | |
| *Autore* | *Antonio Spatuzzi* | |
| **Descrizione** | | | *Lo UC descrive la situazione di fallimento del sistema.* | | | |
| **Attore Principale** | | | **Amministratore** | | | |
| **Entry Condition** | | | Il sistema è stato avviato con una versione aggiornata contenente almeno un errore. | | | |
| **Exit Condition**  On Success | | | Viene caricata una versione precedente del sistema priva di errori. | | | |
| **Flusso di Eventi Principale/Main Scenario** | | | | | | |
| 1 | Amministratore: | L’amministratore preme il pulsante Arresta. | | | | |
| 2 | Sistema: | Il sistema viene terminato. | | | | |
| 3 | Amministratore: | L’amministratore carica la versione precedente del sistema e preme il pulsante Avvia. | | | | |
| 4 | Sistema: | Il sistema viene avviato. | | | | |

# 4. Subsystem services

|  |  |
| --- | --- |
| **Sottosistema Gestione Account** | |
| **Servizi** | **Descrizione** |
| Registrazione | Permette a un utente non registrato di registrare un account. |
| Login | Permette ad un cliente di accedere al sistema con i propri dati. |
| Logout | Permette ad un cliente loggato di uscire correttamente dal sistema. |
| Modifica Profilo | Permette di modificare le proprie credenziali. |
| Cancellazione account | Permette di cancellare il proprio profilo dal sistema. |
| Visualizza Profilo | Permette di visualizzare i dati del proprio profilo. |

|  |  |
| --- | --- |
| **Sottosistema Gestione Acquisti** | |
| **Servizi** | **Descrizione** |
| Acquisto prodotto | Permette di acquistare uno o più prodotti dal carrello. |
| Cancellazione prodotto dal carrello | Permette di eliminare dal carrello tutte le istanze di un certo prodotto. |
| Aggiunta al carrello | Permette di aggiungere un prodotto al carrello. |
| Visualizzazione carrello | Permette di visualizzare il proprio carrello. |
| Cancellazione carrello | Permette di eliminare dal carrello tutti i prodotti. |
| Visualizzazione prodotto | Permette di visualizzare i dettagli di un prodotto. |
| Svuota carrello | Permette di eliminare tutti i prodotti dal carrello. |
| Modifica quantità | Permette di modificare la quantità di un prodotto all’interno del carrello. |

|  |  |
| --- | --- |
| **Sottosistema Gestione Ordini** | |
| **Servizi** | **Descrizione** |
| Visualizza ordini | Permette di visualizzare l’elenco degli ordini effettuati. |
| Visualizza dettagli ordine | Permette di visualizzare i dettagli di un ordine effettuato. |
| Annulla ordine | Permette di annullare un ordine effettuato. |

|  |  |
| --- | --- |
| **Sottosistema Gestione Prodotto** | |
| **Servizi** | **Descrizione** |
| Visualizza dettagli prodotto | Permette di visualizzare i dettagli di un prodotto. |
| Modifica prodotto | Permette di modificare le caratteristiche del prodotto. |
| Aggiunta prodotto | Permette di aggiungere un prodotto inserendo le opportune informazioni. |
| Elimina prodotto | Permette di eliminare un prodotto non più disponibile. |