



S.D.D

SYSTEM DESIGN DOCUMENT

“Maggico Car & Motorbike Parts”

Raffaele Coscione 0512106006

Vincenzo Tortora 0512102104

Francesco Carotenuto 0512104798

Giovanni Renzulli 0512105730

Top Manager

Nome
Prof. Andrea De Lucia

Partecipanti

Nome	Matricola
Raffaele Coscione	0512106006
Francesco Carotenuto	0512104798
Vincenzo Tortora	0512102104
Giovanni Renzulli	0512105730

History

Data	Versione	Cambiamenti	Autori
11/12/2019	1.0	Aggiunta introduzione	Raffaele Coscione
12/12/2019	1.1	Aggiunta Architettura del sistema proposto	Francesco Carotenuto
18/12/2019	1.2	Formattazione completa del documento	Tutti
20/12/2019	1.3	Correzioni EER	Raffaele Coscione
3/1/2020	1.4	Revisione totale del documento	Francesco Carotenuto

Sommario

1.	Introduzione	4
1.1	Scopo del sistema	4
1.2	Obiettivi di design.....	5
1.2.1	Criteri di performance.....	5
1.2.2	Criteri di affidabilità	6
1.2.3	Criteri di manutenibilità	6
1.2.4	Criteri per l'utente finale.....	6
1.3	Definizioni, acronimi e abbreviazioni	7
1.4	Riferimenti	7
1.5	Panoramica	7
2.	Architettura del sistema proposto.....	9
2.1	Panoramica	9
2.2	Decomposizione in sottosistemi	9
2.3	Mapping Hardware/Software	12
2.4	Gestione dei dati persistenti.....	12
2.4.1	Utente	13
2.4.2	Admin.....	13
2.4.3	Cliente	13
2.4.4	Prodotto	14
2.4.5	Carrello.....	14
2.5	Controllo degli accessi e sicurezza	15
2.5	Controllo Globale del software	16
2.6	Condizioni boundary.....	16

1. Introduzione

1.1 Scopo del sistema

L'obiettivo principale del sistema è quello di fornire una piattaforma e-commerce che dia la possibilità, ad un cliente, di acquistare prodotti inerenti al mondo dei motori e, agli amministratori, strumenti per la gestione della piattaforma.

Relativo ad un account può esserci solo un tipo di utente (Cliente o Admin) e questo riconoscimento, fatto in fase di login, della tipologia d'utente dà la possibilità al sistema di creare un'interfaccia adatta al tipo di account loggato.

L'admin ha la possibilità di aggiungere, rimuovere o aggiornare dei prodotti dal catalogo. Inoltre, ha la possibilità di delegare il lavoro di amministratore anche ad altri account registrabili unicamente da se stesso.

Il cliente ha la possibilità scegliere un prodotto, se disponibile nel catalogo, e metterlo nel carrello. Successivamente, visionando il proprio carrello, ha la possibilità di completare l'acquisto o modificare la lista scelta in precedenza.

Uno dei punti fermi del progetto sarà quella di implementare un'interfaccia di facile comprensione da parte della clientela così da fornire un sistema deduttivo e facile nel suo utilizzo.

1.2 Obiettivi di design

Il sistema dovrà poter essere il più efficiente ed intuitivo possibile. Tale efficienza sarà data da brevi tempi di risposta ad ogni genere di input. Si punterà ad aver una buona manutenibilità attraverso il facile inserimento di nuove funzionalità dovuto ad una strutturazione del software semplice da manipolare. Altro aspetto da considerare sono le rigide politiche che eviteranno che l'utente sbagli ad inserire alcuni input. Parte degli aspetti che il sistema avrà sono:

1.2.1 Criteri di performance

Criteri:	Descrizione:
Tempo di risposta:	Maggico assicura una risposta rapida alle richieste dell'utente, la quale verrà gestita ed elaborata entro 10 secondi. Nel caso di una connessione molto lenta tale aspetto potrebbe essere non garantito, ovviamente i tempi di risposta saranno più rapidi quanto più è veloce la tipologia di connessione utilizzata.
Memoria:	La quantità di memoria che verrà utilizzata dal sito web non può essere stimata precisamente. Il sistema potrebbe essere sottoposto alla memorizzazione di centinaia di acquisti con annesse memorizzazioni dei molteplici carrelli dei potenziali utenti, quindi una buona scalabilità in tal senso dovrà essere garantita.

1.2.2 Criteri di affidabilità

Criteri:	Descrizione:
Robustezza:	Maggico gestirà eventuali input errati senza interrompere il funzionamento dell'intero sistema, l'utente verrà avvertito con eventuali messaggi di errore.
Sicurezza:	L'accesso al sistema è controllato da un sistema di autenticazione che categorizza gli utenti non permettendo l'accesso a parti del sistema non autorizzate, oltre alla presenza di filtri che impediranno accessi non previsti.

1.2.3 Criteri di manutenibilità

Criteri:	Descrizione:
Estendibilità:	Il sistema dovrà essere propenso all'inserimento di nuove funzionalità. Quindi, si adotterà una metodologia di programmazione ben strutturata, dunque, di facile comprensione.
Modificabilità:	L'accesso al sistema è controllato da un sistema di autenticazione che categorizza gli utenti non permettendo l'accesso a parti del sistema non autorizzate, oltre alla presenza di filtri che impediranno accessi non previsti.

1.2.4 Criteri per l'utente finale

Criteri:	Descrizione:
Usabilità:	Maggico sarà di semplice utilizzo, grazie ad una grafica semplice ed efficace che guiderà gli studenti nell'acquisto dei prodotti e gli admin nell'amministrazione della piattaforma.

1.3 Definizioni, acronimi e abbreviazioni

- **Maggico:** Nome del sistema che verrà sviluppato.
- **JDBC:** Java Database Connection.
- **DBMS:** Database Management System, Sistema di gestione del database.
- **MySQL:** è il più diffuso DBMS relazionale open source in circolazione.
- **RAD:** Requirements Analysis Document.
- **SDD:** System Design Document.
- **User-friendly:** aggettivo utilizzato per definire un software di facile utilizzo anche per persone non esperte nell'uso del PC.
- **Admin:** Amministratore del sistema in grado di poter aggiungere e rimuovere prodotti dal catalogo ed effettuare registrazioni di nuovi admin.
- **Cliente:** Astrazione semantica di un account quando visita il sito, mette oggetti nel carrello o acquista prodotti
- **Utente:** Astrazione semantica di un account generico, che identifica in egual modo sia un Admin che un Cliente

1.4 Riferimenti

Il contesto è ripreso dal documento di analisi dei requisiti del progetto “Maggico Car & Motorbike Parts”.

È stato anche usato come riferimento il libro:

Object-Oriented Software Engineering: Using UML, Patterns, and Java, 3rd Edition
Prentice Hall, Upper Saddle River, NJ, September 25, 2009.

1.5 Panoramica

In questa sezione presentiamo una panoramica di ciò che riguarderà il documento. Questa forma di indicizzazione serve per orientare il lettore su come si strutturano le fondamenta dell'architettura software che si sta sviluppando, per una maggiore chiarezza.

Di seguito riportiamo le varie parti del contesto che verrà analizzato:

- **Decomposizione del sistema:**
il sistema viene suddiviso in diversi sottosistemi. Ogni sottosistema è caratterizzato da servizi che offre ad altri sottosistemi. L'insieme dei servizi sarà denominato Interfaccia.
- **Mapping Hardware/Software:**
in questa sezione vengono prese decisioni riguardo le piattaforme hardware scelte per il sistema, la comunicazione tra nodi, e come vengano incapsulati i servizi di un sottosistema.

- **Gestione dei dati persistenti:**
in tale sezione vengono individuati gli oggetti che devono essere resi persistenti e quale software verrà utilizzato per tale scopo.
- **Politiche di accesso e Sicurezza:**
in tale sezione si individueranno grazie ad una tabella le operazioni che ogni attore può effettuare.
- **Controllo del software globale:**
che descrive il modo in cui è implementato il controllo globale del software e come si sincronizzano i sottosistemi.
- **Condizioni Boundary:**
che descriverà oltre l'avvio anche la gestione dei fallimenti, interruzione di corrente o anche a errori di progettazione.

2. Architettura del sistema proposto

2.1 Panoramica

Il tipo di sistema proposto è un'applicazione web, composta da un architettura client/server. Tale sistema deve rispondere alle richieste effettuate da parte degli utenti, in base alla tipologia dell'utente che lo utilizza. Nel caso di un utente Amministratore il sistema dovrà mostrare varie opzioni per la manipolazione del catalogo e degli utenti, mentre per il cliente saranno disponibili opzioni di creazione di un account e di acquisto di prodotti tramite l'apposito catalogo.

Le motivazioni che hanno portato alla decisione di utilizzare un architettura client/server sono le seguenti:

Portabilità: Essendo un'applicazione web, potrà essere usata su una varietà di macchine e sistemi operativi e, quindi, sia da computer fissi che dispositivi mobili.

Performance: Il sistema garantirà reattività nel completamento delle task, anche se tale aspetto è molto influenzato dalla qualità della connessione ad internet che l'utente finale possiede.

Scalabilità: il sistema è in grado di supportare e gestire diverse richieste da parte di molti utenti contemporaneamente collegati all'applicazione.

Affidabilità: le componenti sia client che server garantiscono l'affidabilità anche in presenza di guasti e situazioni impreviste, quindi deve essere possibile effettuare dei backup periodici al database.

Nello sviluppo del software utilizzeremo un'architettura di tipo MVC, tale pattern prevede che il software venga diviso in tre parti ognuna delle quali ha un compito diverso. La view si occuperà di curare l'interazione con l'utente, Il controller riceverà informazioni della View per interfacciarsi al model, mentre il model si occuperà della gestione dei dati e, quindi, dell'iterazione con il database sottostante. L'utilizzo di questo modello comporta numerosi vantaggi, primi tra tutti sono la facilità di manutenzione ed estensione. Infatti, essendo uno standard di programmazione, sarà di facile comprensione a programmatori futuri. I model verranno realizzati utilizzando classi Java appropriate, la parte di view verrà implementata utilizzando HTML e Java in pagine JSP e i control saranno realizzati tramite Servlet.

2.2 Decomposizione in sottosistemi

Per la realizzazione del sistema sarà utilizzata una architettura three-tier. Utilizzeremo una particolare architettura multi-tier in cui la logica dell'applicazione verrà suddivisa in tre parti detti layer: 1. Presentation layer: composto da tutte le interfacce grafiche, in particolare dai boundary object come form che verranno compilati dagli utenti.

2. Application layer: composta dagli oggetti che si occuperanno dell'elaborazione dati e di notificare cambiamenti al presentation layer. Inoltre, questo strato interagirà con il database sottostante tramite lo storage layer.

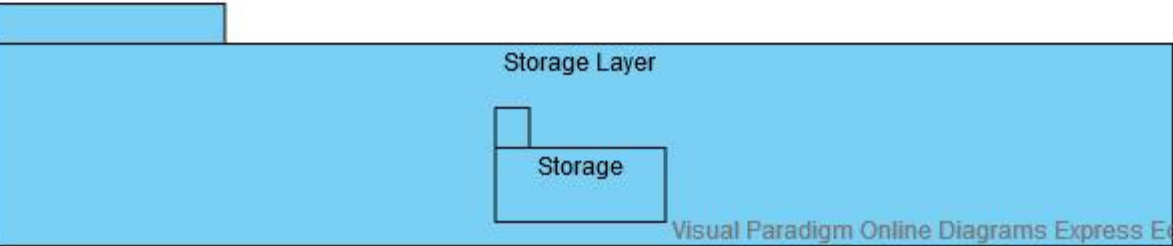
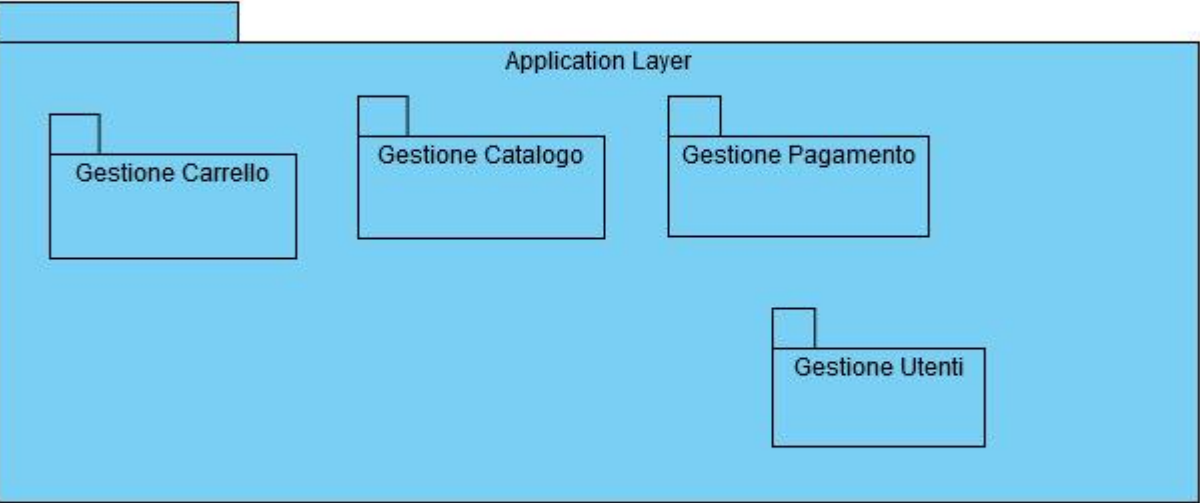
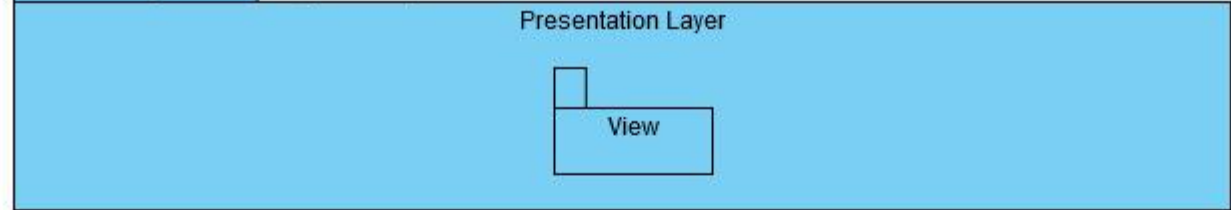
3. Storage layer: Si occupa della memorizzazione di dati persistenti e del loro recupero dal database attraverso l'uso di query.

Il sottosistema del presentation layer è stato suddiviso in quattro sottosistemi.

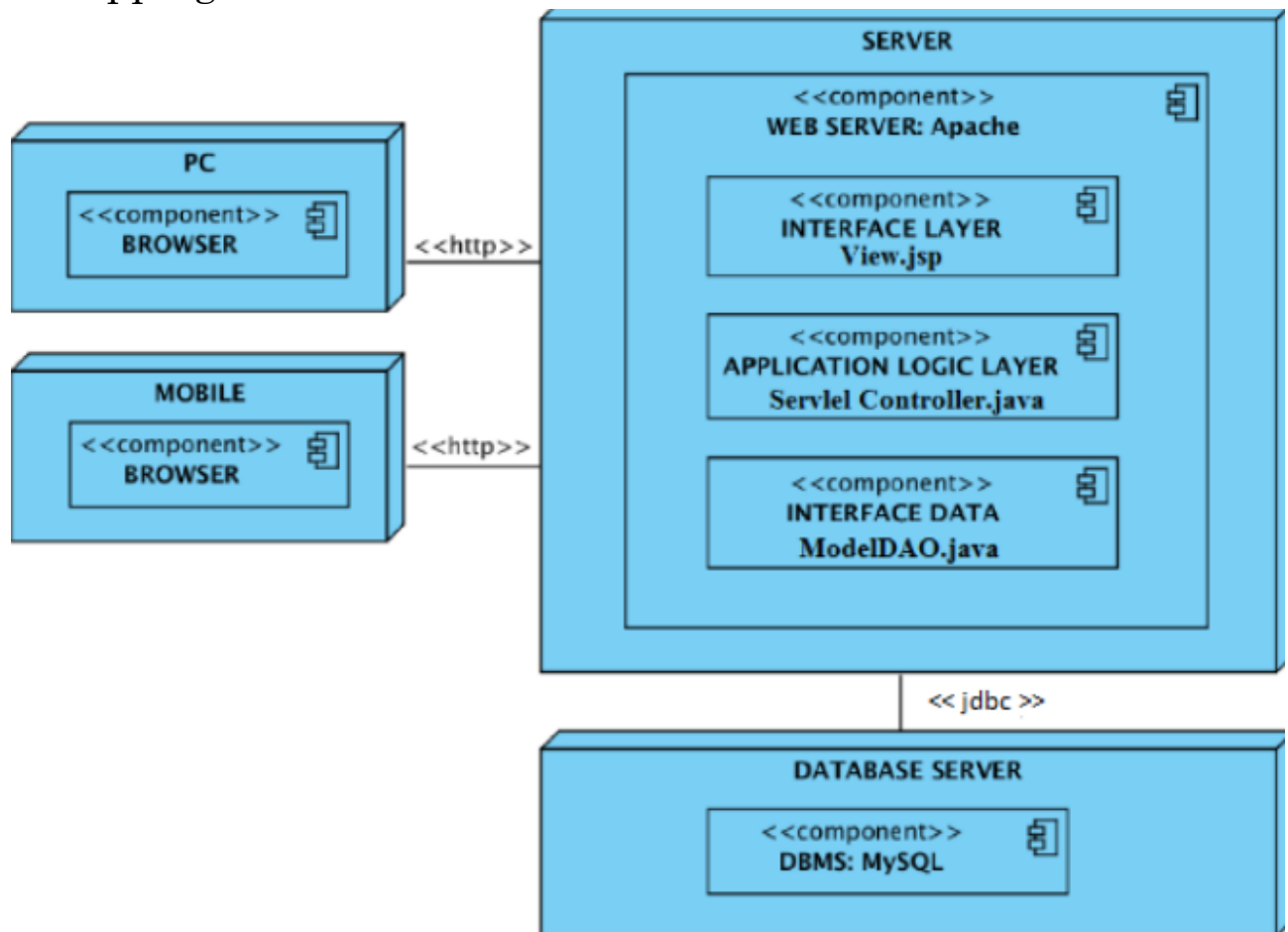
1. Sottosistema Autenticazione: Contiene le pagine di benvenuto.
2. Sottosistema Amministratore: tramite questo sottosistema si potrà accedere alle funzionalità di creazione, aggiornamento e cancellazione dei prodotti; creazione e cancellazione degli admin.
3. Sottosistema Cliente: tale sottosistema si occupa di gestire le funzionalità di modifica del profilo e della gestione del proprio carrello.

Il sottosistema application layer è stato suddiviso in vari sottosistemi

1. Sottosistema Gestione Carrello: comprende tutte le operazioni per la creazione del carrello, per l'aggiunta o la rimozione dei prodotti.
2. Sottosistema Gestione Catalogo: comprende tutte le operazioni per la creazione, aggiunta e rimozione di prodotti dal catalogo
3. Sottosistema Gestione Utente: include tutte le operazioni di creazione di un utente, visualizzazione delle informazioni personali, eliminazione di un utente e la verifica dei dati in fase di Login.
4. Sottosistema Gestione Pagamento: include tutte le operazioni per il completamento della transazione per l'acquisto di un corso (servizio esterno).



2.3 Mapping Hardware/Software

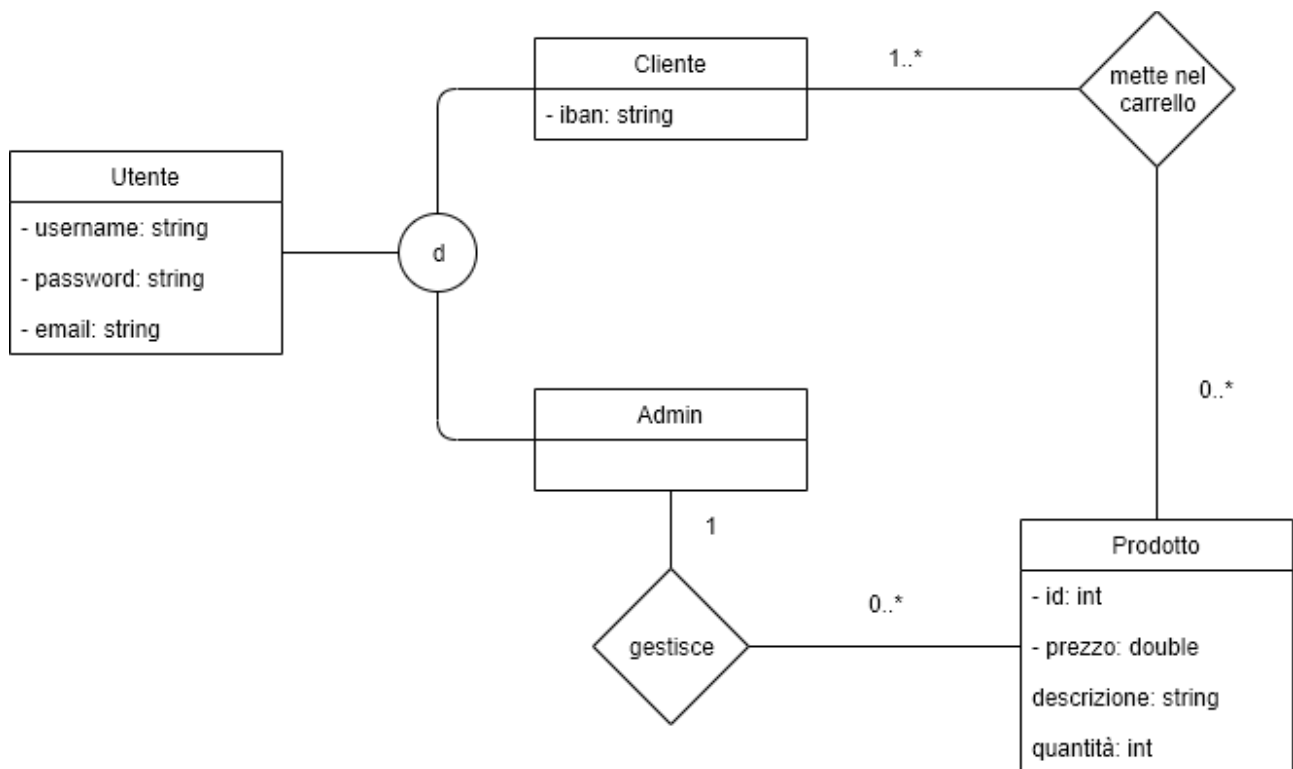


- Web Server: Il server utilizzato è Apache versione 9.
- Interface layer: L'utente utilizza il sistema mediante un Browser installato all'interno del suo calcolatore (ad es. Opera, Firefox, Chrome).
- Application Logic layer: Il sistema, e quindi le funzionalità, sono implementate in linguaggio HTML5 + java.
- Storage layer: Rappresenta il collegamento con il server da parte del sistema e si occupa di tutte le richieste di accesso e modifiche sui dati permanenti presenti nel database.

2.4 Gestione dei dati persistenti

All'interno di questa sezione verrà illustrato il procedimento di design sui dati persistenti partendo dal diagramma delle classi composto dagli entity per poi arrivare alla progettazione di un database relazionale.

Di seguito viene riportato il diagramma EER e le tabelle del database affiliate:



2.4.1 Utente

Nome	Tipo	Vincoli	Key
Username	VARCHAR(45)	NOT NULL	PRIMARY KEY
Password	VARCHAR(45)	NOT NULL	
email	VARCHAR(45)	NOT NULL	

2.4.2 Admin

Nome	Tipo	Vincoli	Key
Username	VARCHAR(45)	NOT NULL	FOREIGN KEY

2.4.3 Cliente

Nome	Tipo	Vincoli	Key
Username	VARCHAR(45)	NOT NULL	FOREIGN KEY
Iban	CHAR(27)	NOT NULL	

2.4.4 Prodotto

Nome	Tipo	Vincoli	Key
Id_prod	Int	NOT NULL	PRIMARY KEY
Prezzo	Double	NOT NULL	
Descrizione	VARCHAR(100)	NOT NULL	
Qt_prodotto	Int	NOT NULL	

2.4.5 Carrello

Nome	Tipo	Vincoli	Key
Username	VARCHAR(45)	NOT NULL	FOREIGN KEY
Id_prod	Int	NOT NULL	FOREIGN KEY

2.5 Controllo degli accessi e sicurezza

All'interno del sistema i vari attori hanno il permesso di eseguire operazioni diverse sui vari sottosistemi, sono escluse le operazioni che non verranno implementate in questa versione del software, quali sistema di gestione delle e-mail e dei pagamenti dei corsi. Per schematizzare meglio il controllo si è deciso di usare la matrice degli accessi riportata di seguito:

Attori/Sottosistemi	Amministratore	Clienti
Utenti	*Login *Logout *Visualizzazione dati personali *Modifica dati personali *Creazione Amministratore	*Login *Logout *Visualizzazione dati personali *Modifica dati personali
Catalogo	*Visualizzare prodotti dal catalogo *Aggiungere prodotti al catalogo *Rimuovere prodotti dal catalogo *Aggiornare prodotti dal catalogo	*Visualizzare prodotti dal catalogo
Carello		*Aggiungere un prodotto al carrello *Rimuovere Un prodotto dal carrello
Pagamento		*Effettuare un pagamento

2.5 Controllo Globale del software

Il flusso di controllo globale è la sequenza di azioni nel sistema. Il sistema ha un flusso guidato di eventi; le funzionalità richiedono un'interazione continua da parte dell'utente; per questo motivo, il controllo del flusso globale che utilizziamo è di tipo procedure-driven. Quindi non abbiamo una sequenza di operazioni prestabilite ma, è l'utente che sceglie l'operazione da eseguire

2.6 Condizioni boundary

Le condizioni limite hanno a che vedere con l'accensione e lo spegnimento del sistema per quanto riguarda il lato Server, mentre dal lato Client si riferiscono agli errori di connessione al server.

ID:	UC_STARTUP				
CASO D'USO:	Startup del Server				
PARTECIPANTI:	Amministratore				
CONDIZIONE DI ENTRATA:	l'amministratore accede al sistema				
CONDIZIONE DI USCITA:	Il server è attivo ed i relativi servizi disponibile				
FLUSSO DI EVENTI:	<table><thead><tr><th>ATTORE</th><th>SISTEMA</th></tr></thead><tbody><tr><td>L'amministratore clicca Sul pulsante "avvia"</td><td>Il sistema si avvia e attiva i Servizi in remoto andando A renderli disponibili per le Richieste. Il Sistema notifica che l'operazione È avvenuta con successo.</td></tr></tbody></table>	ATTORE	SISTEMA	L'amministratore clicca Sul pulsante "avvia"	Il sistema si avvia e attiva i Servizi in remoto andando A renderli disponibili per le Richieste. Il Sistema notifica che l'operazione È avvenuta con successo.
ATTORE	SISTEMA				
L'amministratore clicca Sul pulsante "avvia"	Il sistema si avvia e attiva i Servizi in remoto andando A renderli disponibili per le Richieste. Il Sistema notifica che l'operazione È avvenuta con successo.				
Eccezioni:	Durante la fase di avvio del server, se si verifica un errore, l'amministratore riceverà una notifica di errore.				

ID:	UC_SHUTDOWN				
CASO D'USO:	Shutdown del Server				
PARTECIPANTI:	Amministratore				
CONDIZIONE DI ENTRATA:	l'amministratore accede al sistema				
CONDIZIONE DI USCITA:	Il server è attivo ed i relativi servizi disponibile				
FLUSSO DI EVENTI:	<table> <tr> <th>ATTORE</th><th>SISTEMA</th></tr> <tr> <td>L'amministratore clicca Sul pulsante "spegni"</td><td> <p>Il sistema effettua una scansione per verificare Che non ci siano richieste In attesa, se vi sono, porta A termine tali richieste e, quando non ne ha più in sospeso, notifica all'amministratore l'inizio della procedura di spegnimento che sta eseguendo per poi terminare.</p> </td></tr> </table>	ATTORE	SISTEMA	L'amministratore clicca Sul pulsante "spegni"	<p>Il sistema effettua una scansione per verificare Che non ci siano richieste In attesa, se vi sono, porta A termine tali richieste e, quando non ne ha più in sospeso, notifica all'amministratore l'inizio della procedura di spegnimento che sta eseguendo per poi terminare.</p>
ATTORE	SISTEMA				
L'amministratore clicca Sul pulsante "spegni"	<p>Il sistema effettua una scansione per verificare Che non ci siano richieste In attesa, se vi sono, porta A termine tali richieste e, quando non ne ha più in sospeso, notifica all'amministratore l'inizio della procedura di spegnimento che sta eseguendo per poi terminare.</p>				
Eccezioni:	Durante la fase di avvio del server, se si verifica un errore, l'amministratore riceverà una notifica di errore.				