# System Design Document

3. Architettura del sistema proposto

3.1 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.

3.2 Decomposizione in sottisistemi

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.

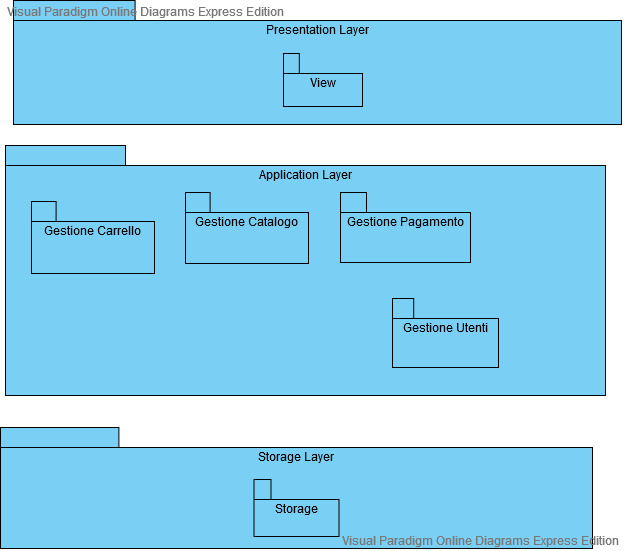
Il sottosistema application layer è stato suddiviso in vari sottosistemi

1.Sottosistema Gestione Carello: comprende tutte le operazioni per la creazione del carrello, per l’aggiunta o la rimozione dei prodotti.

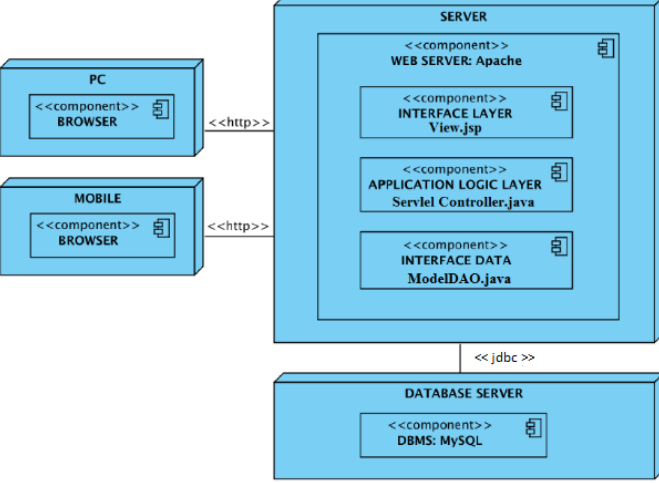
2.Sottosistema Gestione Catalogo: comprende tutte le operazioni per la creazione, aggiunta e rimozione di prodotti dal catalogo

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).



* 1. Mapping Hardware/Software



3.3.1 Web Server: Il server utilizzato è Apache versione 9.

3.3.2 Interface layer: L’utente utilizza il sistema mediante un Browser installato all’interno del suo calcolatore (ad es. Opera, Firefox, Chrome).

3.3.3 Application Logic layer: Il sistema, e quindi le funzionalità, sono implementate in linguaggio HTML5 + java.

3.3.4 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.

3.6 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  \*Rimuover prodotti dal catalogo  \*Aggiornare prodotti dal catalogo | \*Visualizzare prodotti dal catalogo |
| Carello |  | \*Aggiunger un prodotto al carrello  \*Rimuovere Un prodotto dal carello |
| Pagamento |  | \*Effettuare un pagamento |

3.7 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 richiedonoun’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

3.8 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: | 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: | ATTORE SISTEMA  L’amministratore clicca  Sul pulsante “spegni”  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. |
| Eccezioni: | Durante la fase di avvio del server, se si verifica un errore, l’amministratore riceverà una notifica di errore. |