

System Design Document

Indice

Obiettivi di progettazione.....	1
Panoramica.....	3
Decomposizione del sottosistema.....	4
Hardware/Software mapping.....	5
Architettura fisica.....	5
Gestione della persistenza.....	5
Identificazione oggetti persistenti.....	5
Scelta di una strategia di memorizzazione.....	5
Controllo degli accessi e sicurezza.....	6
Controllo del flusso globale.....	6
Servizi forniti dai sottosistemi.....	7

Obiettivi di progettazione

Facile usabilità per l'utente

Il sistema deve fornire un'interfaccia utente semplice ed intuitiva. Inoltre, il sito deve adattarsi ad ogni possibile dimensione della finestra del Web browser.

Bassi tempi di risposta

Il sistema deve rispondere agli input degli utenti in tempi minori di un secondo.

Elevato throughput

Il sistema deve garantire un elevato throughput permettendo l'elaborazione simultanea di più richieste tramite il multi-threading.

Robustezza

Il sistema deve saper gestire i casi in cui l'utente immette input errati.

Alta disponibilità

Il sistema deve essere sempre disponibile.

Sicurezza

Il sistema deve rispettare il protocollo HTTPS garantendo transizioni sicure. Inoltre il sistema dovrebbe far sì che ogni tipologia di utente possa accedere solo alle pagine a lui dedicate.

Adattabilità

Il sistema deve poter essere in grado di adattarsi a differenti tipologie di siti di e-commerce.

Portabilità

Il sistema deve essere accessibile da ogni tipo di dispositivo e Web browser che supporti javascript.

Scalabilità

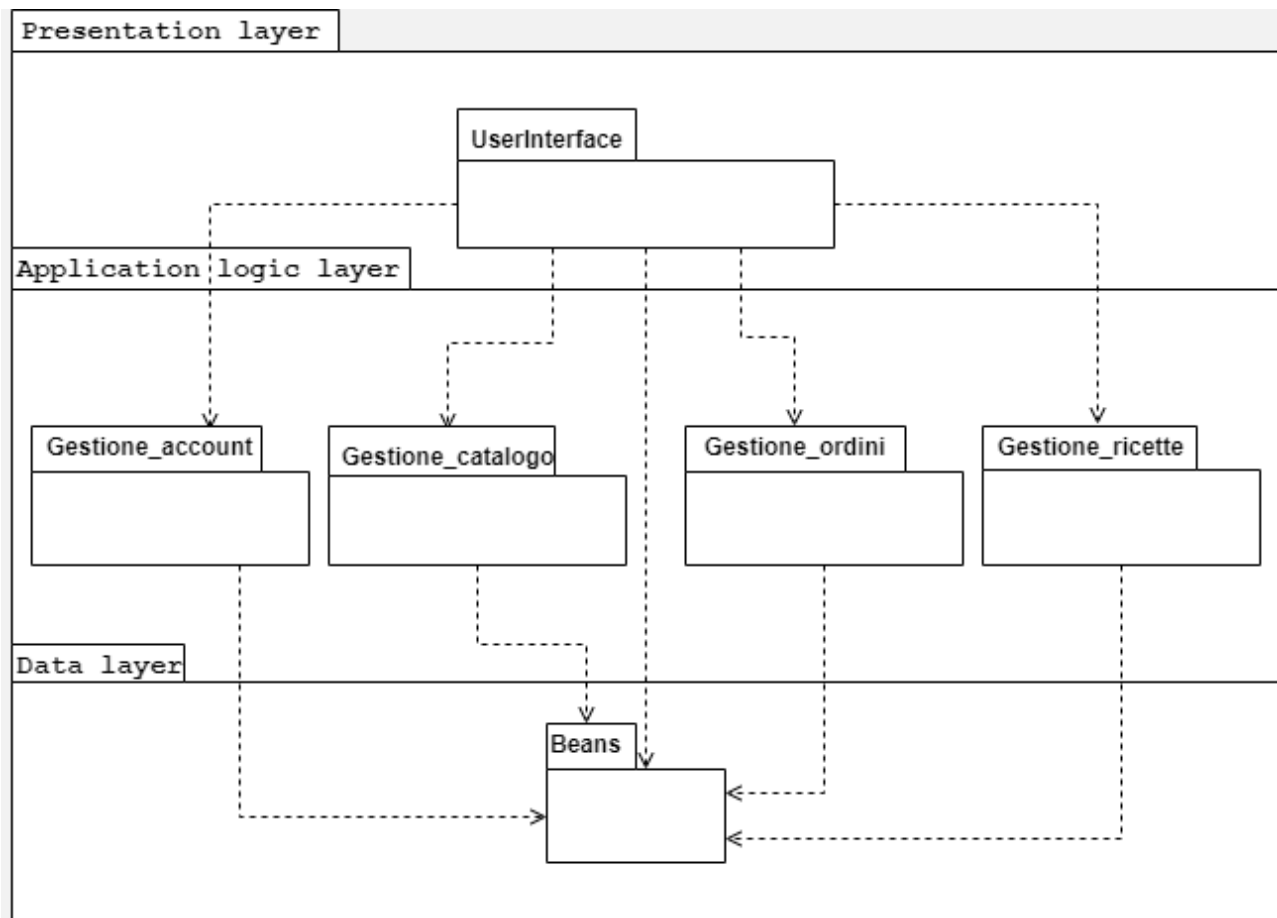
Il sistema deve poter supportare l'evenienza di molteplici accessi da parte degli utenti, garantendo tempi costanti.

Panoramica

Si sceglie di suddividere logicamente il sottosistema in tre livelli:

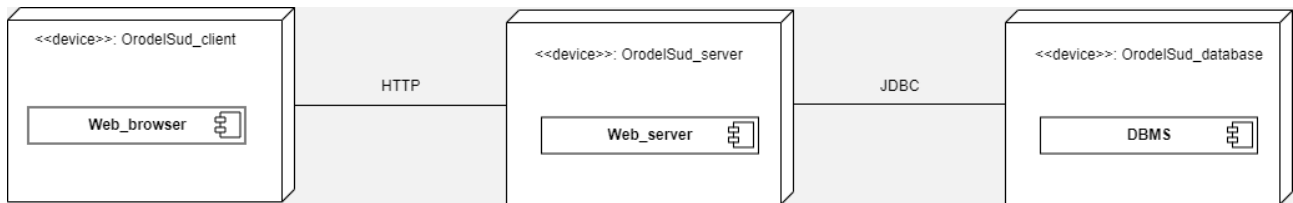
- Presentation layer: esso contiene il modulo "User Interface" che comprende tutti i boundary ed i control object;
- Application logic layer: comprende i moduli "Gestione Account", "Gestione Catalogo", "Gestione Ordini" e "Gestione Ricette". Ciascuno di questi moduli ha il compito di eseguire operazioni CRUD (Create Read Update Delete) relative alle varie entità;
- Data layer: è composto dal modulo "Beans" il quale contiene i vari bean del sistema.

Decomposizione del sottosistema



Hardware/Software mapping

Architettura fisica



Gestione della persistenza

Identificazione oggetti persistenti

Gli oggetti compresi nel sottosistema “Gestione Account”, quali: Cliente, Azienda, Admin, Carta Di Credito, sono persistenti poiché le informazioni legate a queste entità non devono essere volatili in quanto sono necessarie per le funzionalità di login, acquisto, gestione del catalogo e/o del sito. Fa eccezione la classe Visitatore, in quanto non ha informazioni specifiche associate ad essa. Discorso analogo per gli oggetti compresi nel sottosistema “Gestione Ricette” (Recensione, Ricette), nel sottosistema “Gestione Catalogo” (Prodotto) e nel sottosistema “Gestione Acquisto” (Ordine). L’unico oggetto di questi sottosistemi a non essere memorizzato è Carrello, in quanto le informazioni ad esso associate sono legate alla sessione dell’utente.

Scelta di una strategia di memorizzazione

Si sceglie di adottare un database relazionale poiché il sito prevede grandi quantità di dati persistenti e quindi risulta più gestibile il trattamento di questi. Inoltre, dato che il sito prevede accessi multipli contemporanei da parte di diversi client, è fondamentale una gestione della concorrenza la quale è fornita automaticamente dal RDBMS.

Controllo degli accessi e sicurezza

<i>Attori /Oggetti</i>	Gestore Account	Gestore Prodotti	Gestore Carta Di Credito	Carrello	Gestore Recensione	Gestore Ordine
<i>Cliente</i>	View ModifyInfo Login	View	Create Delete	AddProduct DeleteProduct ChangeAmount Purchase	Create	Create View Refuse
<i>Visitatore</i>	Create	View		AddProduct DeleteProduct ChangeAmount	View	
<i>Gestore Prodotti</i>	ModifyInfo View Login	View Modifyinfo Delete			View	
<i>Gestore Ordini</i>	ModifyInfo View Login				View	View
<i>Azienda</i>	View ModifyInfo Login	Create View Modifyinfo Delete			View	View

Il visitatore è un attore che non ha effettuato l'autenticazione al sistema. Tutti gli altri attori devono autenticarsi prima di poter svolgere le loro mansioni. Scegliamo un meccanismo di autenticazione basato su *indirizzo email* e *password*. Nel sito è presente un meccanismo di controllo che non permette l'accesso da parte di utenti non autorizzati a sezioni riservate. Viene istanziato un oggetto di sessione per ciascun utente che effettua il login, che tiene traccia delle sue informazioni.

Controllo del flusso globale

Dato che l'ambiente di esecuzione del sistema lato server è il container Apache Tomcat, quando quest'ultimo riceve una richiesta la processa e la inoltra al Control Object appropriato. Quindi, si è scelto un flusso di controllo decentralizzato basato su eventi che segue il modello MVC. Inoltre il container istanzia un nuovo thread per ciascuna richiesta permettendo così la gestione di richieste concorrenti.

Servizi forniti dai sottosistemi

- **Gestione Account:**
Registrazione, Login, Visualizzazione dati personali, Visualizzazione dati aziendali, Modifica dati personali, Modifica dati aziendali, Aggiunta carta di credito, Rimozione carta di credito.
- **Gestione Ricette:**
Aggiunta recensione, Eliminazione recensione.
- **Gestione Ordini:**
Acquisto, Modifica quantità di prodotto nel carrello, Visualizzazione ordini, Annullamento ordine.
- **Gestione Catalogo:**
Aggiunta prodotto, Eliminazione prodotto, Modifica prodotto, Aggiunta di un prodotto al carrello, Eliminazione di un prodotto dal carrello.