

SYSTEM DESIGN DOCUMENT



NETPHARMA

La tua farmacia online

Studenti

Bene Sabato 05121**06213**

Cozzolino Lidia 05121**08934**

Napoli Riccardo 05121**09152**

Penna Alessandro 05121**06225**

SYSTEM DESIGN DOCUMENT

1.	Introduzione.....	3
2.	Architettura del Software proposto.....	6
3.	Servizi dei sottosistemi.....	10

System design document

1. Introduzione

1.1 Scopo del sistema

L'obiettivo del progetto è quello di realizzare una piattaforma e-commerce dedicata alla vendita di prodotti dedicati alla salute, che consenta quindi ai clienti di poter acquistare i farmaci e gli altri prodotti di cui necessitano direttamente dalla propria abitazione, consentendo anche ai gestori della piattaforma di gestirla facilmente.

La piattaforma deve permettere ai clienti di poter scegliere il prodotto più adatto alle loro esigenze anche tramite la visione di recensioni lasciate da altri clienti che devono però essere moderate al fine di evitarne un utilizzo scorretto.

1.2 Design Goals

Il sistema NetPharma deve avere un alto livello di usabilità garantendo quindi il corretto utilizzo della piattaforma anche da utenti meno esperti possedendo quindi interfacce grafiche intuitive. Inoltre trattandosi di un sistema di e-commerce deve garantire alti standard di affidabilità attuando dei sistemi che prevengono il malfunzionamento del sistema in caso di errori.

Infine devono essere considerati anche requisiti di performance e manutenibilità dovendo la piattaforma essere in grado di gestire correttamente un discreto numero di accessi simultanei e avere la possibilità di poter essere mantenuta osservando regole di buona programmazione. Le classi da considerare sono quindi : Usabilità, Affidabilità, Performance e Manutenzione

1.2.1 – Criteri di usabilità

GRAFICA	<p>NetParma deve rendere le operazioni accessibili a qualsiasi categoria di utente indipendentemente dalle loro abilità informatiche tramite l'utilizzo di una interfaccia grafica minimale e di facile comprensione, nello specifico verranno utilizzati fogli di stile css ed il framework BootStrap, la grafica ha priorità più elevata rispetto a tutto il resto, anche a discapito di performance o manutenibilità.</p> <p>Priorità : Alta</p>
ICONE	<p>Le opportune funzioni dell'interfaccia devono essere accompagnate dalle icone standard ad esse associate come quella del carrello, del log-in e del log-out.</p> <p>Le icone verranno implementate utilizzando BootStrap.</p> <p>Priorità : Alta</p>
EVIDENZIARE GLI ERRORI	<p>Il sistema deve riconoscere ed evidenziare gli errori di compilazione di un form con una apposita colorazione che consenta all'utente di capire la provenienza di un determinato errore, il sistema deve inoltre fornire anche suggerimenti utili a correggerlo utilizzando JavaScript.</p> <p>Priorità : Alta</p>

RESPONSIVENESS	<p>L'interfaccia grafica del sistema deve adattarsi alle grandezze dei principali dispositivi in commercio quali: Computer, Tablet e Smartphone.</p> <p>La responsiveness verrà implementata tramite framework BootStrap.</p> <p>Priorità : Alta</p>
----------------	--

1.2.2 – Criteri di affidabilità

FAULT AVOIDANCE	<p>Il sistema deve essere in grado di scartare eventuali dati errati inseriti, non memorizzarli, non andare in crash a seguito del loro inserimento e di rispondere in maniera adeguata.</p> <p>Questi controlli verranno implementati sempre con Java in modo che sia il server a poterli effettuare, per i form più utilizzati dall'utenza saranno effettuati anche controlli con JavaScript per evitare di sovraccaricare troppo il server quando non necessario. Priorità : Alta</p>
SICUREZZA DEI DATI	<p>Le credenziali di accesso devono essere criptate all'interno del database e di queste non deve essere possibile il recupero ma soltanto la sostituzione o il reset.</p> <p>Questi dati sensibili saranno criptati in MD5 direttamente all'interno del database e manipolati nel codice utilizzando le apposite funzioni di libreria.</p> <p>La sicurezza dei dati ha priorità sulle performance.</p> <p>Priorità : Alta</p>

1.2.3 – Criteri di performance

TEMPO DI RISPOSTA	<p>Il sistema deve essere reattivo per tutte le operazioni che riguardano l'acquisto e avere comunque sempre un tempo di risposta inferiore a 3 secondi indipendentemente dalla quantità di spazio occupata all'interno del database.</p> <p>Verranno quindi ridotti al minimo i cicli e si preferiranno algoritmi con complessità asintotica lineare o inferiore dove possibile.</p> <p>Si provvederà anche ad utilizzare strutture dati opportune per minimizzare gli sprechi di memoria.</p> <p>Priorità : Media</p>
THROUGHPUT	<p>Il sistema deve essere in grado di gestire picchi di carico fino ad un massimo di 300 utenti connessi in simultanea che effettuano operazioni senza mostrare rallentamenti. Garantendo quindi in condizioni di normalità determinati tempi di latenza e una sensazione di fluidità. Verrà quindi utilizzato un ConnectionPool efficiente che sfrutterà la programmazione multithread.</p> <p>Priorità : Media</p>

1.2.4 – Criteri di manutenzione

DESCRIZIONE DELLA LOGICA	Il codice della piattaforma deve avere dei commenti utili alla comprensione delle principali funzionalità. I commenti verranno implementati utilizzando JavaDoc. Priorità : Media
IDENTIFICAZIONE	Le variabili utilizzate all'interno del codice devono avere nomi univoci e facilmente identificabili al fine di identificare facilmente il loro scopo Priorità : Alta

1.3 Acronimi e abbreviazioni

FAULT : Qualsiasi errore o malfunzionamento critico che nel caso non fosse adeguatamente gestito porterebbe all'arresto del sistema.

RESPONSIVENESS : Capacità di una pagina web di adattarsi al tipo di dispositivo in uso. E-

COMMERCE : Negozio online accessibile tramite un computer o qualsiasi altro dispositivo simile.

RAD : Documento di analisi dei requisiti .

THROUGHPUT : In termini generali indica la potenza effettiva di un canale di comunicazione o di un software applicativo.

1.4 Riferimenti

- Testo Object-Oriented Software Engineering Using UML, Patterns, and Java
- RAD (NetPharma)
- Problem statement (NetPharma)

1.5 Panoramica

Il documento si divide in cinque parti:

- Nella prima parte sono introdotti gli obiettivi di design e forniti i riferimenti utilizzati per la creazione
- Nella sezione successiva sono invece illustrate le caratteristiche del sistema corrente.
- La terza parte invece rappresenta il software proposto e vengono descritti i seguenti elementi:
 - **DECOMPOSIZIONE IN SOTTOSISTEMI** : Il sistema viene suddiviso in diversi sottosistemi dove ogni sottosistema è caratterizzato dai servizi che offre agli altri sottosistemi
 - **MAPPING HARDWARE/SOFTWARE** : Si decide l'hardware dove il sistema deve funzionare e si mappano le componenti su di essa
 - **GESTIONE DEI DATI PERSISTENTI** : Descrive i dati memorizzati dal sistema e il database usato per memorizzarli
 - **CONTROLLO DEGLI ACCESSI E SICUREZZA**: Descrive la matrice degli accessi

- CONTROLLO GLOBALE DEL SOFTWARE : Descrive come si sincronizzano i sottosistemi

- La quarta parte è per i servizi dei sottosistemi che descrive in termini di operazioni i servizi forniti da singoli sottosistemi
- L'ultima parte del documento è il glossario che si occupa di illustrare il significato di alcuni termini utilizzati.

2. Architettura del Software proposto

2.1 Panoramica

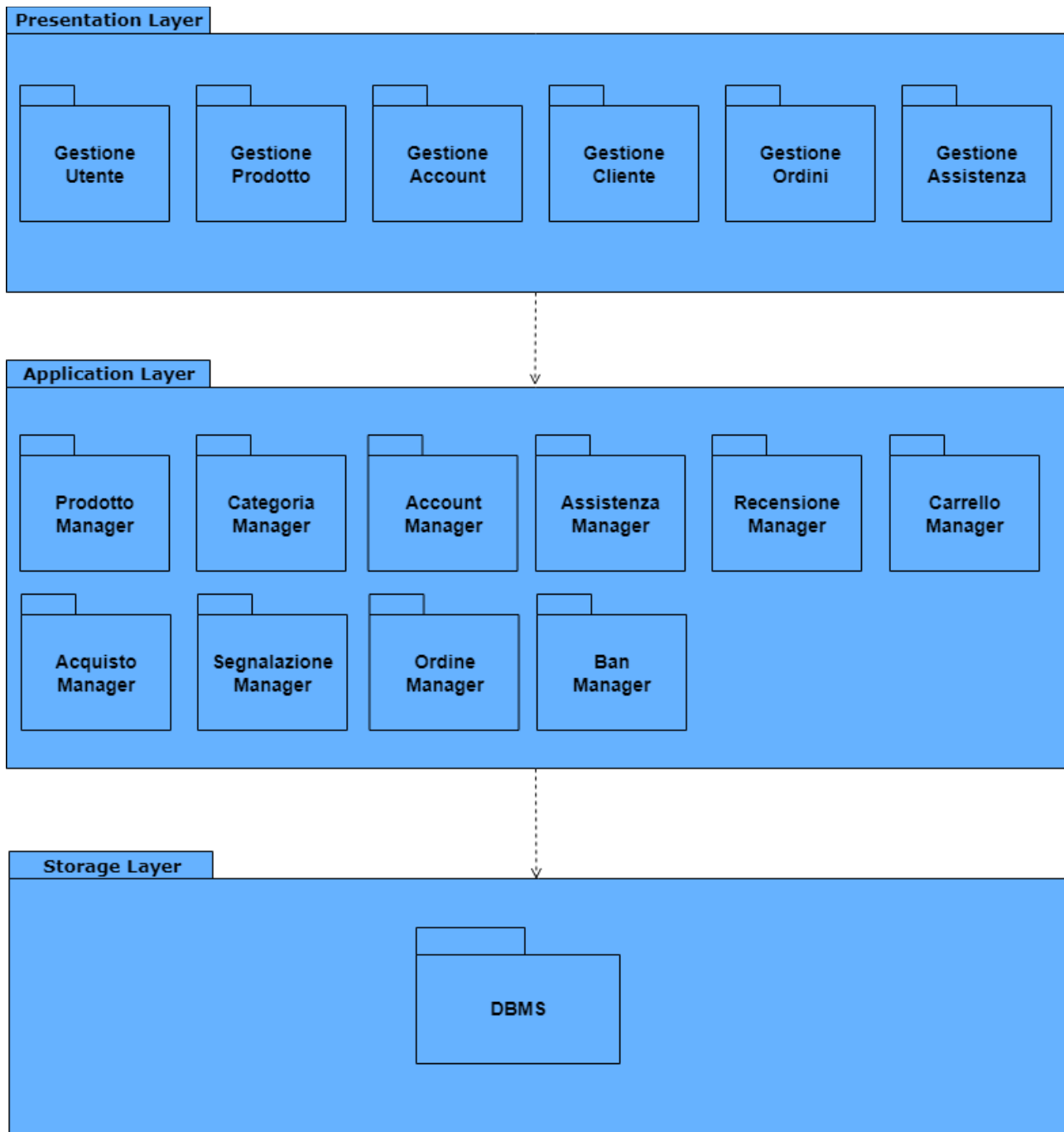
L'architettura del sistema NetPharma è di tipo client/server in cui il client può essere definito come il consumatore del servizio e il server come il fornitore del servizio.

Nello sviluppo del nostro sistema utilizzeremo un'architettura di tipo Model-View-Controller (MVC). L'architettura MVC si basa su tre componenti principali:

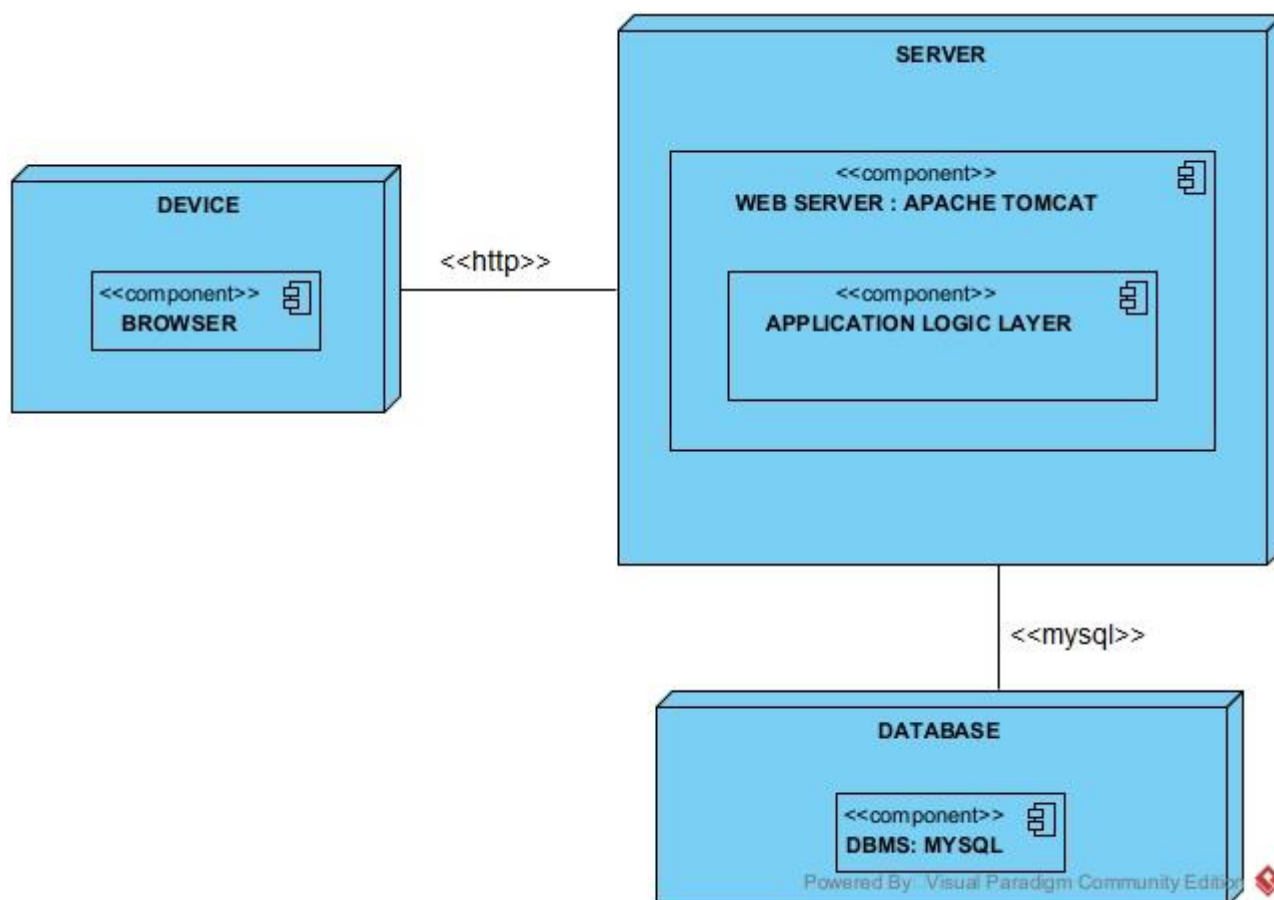
- Model (M): specifica le procedure di estrazione da DB ed elaborazione dei dati.
- View (V): specifica la modalità e la forma di presentazione dei dati all'utente.
- Controller (C): specifica il flusso dell'applicazione e controlla l'interazione tra gli altri livelli.

L'utilizzo del modello MVC comporta numerosi vantaggi, quali, tra i più importanti, la possibilità di suddividere il lavoro nel caso ci debbano lavorare più persone (o gruppi di persone), contemporaneamente, con competenze diverse, agevolazione a un eventuale lavoro di manutenzione dato l'utilizzo di un modello e di regole standard che ne accentuano la comprensione. Per natura dell'architettura MVC si ha basso accoppiamento tra model, view e controller poiché una modifica ad un sottosistema non influenza nessun altro sottosistema. Inoltre, l'architettura MVC favorisce la cosiddetta alta coesione poiché consente il raggruppamento logico di azioni correlate su una stessa componente.

2.2 Decomposizione del sistema



2.3 Mapping Hardware



WEB SERVER :

Il server utilizzato è Apache Tomcat versione 9.X.

INTERFACE LAYER :

L'utente utilizza il sistema dal proprio browser installato all'interno del device.

APPLICATION LOGIC LAYER :

Il sistema e le sue funzionalità sono implementate in linguaggio JavaServlet. La parte grafica verrà riprodotta in HTML e JSP

DATABASE SERVER :

Il DBMS utilizzato è MySQL.

2.4 Gestione dati persistenti

La documentazione in merito è stata inserita in un documento a parte, il cui nome è "Gestione dati persistenti".

2.5 Controllo degli accessi e sicurezza

L'accesso alla piattaforma per tutte le tipologie di utenti comprese le figure amministrative è effettuato tramite l'inserimento delle apposite credenziali (USERNAME,PASSWORD) che verranno richieste allo scadere della sessione in cui si opera.

Objects	Account	Prodotto	Recensione	Ticket	Carrello	Ordine
Actors						
Cliente	modifica_profilo elimina_account log-out	viewProducts addToCart leaveReview searchProduct	submitReview viewSiteReviews viewProductReviews reportReview	openTicket checkStatus removeTicket	check-out removeFrom increaseQuantity decreaseQuantity modifyQuantity	requestOrder
Utente visitatore	sign-up log-in recupera_password	viewProducts searchProduct	viewSiteReviews viewProductReviews		removeFrom increaseQuantity decreaseQuantity modifyQuantity	
Gestore degli utenti	delete_user ban_user unban_user		removeProductReview removeSiteReview			
Gestore degli ordini	Log-out					sendOrders viewOrders
Gestore del catalogo	Log-out	addProduct removeProduct editProduct addCategory removeCategory				
Farmacista	Log-out	viewProducts searchProduct		manageTickets		

2.6 – Controllo del Software Globale

In "NetPharma" Il controllo del software globale viene effettuato dal Web Server, che si occupa di smistare le varie richieste alle Java Servlet appropriate. Le Java Servlet gestiscono la richiesta generando una risposta che viene successivamente inclusa in una pagina JSP per essere visualizzata dall'utente come pagina html.

3. Servizi dei sottosistemi

Gestione Account	
Registrazione	Consente ad un utente visitatore di registrarsi al sistema.
Login	Consente ad un utente registrato di effettuare l'accesso al sistema.
Logout	Consente ad un utente che ha effettuato l'accesso di uscire dal sistema.
Visualizza Profilo	Consente ad un utente registrato di visualizzare le proprie informazioni personali.
Elimina Account	Consente ad un utente registrato di eliminare il proprio account definitivamente.

Gestione Assistenza	
Apri Ticket	Consente ad un cliente di contattare il farmacista di supporto.
Visualizza Tickets	Consente al farmacista di supporto di visualizzare l'elenco dei tickets a cui rispondere.
Rispondi Ticket	Consente al farmacista di rispondere a un ticket.

Gestione Utente	
Visualizza Carrello	Consente ad un utente di visualizzare il proprio carrello.
Aggiungi Prodotto al Carrello	Consente ad un utente di aggiungere un prodotto al carrello.
Rimuovi Prodotto dal Carrello	Consente ad un utente di rimuovere un prodotto dal carrello.
Modifica Quantità Prodotto	Consente ad un utente di modificare la quantità di un prodotto che si vuole acquistare.
Ricerca Prodotto	Consente ad un utente di ricercare un prodotto.
Visualizza Recensioni Sito	Consente ad un utente di visualizzare le recensioni del sito.
Visualizza Prodotti per Categoria	Consente ad un utente di visualizzare l'elenco dei prodotti di una determinata categoria.
Visualizza Prodotto	Consente ad un utente di visualizzare le informazioni di un prodotto.

Gestione Prodotti	
Aggiungi Prodotto	Consente al gestore del catalogo di aggiungere un prodotto sotto una determinata categoria.
Rimuovi Prodotto	Consente al gestore del catalogo di rimuovere un prodotto.
Modifica Prodotto	Consente al gestore del catalogo di modificare i dettagli di un prodotto.
Aggiungi Categoria	Consente al gestore del catalogo di aggiungere una categoria di prodotti.
Rimuovi Categoria	Consente al gestore del catalogo di rimuovere una categoria di prodotti.
Modifica Categoria	Consente al gestore del catalogo di modificare una categoria di prodotti.

Gestione Cliente	
Applica Ban Recensioni	Consente al gestore degli utenti di applicare un ban temporaneo dalle recensioni.
Rimuovi Ban Recensioni	Consente al gestore degli utenti di rimuovere un ban dalle recensioni ad un utente.
Visualizza Checkout	Consente ad un cliente di visualizzare il check-out.
Inserisci Recensione Sito	Consente ad un cliente di inserire una recensione per il sito.
Inserisci Recensione Prodotto	Consente ad un cliente di inserire una recensione per un prodotto.
Segnala Recensione	Consente ad un cliente di segnalare una recensione.
Visualizza Lista Segnalazioni	Consente al gestore degli utenti di visualizzare tutte le segnalazioni ricevute.
Rimuovi Segnalazione	Consente al gestore degli utenti di rimuovere una segnalazione.
Visualizza Segnalazione	Consente al gestore degli utenti di visualizzare una segnalazione.
Acquisto Prodotto	Consente ad un cliente di acquistare un prodotto.
Rimuovi Recensione Sito	Consente al gestore degli utenti di rimuovere una recensione per il sito.
Rimuovi Recensione Prodotto	Consente al gestore degli utenti di rimuovere una recensione per il prodotto.

Controllo Ordini	
Visualizza Ordini da Spedire	Consente al gestore degli ordini di visualizzare gli ordini da spedire.
Spedisci Ordine	Consente al gestore degli ordini di rimuovere un ordine dall'elenco degli ordini da spedire.
Visualizza Storico Ordini	Consente al gestore degli ordini di visualizzare lo storico degli ordini effettuati e da spedire.

Glossario

Log-in: Procedura con la quale, grazie a dei dati di autenticazione, si effettua l'accesso al sistema.

Log-out: Procedura di disconnessione dal sistema.

Form: Spazio composto da campi predefiniti che permette ad un utente di inserire informazioni da inviare ad un server.

Fault: Qualsiasi errore o malfunzionamento critico che nel caso non fosse adeguatamente gestito porterebbe all'arresto del sistema.

Responsiveness: Capacità di una pagina web di adattarsi al tipo di dispositivo in uso.

E-commerce: Negozio online accessibile tramite un computer o qualsiasi altro dispositivo.

RAD: Documento di analisi dei requisiti.

Throughput: Potenza effettiva di un canale di comunicazione o di un software applicativo.

Start-up: Procedura di avvio di un dispositivo o sistema.

Shutdown: Procedura di arresto di un dispositivo o sistema.

MVC: Architettura Model-View-Controller.

Client: Software che usufruisce dei servizi di un server.

Server: Software che offre un servizio specifico richiesto da client.

Modello Client/Server: Architettura di rete nella quale genericamente un computer client o terminale si connette ad un server per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa risorsa hardware/software con altri client, appoggiandosi alla sottostante architettura protocollare.

Model: Specifica le procedure di estrazione dal database e di elaborazione dei dati.

View: Specifica la modalità e la forma di presentazione dei dati all'utente.

Controller: Specifica il flusso dell'applicazione e controlla l'interazione tra gli altri livelli.

Java Servlet: Software scritti in linguaggio Java che operano all'interno di un Web Server che gestisce richieste da client.

JSP: Java Server Pages, tecnologia di Java per lo sviluppo della logica di presentazione di applicazioni web.

HTML: HyperText Markup Language, linguaggio di markup utilizzato per definire la struttura logica di una pagina web.