

Università degli Studi di Salerno

Corso di Ingegneria del Software

System Design Document

Versione 2.0



Paninoteca 80 Fame

Partecipanti:

Nome	Matricola
Gianluca Longobardi	0512103612
Andreana Balbi	0512103732

Revision History

Data	Versione	Descrizione	Autore
24/10/2017	1.0	Prima Stesura del Documento	Longobardi/Balbi
02/11/2017	1.1	Revisione Completa del Documento	Longobardi/Balbi
20/12/2017	2.0	Seconda Revisione	Longobardi/Balbi

Indice

1. Introduzione

- 1.1. Scopo del sistema
- 1.2. Obiettivi di design
 - 1.2.1 Criteri di Performance
 - 1.2.2 Criteri di Affidabilità
 - 1.2.3 Criteri di Costi
 - 1.2.4 Criteri di Manutenzione
 - 1.2.5 Criteri per l'Utente Finale
- 1.3 Definizioni, acronimi e abbreviazioni
- 1.4 Riferimenti
- 1.5 Panoramica

2. Architettura del software proposto

- 3.1. Panoramica
- 3.2. Decomposizione del sistema
- 3.3. Mapping Hardware/Software
- 3.4. Gestione dei dati persistenti
- 3.5. Controllo degli accessi e della sicurezza
- 3.6. Controllo del software globale
- 3.7. Condizioni Boundary
 - 3.7.1 Avvio del sistema
 - 3.7.2 Terminazione del sistema
 - 3.7.3 Fallimento del sistema

3. Servizi dei sottosistemi

- 4.1 Gestione Autenticazione
- 4.2 Gestione Registrazione
- 4.4 Gestione Prodotti
- 4.5 Gestione Utenti
- 4.6 Gestione Prenotazioni

4. Glossario

1.INTRODUZIONE

1.1. *Scopo del sistema*

Al giorno d'oggi il progresso tecnologico è in continua crescita e riguarda qualsiasi ambito. Tutti si avvicinano al mondo di internet e sono anche molte le persone che vogliono imparare ad interagire con un mondo in continuo sviluppo. Per quanto riguarda l'ambito commerciale ci sono persone che hanno già un'attività avviata e vogliono poter offrire ai loro clienti tutti i benefici di uno store online, ma sono molti i problemi a cui vanno in contro. Ci sono anche alcune difficoltà di cui bisogna tener conto e che gli utenti potrebbero incontrare, come ad esempio gli utenti inesperti nell'utilizzo di questi siti. Il sistema è studiato per ottimizzare il lavoro, fidelizzare gli utenti e soprattutto per gestire il magazzino dei prodotti. Lo scopo è quello di cercare di ottimizzare il lavoro e fornire un adeguato servizio in modo tale da ridurre i tempi lavorativi e allo stesso tempo soddisfare velocemente le esigenze richieste.

1.2. *Obiettivi di design*

Il sistema 80_Fame deve essere il più efficiente possibile e facile da utilizzare.

L'efficienza viene garantita in base ai tempi di risposta ad ogni genere di input ma anche in base alla possibilità di garantire una tolleranza degli errori. In più si cercherà di garantire una manutenibilità attraverso l'inserimento di nuove funzionalità e sarà intuitivo tramite l'utilizzo di semplici interfacce.

Per ottenere gli obiettivi finali vanno seguiti dei criteri di progettazione tenendo presente: Performance, Affidabilità, Costi, Manutenzione e Utente Finale.

1.2.1 Criteri di performance

Tempo di Risposta	80_Fame deve essere reattivo per tutte le operazioni più immediate come l'inserimento di un prodotto. Per il caricamento di più prodotti o la sincronizzazione da uno store al database interno, il sistema deve garantire tempi di risposta ragionevoli, ma essendo un sistema web, molto la performance dipenderà dalla qualità della connessione.
Throughput	I picchi di carico devono essere gestiti dal Sistema senza rallentamenti, garantendo la massima fluidità e un ritardo molto basso. Nei momenti di maggiore carico il Sistema deve garantire consistenza in tutte le chiamate al database.
Memoria	Il Sistema utilizza un database per memorizzare tutti i dati. La quantità di dati non influisce sulla performance del Sistema perché vengono utilizzate delle strategie per distribuire il carico e ridurlo.

1.2.2 Criteri di affidabilità

Robustezza	I componenti devono essere affidabili , in caso di guasti o problemi dovuti all'hardware devono essere in grado di mantenere I dati.
Disponibilità	80_Fame deve essere disponibile all'uso ,24h da parte degli utenti grazie a un servizio sempre attivo.
Tolleranza all'errore	Il Sistema deve funzionare anche in condizioni di errore. Ciò è reso possibile grazie al fatto che il Sistema viene modulato e diviso in modo tale che se si verifica un errore in una parte questo non incide sulle altre funzionalità
Sicurezza	Le tecniche per garantire sono basate prevalentemente su un login, la quale permette il riconoscimento dell'utente.Tutti gli accessi avvengono tramite un operazione di controllo, in particolare l'amministratore avrà i massimi permessi poichè ha la facoltà di utilizzare qualsiasi tipo di funzione disponibile.Infine un utente ha la facoltà di utilizzare qualsiasi tipo di funzione che riguarda la propria gestione e organizzazione dei prodotti. Inoltre le password non sono rese chiare in nessun campo di nessuna tabella che tiene traccia di queste informazioni all'interno del database

1.2.3 Criteri di costo

Sviluppo	I costi di sviluppo prevedono l'acquisto del server per la fase di sviluppo.Per quanto riguarda il dbms utilizzeremo mysql che è open source.
Deployment	Non sono previsti costi di deployment

1.2.4 Criteri di manutenzione

Estensibilità	La progettazione sarà introdotta in modo tale da agevolare l'introduzione di nuove funzionalità, utilizzando il linguaggio HTML e I fogli CSS
Modificabilità	Deve essere possibile intervenire sul codice per apportare delle modifiche o correggere eventuali bugs o implementare nuove funzionalità. Il codice deve essere leggibile per rendere più semplice la modifica
Legibilità	Il codice deve essere ben strutturato o con apposite commenti
Tracciabilità dei requisiti	Grazie alla tracciabilità dei requisiti, è possibile effettuare le modifiche necessarie per il funzionamento del Sistema, valutando I costi e I rischi che ne seguono

1.2.5 Criteri per l'utente finale

Usabilità	80_Fame deve essere facile da apprendere ovvero, l'utente deve essere in grado di interagire con il Sistema e utilizzare le funzionalità a disposizione in modo rapido, deve essere flessibile e robusto ovvero, l'utente deve essere in grado di capire quando sta commettendo degli errori per poter intervenire
------------------	--

1.3. Definizioni, acronimi, abbreviazioni

- **80_Fame**: nome del sistema in sviluppo
- **Utente**: attore del sistema che effettua operazioni sul sistema
- **Amministratore**: attore del sistema che si occupa della gestione e della manutenzione
- **Login**: attività di accesso all'account
- **Logout**: attività di uscita dell'account connesso
- **User-Friendly**: è un aggettivo che viene utilizzato per definire un software di facile utilizzo soprattutto per persone che non sono esperte nell'utilizzo dei computer
- **RAD**: Requirements Analysis Document

- **SDD:** System Design Document
- **DBMS:** Database Management System
- **URL:** Uniform Resource Locator, è una sequenza di caratteri che identifica in modo univoco un indirizzo che appartiene a una risorsa di internet
- **SQL:** Structed Query Language, è un linguaggio di database

1.4 *Panoramica*

Il documento è composto da diverse parti, la prima parte riguarda gli obiettivi di design, mentre la seconda parte che verrà successivamente introdotta, riguarda brevemente l'architettura del sistema corrente prima e dopo il re-engineering.

Prima di iniziare ad introdurre la spiegazione dell'architettura è necessario fare un resoconto delle attività di system design che compongono l'architettura software del sistema.

- **Decomposizione del sistema:** Il sistema viene decomposto in sottosistema in modo tale da poter assegnare lo sviluppo di diverse parti di software facili da sviluppare ai vari componenti del team. Un sottosistema è un insieme di classi, associazioni, operazioni, eventi e vincoli che sono in relazione tra di loro, con lo scopo di fornire un servizio. Ogni sottosistema è caratterizzato da servizi che offre ad altri sottosistemi.
- **Mapping Hardware/Software:** descrive in che modo i sottosistemi vengono assegnati all'hardware e alle componenti "off-the-shelf". Elenca tutti i problemi relativi al riuso del software.
- **Gestione dati persistenti:** descrive i dati persistenti memorizzati dal sistema.
- **Politiche di accesso e sicurezza:** descrive il modello utente del sistema, stabilendo in modo preciso le informazioni e le operazioni effettuate da ogni singolo attore tramite le diverse politiche di accesso con le quali si autenticano al sistema.
- **Flusso di controllo globale:** descrive quali operazioni eseguire e in che ordine per poter stabilire e garantire il corretto flusso di controllo del sistema.
- **Condizioni Limite:** descrive lo start-up, lo shutdown e i comportamenti errati del sistema.

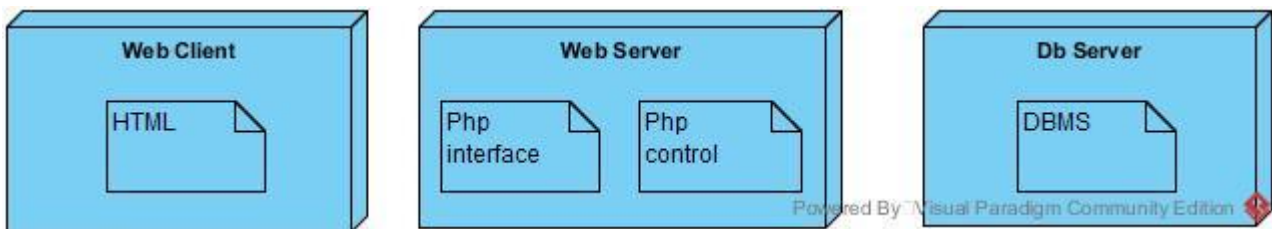
2.ARCHITETTURA DEL SOFTWARE PROPOSTO

3.1 *Panoramica*

Il sistema proposto è un web-based per la gestione di una paninoteca online. Le tipologie di utenti si suddividono in amministratore e utente. L'amministratore è colui che usufruirà di tutte le funzionalità del sistema in quanto è colui che rappresenta l'amministratore del sistema. L'utente potrà gestire solamente i proprio prodotti e i propri acquisti.

L'architettura del sistema si presenta come un'architettura Client/Server per i seguenti motivi:

- **Portabilità:** il sistema potrà essere utilizzato su diverse macchine e diversi sistemi operativi.
- **Trasparenza:** il sistema è in grado di fornire i propri servizi al singolo utente senza interferire con gli altri utenti del sistema.
- **Performance:** il sistema garantirà un elevata reattività per i task degli utenti collegati ma essendo web-based il tutto dipenderà anche dalla qualità della connessione.
- **Scalabilità:** il sistema sarà in gradi di gestire un elevato numero di utenti connessi che effettuano tutte le operazioni contemporaneamente.
- **Flessibilità:** il sistema fornirà un'interfaccia grafica intuitiva e facile da utilizzare, con funzionalità specifiche per il ruolo dell'utente che ne effettua l'accesso.
- **Affidabilità:** i componenti client e server devono essere in grado di mantenere i dati anche in caso di guasto.



3.2 *Decomposizione del Sistema*

Per realizzare il sistema 80_Fame è stato utilizzato lo stile architetturale three-tier in versione Client/server. L'architettura three-tier ("a tre strati") indica una particolare architettura software di tipo multi-tier per l'esecuzione di un'applicazione web che prevede la suddivisione dell'applicazione in tre strati dedicati rispettivamente alla interfaccia utente, alla logica funzionale e alla gestione dei dati persistenti. In particolare i tre strati si occupano di differenti funzionalità del sistema di seguito descritte.

- **Il Presentation Layer** Include tutte le interfacce grafiche e in generale i *boundary objects*, come le form con cui interagisce l'utente. L' interfaccia verso l'utente è rappresentata da un Web server e da eventuali contenuti statici (es. pagine HTML).

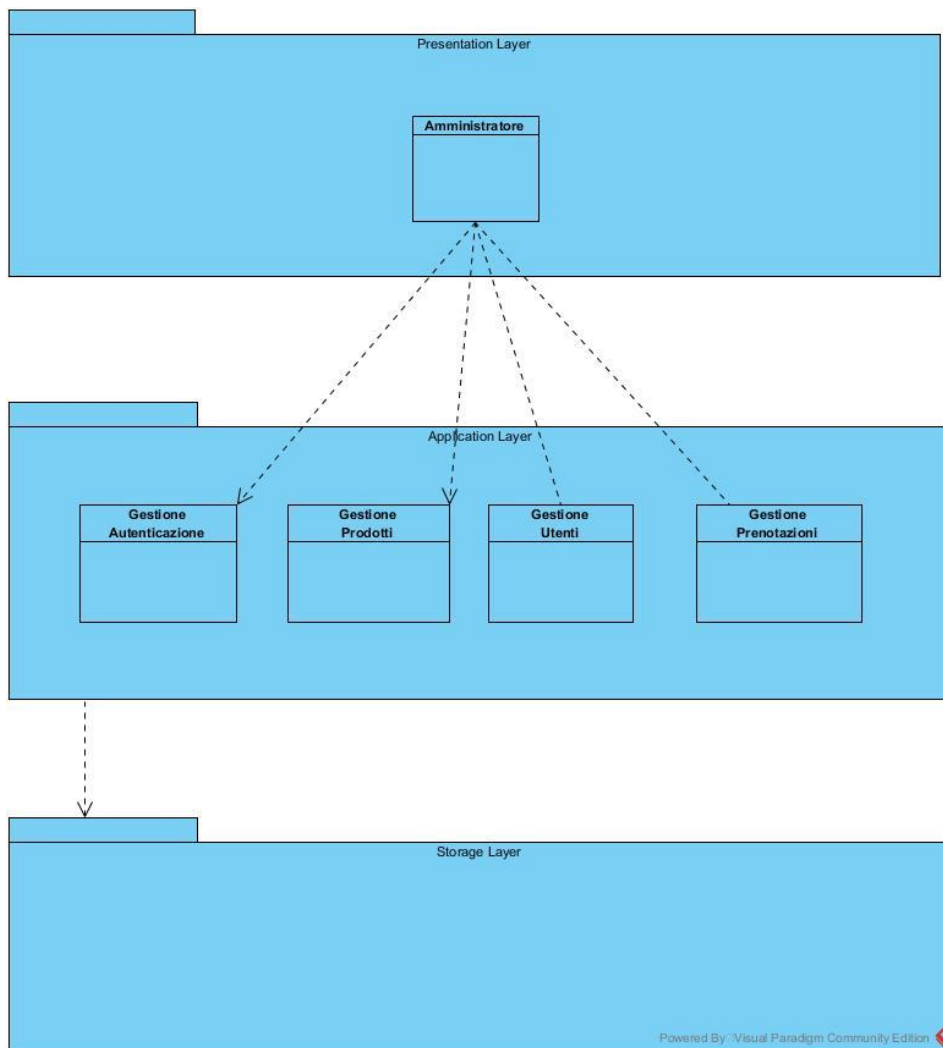
- **L'application Layer** include tutti gli oggetti relativi al controllo e all'elaborazione dei dati. Questo avviene interrogando il database tramite lo storage layer per generare contenuti dinamici e accedere a dati persistenti.
- **Lo storage Layer** effettua la memorizzazione, il recupero e l'interrogazione degli oggetti persistenti. I dati, i quali possono essere acceduti dall'application layer, sono depositati in maniera persistente su un database tramite DBMS.

Descrizione delle gestioni del sistema:

- **Gestione Autenticazione:** Questa funzionalità raccoglie tutte le informazioni per gestire l'autenticazione degli utenti su 80_Fame.
- **Gestione Registrazione:** Questa funzionalità permette di registrarsi fornendo i propri dati.
- **Gestione Prodotti:** Questa funzionalità offre la possibilità di gestire i prodotti nel sistema
- **Gestione Utenti:** Questa funzionalità raccoglie tutte le operazioni necessarie per gestire gli utenti.
- **Gestione Prenotazioni:** Questa funzionalità consente all'utente di effettuare le prenotazioni.

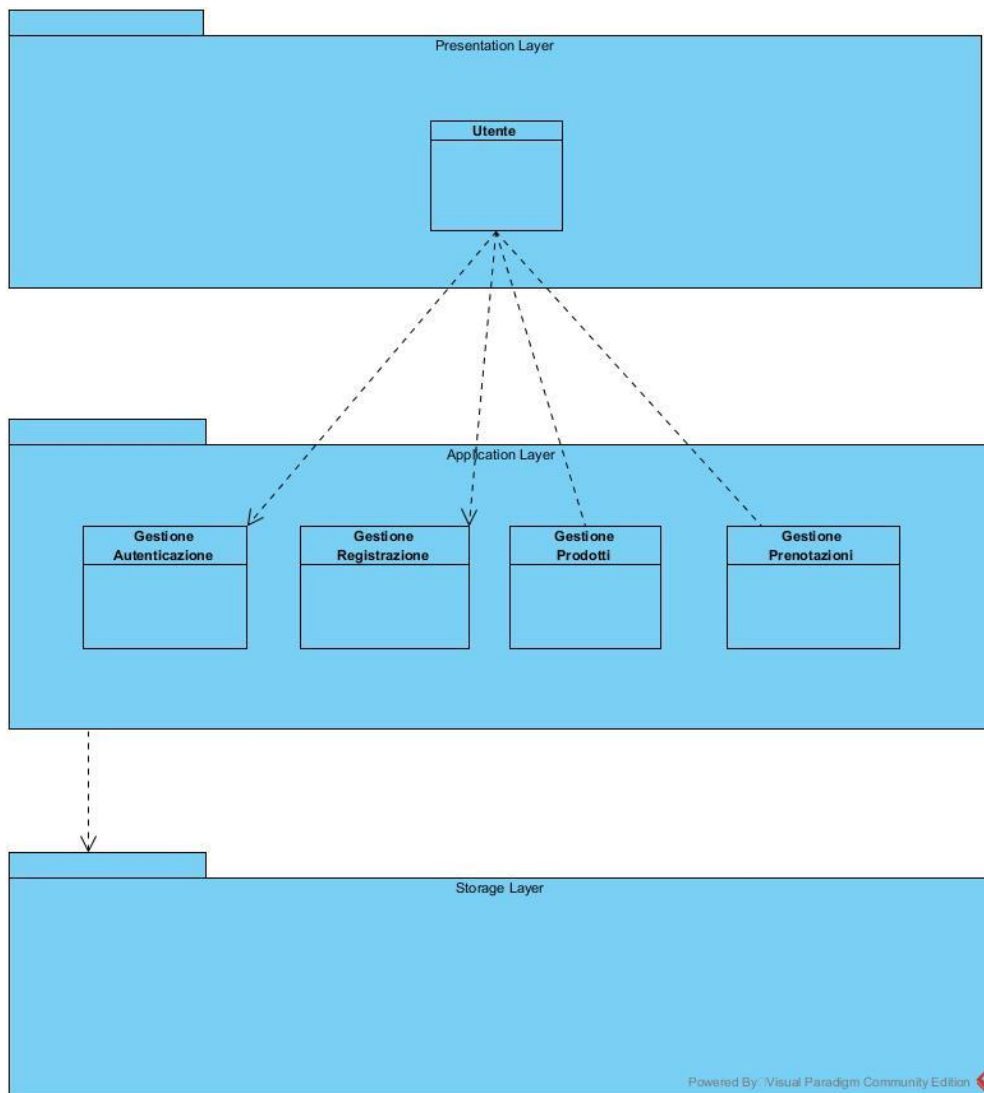
In seguito verrà mostrato nello specifico ogni singolo layer:

L'amministratore



- **Gestione Autenticazione:** L'amministratore può effettuare l'accesso al sistema utilizzando le proprie credenziali e di uscire dal sistema.
- **Gestione Prodotti:** L'amministratore può visualizzare le informazioni di tutti i prodotti presenti sul sistema, può inserire un nuovo prodotto, può rimuovere uno o più prodotti, può modificare le informazioni di un prodotto.
- **Gestione Utenti:** L'amministratore può visualizzare le informazioni di tutti gli utenti presenti sul sistema e può rimuovere uno o più utenti.

L'utente

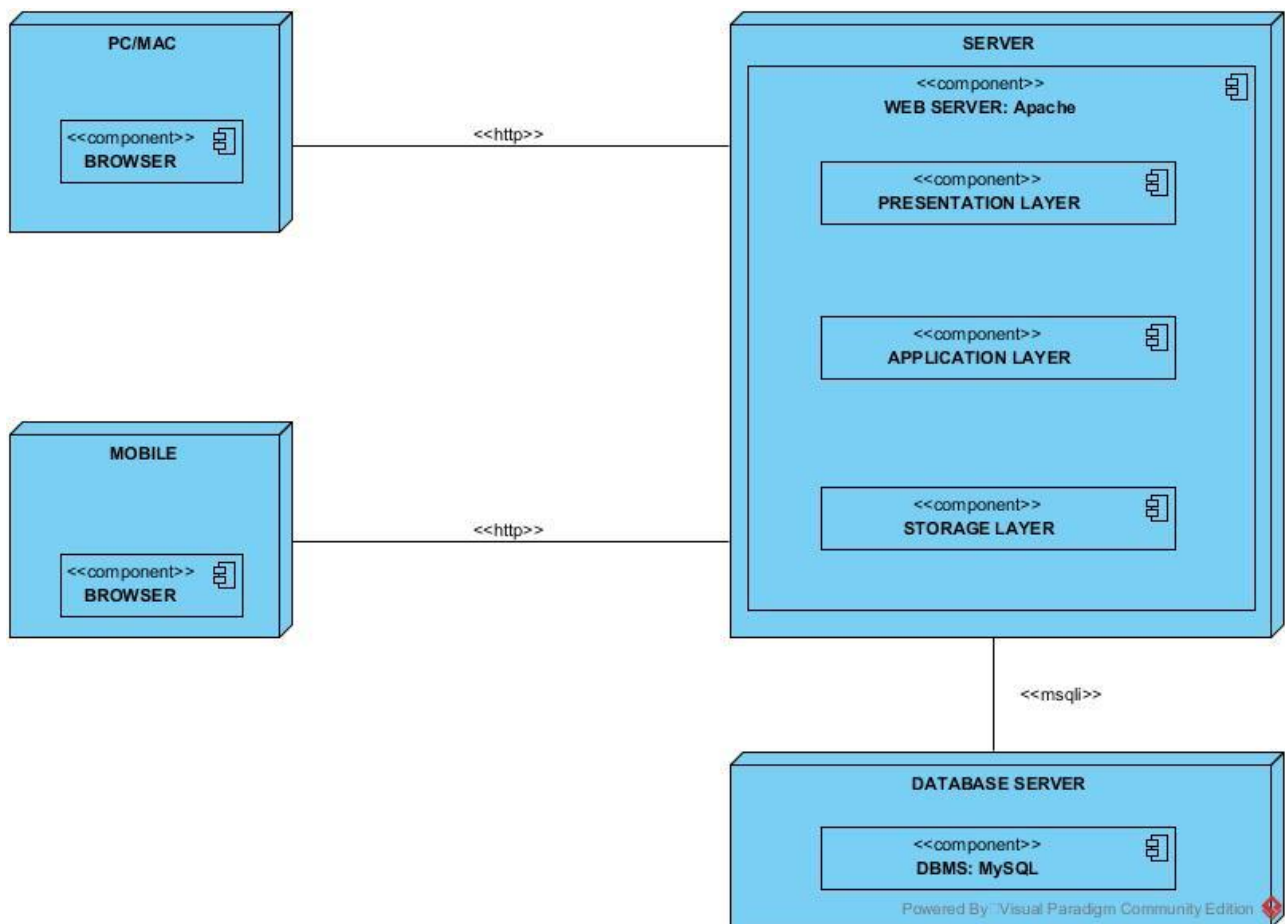


- **Gestione Autenticazione:** L'utente può effettuare l'accesso al sistema utilizzando le proprie credenziali e di uscire dal sistema.
- **Gestione Registrazione:** L'utente può iscriversi al sito in modo da potersi autenticarsi in futuro
- **Gestione Prodotti:** L'utente può visualizzare le informazioni dei prodotti presenti nel sistema inseriti
- **Gestione Prenotazioni:** L'utente può inserire tutte le informazioni necessarie per effettuare le prenotazioni

3.3 Mapping Hardware/Software

La struttura hardware è costituita da un server e dai client cioè un qualsiasi utente può collegarsi al browser per accedere alla Paninoteca_80Fame. I client si collegano al server e al database, il tipo di utente viene determinato in fase di autenticazione controllando nel database il tipo di utente che corrisponde all'Username inserito in quel client. I client dovranno effettuare delle richieste al server per eseguire le operazioni legate alle loro rispettive funzionalità. Il client ed il server saranno connesse tramite una rete che utilizzerà il protocollo TCP/IP.

Di seguito verrà mostrato il diagramma che mostra le funzionalità e i collegamenti hardware:



Presentation layer: l'utente utilizza il sistema mediante un browser installato all'interno del suo computer/mobile (es. Firefox, Chrome).

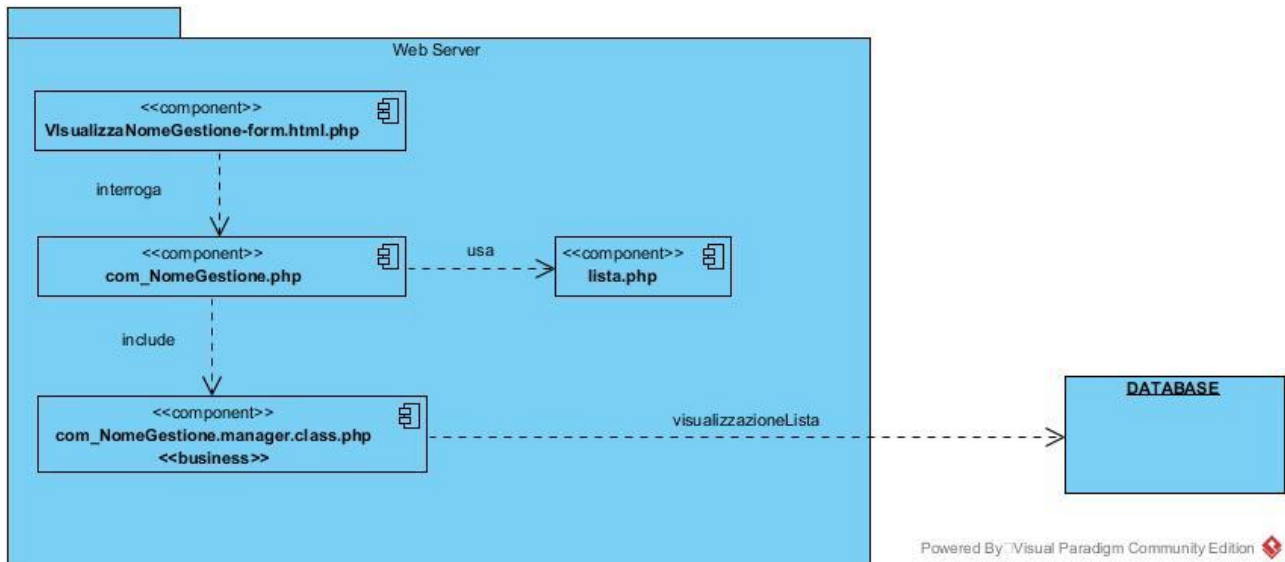
Application layer: il codice è scritto in linguaggio HTML, PHP e il codice risultante viene inviato al browser del client.

Storage layer: rappresenta il collegamento con il server da parte del sistema e si occupa delle richieste di accesso e di modifica sui dati stabili presenti nel database.

Database Server: Il DBMS usato è MySQL che permette l'interazione tra sistema e database.

Diagrammi a Design-Time:

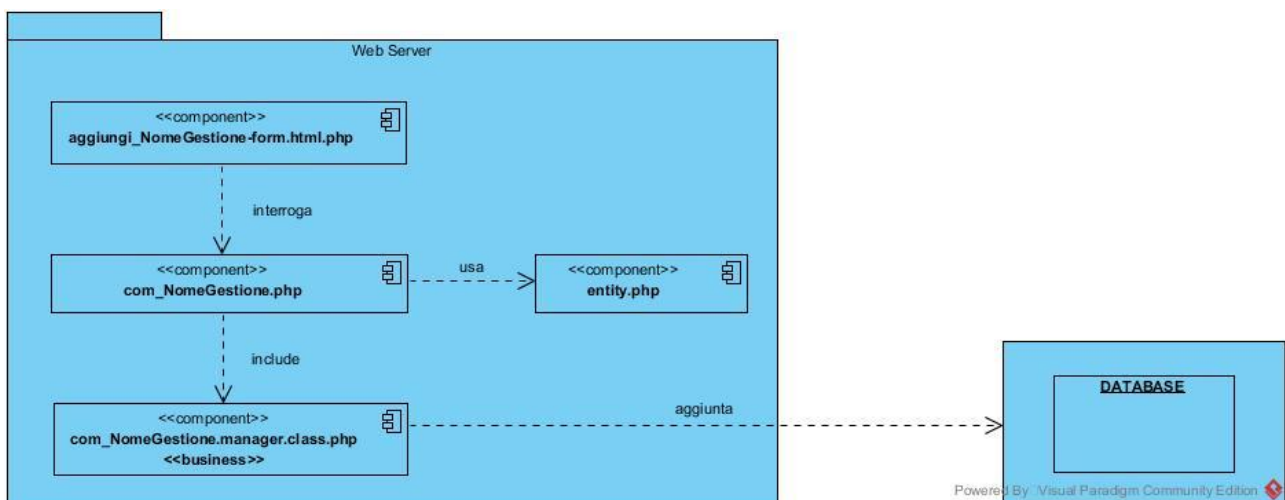
Visualizza



Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole visualizzare una lista di entità pre-esistenti si riferisce ai seguenti sequence diagram :

- SD_2.1: visualizza prodotti
- SD_3.1: visualizza utenti

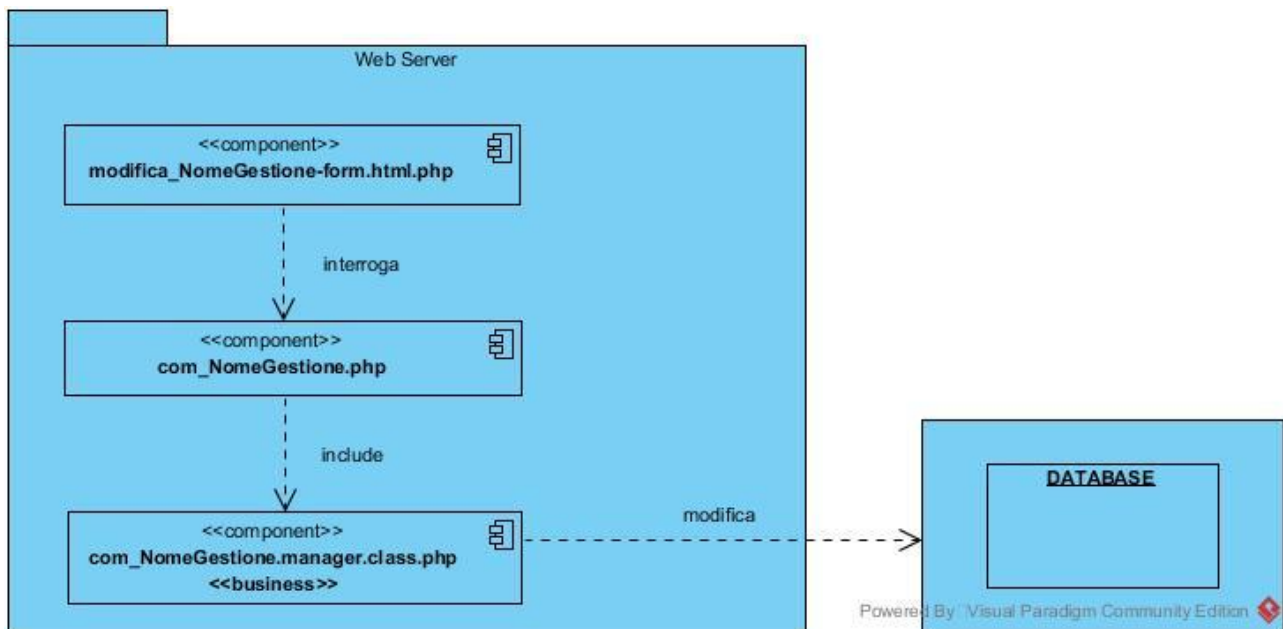
Aggiunta



Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole effettuare un'aggiunta si riferisce al seguente sequence diagram :

- SD_2.2: aggiunta manuale

Modifica



Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole effettuare una modifica su un'entità già esistente si riferisce al seguente sequence diagram:

- SD_2.4: modifica prodotti

Diagrammi a Run-Time:

Al fine della comprensione dei diagrammi che descrivono staticamente il comportamento del sistema a run-time, si definiscono i seguenti concetti:

Client page: si intende la pagina visualizzata dall'utente.

Form: si intende il form incluso nella client page che l'utente utilizza per interagire con il web server.

Controller: classe che si occupa delle operazioni sulle entità del sistema e include il business che si occupa di interagire con il database

View: classe che si occupa di interagire con il web client visualizzando le informazioni elaborate dal controller.

3.4 Gestione dei dati persistenti

Si rimanda al documento: "SDD-GestioneDatiPersistenti+Paninoteca_80Fame" allegato al presente.

3.5 Controllo degli accessi e della sicurezza

Paninoteca_80fame è un sistema che presenta diversi attori che hanno il permesso di eseguire alcune operazioni nel sistema. Abbiamo così suddiviso per tipologia di utente le azioni consentite, in modo da ottenere una visione più compatta e dettagliata grazie ad una matrice degli accessi riportata di seguito:

Oggetti \ Attori	Amministratore	Utente
Autenticazione	<ul style="list-style-type: none">• Login	<ul style="list-style-type: none">• Login
Registrazione		<ul style="list-style-type: none">• Registrazione
Prodotti	<ul style="list-style-type: none">• Visualizza prodotti• Aggiunta manuale• Rimozione prodotto• Modifica prodotto	<ul style="list-style-type: none">• Visualizzi Prodotti
Utenti	<ul style="list-style-type: none">• Visualizza utenti	
Prenotazioni		<ul style="list-style-type: none">• Prenota prodotto

3.6 Controllo del software globale

Il controllo del flusso software viene gestito da classi che interagiscono con il client, il quale ha un web browser, svolgendo le diverse operazioni. Il server smista ogni nuova richiesta alla classe adeguata, inoltrando poi la risposta al client.

3.7 Condizioni Boundary

Le condizioni limite sono l'accensione e lo spegnimento del sistema per quanto riguarda il lato Server. Invece dal lato Client si riferiscono agli errori di connessione al server.

3.7.1 Avvio del sistema

Il sistema dopo essersi avviato presenta un'interfaccia ai client. Una volta effettuata la fase di login ogni utente può accedere alle funzionalità disponibili.

3.7.2 Terminazione del sistema

Dopo aver usufruito di tutte le funzionalità del sistema, l'utente può far terminare la sua permanenza nel sistema eseguendo un'operazione di logout. Inoltre, il sistema può anche terminare in caso di manutenzione del sistema, esecuzioni di operazioni sospette da parte degli utenti ecc...

3.7.3 Fallimento del sistema

Nel caso si verifichi un errore dovuto all'hardware o al software si cercherà di ripristinare il sistema.

I dati poiché sono gestiti dal DBMS non c'è alcuna possibilità che vengano persi. Tuttavia, non è da escludere la perdita dei dati, in quanto si può verificare un guasto al supporto di memorizzazione dei dati. Per minimizzare questo problema, il sistema sarà sottoposto periodicamente a controlli.

Nome Scenario	SC_Startup Server
Istanze di Attori Partecipanti	Andrea: Amministratore
Flusso di Eventi Amministratore	<ol style="list-style-type: none">1. Andrea decide di voler avviare il sistema e quindi clicca sul pulsante "Avvia".2. Il sistema, con le opportune procedure di avvio, attiva i server e i relativi servizi in remoto rendendosi disponibile ad eventuali richieste.3. Il sistema notifica il successo della procedura.

Nome Scenario	SC_Shutdown Server
Istanze di Attori Partecipanti	Andrea: Amministratore
Flusso di Eventi Amministratore	<ol style="list-style-type: none">1. Andrea decide di voler arrestare il sistema e quindi accede alla pagina dedicata e clicca sul pulsante "Arresta".2. Il sistema effettua una scansione per verificare se ci sono ancora client connessi.3. Il sistema termina le eventuali connessioni ancora attive.4. Tramite le opportune procedure di arresto il sistema disattiva i servizi in remoto e il server.5. Il sistema notifica il successo della procedura.

ID	UC_Startup	
Nome Use Case	Startup Server	
Partecipanti	Amministratore	
Condizione d'ingresso	L'amministratore accede al sistema	
Flusso di Eventi	<p>Amministratore</p> <p>L'amministratore accede al Sistema e clicca il pulsante "Avvia"</p>	<p>Sistema</p> <p>Paninoteca 80_Fame attiva I propri servizi rendendosi disponibile in tutte le sue funzionalità e notifica il successo dell'operazione</p>
Condizione d'uscita	Il Sistema è attivo con tutte le sue funzionalità	
Eccezioni		
Requisiti di qualità		

ID	UC_Shutdown	
Nome Use Case	Shutdown Server	
Partecipanti	Amministratore	
Condizione d'ingresso	L'amministratore accede al sistema	
Flusso di Eventi	<p>Amministratore</p> <p>L'amministratore accede al Sistema e clicca il pulsante "Spegni"</p>	<p>Sistema</p> <p>Paninoteca 80_Fame verifica se ci sono client connessi, li disconnette e avvia la procedura d'arresto. Il Sistema notifica il successo dell'operazione.</p>
Condizione d'uscita	Il Sistema viene arrestato correttamente	
Eccezioni	Errore Shutdown	
Requisiti di qualità		

--	--

3. Servizi dei sottosistemi

4.1 Gestione Autenticazione

Sottosistema	Gestione Autenticazione
Descrizione	Gestisce le operazioni di autenticazione di tutti gli attori e le operazioni necessarie al controllo dei dati inseriti.
Servizi offerti	
Login	Questa funzionalità permette di effettuare l'accesso al Sistema, inserendo le proprie credenziali, in modo da poter sfruttare le varie funzionalità del Sistema
Logout	Questa funzionalità permette di uscire dal Sistema
Dati non Trovati	Questa funzionalità permette al Sistema di restituire un errore nel caso in cui non sono presenti dati inerenti durante la fase di login
Errori sui Dati	Questa funzionalità permette al Sistema di restituire un errore nel caso in cui I dati immessi per il login presentano caratteri non consentiti

4.2 Gestione Registrazione

Sottosistema	Gestione Registrazione
Descrizione	Gestisce le operazioni di registrazione di tutti gli attori.
Servizi offerti	
Registrazione utente	Iscrizione al Sistema da parte dell'utente.

4.3 Gestione Prodotti

Sottosistema	Gestione Prodotti
Descrizione	Gestisce tutti i prodotti presenti nel Sistema.
Servizi offerti	
Visualizza Prodotti	Permette di visualizzare i prodotti presenti nel sistema
Aggiunta manuale	Permette l'inserimento di un nuovo prodotto, da parte dell'amministratore, tramite la compilazione di un form con tutte le informazioni del prodotto.
Rimozione prodotto	Permette la rimozione di più prodotti dal Sistema da parte dell'amministratore
Modifica Prodotto	Permette la modifica di un prodotto, da parte dell'amministratore, compilando un form con le informazioni modificate del prodotto.

4.4 Gestione Utenti

Sottosistema	Gestione Utenti
Descrizione	Permette la gestione degli utenti
Servizi offerti	

Visualizza utenti	Permette di visualizzare le informazioni sugli utenti registrati al sistema
-------------------	---

4.5 Gestione Prenotazioni

Sottosistema	Gestione Prenotazioni
Descrizione	Permette la prenotazione dei prodotti presenti nel Sistema
Servizi offerti	
Prenota prodotto	Funzionalità che permette la prenotazione dei prodotti presenti nel Sistema da parte degli utenti.

4.Glossario

Paninoteca_80Fame: nome del sistema che verrà sviluppato;

Utente: il termine identifica un attore del sistema che può usufruire dei servizi offerti

Utente loggato: il termine identifica un utente che ha eseguito il login correttamente;

Utente registrato: il termine identifica utente che ha effettuato la registrazione sul Sistema;

Amministratore: il termine identifica il creatore del sito che ha accesso al codice sorgente e a varie funzionalità;

Prodotti: il termine identifica un oggetto venduto dal sistema;

Aggiunta Manuale: il termine identifica una funzione del sistema che permette all'amministratore di inserire un prodotto non presente, fornendo tutti i parametri manualmente

Modifica Prodotto: il termine identifica una funzionalità del sistema consentita al solo amministratore che gli permette di modificare i prodotti;

Rimozione Prodotto: il termine identifica una funzionalità del sistema consentita al solo amministratore che gli permette di rimuovere i prodotti;

Prenotazione: il termine identifica una funzionalità del sistema che permette agli utenti di poter prenotare qualsiasi prodotto;

Rad: Documento di Analisi dei Requisiti

DBMS: Sistema di gestione di basi di dati

DataBase: Insieme organizzato di dati persistenti