

Università degli Studi di Salerno

Corso di Ingegneria del Software

**Pizzeria “La Saporita”
System Design Document**

Versione 2.1



Partecipanti:

Nome	Matricola
Michela Giovanna Scarpone	0512104490
Raffaella Romano	0512103774

Scritto da:	Scarpone/Romano
-------------	-----------------

Revision History

Data	Versione	Descrizione	Autore
13/11/18	1.0	Prima revisione del documento	Sarpone/Romano
29/12/18	2.0	Seconda revisione del documento	
23/01/19	2.1	Ultima revisione del documento	Sarpone/Romano

Indice

- *1.Introduzione*
 - Scopo del sistema
 - Obiettivi di Design
 - Criteri di performance
 - Criteri di affidabilità
 - Criteri di costo
 - Criteri di manutenzione
 - Criteri per l'utente finale
- *2.Definizioni,acronimini,abbreviazioni*
- *3.Riferimenti*
- *4.Panoramica*
- *5.Architettura Software Proposta*
 - Panoramica
 - Decomposizione del sistema
 - Mapping Hardware/Software
 - Diagrammi a Design-Time
 - Controllo degli accessi e della sicurezza
 - Controllo globale del software
 - Condizioni di boundary
 - Avvio del sistema
 - Terminazione del sistema
 - Fallimento del sistema
 - Servizi dei sottosistemi
 - Gestione Autenticazione
 - Gestione Registrazione
 - Gestione Prodotti
 - Gestione Utenti
 - Gestione Ordini
- *6.Glossario*

1.Introduzione

1.1 Scopo del sistema

Oggigiorno il progresso tecnologico è in continua crescita e riguarda qualsiasi ambito. Per quanto riguarda l'ambito commerciale, diversi sono negozianti che vogliono poter offrire ai loro clienti tutti i benefici di uno store online, ma sono molti i problemi a cui vanno in contro. Una delle maggiori difficoltà è l'utilizzo dello store da parte di persone poco esperte nell'utilizzo dei siti online. Pizzeria LaSaporita ha l'obiettivo di semplificare l'esperienza di acquisto in pizzeria, riducendo i tempi di attesa e ottimizzando l'organizzazione delle prenotazioni. In tal modo si riducono le attese per la consegna dei prodotti e il sovraccarico di richieste, schedulando in miglior modo i tempi lavorativi.

1.2 Obiettivi di Design

Il sistema Pizzeria_LaSaporita deve essere il più efficiente e intuitivo possibile . 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	Pizzeria_LaSaporita deve essere reattivo per tutte le operazioni più immediate come l'inserimento di un prodotto. Per la sincronizzazione dallo store al database interno, il sistema deve garantire tempi di risposta ragionevoli, ma essendo un sistema web, 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 latenza bassa. 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 si utilizzano delle strategie per favorire le prestazioni, 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	Pizzeria_LaSaporita deve essere disponibile all'uso durante le ore lavorative
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 poiché 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

1.2.2 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	Pizzeria_LaSaporita deve essere disponibile all'uso durante le ore lavorative
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 poiché 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'affitto 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
Leggibilità	Il codice deve essere ben strutturato o con appositi 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	Pizzeria_LaSaporita deve essere facile da apprendere, quindi l'utente deve essere in grado di interagire con il Sistema e utilizzare le funzionalità a disposizione in modo rapido; flessibile e robusto ovvero l'utente deve essere in grado di capire quando sta commettendo degli errori per poter intervenire
------------------	---

2. Definizioni, acronimi, abbreviazioni

- **Pizzeria_LaSaporita:** nome del sistema in sviluppo
- **Utente:** attore del sistema che effettua operazioni sul sistema
- **Admin:** attore del sistema che si occupa della gestione e della manutenzione del sistema.
- **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
- **ULR:** 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

3. Riferimenti

Documento RAD del progetto pizzeria_LaSaporita

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-engegnering.

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.

5..ARCHITETTURA DEL SOFTWARE PROPOSTO

5.1 Panoramica

Il sistema proposto è un web-based per la gestione di una pizzeria al taglio online. Le tipologie di utenti si suddividono in amministratore, responsabile e utente. L'amministratore è colui che usufruirà di tutte le funzionalità del sistema e gestirà eventuali problemi che vi si possono presentare. Il responsabile si occupa della registrazione degli utenti. L'utente potrà gestire solamente i propri prodotti e i propri acquisti.

L'architettura del sistema si presenta come un'architettura Client/Server. Il server invia le richieste da parte del Client e cercherà di rispondere in un tempo ristretto.

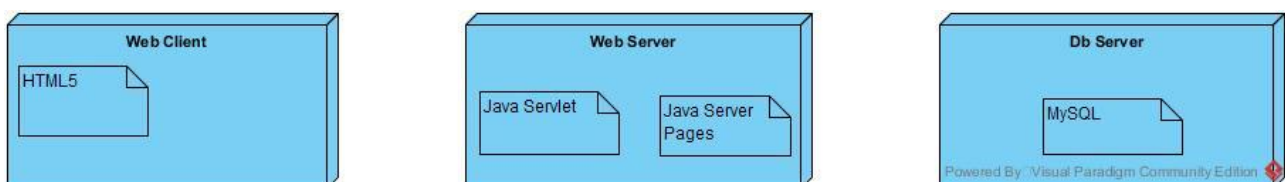
I motivi di questa scelta sono:

- **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 grado 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.

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 mandarli al model o viceversa, mentre il model si occuperà della gestione dei dati e quindi dell'interazione 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 in pagine jsp e i control saranno realizzati tramite servlet.



5.2 Decomposizione del Sistema

Per realizzare il sistema Pizzeria_LaSaporita 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.

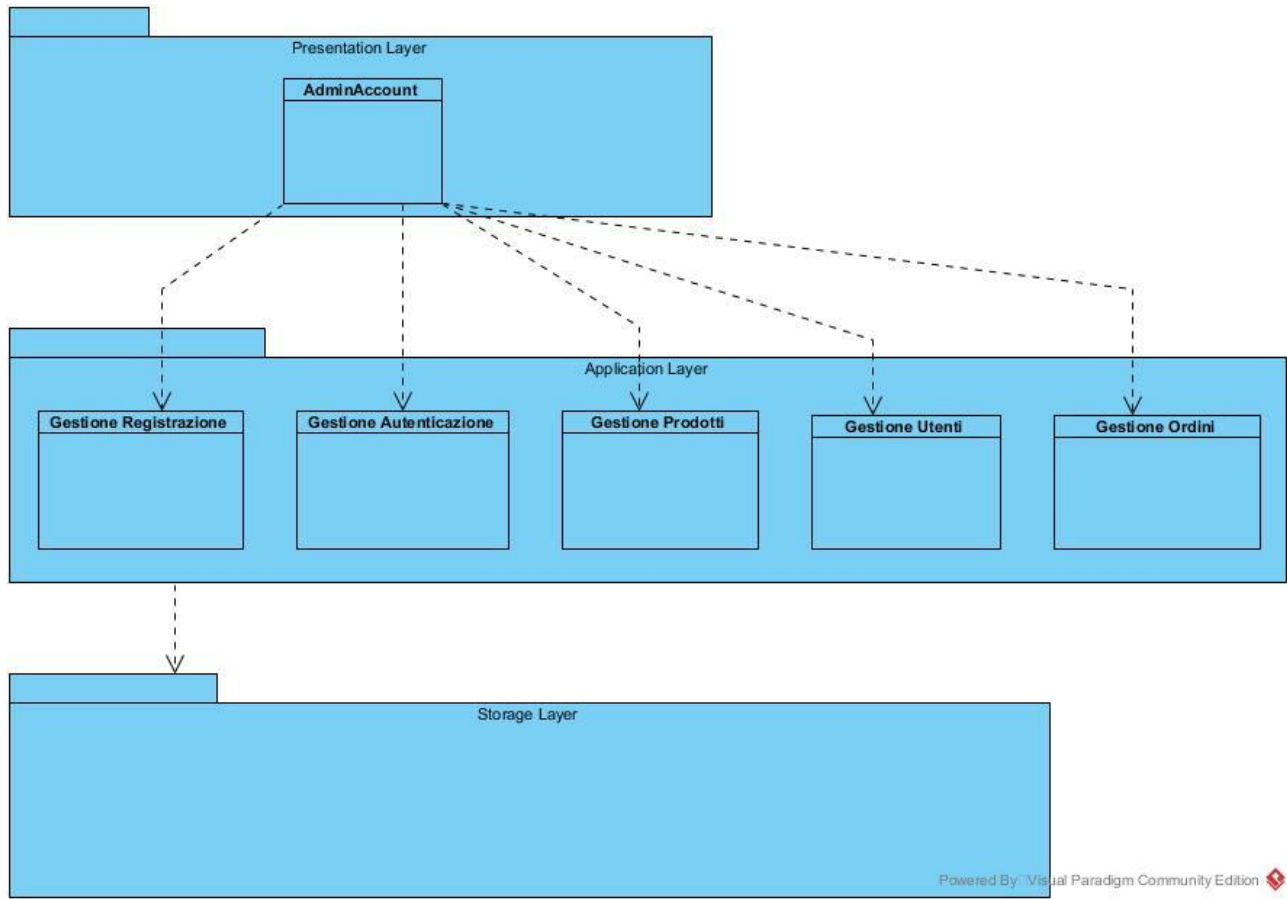
Descrizione delle gestioni del sistema:

Il sottosistema application layer è stato suddiviso in vari sottosistemi:

- **Gestione Autenticazione:** Questa funzionalità raccoglie tutte le informazioni per gestire l'autenticazione degli utenti su LaSaporita
- **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 Ordini:** Questa funzionalità consente all'utente di effettuare le prenotazioni.

Il sottosistema presentation layer è stato suddiviso in due sottosistemi:

AdminAccount



Gestione Autenticazione: L'admin può effettuare l'accesso al sistema utilizzando le proprie credenziali e di uscire dal sistema.

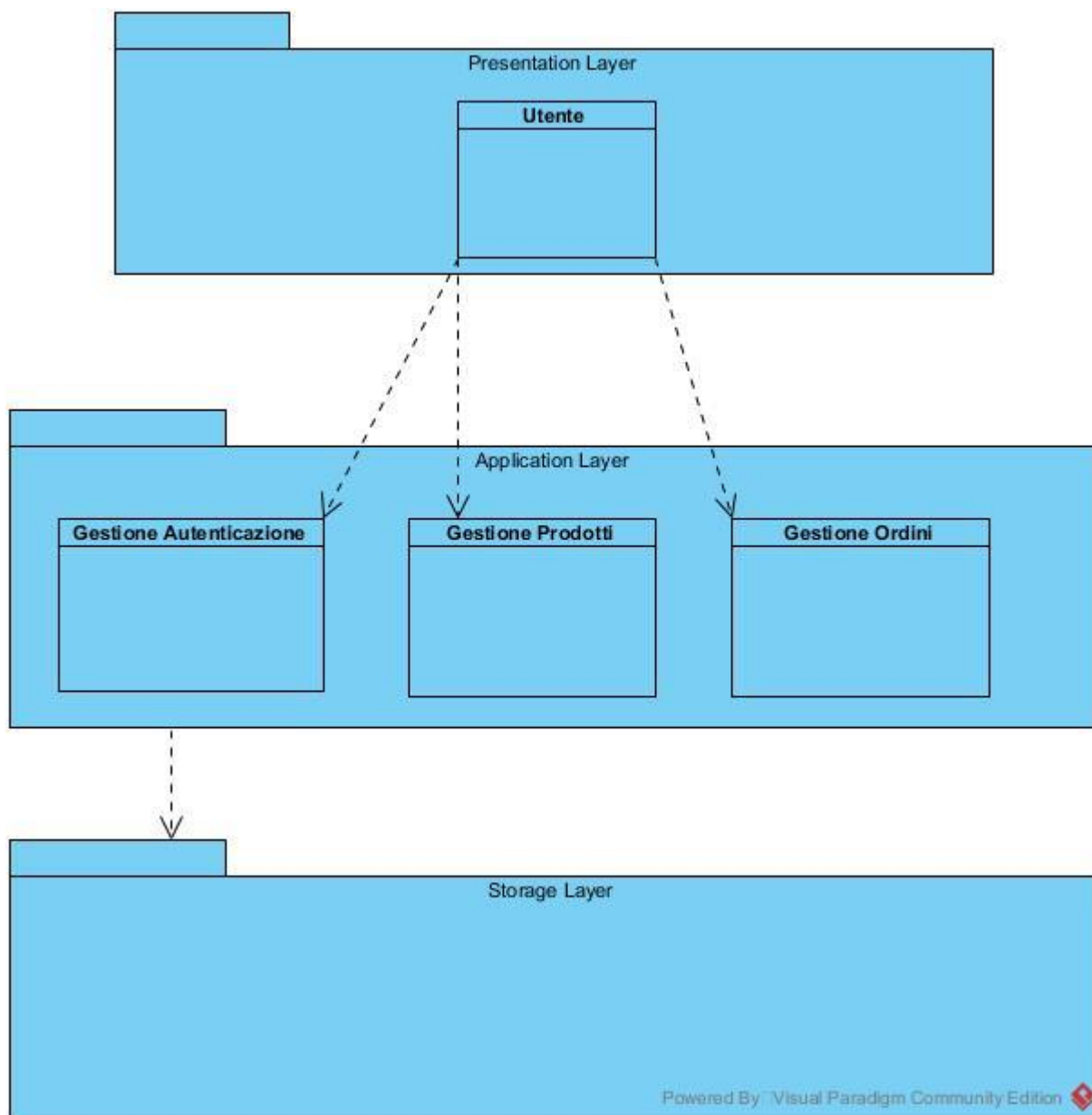
Gestione Prodotti: L'admin può visualizzare le informazioni di tutti i prodotti presenti sul sistema, può inserire un nuovo prodotto o rimuovere uno già esistente.

Gestione Utenti: L'admin può visualizzare le informazioni di tutti gli utenti presenti sul sistema e può rimuovere uno o più utenti.

Gestione Ordini: L'admin può visualizzare le informazioni degli ordini effettuati da ogni singolo utente.

Gestione Registrazione: L'admin iscrive nuovi utenti al sito in modo da permettere l'autenticazione in futuro a quest'ultimo.

L'utente



Gestione Autenticazione: L'utente può effettuare l'accesso al sistema utilizzando le proprie credenziali e di uscire dal sistema.

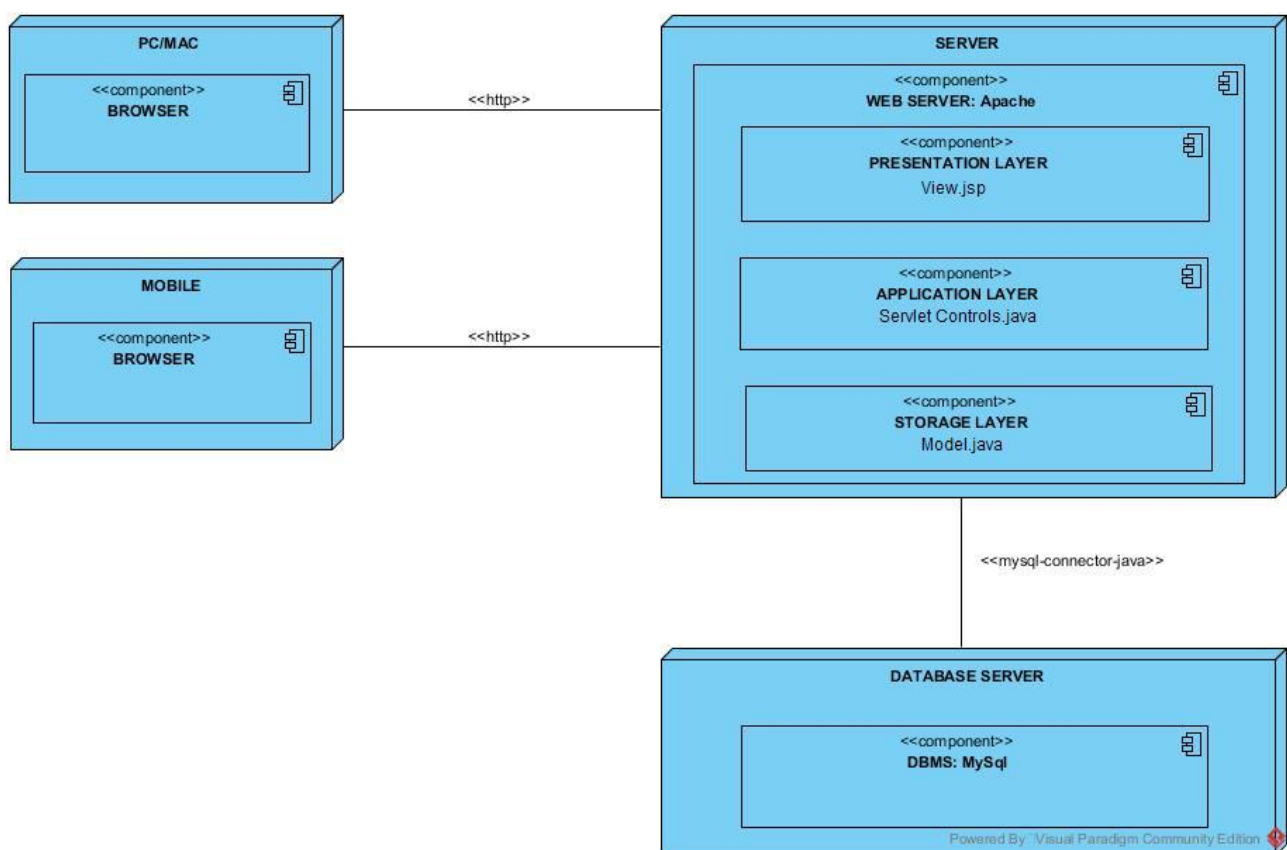
Gestione Prodotti: L'utente può cercare e visualizzare un determinato prodotto o una lista di prodotti.

Gestione Ordini: L'utente può inserire tutte le informazioni necessarie per effettuare gli ordini.

5.3 Mapping Hardware/Software

La struttura hardware è costituita da un server e dai client cioè un qualsiasi utente può collegarsi al browser per accedere a Pizzeria_LaSaporita. 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:



Web Server

Il server utilizzato è Apache.

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

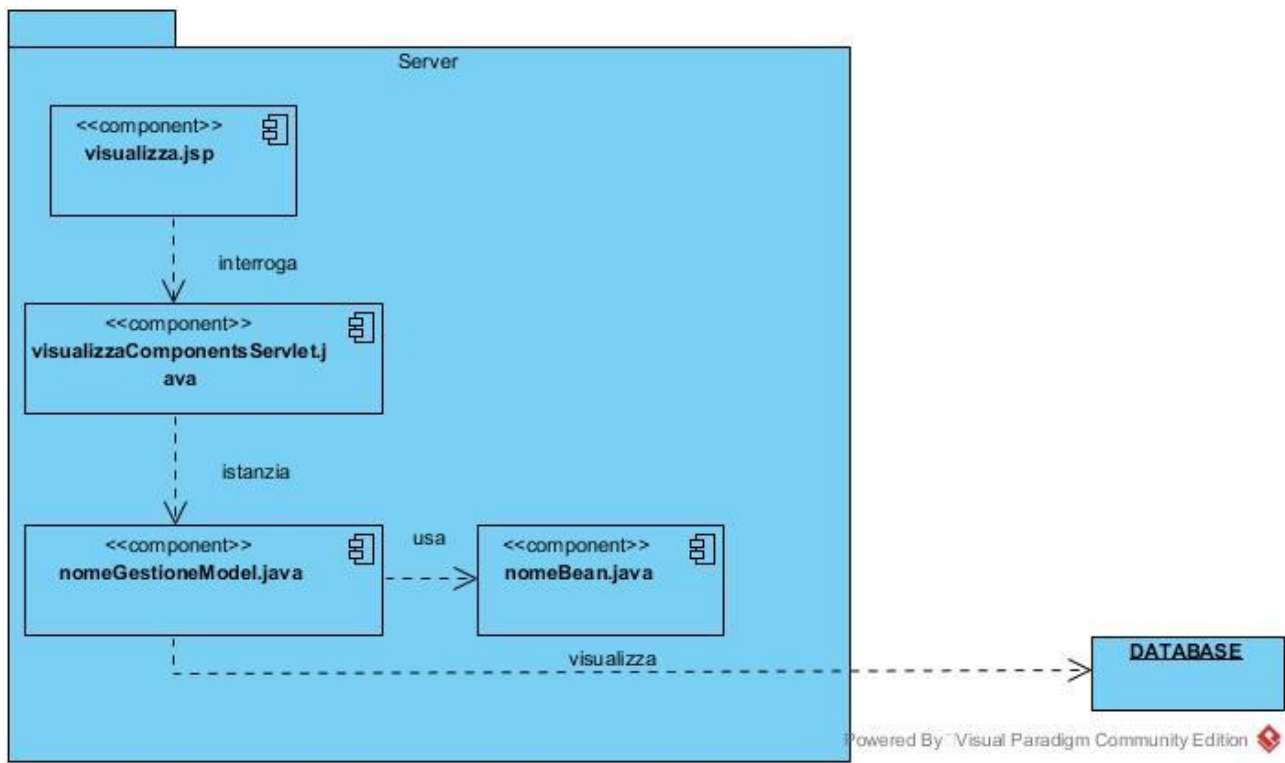
Application layer: il codice è scritto in linguaggio HTML5+java.

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.

5.4 Diagrammi a Design-Time:

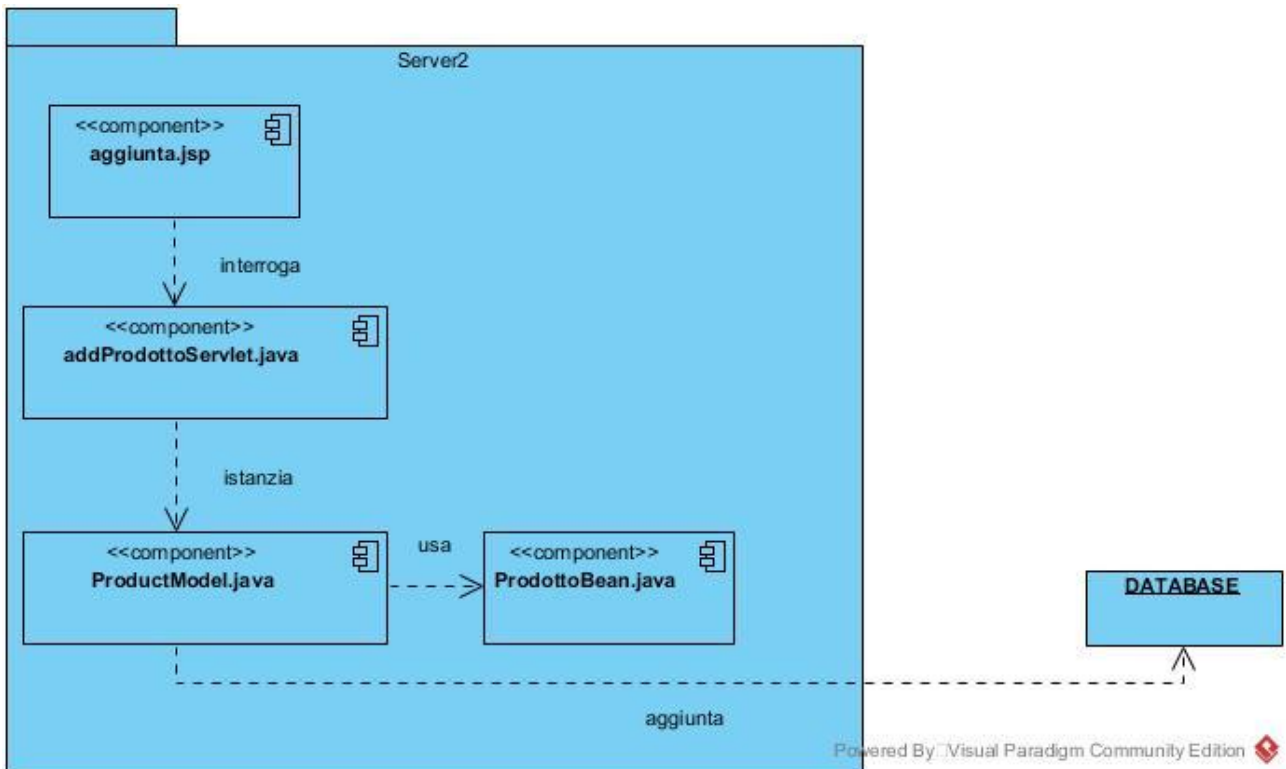
VISUALIZZA



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

- SD_3.0:visualizza utenti
- SD_4.1:visualizza ordini
- SD_2.1:visualizza prodotti

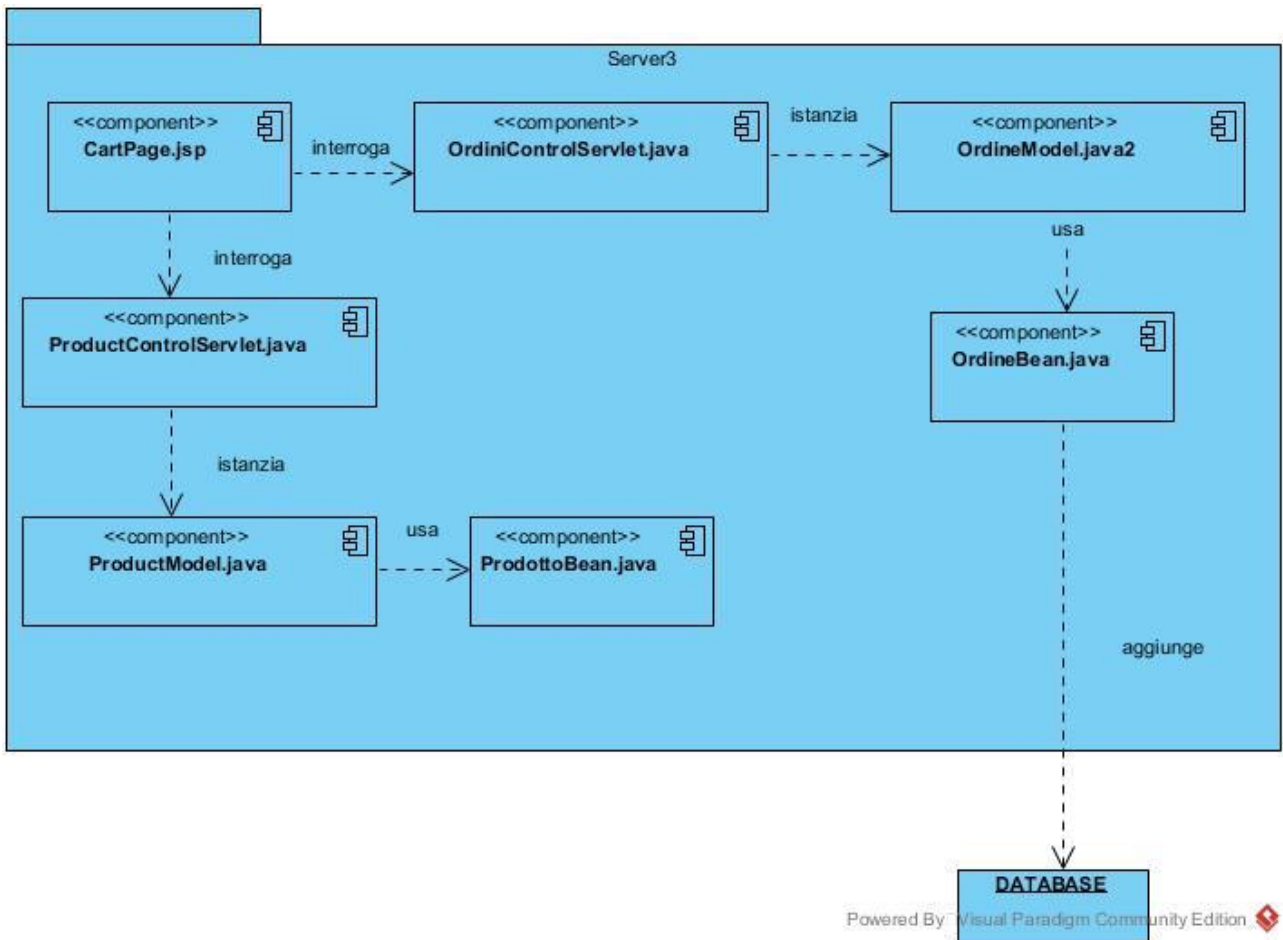
AGGIUNTA



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

-SD_2.2:aggiunta prodotto

ORDINE



Il precedente diagramma che descrive staticamente cosa succede a design time quando si vuole effettuare l'ordine di un prodotto:

-SD_4.0:prenotare prodotto/i;

5.5 Controllo degli accessi e della sicurezza

Pizzeria_LaSaporita è un sistema che presenta diversi attori, i quali hanno il permesso di eseguire operazioni differenti nel sistema. Per schematizzare meglio il controllo si è deciso di usare la matrice degli accessi riportata di seguito:

ATTORE SOTTOSISTEMA	Cliente	Admin
Autenticazione	Login Logout	Login Logout
Registrazione		Registra Utente
Prodotti	Visualizza Prodotti Ricerca Prodotti	Visualizza Prodotti Ricerca Prodotti Aggiungi Prodotti Rimuovi Prodotti
Utenti		Visualizza Utenti Rimuovi Utenti
Ordini	Effettua Ordine	Visualizza Ordine Rimuovi Ordine

4.6 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 event-driven. Quindi non abbiamo una sequenza di operazioni prestabilite ma, è l'utente che sceglie l'operazione da eseguire.

5.7 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.

5.7.1 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.

5.7.2 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...

5.7.3 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	Raffaella:Admin
Flusso di Eventi Amministratore	<ol style="list-style-type: none">1. Raffaella 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,mostrando la home del sistema.

Nome Scenario	SC_Shutdown Server
Istanze di Attori Partecipanti	Raffaella:Admin
Flusso di Eventi Amministratore	<ol style="list-style-type: none"> 1. Raffaella 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	Admin	
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>Pizzeria_LaSaporita 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	Admin	
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>Pizzeria_LaSaporita 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		

5.8 Servizi dei sottosistemi

5.8.1 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

5.8.2 Gestione Registrazione

Sottosistema	Gestione Registrazione
Descrizione	Gestisce le operazioni di registrazione di tutti gli attori.
Servizi offerti	
Registra utente	Permette all'admin di registrare un nuovo utente.

5.8.3 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'admin, 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'admin.

5.8.4 Gestione Utenti

Sottosistema	Gestione Utenti
Descrizione	Permette la gestione degli utenti
Servizi offerti	
Rimozione utenti	Permette di eliminare uno o più utenti registrati nel sistema.
Visualizza utenti	Permette di visualizzare le informazioni sugli utenti registrati nel sistema

5.8.5 Gestione Ordini

Sottosistema	Gestione Ordini
Descrizione	Permette l'ordinazione di uno o più prodotti presenti nel Sistema
Servizi offerti	
Effettua Ordine	Funzionalità che permette la prenotazione di uno o più prodotti presenti nel Sistema da parte degli utenti.

6.Glossario

Pizzeria_LaSaporita: nome del sistema che verrà sviluppato;

Utente: il termine identifica un attore del sistema che può usufruire dei servizi offerti,nello specifico si riferisce al cliente.

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;

Admin: generalizzazione dell'attore amministratore e responsabile;

Prodotti: il termine identifica un oggetto venduto dal sistema;

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

Rimozione Prodotto: il termine identifica una funzionalità del sistema consentita all'admin 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