**System Design Document (SDD)**

**Prezzi Pazzi**

1. **Introduzione**
   1. **Scopo del sistema**

L’Obiettivo del progetto “Prezzi Pazzi” è quello di sviluppare un’applicazione Java che permette di:

1. Effettuare acquisti di prodotti di diversa natura.
2. Offrire un interfaccia semplice, chiara ed intuitiva che stimoli l’utente ad usufruire del servi
3. Ottimizzare e velocizzare gli acquisti da parte degli utenti del sito.
4. Offrire l’opportunità all’amministratore di inserire/eliminare offerte.
   1. **Obiettivi del design**
5. Tempo di risposta: Mediamente una richiesta da parte del cliente è soddisfatta in un tempo di risposta inferiore ai 10 secondi. Queste prestazioni sono garantite sviluppando classi che eseguono meno cicli possibili che aumentano il costo computazionale.
6. Usabilità: il sistema verrà utilizzato da diversi tipi di utenti perché ognuno avrà bisogno di accedere a determinate informazioni ed effettuare determinate azioni.
7. Efficienza: si presuppone che il sistema svolga i propri compiti in tempi accettabili e utilizzando nel miglior modo le risorse disponibili.
8. Robustezza: il software deve essere facilmente modificabile a fronte di ogni evenienza.
9. Funzionalità:il sistema deve soddisfare le esigenze esplicite e implicite che vengono evidenziate in fase di raccolta e analisi dei requisiti.
10. Manutenibilità: il software deve essere facilmente modificabile in base alle esigenze dei clienti o per una futura revisione da parte dei progettisti.
11. Estendibilità:E' consentito, in quanto è possibile aggiungere in futuro, nuove funzionalità al sistema, oppure creare nuove classi, con l’estensione di quelle già esistenti.
    1. **Definizioni, acronimi e abbreviazioni**

**Acronimi**

*RAD*: Requirement Analysis Document

*SDD*: System Design Document

*Java SE*: Java Standard Edition

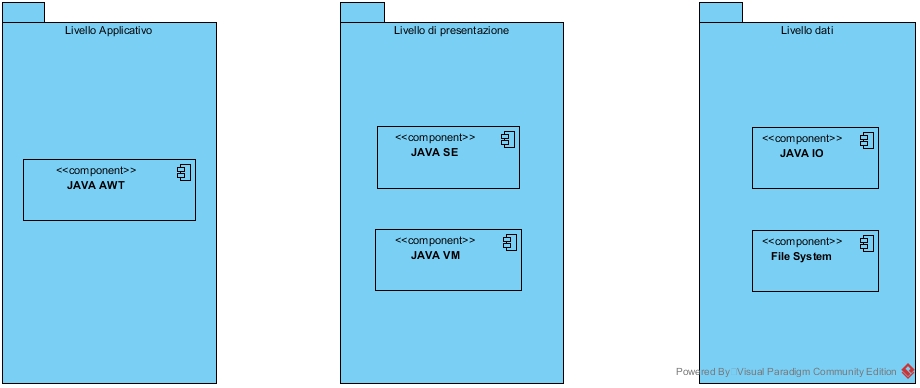
* 1. **Riferimenti**
* Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Prentice- Hall, 2003.
* Kevin Tatroe, Peter MacIntyre, Rasmus Lerdorf , Programming PHP, Creating Dynamic Web Pages O'REILLY, febbraio 2013, Third Edition.
* Jim Arlow, Ila Neustadt, UML e Unified Process, McGraw-Hill Italia Ian Sommerville,I. Sommerville, Software Engineering (9th edition), Addison Wesely.
* Roger S. Pressman, Principi di Ingegneria del Software (terza edizione, 2000), Mc Graw
* R.A.D “Progetti Pazzi”.

1. **Sistema corrente**

Il sistema da noi proposto è implementato dal nulla, in quanto nessun sistema esiste in precedenza.

1. **Architettura proposta**
   1. **Overview**

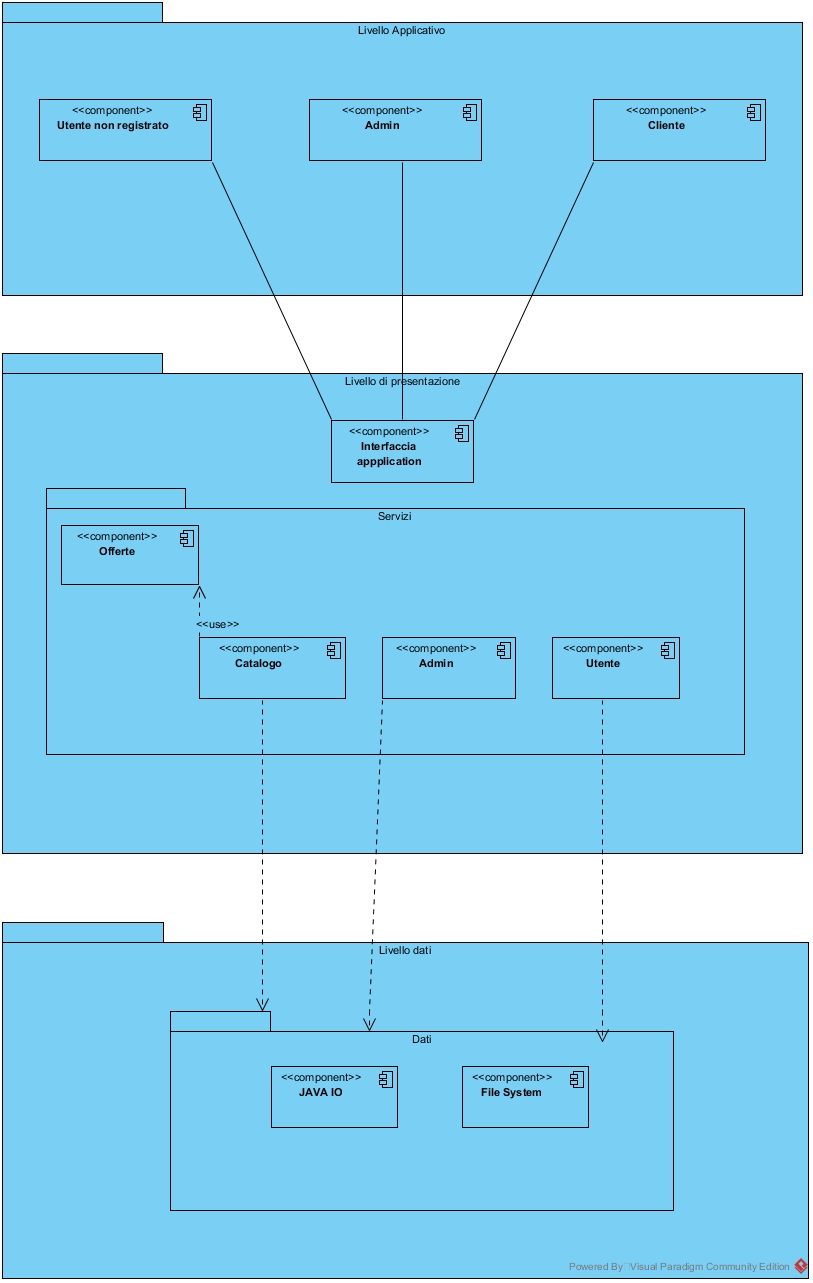
L’architettura del sistema proposto prevede una decomposizione in tre livelli logici. Lo scopo della suddivisione in livelli è quello di facilitare l’implementazione e ridurre la complessità generale del sistema. Il sistema comprenderà un livello di presentazione, un livello applicativo e un livello di gestione dei dati persistenti.



* 1. **Decomposizione in sottosistemi**

Il sistema è diviso in tre livelli:

* **Livello di applicativo**: rappresenta l’interfaccia grafica che consente al cliente di interagire con il sistema per poter usufruire dei suoi servizi. Nella nostra architettura, questo componente ricopre il ruolo del client che tramite l’interfaccia interagisce con il controllo.
* **Livello di presentazione**: ha il compito di eseguire le elaborazioni, generando risposte da inviare al client che le richiede. Nella nostra architettura, esso è visto come uno middelware tra l’Interfaccia grafica e i dati. L’interfaccia richiede una funzionalità a questo componente, che sfrutta il livello dati per assolvere al compito richiesto.
* **Livello Dati**: si occupa della gestione dei dati. Riceve interrogazioni dal livello di presentazione che chiede dati per compiere alcune funzionalità, ed ha il compito di rendere persistente il sistema.



**3.3 Mapping Hardware/Software**

Il sistema che sarà realizzato si basa sulla seguente Architettura

**Memorizzazione Dati:** File

**Linguaggi di programmazione utilizzati:** JAVA SE.

Controller

Gestione Autenticazione

UI



Gestione Offerte

Gestione Dati

Interfaccia Amministratore

Interfaccia Utente non registrato

File System

**3.4 Gestione dei dati persistenti**

I dati persistenti devono essere inseriti in opportuni file memorizzati nel file system della macchina che ospita la nostra applicazione.

* Le credenziali di tutti gli attori del sistema saranno conservati in file
* Tutte le offerte saranno salvate in un catalogo (file .dat)
* Il credito di ogni utente sarà mantenuto in un file
* L’acquisto di ogni utente verrà salvato in file dedicato
* Nel momento della registrazione verrà aggiunta una sezione riservata nel file degli utenti

**3.5 Controllo degli accessi e sicurezza**

All’interno del sistema saranno controllati gli accessi degli utenti tramite un meccanismo di login.

* L’Amministratore accederà al sito tramite la sua pagina di login ed avrà i permessi speciali per eliminare/aggiungere offerte e gestire sconti
* Il Cliente potrà accedere al sito con le credenziali che egli sceglierà nel momento della registrazione. Per acquistare i prodotti accederà alla sua area utente, e potrà anche fare delle ricariche di credito

Il sistema prevede tre figure di utenti: Cliente, Amministratori di Sistema,Utente non registrato . Ogni attore può accedere alle diverse funzionalità del sistema ma con diritti di accesso regolamentati sulla base delle differenti tipologie di utenza. Per documentare i diritti di accesso e per tenerne traccia all’interno del sistema usufruiamo di due file separati. Quando l’utente non registrato si registra vengono salvati i suoi dati in un file, mentre per creare un nuovo Admin c’è bisogno di salvare i dati in un altro file scrivendo direttamente in esso.

|  |  |  |  |
| --- | --- | --- | --- |
| Attori  Oggetti | Amministratore | Cliente | Utente non registrato |
| Effettua Login | X | X |  |
| Acquisto |  | X |  |
| Registrazione |  |  | X |
| Aggiunta Offerte | X |  |  |
| Eliminazione Offerte | X |  |  |
| Ricarica Credito |  | X |  |
| Consultazione Catalogo | X | X |  |

**3.6 Condizione di limite**

Le boundary conditions di installazione, start up e shut down saranno così definite:

Installazione: richiede la presenza di determinati file inseriti in un particolare percorso di file. Lato client occorre l’installazione della JAVA SDK.

Start-up: comporta il run dell’applicazione.

Shut-down: viene interrotto il servizio fornito dall’applicazione terminando il processo dell’applicazione.