|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**Il mondo della gomma  
SDD  
Versione 1.0**



Data: 24/11/2016

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Polito Marco | 0512101994 |
| Tortoriello Giulio | 0512101736 |
| Alifano Vincenzo | 0512100626 |
| Barone Ferdinando | 0512102330 |

|  |  |
| --- | --- |
| **Scritto da:** | Marco Polito, Giulio Tortoriello, Vincenzo Alifano, Ferdinando Barone |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 12/11/2016 | 0.5 | Introduzione | Marco Polito, Giulio Tortoriello,Barone Ferdinando,Vincenzo Alifano |
| 24/11/2016 | 1.0 | Boundary Condition,Servizi SottoSistemi | Vincenzo Alifano,Tortoriello Giulio |

Indice

1. Introduzione 4

1.1. Scopo del sistema 4

1.2. Obiettivi di design 4

1.3. Definizioni, acronimi e abbreviazioni 5

1.4. Panoramica 5

2. Architettura Sistema corrente 5

3. Architettura Sistema proposto 5

3.1. Panoramica 5

3.2. Decomposizione In Sottosistemi 6

3.3. Hardware/Software mapping 9

3.4. Gestione Dei Dati persistenti 9

3.5. Gestione Di Controllo Globale 9

3.6. Boundary Conditions 9

## 

## Introduzione

## Scopo del sistema

Lo scopo del progetto prevede la realizzazione di un sito web di e-commerce per la vendita di pneumatici gestita da un amministratore, il quale può inserire, modificare ed eliminare gli pneumatici sul catalogo del sito. Può inoltre evadere gli ordini degli utenti.

Gli utenti possono trovare i prodotti che rispondono alle loro esigenze attraverso un form di ricerca e successivamente possono acquistarli online, senza il bisogno di alcuna registrazione.

Il sistema, inoltre, fornisce all’utente una guida che spiega dettagliatamente come trovare le specifiche tecniche dello pneumatico (Es. Larghezza, altezza, diametro, stagionalità, marca produttore).

## Obiettivi di design

Il sistema sarà progettato con un interfaccia user-friendly in modo da agevolare l’utilizzo a persone. Sarà un prodotto molto semplice e utile nell’interazione che permetterà ai propri utenti di trovarsi a proprio agio durante l’utilizzo.

***Criteri di Performance***

**Throughput: Sarà possibile utilizzare il sistema da più macchine contemporaneamente. L’unica limitazione alle prestazioni del sistema sarà l’utilizzo di un database relazione utilizzato come repository centrale.**

**Tempo di risposta: Solitamente una richiesta da parte del client è soddisfatta in un tempo massimo di 4 secondi. Questo tipo di prestazioni sono garantite attraverso la scelta di un server capace di sopportare il carico applicativo massimo e la scelta di gestire i controlli sugli input del client così da segnalare immediatamente all’utente la non correttezza dell’input inserito.**

**Memoria: Prevedendo l’immagazzinamento di dati sul database riguardante ordini e prodotti, si stima siano necessari almeno 10 GB di memoria di massa.**

***Criteri di Affidabilità***

**Affidabilità: Il sistema deve essere affidabile. Bisogna gestire eventuali crash del sistema al fine di non compromettere le sue funzionalità. Il sistema deve garantire il funzionamento 24 ore su 24.**

**Efficienza: Il sistema deve svolgere i propri compiti in tempi accettabili utilizzando nel miglior modo le risorse disponibili.**

**Robustezza: Il sistema deve essere facilmente modificabile per fronteggiare ogni evenienza.**

**Funzionalità: Il sistema deve soddisfare le esigenze dell’utente.**

**Sicurezza: Nel sistema le transazioni devono avvenire in modo sicuro.**

***Criteri di Manutenzione***

**Estendibilità: È consentita, in quanto è possibile aggiungere in futuro nuove funzionalità al sistema (Es. Creare nuove classi estendendo quelle già esistenti).**

**Tracciabilità dei requisiti: La tracciabilità dei requisiti a partire dal codice è relativamente semplice per via della modularità dei componenti.**

## Definizioni, acronimi e abbreviazioni

**Il mondo della gomma:** nome del sistema da rilasciare.

**BROWSER:** navigatore, programma di accesso ad Internet.

**WEB-BASED:** applicazione software accessibile per mezzo di Internet.

**SDD:**System Design Document

## Panoramica

Il presente documento SDD(System Design Document)si occupa di:

Definire gli obiettivi di design del progetto(identificando quali caratteristiche di qualità dovrebbero essere ottimizzate).

Decomporre il sistema in sottosistemi più piccoli.

Il software verrà utilizzato per la gestione di un’azienda di vendita pneumatici. L’applicazione consentirà:

* La gestione di vendita di pneumatici adatti ad ogni tipo di autoveicolo.
* La consultazione dei prodotti disponibili in magazzino.
* La gestione dei dati relativi al cliente.

Inoltre il sistema si presenterà con un’interfaccia molto semplice ed intuitiva in modo che l’utente possa sfruttare al meglio tutte le funzionalità offerte.

## Architettura Sistema corrente

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

## Architettura Sistema proposto

## Panoramica

Il software è scomposto secondo lo schema client/server:

->Client:il browser preferito del cliente.

->Server:le pagine che implementano l’intero sistema.

Sono possibili tre configurazioni:

->Single-Tier: tutto il sistema è distribuito ed eseguito su un’unica macchina,che fungerà sia da client che da server.

->Two-Tier:il sistema è distribuito ed eseguito su due macchine differenti,una eseguirà sia il server che il database,mentre la seconda macchina eseguirà l’applicazione client.

Il sistema è basato su tre layer:

**-Interface Layer:**oggetti con cui l’utente interagisce,quali form,pulsanti e finestre…..

-**Logic Application Layer:**livello che si occupa di gestire e contenere le operazioni che gli utenti richiedono.

-**Storage Layer:**livello che gestisce la memorizzazione dei dati persistenti.

## Decomposizione In Sottosistemi

I tre livelli logici usati sono:

* **Interface Layer**: che include tutti i boundary object,finestre,form ecc.
* **Logic Application Layer:** che include tutti i Control Object e gli Entity Object realizzando così tutti i processi logici e di controllo forniti dal sistema.
* **Storage Layer:** responsabile della gestione e della memorizzazione degli oggetti permanenti del sistema.

L’utente interagirà con le componenti Interface. Il Logic Application riceverà i comandi dell’utente inviati tramite l’Interface,e li metterà in atto richiamando le funzioni del Logic Application e dello Storage.Lo Storage interagirà con il database e notificherà alla Interface i cambiamenti.

I componenti individuati all’interno del sottosistema di Logic Application sono:

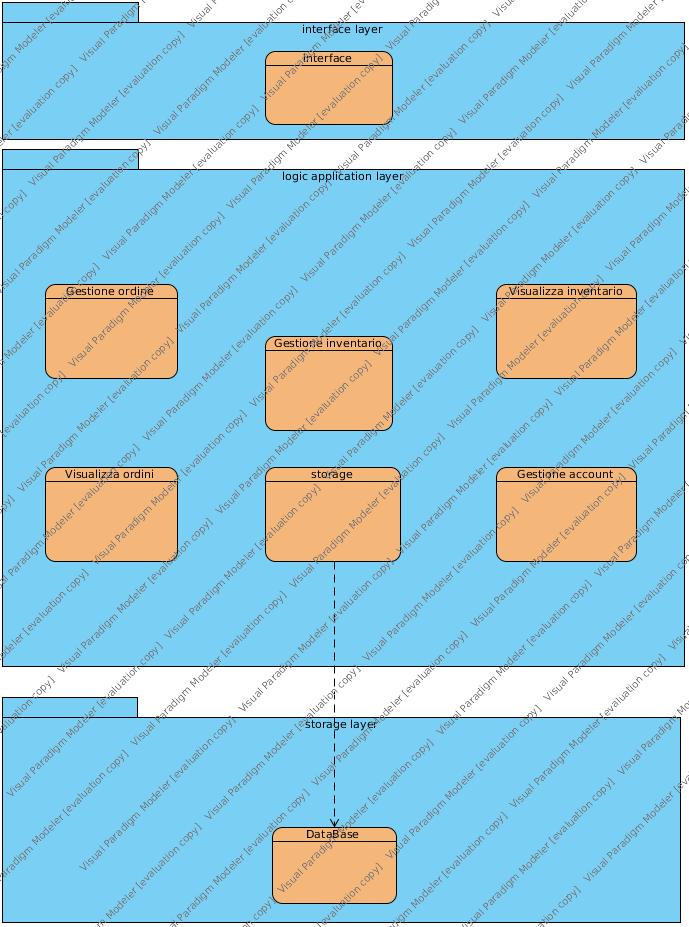
* Visualizza Inventario:Modulo che permette la visualizzazione dei prodotti disponibili in magazzino.
* Visualizza Ordine:Modulo che permette la visualizzazione degli ordini degli utenti.
* Gestione Account:Modulo che si occupa di effettuare il login e il logout dell’amministratore.
* Gestione Ordine:Modulo che permette al gestore di evadere gli ordini effettuati dagli utenti
* Gestione Inventario:Modulo che permette al gestore di effettuare le operazioni di aggiunta,modifica ed eliminazione del prodotto.
* Storage:Modulo che si occupa dell’interazione con il database.
* Control:Componente che si occupa di verificare l’input utente.

Le componenti individuate all’interno del sottosistema di Interface Layer:

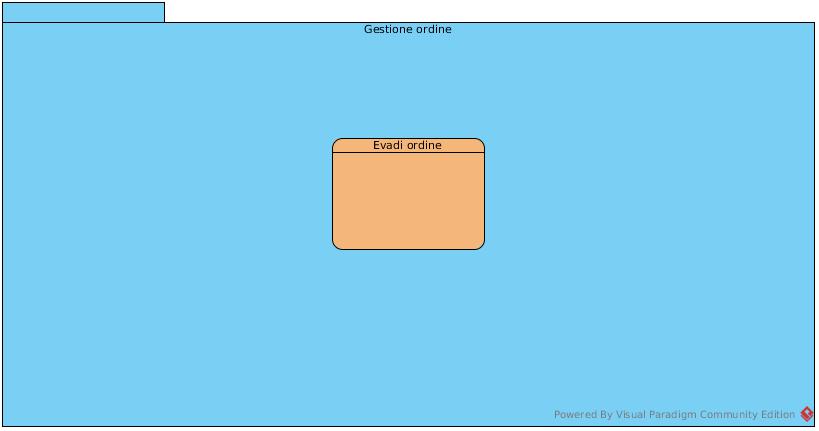
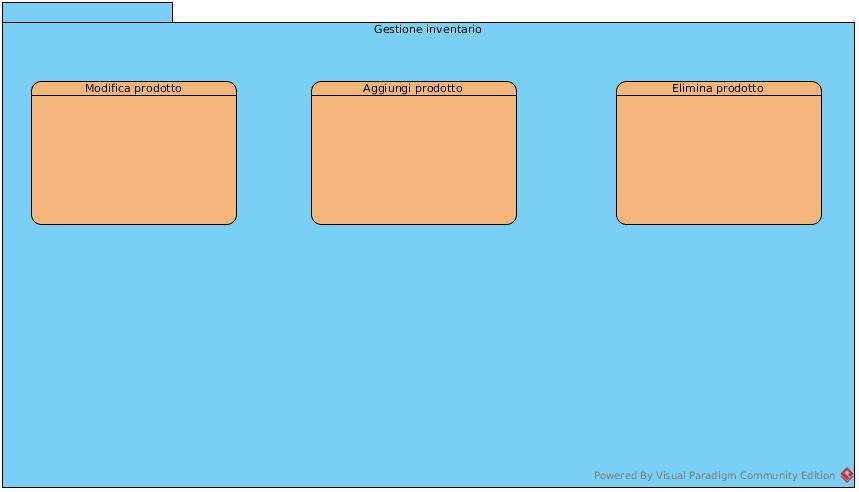
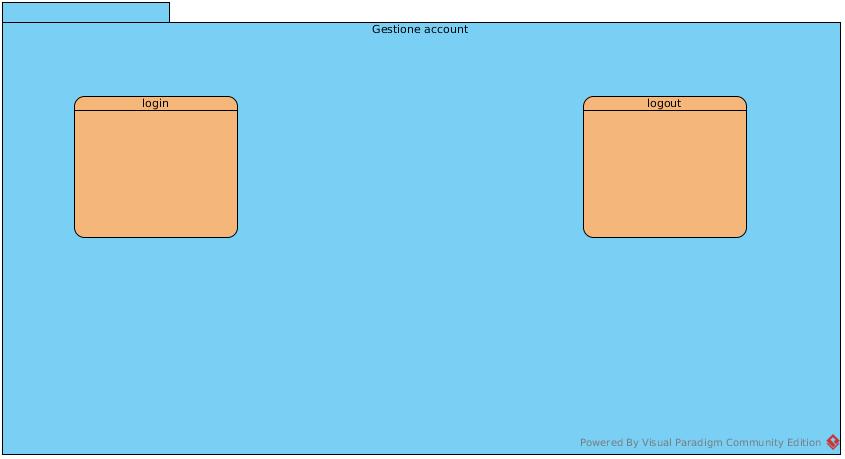
* Interface :Modulo che rappresenta l’Interfaccia grafica con cui l’utente interagirà con il sistema.

Le componenti individuate all’interno del sottosistema di Storage Layer:

* Database:Modulo che inoltra le richieste di recupero dati direttamente al database fisico e ne recupera i dati per poi restituirli allo strato delle applicazioni logiche.



Di seguito vengono specificate per ogni modulo le componenti di cui è formato:



## Hardware/Software mapping

Il sistema che sarà realizzato si basa su un’architettura Web-based, composta da vari

moduli client che richiedono servizi a un modulo server.

**Protocollo Richiesto**: HTTP

**Memorizzazione** **Dati**: DMBS MySQL WebServer: Apache

**Linguaggi di programmazione utilizzati**: Html, CSS , Javascript, Servlet e JSP.

## Gestione Dei Dati persistenti

I dati persistenti conservati nel database sono:

- Prodotto: id prodotto, nome, descrizione, quantità, prezzo, foto, stagionalità, marca.

- Ordine: id ordine, prodotto, dati cliente.

## Gestione Di Controllo Globale

Il flusso di controllo del sistema software è gestito da un “event driven control”.Appena si verifica un evento generato dall’utente tramite l’interfaccia grafica,il controllo viene trasferito al relativo oggetto che lo gestisce. L’oggetto a sua volta manda in esecuzione il sottosistema.

Questo meccanismo è molto flessibile e sfrutta al massimo le potenzialità del linguaggio di programmazione utilizzato per sviluppare il software.

## Boundary Conditions

Il sistema sarà installato in locale e sarà gestito da un amministratore il quale inserirà nel database i prodotti presenti in magazzino. L’utente che accede al sito troverà una Home Page in cui compariranno i diversi form. Il sottosistema accederà al database per mostrare al cliente i dati persistenti. Quindi l’utente avrà a disposizione determinate funzionalità in base alle sue scelte.

* Fallimento Del Sistema: Nel caso in cui si verifichi un crash del sistema i dati non andranno persi poiché gestiti dal DBMS. Se si verificano problemi Hardware i dati andranno persi. Eventuali perdite possono essere prevenute da backup che avvengono periodicamente.