

**GMAfaltiTuozzo**  
**SDD**  
**Versione 1.5**



Data: 25/11/2016

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: <u>System Design Document</u>	Data: 25/11/2016

**Partecipanti:**

Nome	Matricola
Greco Vincenzo	0512103060
Pirozzi Attilio	0512103078
Pizzuti Tullio	0512103000
Tuozzo Antonio	0512103096

<b>Scritto da:</b>	Tuozzo Antonio
--------------------	----------------

## Revision History

Data	Versione	Descrizione	Autore
21/11/2016	1.0	Creazione documento	Antonio Tuozzo
21/11/2016	1.1	Scrittura punti 1 e 2	Antonio Tuozzo e Vincenzo Greco
23/11/2016	1.2	Scrittura punti 3	Tutti
24/11/2016	1.3	Continuazione punto 3 e scrittura punto 4	Tutti
25/11/2016	1.4	Scrittura punto 5	Vincenzo Greco
25/11/2016	1.5	Revisione totale	Tutti

Progetto: GMASflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## Indice

1.	Introduzione.....	4
1.1.	Scopo del sistema.....	4
1.2.	Design Goals .....	4
1.3.	Definizioni, acronimi e terminologia .....	5
1.4.	Referenze.....	5
1.5.	Panoramica.....	6
2.	Architettura del sistema corrente .....	6
3.	Architettura del sistema proposto .....	6
3.1.	Panoramica.....	6
3.2.	Decomposizione in sottosistemi.....	7
3.3.	Mapping hardware/software.....	9
3.4.	Gestione dati persistenti .....	12
3.5.	Controllo degli accessi e sicurezza .....	13
3.6.	Controllo globale del sistema.....	13
3.7.	Condizione Boundary.....	14
3.7.1.	Descrizione.....	14
3.7.2.	Scenari e casi d'uso.....	15
4.	<b>Servizi dei sottosistemi</b> .....	17
4.1.	Amministratore .....	17
4.1.1.	Gestione Prodotti.....	17
4.1.2.	Gestione Magazzino.....	17
4.1.3.	Gestione Storici .....	18
4.1.4.	Gestione Richieste di Carico.....	18
4.1.5.	Gestione Disponibilità.....	18
4.2.	Magazziniere .....	19
4.2.1.	Richieste Carico .....	19
4.2.2.	Disponibilità.....	19
4.2.3.	Gestione Storici .....	19
4.2.4.	Scarico.....	20
5.	<b>Glossario</b> .....	21

Progetto: GMAsfaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

# 1. Introduzione

## 1.1. Scopo del sistema

Lo scopo è di automatizzare il servizio di gestione dei magazzini della società AsfaltiTuozzo. Ha come obiettivo la gestione in maniera agevole delle comunicazioni tra amministratore e magazzinieri, mettendoli in diretto contatto tramite un meccanismo di richieste di carico e scarico della merce.

## 1.2. Design Goals

I design goals rappresentano i requisiti qualitativi desiderati per il sistema GMAsfaltiTuozzo e forniscono un insieme consistente di criteri che devono essere considerati in fase di progettazione. I seguenti design goals, sono basati sui requisiti non funzionali individuati, e sono:

### Criteri di performance

- **Throughput:** L'architettura three-tier impone un hardware performante, in modo tale da poter funzionare con un carico medio di 1000 richieste al giorno.
- **Tempi di risposta:** Il sistema deve mantenere dei tempi di risposta molto brevi, possibilmente inferiore ai 20 secondi, per non far attendere l'utente.

### Criteri di dipendenza

- **Robustezza:** Il sistema deve essere in grado di gestire i possibili input errati dagli utenti, comunicando il problema verificatosi in modo chiaro ed immediato.
- **Affidabilità:** Il sistema deve garantire tutte le operazioni che il sistema stesso offre in maniera affidabile e sicura
- **Sicurezza:** Garantisce accesso alle sole funzionalità consentite allo specifico account

### Criteri di costi

- **Packaging:** Il software dovrà essere installato una sola volta sul server mentre i client potranno accedervi tramite browser.

### Criteri di manutenzione

- **Sopportabilità:** Il sistema sviluppato è adattabile, ovvero offre la possibilità di essere implementato per trattare concetti addizionali del dominio dell'applicazione. Il sistema sarà mantenibile, espandibile o trasferibile, a patto di rispettare i requisiti software necessari all'installazione
- **Tracciabilità dei requisiti:** Le attività di progettazione, definizione e gestione dei requisiti sono fondamentali per un buon progetto. Utile per ridurre costi e rischi

Progetto: GMAsfaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

- **Leggibilità:** Il codice prodotto deve essere correttamente documentato al fine di velocizzare eventuali modifiche o l'estensione di alcune funzionalità del sistema stesso. Dovranno essere seguiti i seguenti criteri:

- Ogni classe deve essere documentata mediante la descrizione della funzionalità che essa offre
- Ogni metodo non banale deve essere documentato
- Per ogni metodo dovranno essere descritti:
  - Parametri in input
  - Parametri in output
  - Descrizione della funzionalità offerta
  - Pre-condizione
  - Post-condizione
  - Eventuali errori

- **Criteri finali**

- **Usabilità:** Il sistema, dovendo essere utilizzato da persone inesperte non deve necessitare di un addestramento specifico all'uso. Ogni funzionalità, deve contenere al suo interno, una breve descrizione di sé stessa. Inoltre, per favorire un utilizzo semplice, deve essere possibile agli utenti, accedere a tutte le funzionalità in pochi click.

## 1.3. Definizioni, acronimi e terminologia

### -Definizioni:

Browser: Microsoft Edge, Firefox, Opera, Google Chrome  
 WebServer: server dove vengono memorizzate le operazioni

### -Acronimi:

RAD: Requirements Analysis Document  
 SDD: System Design Document  
 DB: DataBase  
 DBMS: DataBase Management System

## 1.4. Referenze

Come riferimento prendere visione del documento RAD del sistema GMAsfaltiTuozzo

Progetto: GMAsfaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## 1.5. Panoramica

Questa sezione descrive l'Architettura Software proposta per il sistema in base alle seguenti voci:

- **Decomposizione in sottosistemi:** Viene effettuata una descrizione ad alto livello della suddivisione del sistema in vari sottosistemi.
- **Mapping hardware/software:** Vengono mostrare le scelte intraprese circa le piattaforme hardware e software su cui il sistema dovrà girare. Successivamente sarà necessario:
  1. Specificare quale nodo e responsabile per quale funzionalità;
  2. Mappare le componenti individuate sulle piattaforme scelte.
 A tal fine, la descrizione di tale voce viene affiancata ad un component diagram e ad un deployment diagram per meglio delucidare quanto suddetto.
- **Gestione dei dati Persistenti:** Consente di identificare gli oggetti persistenti.
- **Controllo degli accessi e sicurezza:** Viene specificato con quale meccanismo ci si autentica nel sistema e come esso si interfaccia con il sistema in termini di matrici di accesso.
- **Controllo Software:** Descrive come il controllo globale viene implementato, e come viene gestita la concorrenza.
- **Boundary Conditions:** Vengono descritte le condizioni limite per il sistema GMAsfaltiTuozzo quali, configurazione, accensione e spegnimento del sistema
- **Servizi dei Sottosistemi:** Vengono descritti in termini di operazioni i servizi forniti da ogni sottosistema

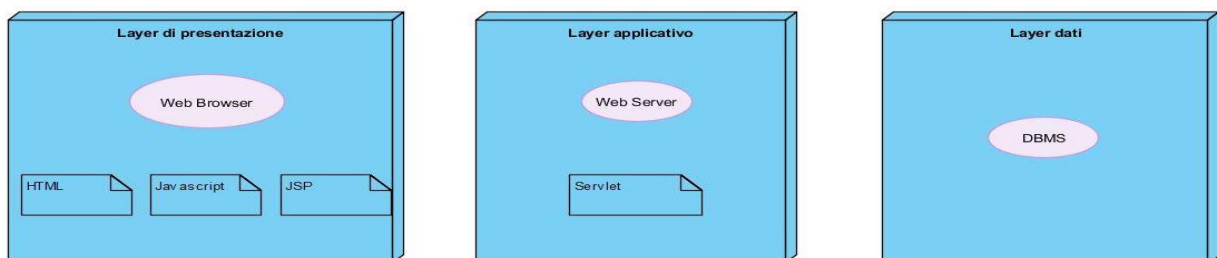
## 2. Architettura del sistema corrente

Il sistema che verrà a crearsi, non si basa o integra nessun sistema preesistente.

## 3. Architettura del sistema proposto

### 3.1. Panoramica

Il sistema software dovrà avere una architettura three-tier (a tre livelli) per soddisfare i requisiti di manutenibilità e supportabilità con un client che implementa il livello di presentazione, un server che implementa la logica applicativa e un DBMS per la gestione dei dati. I tre livelli dovranno essere potenzialmente installabili su tre macchine diverse.

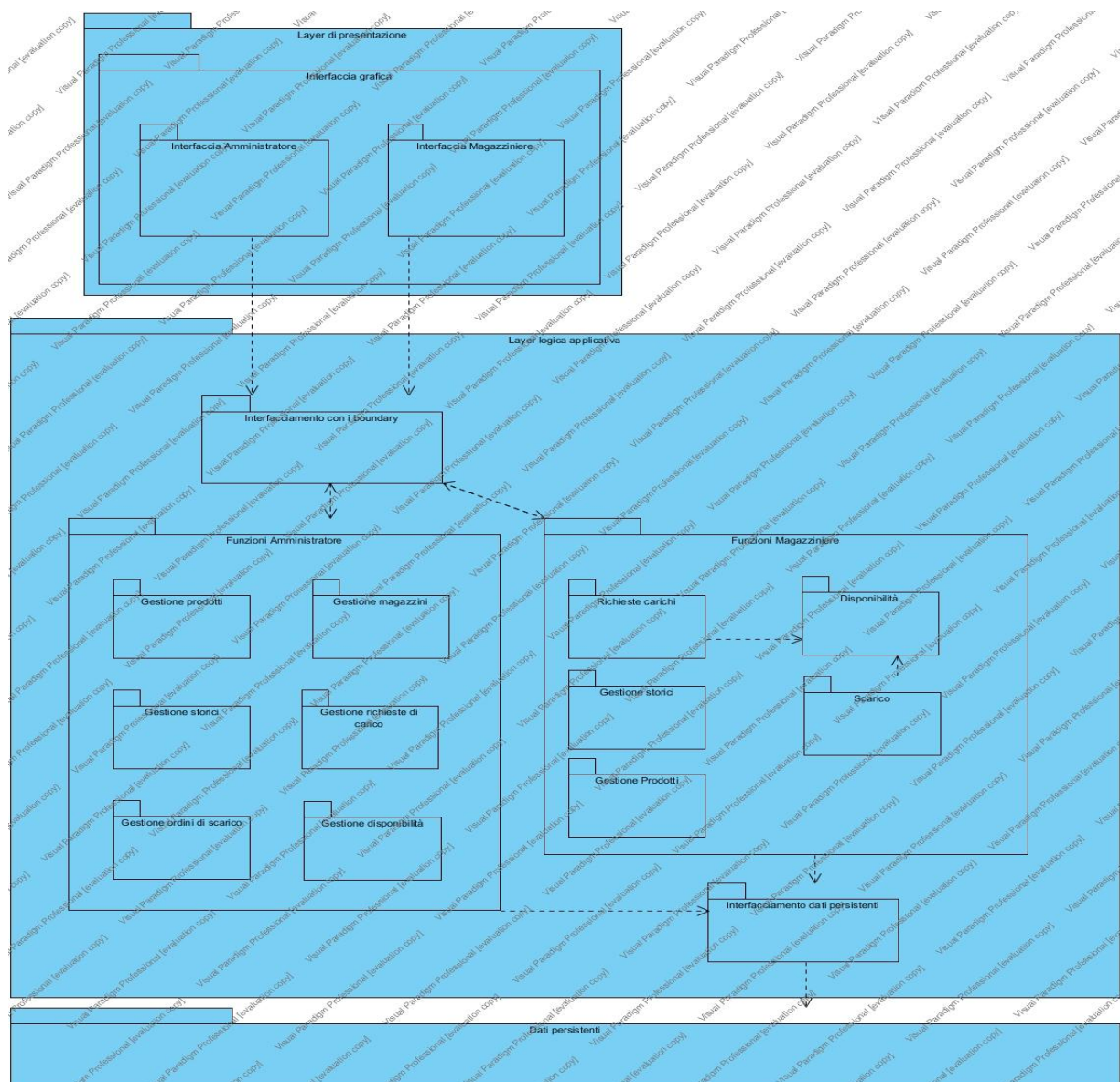


All'interno del livello di presentazione, viene rappresentata l'interfaccia utente, divisa tra Interfaccia Amministratore e Interfaccia Magazziniere.

Il layer della logica applicativa è diviso in base alle funzione dell'amministratore e quelle del magazziniere. Entrambi si interfacciano sia con gli oggetti boundary che con i dati persistenti.

Il livello di gestione della persistenza è costituito da un DBMS che si occupa della conservazione di dati e metadati propri del sistema.

### 3.2. Decomposizione in sottosistemi



Progetto: GMASflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## Interfaccia Grafica

Interfaccia Amministratore: Permette all'amministratore globale di essere autenticato e ne veicola l'accesso alle funzionalità del sistema attraverso un interfaccia a menù, descritta nel documento di analisi dei requisiti.

Interfaccia Magazziniere: Permette al magazziniere di un generico magazzino di essere autenticato e ne veicola l'accesso alle funzionalità del sistema attraverso un interfaccia a menù, descritta nel documento di analisi dei requisiti.

## Funzioni Amministratore

Gestione prodotti: Gestisce le operazione sui prodotti. Permette l'aggiunta, la modifica e la rimozione dei prodotti.

Gestione magazzini: Gestisce le operazione sui magazzini. Permette l'aggiunta, la modifica e la rimozione dei magazzini.

Gestione storici: Permette di visualizzare le movimentazione fatte dai magazzini

Gestione richieste di carico: Visualizza le richieste di carico inviate dai magazzini

Gestione ordini di scarico: Gestisce l'invio degli ordini di scarico

Visualizza disponibilità: Permette di prelevare le disponibilità dei magazzini.

## Funzioni Magazziniere

Richieste di carico: Permette di gestire le richieste di carico da inoltrare all'amministratore.

Gestione disponibilità: Permette di prelevare e aggiornare la disponibilità del magazzino

Gestione ordini di scarico: Permette di scaricare la merce dal magazzino.

Gestione storici: Gestisce la visualizzazione degli storici.

Gestione prodotti: Permette di gestire i prodotti.



Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## Interfacciamento con i Boundary

Sistema di inoltro delle richieste ai control da parte dei boundary. Ha anche una controparte tramite la quale i control istanziano i nuovi boundary da mandare in display per restituire il feedback delle operazioni.

## Interfacciamento con i dati persistenti

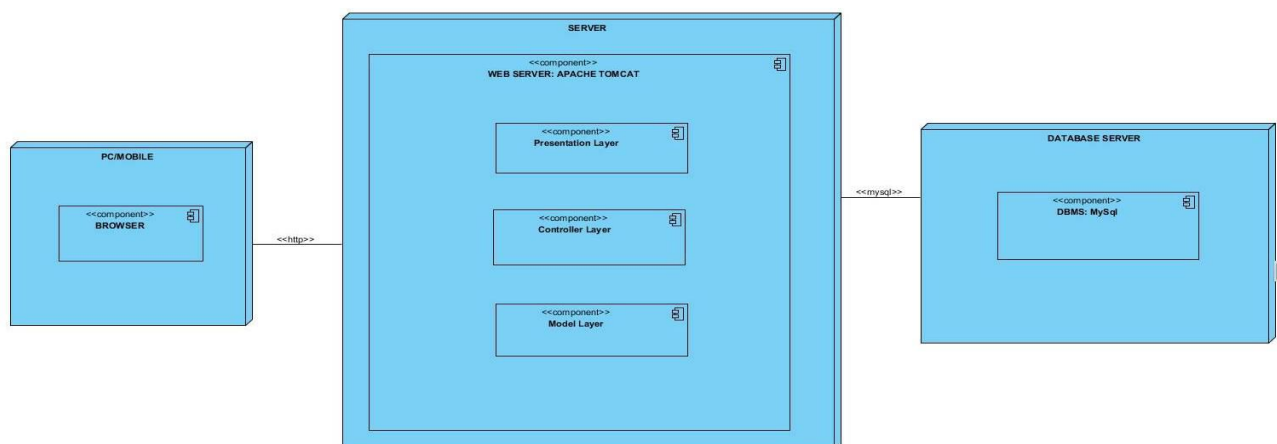
Sistema che si occupa di accedere ai dati memorizzati all'interno dei dati persistenti

## Dati Persistenti

E' responsabile di mantenere i dati persistenti e di assicurare la coerenza dei dati attraverso il controllo dei riferimenti e l'atomicità delle operazioni.

## 3.3. Mapping hardware/software

Il sistema GMAsfaltituozzo sarà web-based. Le funzionalità verranno implementate sul server in modo tale da poter essere accedute via TCP/IP. Al server si collegano i client ed il database, il tipo di utente è determinato in fase di autenticazione controllando nel database il tipo di utente che corrisponde allo Username inserito in quel client. I client dovranno effettuare richieste al server per eseguire le operazioni legate alle loro rispettive funzionalità.



Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## Web Server

Il server utilizzato è Apache Tomcat.

## Presentation Layer

L'utente utilizza il sistema mediante un Browser installato all'interno del suo dispositivo.

## Controller Layer

Il sistema, qe quindi le funzionalità, sono implementate in linguaggio JAVA, che verrà successivamente elaborato e tradotto in pagine JSP per poter essere inviate al browser del Client.

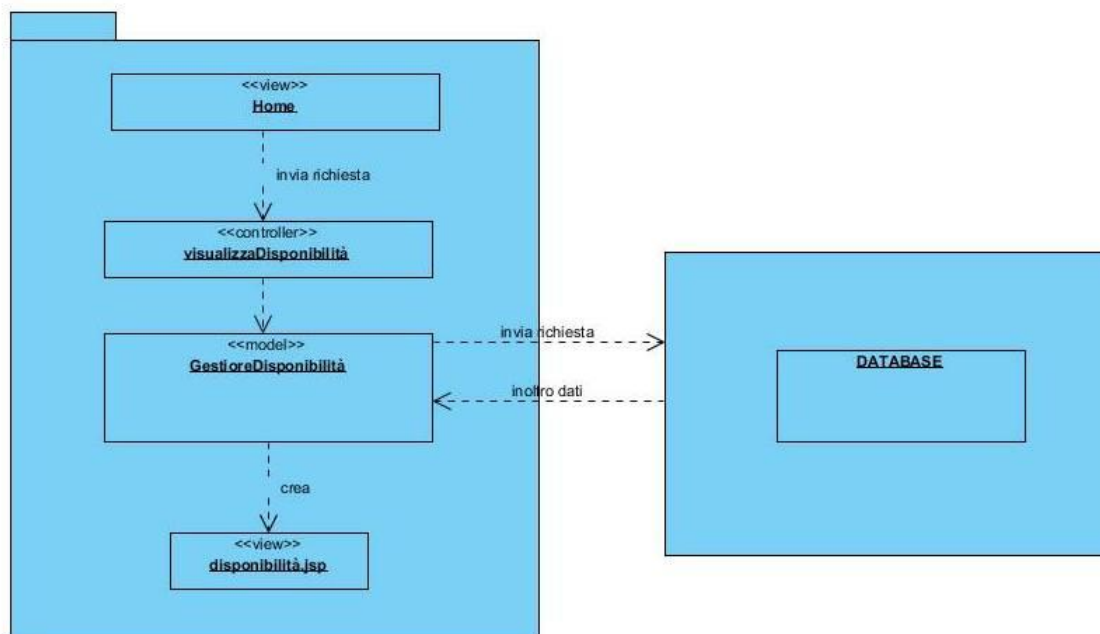
## Model Layer

Rappresenta il collegamento con il server da parte del sistema e si occupa di tutte le richieste di accesso e modifiche sui dati persistenti presenti nel database.

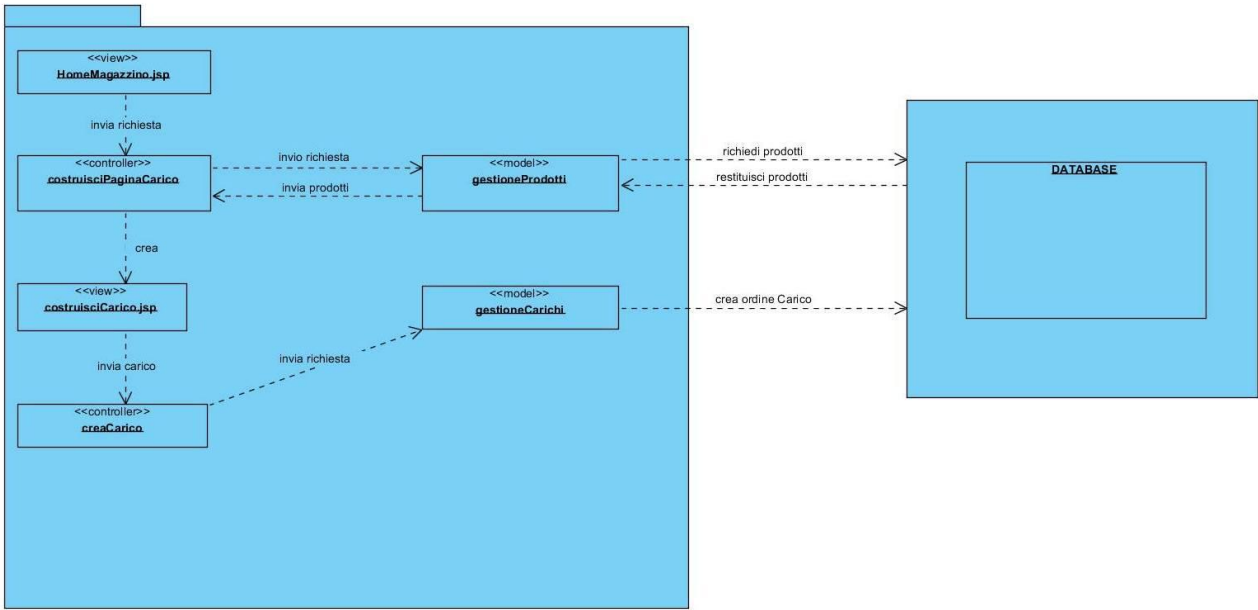
## Database Server

Il DBMS usato è MySQL il quale presenta molte API che permettono l'interazione tra sistema e database.

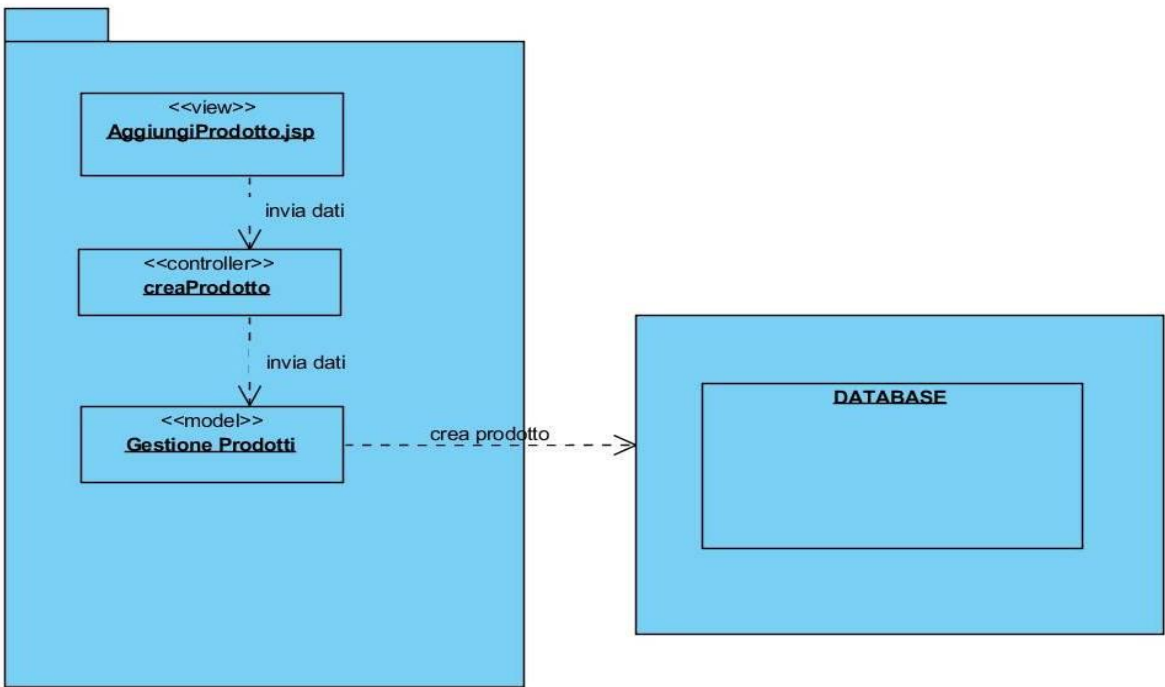
## Visualizza Disponibilità: Amministratore-Magazziniere



Crea ordine di carico: Magazziniere



Aggiungi prodotto: Amministratore



Progetto: GMASflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

### 3.4. Gestione dati persistenti

Sulla base del Requirements Analysis Document, sono stati identificati i seguenti oggetti persistenti:

Amministratore: Persona che dirige e controlla l'azienda fornitrice del servizio. Ha la possibilità di effettuare le operazioni di gestione sui magazzini e del controllo degli ordini di carico. Può decidere di inserire, cancellare e modificare i prodotti per i magazzini. Dirige e controlla l'azienda fornitrice del servizio.

Magazziniere: Responsabile del magazzino che usufruisce del sistema per gestire il magazzino a cui è assegnato.

Magazzino: Struttura usata per allocare e gestire la merce

Prodotto: Merce venduta dall'azienda che viene gestita dal sistema

Richieste Carichi: Operazione effettuata da un magazzino utilizzata per ordinare nuovi prodotti.

Scarico: Operazione imposta da un amministratore ad un magazzino mittente per il rifornimento di un magazzino destinatario.

Gestione storici: Gestisce gli storici delle operazioni effettuate.

Gestione richieste di carico: Gestisce le richieste di carico nel database.

Gestione Ordini Di Scarico: Gestisci gli ordini di scarico nel database.

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

Gestione Disponibilità: Legge la disponibilità dei magazzini nel database

Gestione Magazzini: Gestisce l'entità magazzino nel database

Gestione Prodotti: Gestisce i prodotti nel database.

### 3.5. Controllo degli accessi e sicurezza

Le funzioni del sistema sono state divise in base alla tipologia di utente riconosciuta al momento del login: Amministratore e Magazziniere.

L'amministratore può:

- Gestire i prodotti
- Gestire i magazzini
- Visualizzare gli storici e le disponibilità dei magazzini
- Visualizzare gli ordini di carico provenienti dai magazzini
- Creare ordini di scarico

Il magazziniere può:

- Visualizzare i prodotti
- Visualizzare il proprio storico e disponibilità
- Gestire gli ordini di carico
- Scaricare prodotti

### 3.6. Controllo globale del sistema

Il flusso di controllo sarà gestito nella maniera più consona ad un'applicazione web, quindi sarà di tipo event-driven. La gestione della concorrenza inoltre è un problema corposo da gestire in un approccio event-driven a causa dell'imprevedibilità degli eventi provenienti dai numerosi agenti con cui il sistema si troverà a dialogare: l'integrità dei dati è tuttavia garantita dal DBMS MySQL, tramite la semantica delle transazioni.

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## 3.7. Condizione Boundary

### 3.7.1. Descrizione

Le operazioni di installazione, start e stop del sistema GMAsfaltituozzo definiscono le boundary condition:

- **Installing:** lato server è richiesta la presenza del DBMS e del web server. Lato client è richiesta una connessione a Internet e un Browser compatibile con HTML e Javascript.
- **Configuration:** bisognerà inserire i file del sistema all'interno di una directory raggiungibile tramite web server attivo. E' necessario configurare il database, configurare il file per l'accesso all'amministratore. Successivamente l'amministratore potrà creare i prodotti e i magazzini.
- **Starting:** lato server bisognerà avviare il server per far partire il sistema.
- **Stopping:** lato server bisognerà stoppare il server per fermare il sistema.
- **Fallimento del sistema:** nell'eventualità si verifichi un errore all'interno del sistema si cercherà di ripristinare una configurazione del sistema precedente funzionante. Siccome i dati sono gestiti dal DBMS non c'è alcun rischio di perderli. Tuttavia, non è da escludere la perdita dei dati, in caso di un guasto al supporto di memorizzazione dei dati nel database-server. Per minimizzare questo rischio, si eseguiranno periodicamente dei backup dei dati del sistema.

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

### 3.7.2. Scenari e casi d'uso

#### Configurazione del server

<b>Attori</b>	Amministratore
<b>Entry condition</b>	Il sistema non è stato ancora configurato per il primo avvio
<b>Exit condition</b>	Il sistema può essere avviato
<b>Rilevanza</b>	Elevata
<b>Flusso di eventi</b>	<ol style="list-style-type: none"> <li>1. Copia i file del sistema in una directory raggiungibile dal Web</li> <li>2. Configura il file del server inserendo Username e Password</li> <li>3. Avvia il server</li> </ol>

#### Startup del Server

<b>Attori</b>	Amministratore globale
<b>Entry condition</b>	Il server non è stato ancora avviato
<b>Exit condition</b>	Il server è avviato
<b>Rilevanza</b>	Elevata
<b>Flusso di eventi</b>	<ol style="list-style-type: none"> <li>1. Avvia MySql</li> <li>2. Avvia il Web Server</li> </ol>

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## Shutdown del server

<b>Attori</b>	Amministratore generale
<b>Entry condition</b>	Il server è attivo
<b>Exit condition</b>	Il server non è più attivo
<b>Rilevanza</b>	Elevata
<b>Flusso di dati</b>	<ol style="list-style-type: none"> <li>1. Stoppa il Web Server</li> <li>2. Stoppa il servizio MySql</li> </ol>



Progetto: GMASflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## 4. Servizi dei sottosistemi

### 4.1. Amministratore

#### 4.1.1. Gestione Prodotti

<b>Sottosistema</b>	Gestione Prodotti
<b>Descrizione</b>	Sottosistema che gestisce i prodotti del sistema
<b>Servizi offerti</b>	<b>Descrizione</b>
Crea Prodotto	Permette l'inserimento dei dati di un nuovo prodotto nel sistema
Elimina Prodotto	Permette la cancellazione di un prodotto del sistema
Modifica Prodotto	Permette l'inserimento dei dati di un prodotto da modificare nel sistema

#### 4.1.2. Gestione Magazzino

<b>Sottosistema</b>	Gestione Magazzino
<b>Descrizione</b>	Sottosistema che gestisce i magazzini del sistema
<b>Servizi offerti</b>	<b>Descrizione</b>
Crea Magazzino	Permette l'inserimento dei dati di un nuovo magazzino nel sistema
Elimina Magazzino	Permette la cancellazione di un magazzino del sistema
Modifica Magazzino	Permette l'inserimento dei dati di un magazzino da modificare nel sistema

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

#### 4.1.3. Gestione Storici

<b>Sottosistema</b>	Gestione Storici
<b>Descrizione</b>	Sottosistema che gestisce gli storici delle operazioni effettuate nel tempo da parte dell'amministratore
<b>Servizi offerti</b>	<b>Descrizione</b>
Visualizza storici	Possibilità di visualizzare un listato delle operazioni effettuate in passato

#### 4.1.4. Gestione Richieste di Carico

<b>Sottosistema</b>	Gestione Richieste di Scarico
<b>Descrizione</b>	Sottosistema che gestisce il processo di ordini di scarico
<b>Servizi offerti</b>	<b>Descrizione</b>
Crea ordine	Permette l'inserimento dei dati di un nuovo ordine nel sistema
Modifica stato	Permette l'inserimento dei dati di un ordine da modificare nel sistema

#### 4.1.5. Gestione Disponibilità

<b>Sottosistema</b>	Gestione disponibilità
<b>Descrizione</b>	Sottosistema che visualizza la disponibilità dei prodotti presenti in magazzino da parte dell'amministratore
<b>Servizi offerti</b>	<b>Descrizione</b>
Visualizza disponibilità	Viene visualizzata un listato contenente la disponibilità dei prodotti presenti in magazzino

Progetto: GMASflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## 4.2. Magazziniere

### 4.2.1. Richieste Carico

Sottosistema	Richieste Carico
<b>Descrizione</b>	Sottosistema che gestisce il processo di richieste di carico da parte del magazziniere
<b>Servizi offerti</b>	<b>Descrizione</b>
Invia richiesta	Permette di effettuare una richiesta all'amministratore di un carico di prodotti
Aggiorna stato carico	Permette l'aggiornamento di stato di un carico

### 4.2.2. Disponibilità

Sottosistema	Disponibilità
<b>Descrizione</b>	Sottosistema che visualizza la disponibilità dei prodotti presenti in magazzino da parte del magazziniere
<b>Servizi offerti</b>	<b>Descrizione</b>
Visualizza	Permette la visualizzazione del listato contenente la disponibilità dei prodotti
Aggiorna	Permette l'aggiornamento dei prodotti

### 4.2.3. Gestione Storici

Sottosistema	Gestione storici
<b>Descrizione</b>	Sottosistema che gestisce gli storici delle operazioni effettuate nel tempo da parte del magazziniere
<b>Servizi offerti</b>	<b>Descrizione</b>
Visualizza	Possibilità di visualizzare un listato delle operazioni effettuate in passato
Registra	Possibilità di registrare un'operazione effettuata nel sistema

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

#### 4.2.4. Scarico

<b>Sottosistema</b>	Scarico
<b>Descrizione</b>	Sottosistema che gestisce lo scarico dei prodotti da parte del magazziniere
<b>Servizi offerti</b>	<b>Descrizione</b>
Crea scarico	Permette di inserire i dati per la creazione di un nuovo scarico

Progetto: GMAsflaltiTuozzo	Versione: 1.5
Documento: System Design Document	Data: 25/11/2016

## 5. Glossario

**Atomicità(operazione)** : consiste in un'operazione di esecuzione indivisibile dal punto di vista logico.

**Backup**: si indica la replicazione, su un qualunque supporto di memorizzazione, di materiale informativo archiviato nella memoria di massa dei computer.

**Controller**: riceve i comandi dell'utente (in genere attraverso il view) e li attua modificando lo stato degli altri due componenti.

**Coerenza(dati)**: processo di mantenere l'uniformità dei dati.

**Event-driven**: programma che gestisce le risposte alle azioni generate dall'utente o dal sistema.

**HTML**: L'HyperText Markup Language, in informatica è il linguaggio di markup solitamente usato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web sotto forma di pagine web.

**MySQL**: è un Relational Database Management System (RDBMS) composto da un client a riga di comando e un server.

**Model**: fornisce i metodi per accedere ai dati utili all'applicazione.

**JSP**: In informatica JavaServer Pages, di solito indicato con la sigla JSP, è una tecnologia di programmazione Web in Java per lo sviluppo della logica di presentazione di applicazioni Web, fornendo contenuti dinamici in formato HTML.

**Javascript**: è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client.

**Java**: è un linguaggio di programmazione orientato agli oggetti a tipizzazione statica specificatamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione.

**Servlet**: sono oggetti scritti in linguaggio Java che operano all'interno di un server web.

**View**: visualizza i dati contenuti nel model e si occupa dell'interazione con utenti e agent.

**Web-Server**: è un'applicazione software che, in esecuzione su un server è in grado di gestire le richieste di trasferimento di pagine web di un client, tipicamente un web browser.