

Università degli Studi di Salerno
Corso di Ingegneria del Software

GMAfaltiTuozzo
ODD
Versione 1.0



Data: 13/12/2016

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

Partecipanti:

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

Scritto da:	Antonio Tuozzo, Vincenzo Greco
--------------------	--------------------------------

Revision History

Data	Versione	Descrizione	Autore
21/12/2016	1.0	Creazione documento e scrittura punti 1,2	Tuozzo, Greco
23/12/2016	1.0	Scrittura punto 3,4,5	Pizzuti

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

Indice

1.	Introduzione	4
1.1.	Scelte progettuali per l'Object Design	4
1.2.	Linee guida nella documentazione delle interfacce	5
1.3.	Definizioni, Acronimi e Abbreviazioni	5
1.4.	Riferimenti	6
2.	Packages	6
2.1.	Funzioni Amministratore	6
2.2.	Funzioni Magazziniere	6
3.	Interfaccia delle classi	7
3.1.	Registrazione scarico (Magazziniere)	7
3.1.1.	OperazioneCompletaBean	7
3.1.2.	Scarica e registra control	8
3.1.3.	GetInformation	8
3.2.	Inserisci prodotto (amministratore)	9
3.2.1.	ProdottoBean	9
3.2.2.	Inserisci prodotto control	10
3.2.3.	GetInformation	10
4.	Sequence diagram (low level)	11
4.1.	Registrazione scarico merce	11
4.2.	Registrazione prodotto	12
5.	Database	13
5.1.	DBMS	13
5.2.	Descrizione database	13
5.2.1.	Magazzino	13
5.2.2.	Prodotto	13
5.2.3.	Disponibilità	13
5.2.4.	Operazioni completate	13
5.2.5.	Composizione operazioni completate	13
5.2.6.	Operazioni in sospeso	14
5.2.7.	Composizione operazioni in sospeso	14

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

1. Introduzione

1.1. Scelte progettuali per l'Object Design

Successivamente alla scrittura del RAD e dell'SDD, abbiamo scritto, in linea generale tutte le caratteristiche e le funzionalità del nostro progetto. Abbiamo definito i seguenti compromessi:

- **Prestazioni vs Costi**

Il sistema realizzato è stato sviluppato grazie all'uso di materiale open source minimizzando così i costi e rendendo l'utilizzo più che soddisfacente. In particolar modo ci siamo serviti di un server web locale gratuito sviluppato dalla Apache Software Foundation(Apache Tomcat).

- **Interfaccia vs Tempo di risposta**

Il tempo di risposta tra server e interfaccia è sufficientemente rapido a soddisfare le esigenze dei vari utenti collegati al sistema. Pertanto la grandezza del database è direttamente proporzionale al tempo di risposta e ricerca nel database.

- **Costi vs Mantenimento**

L'utilizzo di risorse open source rendono economici i costi di mantenimento del sistema. Durante lo sviluppo e la progettazione del progetto GMAsflatiTuozzo, si è ritenuto opportuno effettuare delle scelte progettuali mirate soprattutto a ridurre la complessità nello sviluppo e a favorire l'usabilità del sito da parte degli utenti finali indipendentemente dal fatto che si trattassero di amministratori o clienti.

La parte server è stata realizzata utilizzando varie tecnologie di programmazione web:

JSP: in grado di generare pagine web dinamiche

Servlet: capace di gestire le richieste generate da uno o più client

Il database è stato creato in SQL con le relative query sul DBMS grazie all'ausilio del software libero MySQL.

Per la parte client, invece, sono stati utilizzati diversi linguaggi di programmazione:

HTML: per la struttura statica dei documenti;

CSS : utilizzato per definire lo style delle pagine web;

Javascript: utilizzato per rendere dinamiche le pagine HTML.

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

1.2. Linee guida nella documentazione delle interfacce

- Le classi hanno nomi singolari;
- I nomi delle classi identificano le operazioni che implementano;
- I metodi sono chiamati con frasi verbali, mentre i campi ed i parametri con frasi sostantivo;
- Ogni classe deve avere una breve spiegazione del suo scopo;
- La descrizione del metodo deve apparire prima di ogni sua dichiarazione e deve descrivere il suo scopo
- Dovranno essere elencati tutti i parametri di ogni metodi con le eventuali eccezioni relative.

Il sistema è accessibile sia dal magazziniere che dall'amministratore.

Il magazziniere potrà consultare il catalogo dei prodotti presenti nel magazzino, effettuare ordini di carico e di scarico verso altri magazzini o terzi. Potrà anche visualizzare lo storico delle movimentazioni da lui effettuate.

L'amministratore gestirà tutta la parte amministrativa, potrà gestire (aggiunta, cancellazione e modifica) dei prodotti e dei magazzini. Gestirà tutti gli ordini di carico effettuate dai magazzini e potrà inoltre richiedere un ordine di carico dal fornitore unico.

Tutti i tipi di utenti, quindi, saranno coinvolti in query al database ma l'utilizzo delle form rende quest'ultime semplicissime anche per i meno esperti.

1.3. Definizioni, Acronimi e Abbreviazioni

Definizioni:

- **Browser Web:** applicazione che permette l'accesso a Internet
- **WebServer:** server dove vengono memorizzate le operazioni
- **Amministratore:** responsabile della gestione dell'intero sito GMAsflatiTuozzo
- **Magazziniere:** persona che gestisce il magazzino
- **MySQL** è un software per la gestione di database relazionali composto da un client a riga di comando e un server.
- **SQL** è un linguaggio standardizzato per database basati sul modello relazionale in grado di poter:
 - Creare e modificare schemi di database
 - Inserire, cancellare e modificare dati memorizzati
 - Interrogare dati memorizzati
 - Gestire gli utenti e i permessi
- **Query:** In informatica il termine query viene utilizzato per indicare l'interrogazione da parte di un utente di un database, strutturato tipicamente secondo il modello relazionale, per compiere determinate operazioni sui dati.

Acronimi:

- **RAD:** Requirements Analysis Document
- **SDD:** System Design Document

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

- **ODD:** Object Design Document
- **DB:** DataBase
- **DBMS:** DataBase Management System

1.4. Riferimenti

I riferimenti si collegano ai precedenti documenti RAD, SDD.

2. Packages

Questa sezione descrive la decomposizione dei sottosistemi in pacchetti e l'organizzazione dei file di codice. Questo include una panoramica di ogni pacchetto, le sue dipendenze con altri pacchetti, e il suo utilizzo previsto.

2.1. Funzioni Amministratore

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

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

Gestione storici: Permette di visualizzare le movimentazioni 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.

2.2. 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.

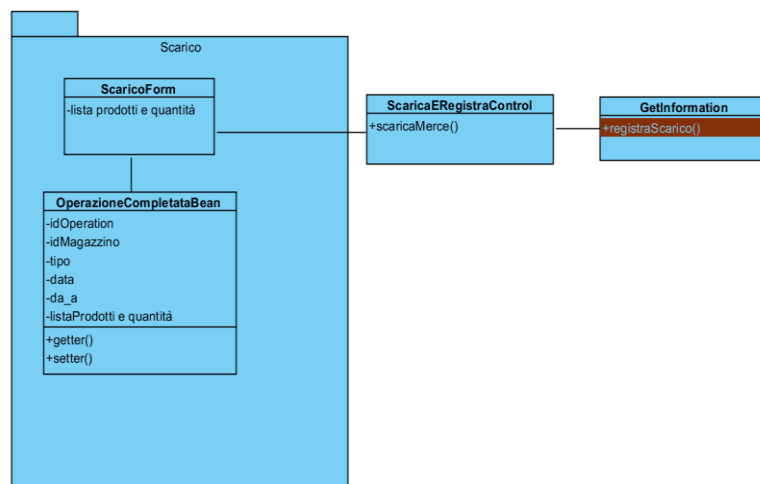
Gestione storici: Gestisce la visualizzazione degli storici.

Gestione prodotti: Permette di gestire i prodotti.

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

3. Interfaccia delle classi

3.1. Registrazione scarico (Magazziniere)



3.1.1. OperazioneCompletataBean

Classe: OperazioneCompletataBean	Autore: Team	
Descrizione	Classe che rappresenta l'entità operazione completata	
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
IdOperazione	Privato	Identifica l'operazione
idMagazzino	Privato	Identifica il magazzino
Tipo	Privato	Identifica il tipo di operazione effettuata
Data	Privato	Giorno e ora in cui è stata effettuata l'operazione
Da_a	Privato	Identifica il terzo con cui si è fatta l'operazione
Listaprodotti e quantità	Privato	Contiene i prodotti riguardanti l'operazione e le rispettive quantità
Metodi		
Getter...	Pubblici	Preleva le informazioni della classe per tutti i suoi attributi
Setter...	Pubblici	Imposta le informazioni della classe per tutti i suoi attributi

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

3.1.2. Scarica e registra control

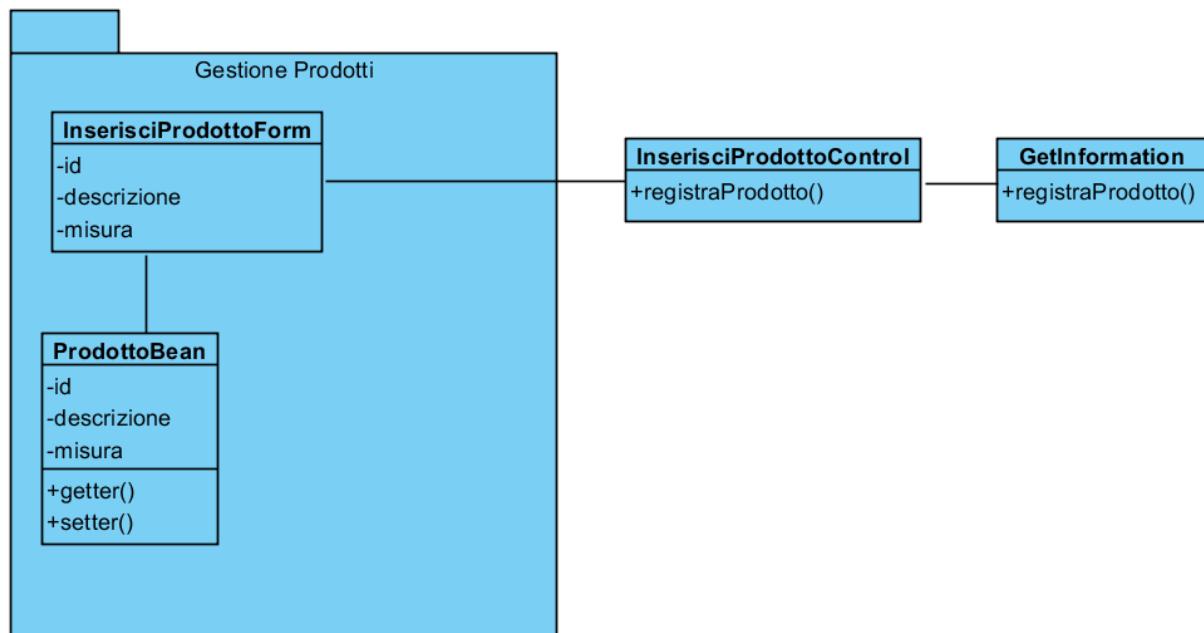
Classe: ScaricaERegistraControl	Autore: Team	
Descrizione	La classe che preso in input l’oggetto operazione bean si interfaccia con getInformation per registrare le informazioni come dati persistenti	
Dipendenza	OperazioneCompletataBean, GetInformation	
Metodi		
Nome	Accesso	Descrizione
scaricaMerce	Pubblico	Prende i dati in input e li invia al gestore dei dati persistenti per scriverli nel database

3.1.3. GetInformation

Classe: GetInformation	Autore: Team	
Descrizione	Classe che si interfaccia con il database e i control object. Gestisce i dati persistenti	
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
Metodi		
....	Pubblici	Contiene i metodi per l'interfacciamento con il database
registraScarico	Pubblico	Registra uno scarico nella tabella delle operazioni completate e riduce la disponibilità del magazzino

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

3.2. Inserisci prodotto (amministratore)



3.2.1. ProdottoBean

Classe: ProdottoBean	Autore: Team	
Descrizione	Contiene le informazioni riguardanti un prodotto	
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
Id	Privato	Contiene l'identificativo del prodotto
Descrizione	Privato	Contiene la descrizione del prodotto
Misura	Privato	Contiene l'unità di misura del prodotto
Metodi		
Getter...	Pubblico	Preleva le informazioni della classe per tutti i suoi attributi
Setter...	Pubblico	Imposta le informazioni della classe per tutti i suoi attributi

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

3.2.2. Inserisci prodotto control

Classe: InserisciProdottoControl	Autore: Team	
Descrizione	La classe che preso in input l'oggetto prodottobean si interfaccia con <code>getInformation</code> per registrare le informazioni come dati persistenti	
Dipendenza	ProdottoBean, GetInformation	
Attributi		
Nome	Accesso	Descrizione
Metodi		
registraProdotto	Pubblico	Prende i dati in input e li invia al gestore dei dati persistenti per scriverli nel database

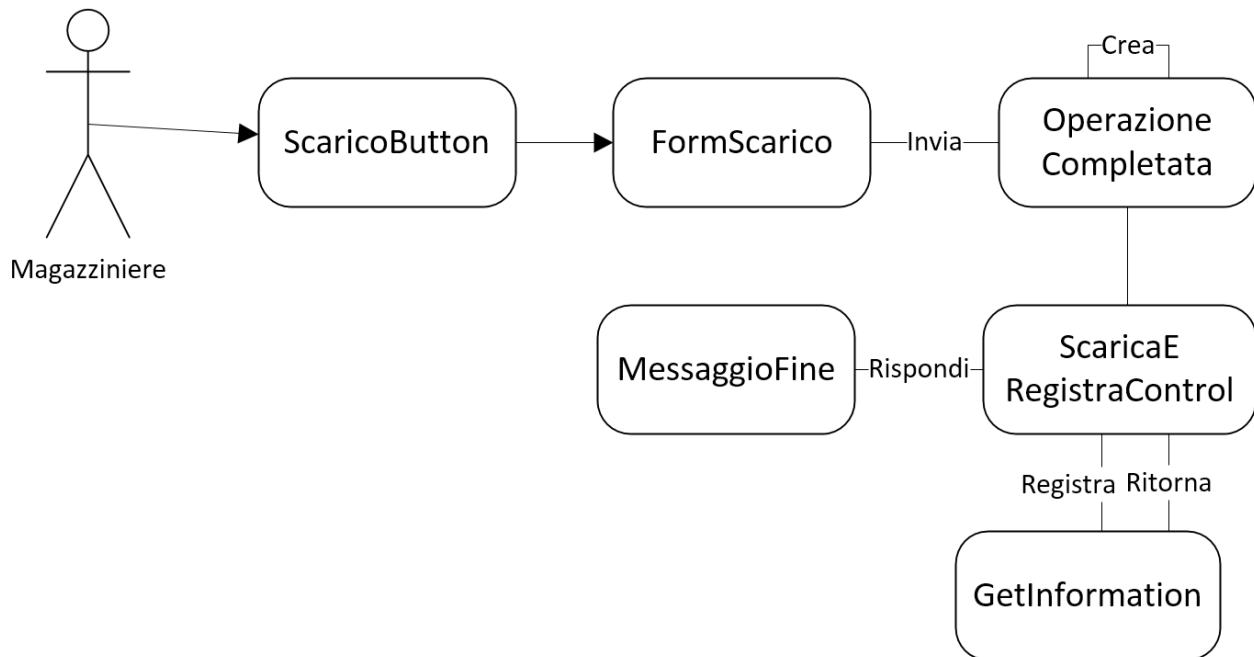
3.2.3. GetInformation

Classe: GetInformation	Autore: Team	
Descrizione	Classe che si interfaccia con il database e i control object. Gestisce i dati persistenti	
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
Metodi		
....	Pubblici	Contiene i metodi per l'interfacciamento con il database
registraProdotto	Pubblico	Registra un nuovo prodotto nel database

Progetto: GMASflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

4. Sequence diagram (low level)

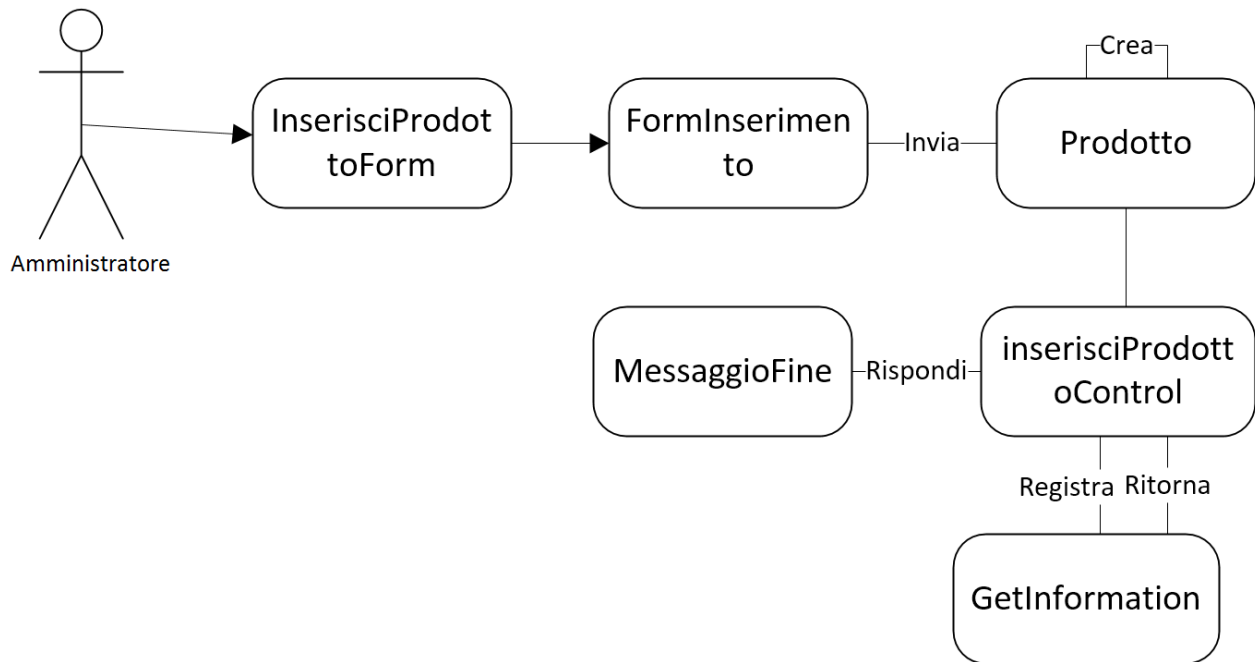
4.1. Registrazione scarico merce



1. L'utente chiede di scaricare delle merci e tramite il bottone richiede la funzionalità
2. L'utente completa la form con i prodotti e le quantità
3. I dati vengono inviati
4. Viene istanziato il nuovo oggetto
5. Passato al control per la registrazione
6. L'oggetto control chiede a get information di registrare
7. Attende una risposta
8. Notifica l'utente con un messaggio

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

4.2. Registrazione prodotto



1. L'utente chiede di registrare un nuovo prodotto e tramite il bottone richiede la funzionalità
2. L'utente completa la form con i prodotti e le quantità
3. I dati vengono inviati
4. Viene istanziato il nuovo oggetto
5. Passato al control per la registrazione
6. L'oggetto control chiede a get information di registrare
7. Attende una risposta
8. Notifica l'utente con un messaggio

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

5. Database

5.1. DBMS

La scelta è ricaduta su MySql poiché è utilizzato da molte industrie; inoltre è gratuito e ha molte feature tra cui la replicazione dei dati. In più consente di gestire l'accesso simultanea grazie alla sua struttura multithread.

5.2. Descrizione database

5.2.1. Magazzino

idMag chiave primaria
 descrizioneM
 città
 via
 cap
 nCivico
 passwordM
 tipo
 Accesso: root

5.2.2. Prodotto

idProduct chiave primaria
 unità di misura
 descrizioneP

5.2.3. Disponibilità

IdMag chiave primaria, chiave esterna
 idProduct chiave primaria, chiave esterna
 quantità

5.2.4. Operazioni completate

idOperazione chiave primaria
 idM chiave esterna
 tipo
 data
 da_a chiave esterna

5.2.5. Composizione operazioni completate

idOperazione chiave primaria, chiave esterna
 idProduct chiave primaria, chiave esterna
 quantità

Progetto: GMAsflatiTuozzo	Versione: 1.0
Documento: <u>Object Design</u>	Data: 21/12/2016

5.2.6. Operazioni in sospeso

idOperazione chiave primaria

idM chiave esterna

tipo

data

da_a chiave esterna

5.2.7. Composizione operazioni in sospeso

idOperazione chiave primaria, chiave esterna

idProduct chiave primaria, chiave esterna

quantità