

Gestione grossista Il Girasole
Object Design Document
Versione 1.3



Data: 19/12/2017

Progetto: Grossista Il Girasole	Versione: 1.3
Documento: Object Design Document	Data: 19/12/2017

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Silvia Castelli	0512103054
Valeria Monzillo	0512103042

Scritto da:	Silvia Castelli e Valeria Monzillo
--------------------	------------------------------------

Revision History

Data	Versione	Descrizione	Autore
11/12/2017	1.0	Scrittura punto 1 e 2	Castelli/Monzillo
12/12/2017	1.1	Scrittura punto 3	Castelli/Monzillo
13/12/2017	1.2	Scrittura punto 4	Castelli/Monzillo
19/12/2017	1.3	Revisione	Castelli/Monzillo

Indice

1.	INTRODUZIONE	4
1.1.	Scelte progettuali per l'Object Design	4
1.2.	Linee guida nella documentazione delle interfacce	4
1.3.	Definizioni, Acronimi e Abbreviazioni	5
1.4.	Riferimenti	5
2.	Packages	5
2.1.	Funzioni Amministratore	6
2.2.	Funzioni Cliente	6
2.3.	Funzioni Dipendente	6
3.	Interfaccia delle classi	6
3.1.	Inserisci Dipendente(Ammministratore)	6
3.1.1.	DipendenteBean	7
3.1.2.	InserisciDipendenteControl	7
3.1.3.	GetInformation	8
3.2.	Inserisci prodotto(Dipendente)	8
3.2.1.	ProductBean	9
3.2.2.	InserisciProdottoControl	9
3.2.3.	GetInformation	9
3.3.	Acquisto Prodotti(Cliente)	10
3.3.1.	ComposizioneVenditaBean	10
3.3.2.	ControlAcquista	11
3.3.3.	GetInformation	11
4.	Sequence Diagram (low level)	12
1.1.	Aggiunta prodotto	12
4.1.	Acquisto prodotto	13
4.2.	Aggiungi dipendente	14
5.	Database	15
5.1.	Descrizione database	15
5.1.1.	Prodotto	15
5.1.2.	Dipendente	15
5.1.3.	Cliente	15
5.1.4.	Fattura	15
1.1.1.	Acquistati	16

1. INTRODUZIONE

1.1. Scelte progettuali per l'Object Design

In seguito alla stesura dei precedenti documenti, RAD e SSD, sono stati definiti a grosse linee le caratteristiche e le funzionalità del nostro progetto.

Quali:

- ***Prestazioni vs Costi***

Il sistema è stato realizzato usufruendo di materiale open source, in modo da prediligere l'efficienza del sistema minimizzando i costi. In particolar modo il server web locale da noi utilizzato è Apache Tomcat, fornito gratuitamente da Apache Software Foundation.

- ***Interfaccia vs Tempo di risposta***

Il software è pensato per garantire tempi di risposta rapidi tra il server e l'interfaccia utente, per poter soddisfare i diversi utenti collegati al sistema. Di conseguenza, la grandezza del database sarà influenzata dal tempo di risposta e dall'inserimento e la ricerca al suo interno.

- ***Costi vs Mantenimento***

Basato su risorse open source, il software non richiederà elevati costi di mantenimento. È stato pensato per ridurre la complessità nello sviluppo e semplificare al meglio l'uso da parte degli utenti, sia clienti, sia dipendenti e sia amministratore.

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.

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 metodo con le eventuali eccezioni relative.

Il sistema è pensato per essere accessibile sia ai dipendenti, sia ai clienti e sia all'amministratore.

I dipendenti potranno visualizzare tutto il catalogo dei prodotti, e se necessario, inserire o

eliminare un prodotto dell'azienda in base alla disponibilità. I clienti potranno visualizzare tutta la merce disponibile e scegliere quella di loro interesse. Per l'acquisto basterà recarsi nell'apposita interfaccia "Carrello" e procedere all'ordine.

L'amministratore sarà invece responsabile della gestione dei dipendenti e delle loro credenziali per accedere al sito. Avrà piena visione dei dipendenti presenti in azienda, potendo inserire o eliminare uno di essi se necessario. Avrà, inoltre, il compito di modificare lo stato degli ordini effettuati dai singoli clienti.

Tutti i tipi di utenti, quindi, utilizzeranno query per interagire con il database, ma attraverso l'utilizzo delle form verranno semplificate per essere accessibili anche per dai 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 Il Girasole
- 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
- **ODD:** Object Design Document
- **DB:** DataBase
- **DBMS:** DataBase Management System

1.4. Riferimenti

I riferimenti si riconducono ai documenti stilati in precedenza: RAD, SDD.

2. Packages

Questa sezione descrive la decomposizione dei sottosistemi in pacchetti e l'organizzazione dei file di codice. Allo stesso tempo fornisce una panoramica di ogni pacchetto, le sue dipendenze con altri pacchetti e il utilizzo previsto per esso

2.1. Funzioni Amministratore

Gestione dipendenti: Gestisce l'organizzazione dei dipendenti. Permette l'inserimento o la rimozione di un dipendente all'interno del sistema.

Visualizza catalogo dipendenti: Permette di visualizzare l'elenco dei dipendenti facenti parte dell'azienda.

Gestione stato ordini: Consente di modificare lo stato di ordine effettuato dal cliente.

Visualizza ordini: Consente all'amministratore di visualizzare tutti gli ordini che sono stati effettuati dai clienti, con i dati ad esso relativi.

2.2. Funzioni Cliente

Visualizza catalogo prodotti: Permette al dipendente di visualizzare tutti i prodotti disponibili in azienda.

Gestione prodotti: Gestisce le operazioni relative ai prodotti, permettendo al dipendente di inserire o rimuovere un prodotto in base alle disponibilità dell'azienda.

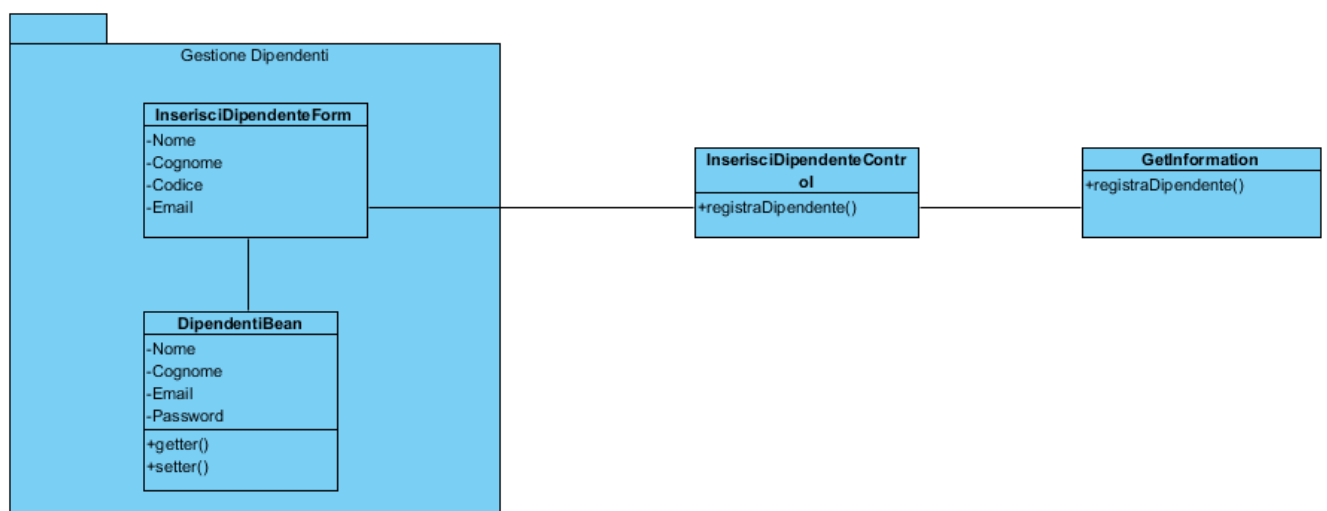
2.3. Funzioni Dipendente

Visualizza prodotti: Visualizza l'offerta dei prodotti presenti in azienda.

Gestione carrello: Potrà acquistare o eliminare i prodotti che sono stati inseriti in precedenza all'interno del carrello.

3. Interfaccia delle classi

3.1. Inserisci Dipendente (Amministratore)



3.1.1. DipendenteBean

Classe: DipendentiBean	Autore: Team
Descrizione	Contiene le informazioni riguardanti i dipendenti
Dipendenza	

Attributi		
Nome	Accesso	Descrizione
Nome	Privato	Contiene il nome del dipendente
Cognome	Privato	Contiene il cognome del dipendente
Codice	Privato	Contiene il codice identificativo del dipendente
Email	Privato	Contiene l'email associata al dipendente
Password	Privato	Contiene la password associata al dipendente
Metodi		
Getter...	Pubblico	Preleva le informazioni della classe per tutti i suoi attributi
Setter...	Pubblico	Imposta le informazione della classe per tutti i suoi attributi.

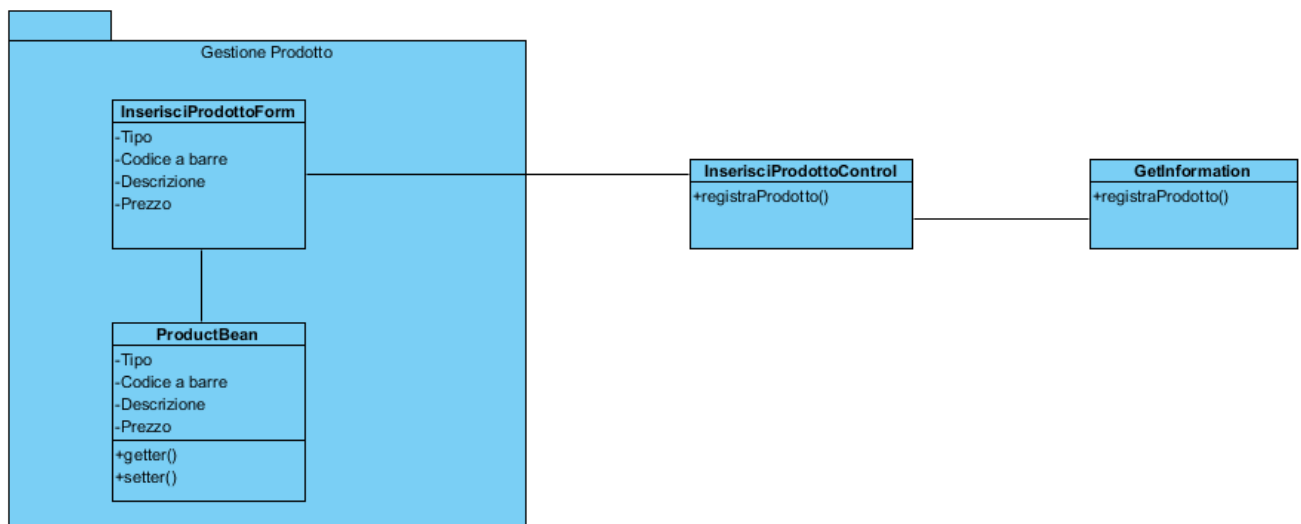
3.1.2. InserisciDipendenteControl

Classe: InserisciDipendenteControl		Autore: Team
Descrizione		La classe che preso in input l’oggetto DipendenteBean si interfaccia con getInformation per registrare le informazioni come dati persistenti
Dipendenza		DipendenteBean,GetInformation
Attributi		
Nome	Accesso	Descrizione
Metodi		
registraDipendente	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 e gestisce i dati persistenti
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
Metodi		
...	Pubblico	Contiene i metodi per l'interfacciamento del database.
registraDipendente	Pubblico	Registra un nuovo dipendente nel database

3.2. Inserisci prodotto(Dipendente)



3.2.1. ProductBean

Classe: ProductBean		Autore: Team
Descrizione		Contiene le informazioni riguardanti i prodotti
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
Tipo	Privato	Contiene il tipo di prodotto
Codice a barre	Privato	Contiene il codice a barre del prodotto
Descrizione	Privato	Contiene la descrizione del prodotto
Prezzo	Privato	Contiene il prezzo 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.

3.2.2. InserisciProdottoControl

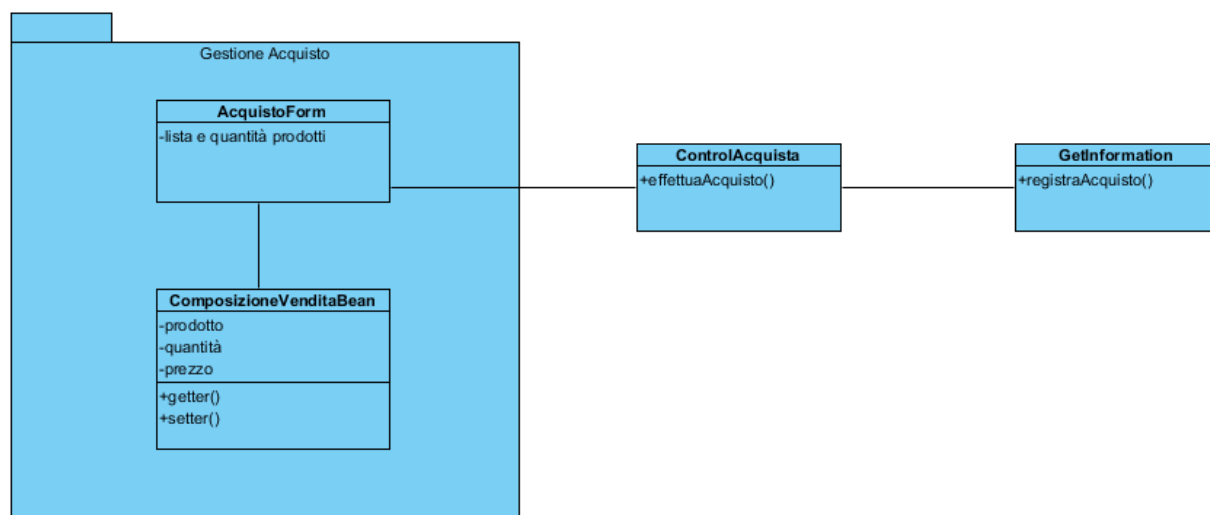
Classe: InserisciProdottoControl		Autore: Team
Descrizione		La classe che preso in input l'oggetto ProductBean si interfaccia con getInformation per registrare le informazioni come dati persistenti
Dipendenza		ProductBean,GetInformation
Attributi		
Nome	Accesso	Descrizione
Metodi		
registraProdotto	Pubblico	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 e gestisce i dati persistenti
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
Metodi		
...	Pubblico	Contiene i metodi per l'interfacciamento del database.

registraDipendente	Pubblico	Registra un nuovo dipendente nel database
---------------------------	----------	---

3.3.Acquisto Prodotti(Cliente)



3.3.1. ComposizioneVenditaBean

Classe: ComposizioneVenditaBean		Autore: Team
Descrizione		Contiene le informazioni riguardanti l'acquisto dei prodotti
Dipendenza		
Attributi		
Nome	Accesso	Descrizione
Prodotti	Privato	Contiene il nome dei prodotti che si sta acquistando
Quantità	Privato	Contiene la quantità del prodotto che si sta acquistando
Prezzo	Privato	Contiene il prezzo dei prodotti
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.

3.3.2. ControlAcquista

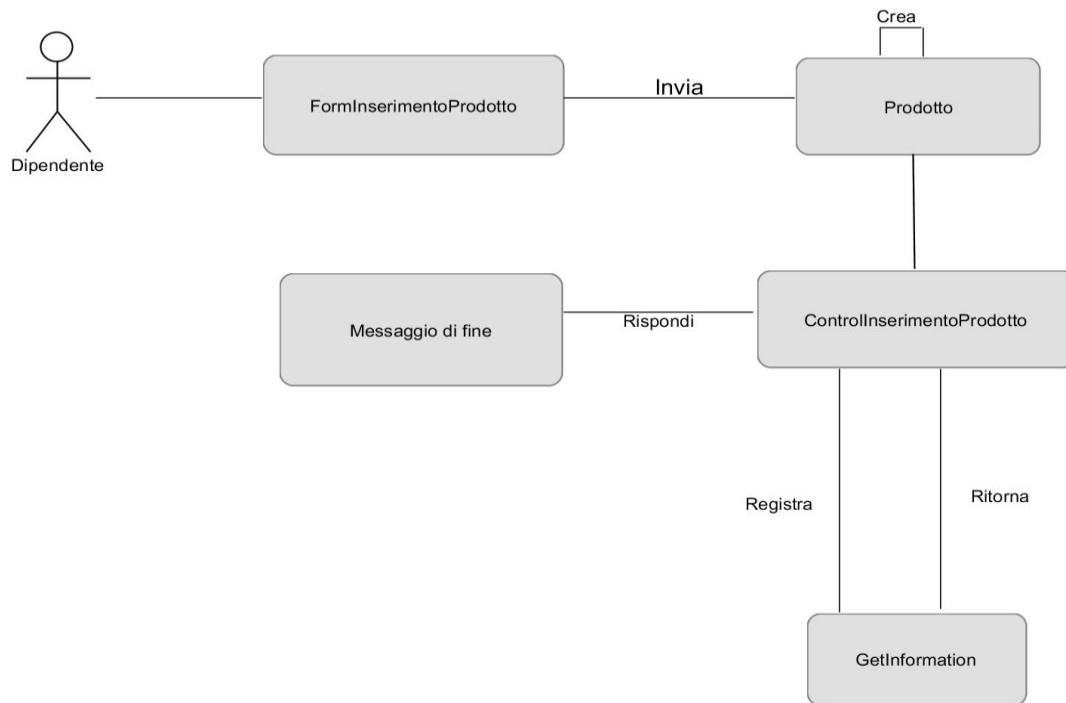
Classe: ControlAcquista		Autore: Team
Descrizione		La classe che preso in input l'oggetto CompraVenditaBean si interfaccia con GetInformation per registrare le informazioni come dati persistenti
Dipendenza		CompraVenditaBean, DBInformation
Attributi		
Nome	Accesso	Descrizione
Metodi		
effettuaAcquisto	Pubblico	Prende i dati in input e li invia al gestore dei dati persistenti per scriverli nel database

3.3.3. GetInformation

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

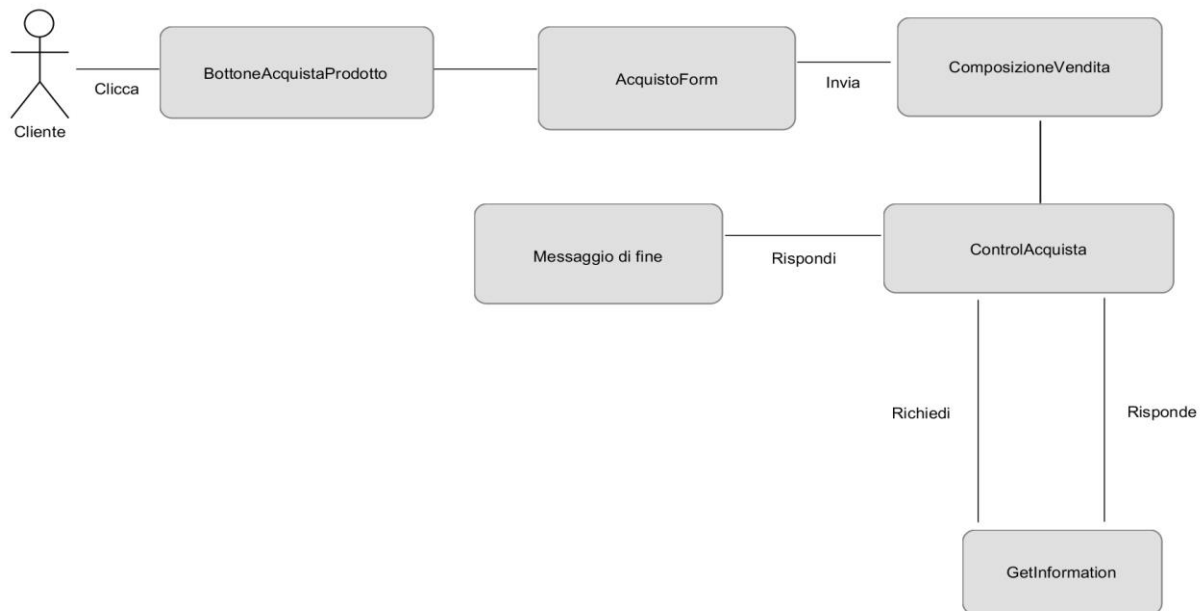
4. Sequence Diagram (low level)

1.1. Aggiunta prodotto



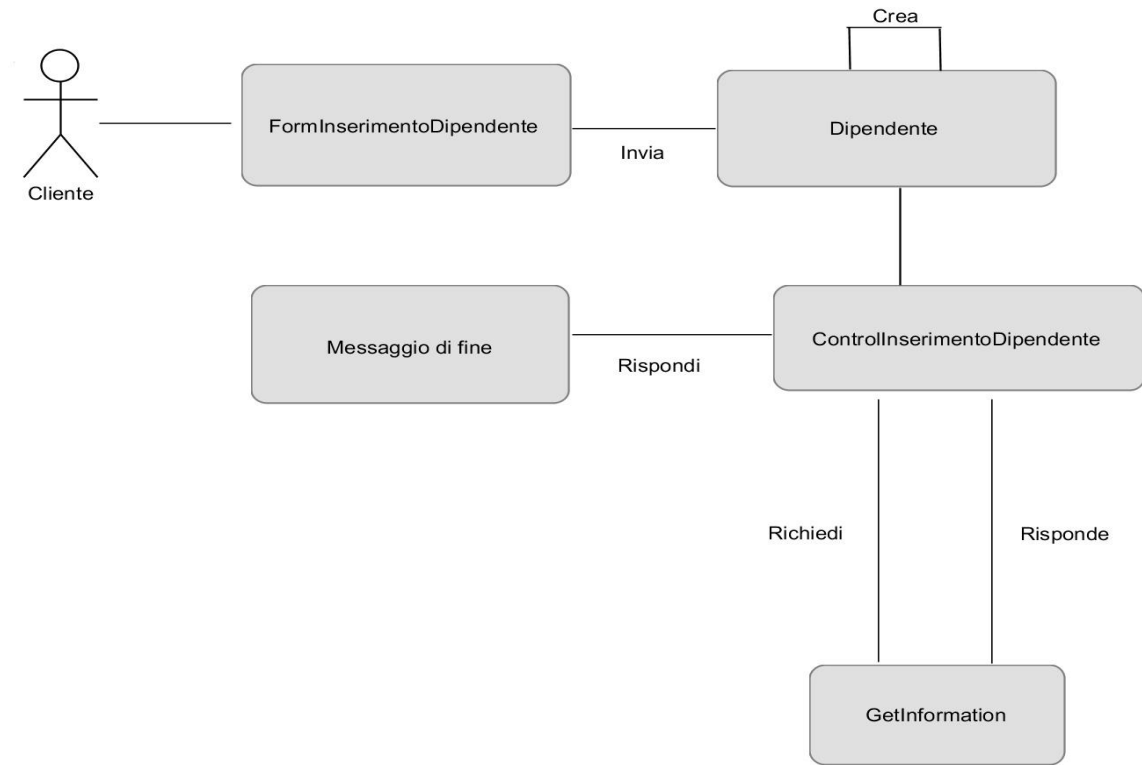
1. L'utente chiede di registrare un prodotto
2. L'utente inserisce i dati nella form
3. I dati vengono inviati
4. Viene instaziato un nuovo oggetto
5. L'oggetto è passato al control
6. Il control chiede al GetInformation di registrarlo
7. Attende una risposta
8. Notifica l'utente con un messaggio

4.1. Acquisto prodotto



1. L'utente clicca sul bottone "Acquista"
2. Vengono inviati i dati al control
3. Il controlAcquisra verifica i dati ricevuti
4. Invia una richiesta al GetInformation
5. Attende una risposta
6. Notifica l'utente con un messaggio

4.2. Aggiungi dipendente



9. L'utente chiede di registrare un dipendente
10. L'utente inserisce i dati nella form
11. I dati vengono inviati
12. Viene instaziato un nuovo oggetto
13. L'oggetto è passato al control
14. Il control chiede al GetInformation di registrarlo
15. Attende una risposta
16. Notifica l'utente con un messaggio

5. Database

Per la creazione e gestione del database viene utilizzato MySQL.

MySQL si presta bene come DBMS poiché è gratuito e quindi è anche il più utilizzato tra le aziende. Inoltre questo DBMS presenta una tecnologia multi-thread che permette agli utenti di accedervi in contemporeanea.

5.1. Descrizione database

5.1.1. Prodotto

Chiave primaria: code

Prezzo

Descrizione

Tipo_prodotto

5.1.2. Dipendente

Chiave primaria: cod

Nome

Cognome

Email

Password

Tipo_d

5.1.3. Cliente

Chiave primaria: codice

Tipo_cliente

Nome

Cognome

Via

Città

Nome_negozio

Partita_IVA

Email

Password

5.1.4. Fattura

Chiave primaria: CodiceVendita

Ora

Totale

C_codice

Data

Stato

1.1.1. Acquistati

Chiave esterna: Codice Vendita

Chiave primaria e esterna: Code

Prezzo

Quantità