

Università degli Studi di Salerno

Corso di Ingegneria del Software

Auto Shop Object Design Versione 1.0



Data: 19/12/2017

Cognome Nome	Matricola
Corrado Mancino Alfredo	0512102506
Carbè Daniele	0512102326
Caloia Gennaro	0512102332

Indice

1. Introduzione

- 1.1. Obiettivi di design trade-off
- 1.2. Linee guida per l'interfaccia
- 1.3 Definizioni, acronimi e abbreviazioni
- 1.4 Riferimenti

2. Packages

- 2.1 Package model
- 2.2 Package control

3. Interfacce delle classi

- 3.1 Classe Dipendente
- 3.2 Classe Auto
- 3.3 Classe Preventivo
- 3.4 Classe RicambiAuto
- 3.5 Classe Ordine
- 3.6 Classe Carrello
- 3.7 Classe Cliente
- 3.8 Classe ProdcutModelAutomobile
- 3.9 Classe ProductModelCarrello
- 3.10 Classe ProcutModelDipendente
- 3.11 Classe ProdcutModelOrdine
- 3.12 Classe ProductModelRifornimento
- 3.13 Classe ProductModelRicambi
- 3.14 Classe ControlRicambi
- 3.15 Classe ControlDipendente
- 3.16 Classe ControlRegister
- 3.17 Classe ConnectionDB
- 3.18 Classe ControlOrdine
- 3.19 Classe ControlRifornimento
- 3.20 Classe ControlAuto

1. Introduzione

1.1 Obiettivi di design Trade-off

Dopo la realizzazione del documento RAD (Requirement Analysis Document) e SDD (System Design Document), abbiamo descritto in linea di massima, quello che sarà il nostro sistema e quindi i nostri obiettivi, tralasciando gli aspetti implementativi.

Prestazioni vs. Costi

Considerando il sistema che stiamo realizzando, possiamo dire che il non eccessivo budget a nostra disposizione ci ha consentito di realizzare il prodotto utilizzando materiale open source partendo da zero minimizzando così i costi e rendendo il sistema più che soddisfacente.

Interfaccia vs. Tempo di risposta

Il tempo di risposta tra server e interfaccia sono più che sufficienti (rapidi) a soddisfare le esigenze dei vari dipendenti e clienti collegati al Sistema.

Ovviamente maggiore sarà la grandezza del database e maggiore sarà il tempo di risposta e ricerca nel database.

Interfaccia vs. Easy-use

L'interfaccia, grazie a form e ad una impostazione semplice e intuitiva, permette un facile utilizzo del sistema.

Costi vs. Mantenimento

Grazie a un uso di materiale open source e l'utilizzo del linguaggio javadoc il sistema può essere facilmente modificato, implementato con nuove funzioni o corretto in presenza di errori.

1.2 Linee guida per l'interfaccia

Il sistema è multi-utente (può accedervi chiunque, sia un semplice cliente o un dipendente). Al cliente, il sistema nasconde la logica delle operazioni, fornendogli solamente la consultazione del catalogo dei prodotti e la possibilità di acquistarli.

Il dipendente invece deve avere accesso all'interfaccia magazzino e gestione dipendenti.

Il dipendente riceve i risultati delle richieste effettuate dal sistema, senza preoccuparsi delle operazioni e di come queste sono state effettuate. Avvenuta l'operazione di vendita di uno o più prodotti il sistema decrementa dal DataBase totale il numero delle quantità vendute del prodotto acquistato (aggiorna quindi il DataBase totale).

1.3 Definizioni, acronimi e abbreviazioni

Non ci sono acronimi, definizioni e abbreviazioni.

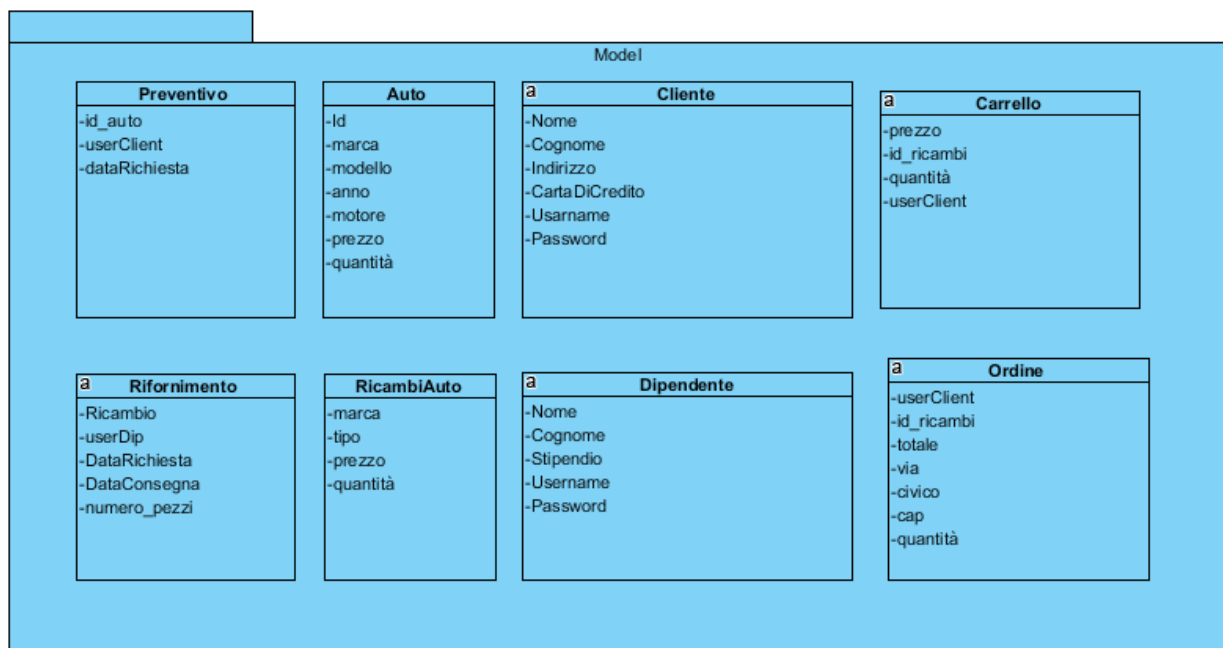
1.4 Riferimenti

- Sommerville, Software Engineering (6th edition, 2001), Addison Wesley.

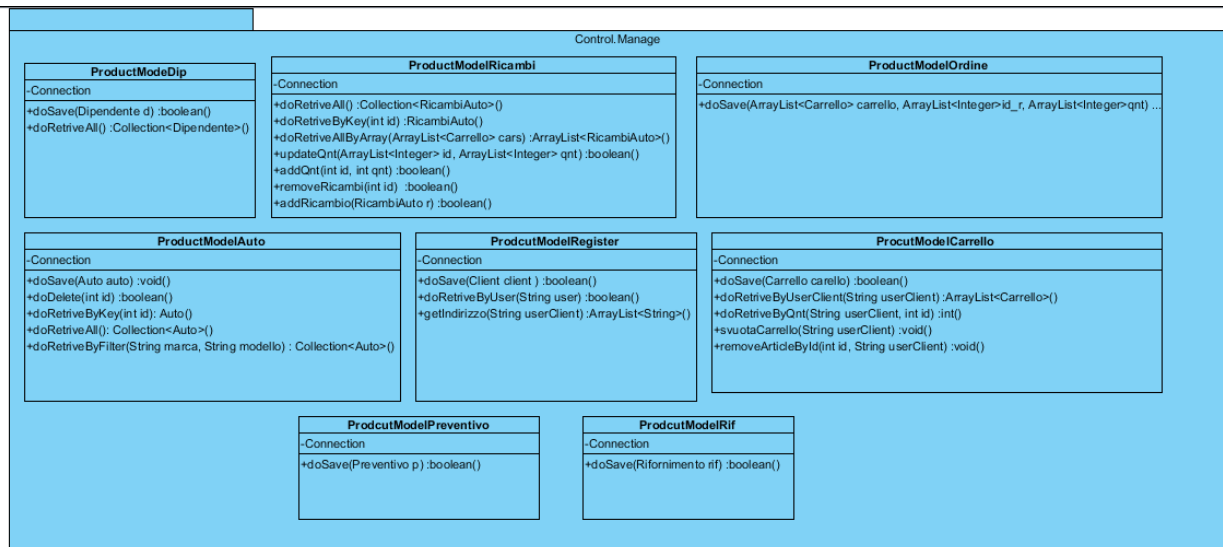
- Roger S. Pressman, Principi di Ingegneria del Software (terza edizione, 2000), Mc Graw Hill Italia.

2. Packages

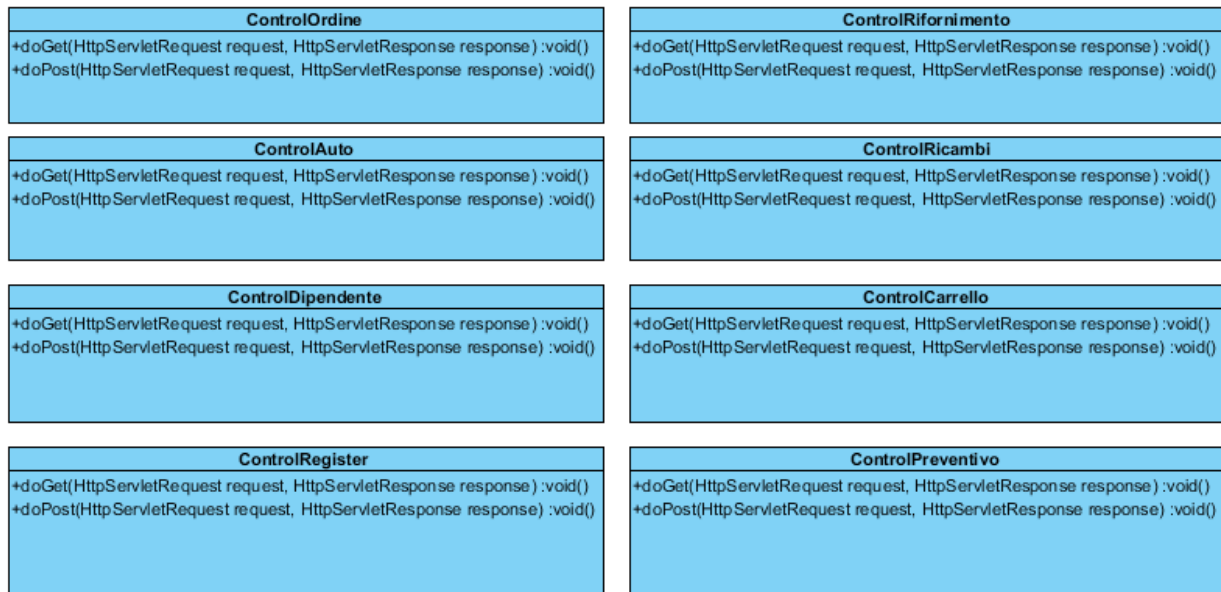
2.1 Package Model



2.2 Package Control.Manage



2.3 Package Control.Servlet



3. Interfacce delle classi

3.1 Classe Dipendente

Nome Classe	Dipendente
Descrizione	Questa classe rappresenta l'astrazione di un dipendente
Pre-Condizioni	<p>context Dipendente : : getNome() pre: Nome != null</p> <p>context Dipendente : : setNome(Nome) pre: Nome != null</p> <p>context Dipendente : : getCognome() pre: Cognome != null</p> <p>context Dipendente : : setCognome(Cognome) pre: Cognome != null</p> <p>context Dipendente : : getId() pre: Id > 0</p> <p>context Dipendente : : setId(Id) pre: Id > 0</p> <p>context Dipendente : : getStipendio() pre: Stipendio != null</p> <p>context Dipendente : : setStipendio(Stipendio) pre: 1000 < Stipendio < 2000</p> <p>context Dipendente : : getUsername() pre: Username != null context Dipendente : : setUsername(Username) pre: Username != null</p> <p>context Dipendente : : getPassword() pre: Password != null</p> <p>context Dipendente : : setPassword>Password) pre: Password != null</p>
	<p>context Dipendente : : setNome(Nome) post: Nome != null</p> <p>context Dipendente : : setCognome(Cognome) post: Cognome != null</p>

Post-Condizioni	<p>context Dipendente : : setId(Id) post: Id > 0</p> <p>context Dipendente : : setStipendio(Stipendio) post: 1000 < Stipendio < 2000</p> <p>context Dipendente : : setUsername(Username) post: Username != null</p> <p>context Dipendente : : setPassword>Password) post: Password != null</p>
Invarianti	<p>context Dipendente inv: Nome != null and Cognome != null and CodiceFiscale != null and Id > 0 and 1000 < Stipendio < 2000 and Username != null and Password != null</p>

3.2 Classe Auto

Nome Classe	Auto
Descrizione	Questa classe rappresenta l'astrazione di un'automobile.
Pre-Condizioni	<p>context Automobile : : getMarca() pre: Marca != null</p> <p>context Automobile : : setMarca(Marca) pre: Marca != null</p> <p>context Automobile : : getModello() pre: Modello != null</p> <p>context Automobile : : setModello(Modello) pre: Modello != null</p> <p>context Automobile : : getAnno() pre: Anno != null</p> <p>context Automobile : : setAnno(Anno) pre: Anno != null</p> <p>context Automobile : : getIdAutomobile() pre: IdAutomobile > 0</p> <p>context Automobile : :</p>

	<p>setIdAutomobile(IdAutomobile) pre: IdAutomobile > 0</p> <p>context Automobile : : getPrezzo() pre: Prezzo > 0</p> <p>context Automobile : : setPrezzo(Prezzo) pre: Prezzo > 0 context Automobile : : getQuantità() pre: Quantità > 0</p> <p>context Automobile : : setQuantità(Quantità) pre: Quantità > 0</p>
Post-Condizioni	<p>context Automobile : : setMarca(Marca) post: Marca != null</p> <p>context Automobile : : setModello(Modello) post: Modello != null</p> <p>context Automobile : : setAnno(Anno) post: Anno != null</p> <p>context Automobile : : setIdAutomobile(IdAutomobile) post: IdAutomobile > 0</p> <p>context Automobile : : setPrezzo(Prezzo) post: Prezzo > 0</p> <p>context Automobile : : setQuantità(Quantità) post: Quantità > 0</p>
Invarianti	<p>context Automobile inv: Marca != null and Modello != null and Anno != null and IdAutomobile > 0 and Prezzo > 0 and Quantità > 0</p>

3.3 Preventivo

Nome Classe	Preventivo
Descrizione	Questa classe rappresenta l'astrazione di un'automobile.
	<p>context Preventivo : : setId_auto(int id) pre: id>0</p> <p>context Preventivo : : getId_auto() pre: id>0</p>

Pre-Condizioni	<p>context Preventivo : : getUserClient() pre: UserClient != null</p> <p>context Preventivo : : setUserClient(UserClient) pre: UserClient != null</p> <p>context Preventivo : : getData() pre: data != null</p> <p>context Preventivo : : setData(Data d) pre: data != null</p>
Post-Condizioni	<p>context Preventivo : : setMarca(Marca) post: Marca != null</p> <p>context Preventivo : : setId_auto(int id) post: id>0</p> <p>context Preventivo : : getId_auto() post: id>0</p> <p>context Preventivo : : getUserClient() post: UserClient != null</p> <p>context Preventivo : : setUserClient(UserClient) post: UserClient != null</p>
Invarianti	context Automobile inv : Marca != null and Modello != null and Anno != null and IdAutomobile > 0 and Prezzo > 0 and Quantità > 0

3.4 Classe RicambiAuto

Nome Classe	RicambiAuto
Descrizione	Questa classe rappresenta l'astrazione di un ricambio.
	<p>context Ricambi : : getTipo() pre: Tipo != null</p> <p>context Ricambi : : setTipo(String tipo) pre: Tipo != null</p>

Pre-Condizioni	context Ricambi : : getModello() pre: marca != null context Ricambi : : setMarca(String marca) pre: marca != null
Post-Condizioni	context Ricambi : : setTipo(String tipo) post: Tipo != null context Ricambi : : setMarca(String marca) post: marca != null
Invarianti	context Ricambi inv : Marca != null and Modello != null and Anno != null and IdAutomobile > 0 and Prezzo > 0 and Quantità > 0 and Tipo != null and Modello != null

3.5 Classe Ordine

Nome Classe	Ordine
Descrizione	Questa classe rappresenta l'astrazione di un ordine.
Pre-Condizioni	context Ordine : : getData() pre: Data != null context Ordine : : setData(Data) pre: Data != null context Ordine : : getIdOrdine() pre: IdOrdine > 0 context Ordine : : setIdOrdine(IdOrdine) pre: IdOrdine > 0 context Ordine : : getTotale() pre: Totale > 0 context Ordine : : setTotale(Totale) pre: Totale > 0 context Ordine : : getIdCliente() pre: IdCliente > 0 context Ordine : : setIdCliente(IdCliente) pre: IdCliente > 0 context Ordine : : getCap() pre: Cap != null

	<p>context Ordine : : setCap(Cap) pre: Cap != null</p> <p>context Ordine : : getIndirizzoConsegna() pre: IndirizzoConsegna != null</p> <p>context Ordine : : setIndirizzoConsegna(IndirizzoConsegna) pre: IndirizzoConsegna != null</p>
Post-Condizioni	<p>context Ordine : : setData(Data) post: Data != null</p> <p>context Ordine : : setTotale(Totale) post: Totale > 0</p> <p>context Ordine : : setIdCliente(IdCliente) post: IdCliente > 0</p> <p>context Ordine : : setCap(Cap) post: Cap != null</p> <p>context Ordine : : setIndirizzoConsegna(IndirizzoConsegna) post: IndirizzoConsegna != null</p>
Invarianti	<p>context inv: Data != null and MetodoPagamento != null and Stato != null and IdOrdine > 0 and Totale > 0 and IdCliente > 0 and IndirizzoConsegna != null and Cap != null</p>

3.6 Classe Carrello

Nome Classe	Carrello
Descrizione	Questa classe rappresenta l'astrazione di un carrello.
Pre-	<p>context Carrello : : getSpedizione() pre: Spedizione != null</p> <p>context Carrello : : setSpedizione(Spedizione) pre: Spedizione != null</p> <p>context Carrello : : getTotale() pre: Totale != null</p> <p>context Carrello : : setTotale(Totale) pre: Totale != null</p>

Condizioni	<p>context Carrello : : getListaRicambi() pre: ListaRicambi != null</p> <p>context Carrello : : delete() pre:</p> <p>context Carrello : : addItem(Ricambio) pre: Ricambio != null and ListaRicambi != null</p> <p>context Carrello : : deleteItem(Ricambio) pre: Ricambio != null and ListaRicambi != null</p> <p>context Carrello : : Modify(Ricambio) pre: Ricambio != null and ListaRicambi != null</p>
Post-Condizioni	<p>context Carrello : : setSpedizione(Spedizione) post: Spedizione != null</p> <p>context Carrello : : setTotale(Totale) post: Totale != null</p> <p>context Carrello : : delete() post: ListaRicambi vuota and Totale = 0</p> <p>context Carrello : : addItem(Ricambio) post: Ricambio in ListaRicambi</p> <p>context Carrello : : deleteItem(Ricambio) post: Ricambio not in ListaRicambi</p>
Invarianti	context Carrello inv : ListaRicambi != null and Totale != null and Spedizione != null

3.7 Classe Cliente

Nome Classe	Cliente
Descrizione	Questa classe rappresenta l'astrazione di un cliente.

Pre-Condizioni	<p>context Cliente : : getNome() pre: Nome != null</p> <p>context Cliente : : setNome(Nome) pre: Nome != null</p> <p>context Cliente : : getCognome() pre: Cognome != null</p> <p>context Cliente : : setCognome(Cognome) pre: Cognome != null</p> <p>context Cliente : : getIdCliente() pre: IdCliente != null</p> <p>context Cliente : : setIdCliente(IdCliente) pre: IdCliente != null</p> <p>context Cliente : : getEmail() pre: Email != null</p> <p>context Cliente : : setEmail(Email) pre: Email != null</p> <p>context Cliente : : getRecapito() pre: Recapito != null</p> <p>context Cliente : : setRecapito(Recapito) pre: Recapito != null</p>
Post-Condizioni	<p>context Cliente : : setNome(Nome) post: Nome != null</p> <p>context Cliente : : setCognome(Cognome) post: Cognome != null</p> <p>context Cliente : : setIdCliente(IdCliente) post: IdCliente != null</p> <p>context Cliente : : setEmail(Email) post: Email != null</p> <p>context Cliente : : setRecapito(Recapito) post: Recapito != null</p>
Invarianti	<p>context Cliente inv: Nome != null and Cognome != null</p>

	null and IdCliente != null and Email != null and Recapito != null
--	---

3.8 Classe ProductModelAutomobile

Nome Classe	ProductModelAutomobile
Descrizione	Questa classe rappresenta l'astrazione della gestione delle automobili.
Pre-Condizioni	<p>context ManageAutomobile : : insertAutomobile(Automobile) pre: Automobile != null</p> <p>context ManageAutomobile : : deleteAutomobile(idAutomobile) pre: idAutomobile != null</p> <p>context ManageAutomobile : : updateMarca(idAutomobile, Marca) pre: idAutomobile != null and Nome != null</p> <p>context ManageAutomobile : : updateModello(idAutomobile, Modello) pre: idAutomobile != null and Modello != null</p> <p>context ManageAutomobile : : updatePrezzo(idAutomobile, Prezzo) pre: idAutomobile != null and Prezzo != null</p> <p>context ManageAutomobile : : updateQuantita(idAutomobile, Quantita) pre: idAutomobile != null and Quantita != null</p> <p>context ManageAutomobile : : getCategory(category) pre: category != null</p> <p>context ManageAutomobile : : getQuantita(idAutomobile) pre: idAutomobile != null</p>

	context ManageAutomobile : : getAutomobile(idAutomobile) pre: idAutomobile != null
Post-Condizioni	
Invarianti	context ManageAutomobile inv :

3.9 Classe ProductModelCarrello

Nome Classe	ProductModelCarrello
Descrizione	Questa classe rappresenta l'astrazione della gestione di un carrello.
Pre-Condizioni	context ManageCarrello : : saveCart(carrello, DatiCliente, i) pre: carrello != null and DatiCliente != null and i != null
Post-Condizioni	
Invarianti	context ManageCarrello inv :

3.10 Classe ProductModelDipendente

Nome Classe	ProductModelDipendente
Descrizione	Questa classe rappresenta l'astrazione della gestione dei dipendenti.
	context ManageDipendente : : getDipendenti() pre: context ManageDipendente : : insertDipendente(Dipendente) pre: Dipendente != null context ManageDipendente : : deleteDipendente(idDipendente) pre: idDipendente != null context ManageDipendente : : updateNome(idDipendente, Nome) pre: idDipendente

Pre- Condizion i	<p>!= null and Nome != null</p> <p>context ManageDipendente : : updateCognome(idDipendente, Cognome) pre: idDipendente != null and Cognome != null</p> <p>context ManageDipendente : : updateCodiceFiscale(idDipendente,CodiceFiscale) pre: Dipendente != null and CF!= null</p> <p>context ManageDipendente : : updateStipendio(idDipendente, Stipendio) pre: idDipendente != null and Stipendio != null</p> <p>context ManageDipendente : : updateTipo(idDipendente, Tipo) pre: idDipendente != null and Tipo != null</p> <p>context ManageDipendente : : updateUsername(idDipendente, Username) pre: idDipendente != null and Username != null</p> <p>context ManageDipendente : : updatePassword(idDipendente, Password) pre: idDipendente != null and Password != null</p> <p>context ManageDipendente : : getDipendente(idDipendente) pre: idDipendente != null</p> <p>context ManageDipendente : : getDipendente(Username, Password) pre: Username != null and Password != null</p> <p>context ManageDipendente : : checkLogin(Username, Password) pre: Username != null and Password != null</p> <p>context ManageDipendente : : getStipendiP(Data) pre: Data != null</p>
	<p>context ManageDipendente : : getDipendente(idDipendente) post: Dipendente !=</p>

Post-Condizioni	null
	context ManageDipendente : : getDipendente(Uname, Password) post: Dipendente != null
	context ManageDipendente : : checkLogin(Uname, Password) post: tipo != null
	context ManageDipendente : : getStipendiP(Data) post: listaDipendenti != null
Invarianti	context ManageDipendente inv :

3.11 Classe ProductModelOrdine

Nome Classe	ProductModelOrdine
Descrizione	Questa classe rappresenta l'astrazione della gestione degli ordini
Pre-Condizioni	context ManageOrdine : : getOrdini() pre: context ManageOrdine : : getInvece() pre: context ManageOrdine : : deleteOrdine(Ordine) pre: Ordine != null context ManageOrdine : updateIdCliente(idOrdine, idCliente) pre: idOrdine != null and idCliente != null context ManageOrdine : updateIndirizzoConsegna(idOrdine, IndirizzoConsegna) pre: idOrdine != null and IndirizzoConsegna != null context ManageOrdine : updateTotale(idOrdine, Totale) pre: idOrdine != null and Totale != null

	<p>context ManageOrdine : updateMetodoPagamento(idOrdine, MetodoPagamento) pre: idOrdine != null and MetodoPagamento != null</p> <p>context ManageOrdine : updateStato(idOrdine, Stato) pre: idOrdine != null and Stato != null</p> <p>context ManageOrdine :getI() pre:</p> <p>context ManageOrdine : evadi(idOrdine) pre: idOrdine != null</p> <p>context ManageOrdine : getOrdine(idOrdine) pre: idOrdine != null</p> <p>context ManageOrdine : dettagliA(idOrdine) pre: idOrdine != null</p> <p>context ManageOrdine : dettagliC(idCliente) pre: idCliente != null</p>
Post-Condizioni	<p>context ManageOrdine : getOrdine(idOrdine) pre: Ordine != null</p> <p>context ManageOrdine : dettagliA(idOrdine) pre: ListaArticoli != null</p> <p>context ManageOrdine : dettagliC(idCliente) pre: Cliente != null</p> <p>context ManageOrdine :getI() post: I != null</p> <p>context ManageOrdine : : getOrdini() post: ListaOrdini != null</p> <p>context ManageOrdine : : getInevasi() post: ListaInevasi != null</p>
Invarianti	context ManageOrdine inv:

3.12 Classe ProductModelRifornimento

Nome Classe	ProdcutModelRifornimento
Descrizione	Questa classe rappresenta l'astrazione della gestione delle richieste di rifornimento.
Pre-Condizioni	<p>context ManageRifornimento : : getRifornimenti() pre:</p> <p>context ManageRifornimento : : insertRifornimento(Rifornimento) pre: Rifornimento != null</p> <p>context ManageRifornimento : : insertComposizione(idRifornimento, Idtshirt, pezzi) pre: idRifornimento != null and Idtshirt != null and pezzi > 0</p> <p>context ManageRifornimento : : deleteRifornimento(Rifornimento) pre: Rifornimento != null</p> <p>context ManageRifornimento : : updateDataEffettuazione(dataEffettuazione, idRifornimento)pre: idRifornimento != null and dataEffettuazione != null</p> <p>context ManageRifornimento : : updateDataRichiesta(dataRichiesta, idRifornimento)pre: idRifornimento != null and dataRichiesta != null</p> <p>context ManageRifornimento : : updateDataConsegna(dataConsegna, idRifornimento)pre: idRifornimento != null and dataConsegna!= null</p> <p>context ManageRifornimento : : evadi(idRifornimento) pre: idRifornimento != null</p> <p>context ManageRifornimento : : getlnevasi() pre:</p> <p>context ManageRifornimento : :</p>

	insertRifornimento(Rifornimento,Stato) pre: Rifornimento != null and Stato != null
Post-Condizioni	context ManageRifornimento : : getInevasi() post: ListaInevasi!=null context ManageRifornimento : : getRifornimenti() post: ListaRifornimenti!=null
Invarianti	context ManageRifornimento inv :

3.13 Classe ProductModelRicambi

Nome Classe	ProductModelRicambi
Descrizione	Questa classe rappresenta l'astrazione della gestione dei ricambi
Pre-Condizioni	<p>context ManageRicambi: : getRicambi() pre:</p> <p>context ManageRicambi : : insertRicambio(Ricambio) pre: Ricambio != null</p> <p>context ManageRicambi : : deleteRicambio(idRicambio) pre: idRicambio != null</p> <p>context ManageRicambi : : updateTipo(idRicambio, tipo) pre: idRicambio != null and tipo != null</p> <p>context ManageRicambi : : updateModello(idRicambio, modello) pre: idRicambio != null and modello != null</p> <p>context ManageRicambi : : updatePeso(idRicambio, peso) pre: idRicambio != null and peso != null</p> <p>context ManageRicambi : : updateLunghezza(idRicambio, lunghezza) pre: idRicambio != null and lunghezza != null</p> <p>context ManageRicambi : : updateLarghezza(idRicambio, larghezza) pre: idRicambio != null and larghezza != null</p>

	context ManageRicambi : : getIdRicambio(idRicambio, tipo, modello, peso, lunghezza, larghezza) pre: idRicambio != null and tipo != null and modello != null and peso != null and lunghezza != null and larghezza != null context ManageRicambi : : getRicambi(idRicambio) pre: idRicambio != null
Post-Condizioni	context ManageRicambi : : getRicambi() post: ListaRicambi != null context ManageRicambi : : getIdRicambi(idRicambio) post: ListaRicambi != null
Invarianti	context ManageRicambi inv :

3.14 Classe ControlRicambi

Nome Classe	ControlRicambi
Descrizione	Questa classe si occupa della gestione dei ricambi.
Pre-Condizioni	context ServletRicambi : : doGet(request, response) pre: request != null and response != null context ServletRicambi : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	context ServletRicambi : : doGet(request, response) pre: request != null and response != null context ServletRicambi : : doPost(request, response) pre: request != null and response != null
Invarianti	context ServletRicambi inv :

3.15 Classe ControlDipendente

Nome Classe	<u>ControlDipendente</u>
Descrizione	Questa classe si occupa dell'inserimento, rimozione e modifica dei dipendenti

Pre-Condizioni	context GestisciDipendente : : doGet(request, response) pre: request != null and response != null context GestisciDipendente : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	context GestisciDipendente : : doGet(request, response) pre: request != null and response != null context GestisciDipendente : : doPost(request, response) pre: request != null and response != null
Invarianti	context GestisciDipendente inv :

3.16 Classe ControlRegister

Nome Classe	ControlRegister
Descrizione	Questa classe si occupa della gestione dei dipendenti.
Pre-Condizioni	context ServletLogin : : doGet(request, response) pre: request != null and response != null context ServletLogin : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	context ServletLogin : : doGet(request, response) pre: request != null and response != null context ServletLogin : : doPost(request, response) pre: request != null and response != null
Invarianti	context ServletLogin inv :

3.17 Classe ConnectionDB

Nome Classe	ConnectionDB
Descrizione	Questa classe si occupa della connessione con il database.
	context DbConnect : : connect() pre:

Pre-Condizioni	context DbConnect : : close() pre: Connection != null context DbConnect : : getConnection() pre: Connection != null
Post-Condizioni	context DbConnect : : getConnection() post: Connection != null
Invarianti	context DbConnect inv :

3.18 Classe ControlOrdine

Nome Classe	ControlOrdine
Descrizione	Questa classe si occupa della gestione degli ordini.
Pre-Condizioni	context GestisciOrdine : : doGet(request, response) pre: request != null and response != null context GestisciOrdine : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	context GestisciOrdine : : doGet(request, response) pre: request != null and response != null context GestisciOrdine : : doPost(request, response) pre: request != null and response != null
Invarianti	context GestisciOrdine inv :

3.19 Classe ControlRifornimento

Nome Classe	ControlRifornimento
Descrizione	Questa classe si occupa della gestione dei rifornimenti.
Pre-Condizioni	context ServletRifornimento : : doGet(request, response) pre: request != null and response != null context ServletRifornimento : : doPost(request, response) pre: request != null and response != null
Post-	context ServletRifornimento : : doGet(request,

Condizioni	response) pre: request != null and response != null context ServletRifornimento : : doPost(request, response) pre: request != null and response != null
Invarianti	context ServletRifornimento inv : GestoreRifornimenti != null

3.20 Classe ControlAuto

Nome Classe	ControlAuto
Descrizione	Questa classe si occupa di prelevare le auto già esistenti nel db
Pre-Condizioni	context ServletInsertAutomobile : : doGet(request, response) pre: request != null and response != null context ServletInsertAutomobile : : doPost(request, response) pre: request != null and response != null
Post-Condizioni	context ServletInsertAutomobile : : doGet(request, response) pre: request != null and response != null context ServletInsertAutomobile : : doPost(request, response) pre: request != null and response != null
Invarianti	context ServletInsertAutomobile inv :