# 

# Entity

|  |  |
| --- | --- |
| Nome Classe | UtenteBean |
| Descrizione | Rappresenta l’oggetto utente |
| Firma dei Metodi | + getPass() : String  + getRole() :void  + getEmail() : String  + getFirstName() : String  + getLastName(): String  + getBirthday(): Date  +setPass (pass: String)  +setRole (role: int)  +setEmail (email: String)  +setFirstName (firstName: String)  +setLastName(String: lastName)  +setBirthday (birthday: Date) |
| Pre-condizioni | **Context** UtenteBean:: setRole(role)  Pre: role==0 OR role==1  **Context** UtenteBean::setEmail(email)  Pre: l’email non deve essere già presente nel DB |
| Post-condizioni | **Context** UtenteBean::setRole(role)  Post: l’utente è standard o admin  **Context** UtenteBean::setEmail(email)  Post: l’email è nel DB |
| Invariante | **Context** UtenteBean  Inv: role==0 or role==1 |

|  |  |
| --- | --- |
| Nome Classe | OcchialeBean |
| Descrizione | Rappresenta l’oggetto occhiale |
| Firma dei Metodi | +getIdGlasses():String  +setIdGlasses(idGlasses:String)  +getNameGlasses():String  +setNameGlasses(nameGlasses:String)  +getBrand():String  +setBrand(brand:String)  +getPrice():int  +setPrice(price:int)  +getAvailability():int  +setAvailability(availability:int)  +getType():String  +setType(type:String)  +getColor():String  +setColor(color:String)  +getCategory():String  +setCategory(category:String)  +getImage():String  +setImage(image:String)  +getImage2():String  +setImage2(image2:String)  +getDescription():String  +setDescription(description:String)  +getQuantity():int  +setQuantity(quantity:int)  +getTotPrezzo():float  +setTotPrezzo(prezzotot:float) |
| Pre-condizioni | **Context** OcchialeBean::setIdGlasses(idOcchiale)  Pre: idOcchiale non deve essere presente nel DB  **Context** OcchialeBean::setQuantity (quantity)  Pre: quantity non deve essere negativa |
| Post-condizioni | **Context** OcchialeBean::setIdGlasses(idOcchiale)  Post: l’idOcchiale è presente nel DB |
| Invariante | **Context** OcchialeBean  Inv: self.quantity>=0 |

|  |  |
| --- | --- |
| Nome Classe | OrdineBean |
| Descrizione | Rappresenta l’oggetto ordine |
| Firma dei Metodi | +getIdOrder():UUID  +setIdOrder(idOrder:UUID)  +getDate():Date  +getStato():String  +setStato(s:String)  +setDate(date:Date)  +getEmail():String  +getTot():int  +setTot(t:int)  +setEmail(email:String) |
| Pre-condizioni | **Context** OrdineBean::setStato(stato)  Pre: stato==”confermato” or stato==”in elaborazione” or stato==”spedito”  **Context** OrdineBean::setIdOrder(idOrder)  Pre: idOrder non deve essere già presente in DB  **Context** OrdineBean::setEmail(email:String)  Pre: email deve avere una corrispondeza nel DB come chiave di un UtenteBean |
| Post-condizioni | **Context** OrdineBean::setIdOrder(idOrder)  Post: idOrder è presente in DB  **Context** OrdineBean::setEmail(email:String)  Post: email corrisponde ad un utente presente nel DB |
| Invariante | **Context** OrdineBean  Inv: self.stato==”confermato” or self.stato==”in elaborazione” or self.stato==”spedito” |

|  |  |
| --- | --- |
| Nome Classe | OcchialiOrdineBean |
| Descrizione | Rappresenta i vari occhiali presenti in un Ordine |
| Firma dei Metodi | +getIdOcchialeOrdine():int  +getIdOrdine():UUID  +getOcchiale ():OcchialeBean  +getIdocchiale ():String  +setIdOcchiale (id:String)  +getPrezzoEffettivo():float  +getIva():float  +getQuantita():int  +setIdOcchialeOrdine(idOcchialeOrdine:int)  +setIdOrdine(idOrdine:UUID)  + setOcchiale (occhiale:OcchialeBean)  +setPrezzoEffettivo(prezzoEffettivo:int)  +setQuantita(quantita:int)  +setPaymentType(type: String)  +getPaymentType(): String |
| Pre-condizioni | **Context** OcchialeOrdineBean::setIdOcchialeOrdine(idOcchialeOrdine)  Pre: idOcchialeOrdine non deve essere già presente nel DB  **Context** OcchialeOrdineBean::setIdOcchiale (idOcchiale)  Pre: idOcchiale deve avere una corrispondeza nel DB come chiave di un OcchialeBean  **Context** OcchialeOrdineBean::setIdOrdine (idOrdine)  Pre: idOrdine deve avere una corrispondeza nel DB come chiave di un OrdineBean |
| Post-condizioni | **Context** OcchialeOrdineBean::setIdOcchialeOrdine(idOcchialeOrdine)  Post: idOcchialeOrdine presente nel DB  **Context** OcchialeOrdineBean::setIdOcchiale (idOcchiale)  Post: idOcchiale ha una corrispondeza nel DB come chiave di un OcchialeBean  **Context** OcchialeOrdineBean::setIdOrdine (idOrdine)  Post: idOrdine ha una corrispondeza nel DB come chiave di un OrdineBean |
| Invariante | **Context** OcchialeOrdineBean  Inv: self.prezzoEffettivo>=0 |

|  |  |
| --- | --- |
| Nome Classe | IndirizzoBean |
| Descrizione | Rappresenta l’oggetto indirizzo |
| Firma dei Metodi | +getNome(): String  +setNome(nome: String)  +getCognome(): String  +setCognome(cognome: String)  +getIdIndirizzo():int  +setIdIndirizzo(idIndirizzo:int)  +getAddress():String  +setAddress(address:String)  +getStatus():int  +setStatus(i:int)  +getCity():String  +setCity(city:String)  +getProvince():String  +setProvince(province:String)  +getCap():int  +setCap(cap:int)  +getEmail():String  +setEmail(email:String)  +getTelefono():String  +setTelefono(telefono:String) |
| Pre-condizioni | **Context** IndirizzoBean::setIdIndirizzo(idIndirizzo)  Pre: idIndirizzo non è già presente nel DB  **Context** IndirizzoBean::setEmail (email)  Pre: email deve avere una corrispondeza nel DB come chiave di un UtenteBean |
| Post-condizioni | **Context** IndirizzoBean::setIdIndirizzo(idIndirizzo)  Post: idIndirizzo è presente nel DB  **Context** IndirizzoBean::setEmail (email)  Post: email ha una corrispondeza nel DB come chiave di un UtenteBean |
| Invariante | **Context** IndirizzoBean  Inv: self.Status==1 or self.Status==1 |

|  |  |
| --- | --- |
| Nome Classe | Carrello |
| Descrizione | Rappresenta l’oggetto carrello |
| Firma dei Metodi | +addCarrello(prod:OcchialeBean)  +deleteProduct(prod:OcchialeBean)  +prendiProdotto(id:String):OcchialeBean  +delete()  +getCarrello():ArrayList<OcchialeBean>  +searchProdotto(code:String):boolean  +getDimensione():int  +getPrezzoTotale(quant:int,id:String):float  +calcolaPrezzoTotale(id:String):float  +getPrezzoTotale():float  +modificaQuantita(id:String,quant:int)  +setTotale(t:float)  +getTotale():float  +getIva():float  +setIva(i:float)  +getPrezzoEffettivo():float  +setPrezzoEffettivo(p:float) |
| Pre-condizioni | **Context** Carrello::prendiProdotto(idOcchiale)  Pre: idOcchiale deve essere presente nel DB  **Context** Carrello::modificaQuantità (idOcchiale,quantità)  Pre: idOcchiale deve essere presente nel DB e quantità deve esser >=0 |
| Post-condizioni | **Context** Carrello::prendiProdotto(idOcchiale)  Post: Un OcchialeBean viene utilizzato |
| Invariante | **Context** Carrello  Inv: self.dimensioneCarrello>0 |

|  |  |
| --- | --- |
| Nome Classe | TipologiaPagamentoBean |
| Descrizione | Rappresenta l’oggetto Pagamento |
| Firma dei Metodi | +getIdPaymentType():int  +setIdPaymentType(idPaymentType:int)  +getPaymentName():String  +setPaymentName(paymentName:String)  +getIdOrder()  +setIdOrder(idOrder: UUID) |
| Pre-condizioni | **Context** TipologiaPagamentoBean::setIdPaymentType |
| Post-condizioni | **Context** TipologiaPagamentoBean:: |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | SpedizioneBean |
| Descrizione | Rappresenta l’oggetto Spedizione |
| Firma dei Metodi | +getIdShipment():int  +setIdShipment(idShipment:int)  +getDateShipment():Date  +setDateShipment(dateShipment:Date)  +getIdOrder():int  +setIdOrder(idOrder:int)  +getShipmentType():int  +setShipmentType(shipmentType:int) |
| Pre-condizioni | **Context** SpedizioneBean::setIdShipment (idShipment)  Pre: idShipment deve essere presente nel DB  **Context** SpedizioneBean::setIdOrder (idOrder)  Pre: idOrder deve avere una corrispondeza nel DB come chiave di un Ordine |
| Post-condizioni | **Context** SpedizioneBean::setIdShipment (idShipment)  Post: idShipment è presente nel DB  **Context** SpedizioneBean::setIdOrder (idOrder)  Post: idOrder ha una corrispondeza nel DB come chiave di un Ordine |
| Invariante | **Context** SpedizioneBean  Inv: self.shipmentTyep=0 or self.shipmentTyep=1 |

Manager

|  |  |
| --- | --- |
| Nome Classe | Utente Dao |
| Descrizione | Rappresenta il manager che si occupa di interagire con il DB gestendo le query. |
| Firma dei Metodi | +doRetrieveByKey(keys:ArrayList<String>):UtenteBean  +esisteEmail(email:String):boolean  +doRetrieveByMail(email:String):UtenteBean  +cercaIndirizzo(email:String):IndirizziBean  +doRetrieveAll(order:String):Collection<UtenteBean>  +changePassword(email:String, pass:String)  +doDelete(bean:UtenteBean)  +doSave(utente:UtenteBean) |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | OcchialeDao |
| Descrizione | Rappresenta il manager che si occupa di interagire con il DB gestendo le query. |
| Firma dei Metodi | +doRetrieveByKey(keys:ArrayList<String>):Collection<OcchialeBean>  +doRetrieveBySex(sex:String):Collection<OcchialeBean>  +doRetrieveOcchiale (id:String): OcchialeBean  +singleProduct( keys:ArrayList<String>):OcchialeBean  +doRetrieveAll():Collection<OcchialeBean>  +doRetrieveByBrand(brand:String):Collection<OcchialeBean>  +decreaseAvailability(occhiale: OcchialeBean):void  +doSave(occhiale:OcchialeBean):void  +doDelete(occhiale:OcchialeBean):void  +doUpdate(occhiale:OcchialeBean):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | IndirizziDao |
| Descrizione | Rappresenta il manager che si occupa di interagire con il DB gestendo le query. |
| Firma dei Metodi | +doRetrieveActive(email: String): IndirizziBean  +search(email: String, via: String): IndirizziBean  +doRetrieveAllAddress(email: String): Collection<IndirizziBean>  +doRetrieveAll(order: String): Collection<IndirizziBean>  +doUpdate(indirizzo: IndirizziBean): void  +doDelete(indirizzo: IndirizziBean): void  +doSave(indirizzo: IndirizziBean):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | OrdineDao |
| Descrizione | Rappresenta il manager che si occupa di interagire con il DB gestendo le query. |
| Firma dei Metodi | +doSave(ordine:OrdineBean):void  +doRetrieveByKey(idOrdine:String):OrdineBean  +doRetrieveByUser(email:String):ArrayList<OrdineBean>  +doRetrieveAll(order:String):Collection<OrdineBean>  +doUpdate(ordine:OrdineBean):void  +doDelete(ordine:OrdineBean):void  +doRetriveByUser(user:UtenteBean,order:String):ArrayList<OrdineBean>  +doRetriveByDate(init:Date, end:Date, skip:int, limit:int):ArrayList<OrdineBean> |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | OcchialeOrdineDao |
| Descrizione | Rappresenta il manager che si occupa di interagire con il DB gestendo le query. |
| Firma dei Metodi | +doSave(occhialeOrdine:OcchialeOrdineBean):void  +doDelete(idOcchialeOrdine:String):boolean  +doRetrieveByKey(idOcchialeOrdine:String):OcchialeOrdineBean  +doRetrieveAll(order:String):Collection<OcchialeOrdineBean>  +doRetrivebyOrder(ordine:String,data:DataSource):ArrayList<OcchialeOrdineBean> |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | SpedizioneDao |
| Descrizione | Rappresenta il manager che si occupa di interagire con il DB gestendo le query. |
| Firma dei Metodi | +doRetrieveByKey(keys:ArrayList<String>):SpedizioneBean  +doRetrieveAll(order:String):Collection<SpedizioneBean>  +doUpdate(spedizione:SpedizioneBean):void  +doDelete(bean:SpedizioneBean):void  +doSave(spedizione:SpedizioneBean):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | TipologiaPagamentoDao |
| Descrizione | Rappresenta il manager che si occupa di interagire con il DB gestendo le query. |
| Firma dei Metodi | +doRetrieveByKey(keys:ArrayList<String>):TipologiaPagamentoBean  +doRetrieveAll(order:String):Collection< TipologiaPagamentoBean >  +doUpdate(bean: TipologiaPagamentoBean):void  +doDelete(bean: TipologiaPagamentoBean):void  +doSave(bean: TipologiaPagamentoBean):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

CONTROL

|  |  |
| --- | --- |
| Nome Classe | LoginServlet |
| Descrizione | Questa classe è un control che si occupa di passare a UtenteDao i dati dell’utente usando la query di Retrieve |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  # doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | RegistServlet |
| Descrizione | Questa classe è un control che si occupa di passare a UtenteDao i dati dell’utente per inserire il nuovo utente |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  # doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | LogoutServlet |
| Descrizione | Questa classe è un control che si occupa di effettua il logout dell’utente dal sito |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  # doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | CategoryServlet |
| Descrizione | Questa classe è un control che si occupa della visualizzazione degli occhiali in base alla scelta dell’utente |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | Checkout |
| Descrizione | Questa classe è un control che si occupa di passare le informazioni di un acquisto a OrdineDao, OcchialeOrdineDao e OcchialeDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | CryptoPw |
| Descrizione | Questa classe è un control che si occupa di crittografare le password |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | EsisteEmail |
| Descrizione | Questa classe è un control che si occupa di passare la mail ad UtenteDao per vedere se esiste già in fase di registrazione |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ForgetPass |
| Descrizione | Questa classe è un control che si occupa di inserire una nuova password |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | IndirizzoServlet |
| Descrizione | Questa classe è un control che si occupa di recuperare gli indirizzi usando IndirizziDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | OcchialeControl |
| Descrizione | Questa classe è un control che si occupa di occupa di prelevare l’intero catalogo di Occhiali per visualizzarli nell’homepage |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |

|  |  |
| --- | --- |
| Nome Classe | Search |
| Descrizione | Questa classe è un control che permette di ottenere determinati occhiali passando a OcchialiDao il brand scelto dall’utente |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletAddAddress |
| Descrizione | Questa classe è un control che si occupa di passare un nuovo indirizzo a IndirizzoDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | SearchClienteAdmin |
| Descrizione | Questa classe è un control che si occupa di passare i dati di un utente da cercare a UtenteDao e ottenere i suoi ordini , interfacciandosi con OrdineDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ListaOrdiniDataAdmin |
| Descrizione | Questa classe è un control che si occupa di passare i dati inseriti dall’admin ad OrdineDao per visualizzare gli ordini di una certa data |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletAggiungiprodAdmin |
| Descrizione | Questa classe è un control che si occupa di passare i dati di un nuovo occhiale a OcchialeDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletAmministratore |
| Descrizione | Questa classe è un control che si occupa di restituire un catalogo di occhiali |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletCarrello |
| Descrizione | Questa classe è un control che si occupa di passare la quantità scelta dall’utente al Carrello |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletControlloAdmin |
| Descrizione | Questa classe è un control che si occupa di gestire l’apertura delle 3 pagine azione dell’Admin |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletDettagliOrdine |
| Descrizione | Questa classe è un control che si occupa di recuperare i dettagli da OrdineDao tramite OcchialeOrdineDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |
|  |  |

|  |  |
| --- | --- |
| Nome Classe | ServletEliminaProdAdmin |
| Descrizione | Questa classe è un control che si occupa di rimuovere passare l’id di un Occhiale a OcchialeDao, che eliminerà il prodotto dal DB |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletEliminaProdotto |
| Descrizione | Questa classe è un control che si occupa di eliminare un occhiale dal Carrello, scelto dall’utente |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletModificaAmministratore |
| Descrizione | Questa classe è un control che si occupa di passare i dati nuovi di un Occhiale esistente a OcchialeDao che farà la update dell’occhiale in questione |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletOrdine |
| Descrizione | Questa classe è un control che si occupa di restituire e visualizzare la lista di ordini tramite OrdineDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletProdotto |
| Descrizione | Questa classe è un control che si occupa di ottenere i dettagli di un occhiale passando l’id a OcchialeDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletProfile |
| Descrizione | Questa classe è un control che si occupa di ottenere le informazioni dell’utente loggato, usando UtenteDao |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ServletCercaCliente |
| Descrizione | Questa classe è un control che si occupa di passare l’email a OrdineDao per ottenere la lista dei suoi ordini |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe | ShopControl |
| Descrizione | Questa classe è un control che si occupa della visualizzazione dell’intero catalogo |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |

|  |  |
| --- | --- |
| Nome Classe |  |
| Descrizione | Questa classe è un control che si occupa di |
| Firma dei Metodi | # doPost(request: HttpServletRequest, response: HttpServletResponse):void  + doGet(request: HttpServletRequest, response: HttpServletResponse):void |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invariante |  |