

## UtenteRegistrato

- nome : String
- cognome : String
- email : String
- password : String
- genere : String
- username : String
- nascita : Date
- ruolo : String

- + getNome() : String
- + getCognome() : String
- + getEmail() : String
- + getPassword() : String
- + getGenere() : String
- + getUsername() : String
- + getNascita() : Date
- + getRuolo() : String
- + setName(nome : String) : void
- + setCognome(cognome : String) : void
- + setEmail(email : String) : void
- + setPassword(password : String) : void
- + setGenere(genere : String) : void
- + setUsername(username : String) : void
- + setNascita(data : Date) : void
- + setRuolo (ruolo : String) : void

**context** UtenteRegistrato::getNome()

**post:** result = nome

**context** UtenteRegistrato::getCognome()

**post:** result = cognome

**context** UtenteRegistrato::getEmail()

**post:** result = email

**context** UtenteRegistrato::getPassword()

**post:** result = password

**context** UtenteRegistrato::getGenere()

**post:** result = genere

**context** UtenteRegistrato::getUsername()

**post:** result = username

**context** UtenteRegistrato::getNascita()

**post:** result = nascita

**context** UtenteRegistrato::getRuolo()

**post:** result = ruolo

**context** UtenteRegistrato::setName(nome)

**post:** self.nome = nome

**context** UtenteRegistrato::setCognome(cognome)

**post:** self.cognome = cognome

**context** UtenteRegistrato::setEmail(email)

**post:** self.email = email

**context** UtenteRegistrato::setPassword(password)

**post:** self.password = password

**context** UtenteRegistrato::setGenere(genere)

**post:** self.genere = genere

**context** UtenteRegistrato::setUsername(username)

**post:** self.username = username

```

context UtenteRegistrato::setNascita(nascita)
    post: self.nascita = nascita
context UtenteRegistrato::setRuolo(ruolo)
    post: self.ruolo = ruolo

```

UtenteRegistratoModel
//
+ doRetrieveByKey(username : String) : UtenteRegistrato + doRetrieveAll(order : String) : Collection<UtenteRegistrato> + doSave(user : UtenteRegistrato) : void + doUpdate(user : UtenteRegistrato) : void + doDelete(user : UtenteRegistrato) : void

```

context UtenteRegistratoModel::doRetrieveByKey(username)
    pre: username != null
    post: result = database.UtenteRegistrato -> select(user | user.username = username)
context UtenteRegistratoModel::doRetrieveAll()
    post: result = database.UtenteRegistrato
context UtenteRegistratoModel::doSave(user)
    pre: user.nome != null and user.cognome != null and user.email != null
    and user.password != null and user.genere != null and user.username != null
    and user.nascita != null and user.ruolo != null
    post: database.UtenteRegistrato -> exists(u | u.username = user.username)
context UtenteRegistratoModel::doDelete(user)
    pre: database.UtenteRegistrato -> exists(u | u.username = user.username)
    post: not(database.UtenteRegistrato -> exists(u | u.username = user.username))

```

DatiSpedizione
- id : int - nomeRicevente : String - cognomeRicevente : String - email : String - cellulare : String - nCivico : int - via : String - city : String - paese : String - provincia : String - cap : String - cliente : UtenteRegistrato

## DatiSpedizione

```
+ getId() : int
+ getNomeRicevente() : String
+ getCognomeRicevente() : String
+ getEmail() : String
+ getCellulare() : String
+ getNCivico() : int
+ getVia() : String
+ getCity() : String
+ getPaese() : String
+ getProvincia() : String
+ getCap() : String
+ getCliente() : UtenteRegistrato
+ setId(id : int) : void
+ setNomeRicevente(nome : String) : void
+ setCognomeRicevente(cognome : String) : void
+ setEmail(email : String) : void
+ setCellulare(cellulare : String) : void
+ setNCivico(nCivico : int) : void
+ setVia(via : String) : void
+ setCity(city : String) : void
+ setPaese(paese : String) : void
+ setProvincia(provincia : String) : void
+ setCap(cap : String) : void
+ setCliente(cliente : UtenteRegistrato) : void
```

```
context DatiSpedizione::getId()
    post: result = id
context DatiSpedizione::getNomeRicevente()
    post: result = nomeRicevente
context DatiSpedizione::getCognomeRicevente()
    post: result = cognomeRicevente
context DatiSpedizione::getEmail()
    post: result = email
context DatiSpedizione::getCellulare()
    post: result = cellulare
context DatiSpedizione::getNCivico()
    post: result = nCivico
context DatiSpedizione::getVia()
    post: result = via
context DatiSpedizione::getCity()
    post: result = city
context DatiSpedizione::getPaese()
    post: result = paese
context DatiSpedizione::getProvincia()
    post: result = provincia
context DatiSpedizione::getCap()
    post: result = cap
context DatiSpedizione::getCliente()
    post: result = cliente
context DatiSpedizione::setId(id)
    post: self.id = id
context DatiSpedizione::setNomeRicevente(nomeRicevente)
    post: self.nomeRicevente = nomeRicevente
context DatiSpedizione::setCognomeRicevente(cognomeRicevente)
    post: self.cognomeRicevente = cognomeRicevente
```

```

context DatiSpedizione::setEmail(email)
    post: self.email = email
context DatiSpedizione::setCellulare(cellulare)
    post: self.cellulare = cellulare
context DatiSpedizione::setNCivico(nCivico)
    post: rself.nCivico = nCivico
context DatiSpedizione::setVia(via)
    post: self.via = via
context DatiSpedizione::setCity(city)
    post: self.city = city
context DatiSpedizione::setPaese(paese)
    post: self.paese = paese
context DatiSpedizione::setProvincia(provincia)
    post: self.provincia = provincia
context DatiSpedizione::setCap(cap)
    post: self.cap = cap
context DatiSpedizione::setCliente(cliente)
    post: self.cliente = cliente

```

DatiSpedizioneModel
//
+ doRetrieveByOrdine(ordine : Ordine) + doSave(dati : DatiSpedizione) : void + doDelete(dati: DatiSpedizione) : void

```

context DatiSpedizioneModel::doRetrieveByID(id)
    pre: id != null
    post: result = database.DatiSpedizione -> select(d | d.id = id)
context DatiSpedizioneModel::doSave(dati)
    pre: dati.nomeRicevente != null and dati.cognomeRicevente != null and dati.email != null
    and user.password != null and user.genere != null and user.username != null
    and dati.cellulare != null and dati.nCivico != null and dati.Via != null and dati.city != null
    and dati.Paese != null and dati.Provincia != null and dati.cap != null and dati.cliente != null
    post: database.DatiSpedizione -> exists(d | d = dati) and dati.id != null
context DatiSpedizioneModel::doDelete(dati)
    pre: dati.id != null and dati.cognomeRicevente
    post: not(database.DatiSpedizioneModel -> exists(d | d.id = dati.id))

```

CartItem
- prodotto : Prodotto - quantity : int

CartItem
+ getProdotto() : String + getQuantity() : String + setProdotto(prodotto : Prodotto) : void + setQuantity(quantità : int) : void + increaseQuantity() : void + decreaseQuantity() : void

```

context CartItem::getProdotto()
    post: result = prodotto
context CartItem::getQuantity()
    post: result = quantity
context CartItem::setProdotto(prodotto)
    post: self.prodotto = prodotto
context CartItem::setQuantity(quantity)
    post: self.quantity = quantity
context CartItem::increaseQuantity()
    post: getQuantity() = @pre.getQuantity() + 1
context CartItem::decreaseQuantity()
    post: getQuantity() = @pre.getQuantity() - 1

```

Carrello
- cartItems : ArrayList<CartItem>
+ getItems() : ArrayList<CartItem> + addItem(prodotto : Prodotto, quantity : int) : void + removeItem(prodotto : Prodotto) : void + getTotalPrice() : float

```

context Carrello::getItems()
    post: result = cartItems
context Carrello::addItem(prodotto)
    post: cartItems -> exists(c | c.prodotto.id = prodotto.id)
context Carrello::removeItem(prodotto)
    post: not(cartItems -> exists(c | c.prodotto.id = prodotto.id))
context Carrello::getTotalPrice()
    post: result = sum(cartItem.getProdotto().getPrice() | getItems() ->
includes(cartItem))

```

CarrelloModel
//
+ doRetrieveByCliente(cliente : UtenteRegistrato) : Carrello + doSave(carrello : Carrello, cliente : UtenteRegistrato) : void + doUpdate(carrello : Carrello, cliente : UtenteRegistrato) : void + doDelete(carrello : Carrello, cliente : UtenteRegistrato) : void

```

context CarrelloModel::doRetrieveByCliente(cliente)
    pre: cliente.username != null
    post: result = database.Carrello -> select(c | c.cliente = cliente.username)
// context CarrelloModel::doSave(carrello)

// context CarrelloModel::doUpdate(carrello)

```

Utility
//
+ checkNome(nome : String) : boolean + checkEmail(email : String) : boolean + checkPassword(password : String) : boolean

```

context Utility::checkNome(nome)
    post: result = True if isalpha(nome) else False
context Utility::checkEmail(email)
    post: result = True if email.match(x@x.x) else False
context Utility::checkPassword(password)
    post: result = True if [ len(password) >= 8 and containsUppercase(password)
        and containsLowercase(password) and containsNumber(password)
        and contains(password, "@ ! # $ % ' - / = ^ \ _ ` { } ~ +") ] else False

```