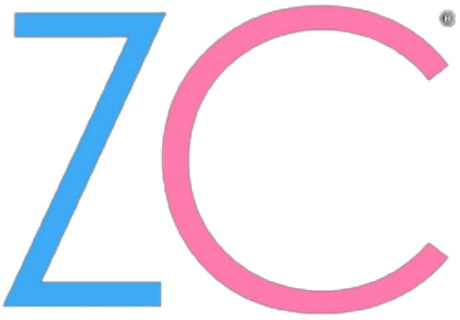
UNIVERITÀ DEGLI STUDI
DI SALERNO

Corso di Ingegneria del Software
Prof: Andrea De Lucia

ZoomClick Eventi



OBJECT DESIGN DOCUMENT

Matricola	*Partecipanti*
05121109837	De Maio Dario
05121112615	Santangelo Angelo

Versione	Autore	Descrizione
1.0	D.M.D. S.A.	Prima stesura dell'Object Design Document: Introduzione, trade-off
2.0	D.M.D.	linee guida per la documentazione delle interfacce, definizione, acronimi, riferimenti
2.0	S.A.	Organizzazione directory
2.1	S.A.	Glossario delle interfacce
3.0	D.M.D. S.A.	Revisione organizzazione directory
3.1	D.M.D.	OCL: classe cliente, gestore pacchetti
3.1	S.A.	OCL: classe party
3.2	D.M.D.	OCL: classe gestore impiegati, artista
3.2	S.A.	OCL: UserManager, PartyManager
3.3	D.M.D.	OCL: PacchettoManager
3.4	D.M.D S.A	Revisione totale documenti

INDICE

1. Introduzione

- ## Panoramica generale

  Nel documento di Object Design saranno spiegate le scelte finali prima dell'implementazione della piattaforma ZoomClickEventi.

  Questo documento non solo presenta i principali trade-off di progettazione, ma anche la specifica delle interfacce dei moduli e l'organizzazione delle classi relative alla porzione di sistema che verrà implementato.

- ## Trade-off

  Oltre ai trade-off presenti nel documento System Design Document, in questa fase ne vengono evidenziati ulteriori.

  1. ### Tempo di risposta vs Spazio occupato

     Onde evitare di dover effettuare un continuo accesso al database, per far sì che un utente possa visualizzare le proprie informazioni (ordini, impiegati dell'agenzia, pacchetti presenti nel catalogo), è stata prevista una struttura dati che si occuperà di mantenere in memoria tali dati.

  2. ### Interfaccia vs Usabilità

     L'interfaccia grafica deve essere molto chiara, deve far uso di pulsanti disposti in maniera da rendere semplice l'utilizzo del sistema.

  3. ### Prestazioni vs Costi

     Il sistema prevede l'uso di file di stile semplici ed open-source per mantenere le prestazioni elevate.

- ## Linee guida per la documentazione delle interfacce

  Le interfacce delle classi di tutti i vari moduli che saranno sottoposti all'implementazione verranno implementate seguendo le seguenti regole:

  1. Le classi sono identificate da nomi singolari;

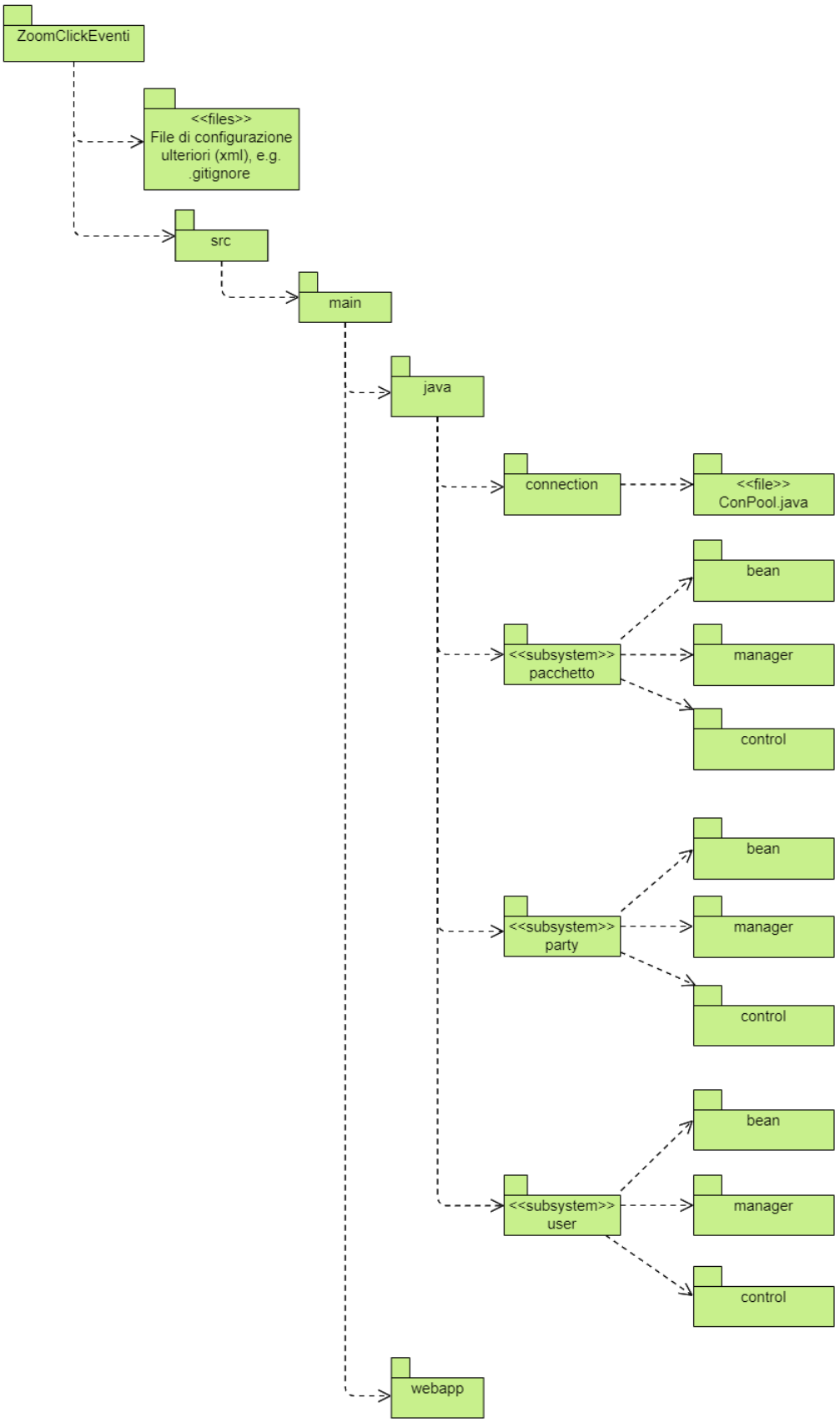
2. I nomi degli attributi e dei metodi seguiranno la convenzione “camel-case”;
3. Tutti i metodi getters/setters correlati ad un attributo saranno esplicitati con il nome dell’attributo che segue get/set (es. get...()/set...());
4. Eventuali errori verranno gestiti attraverso le eccezioni, senza alcun valore di ritorno;
5. I nomi delle classi Control e Manager saranno preceduti dal nome della classe di riferimento.
6. Precondizioni, postcondizioni ed invarianti di classe saranno esplicitati mediante OCL;
7. I nomi delle operazioni seguono lo standard “entity:operation”;

- **Definizioni, acronimi, abbreviazioni**
  RAD: Requirement Analysis Document;
  ODD: Object Design Document;
  SDD: System Design Document;
  OCL: Object Constraint Language;
- **Riferimenti**
  Questo documento utilizza riferimenti a concetti introdotti ed esplicitati nel RAD e nel SDD di ZoomClickEventi.

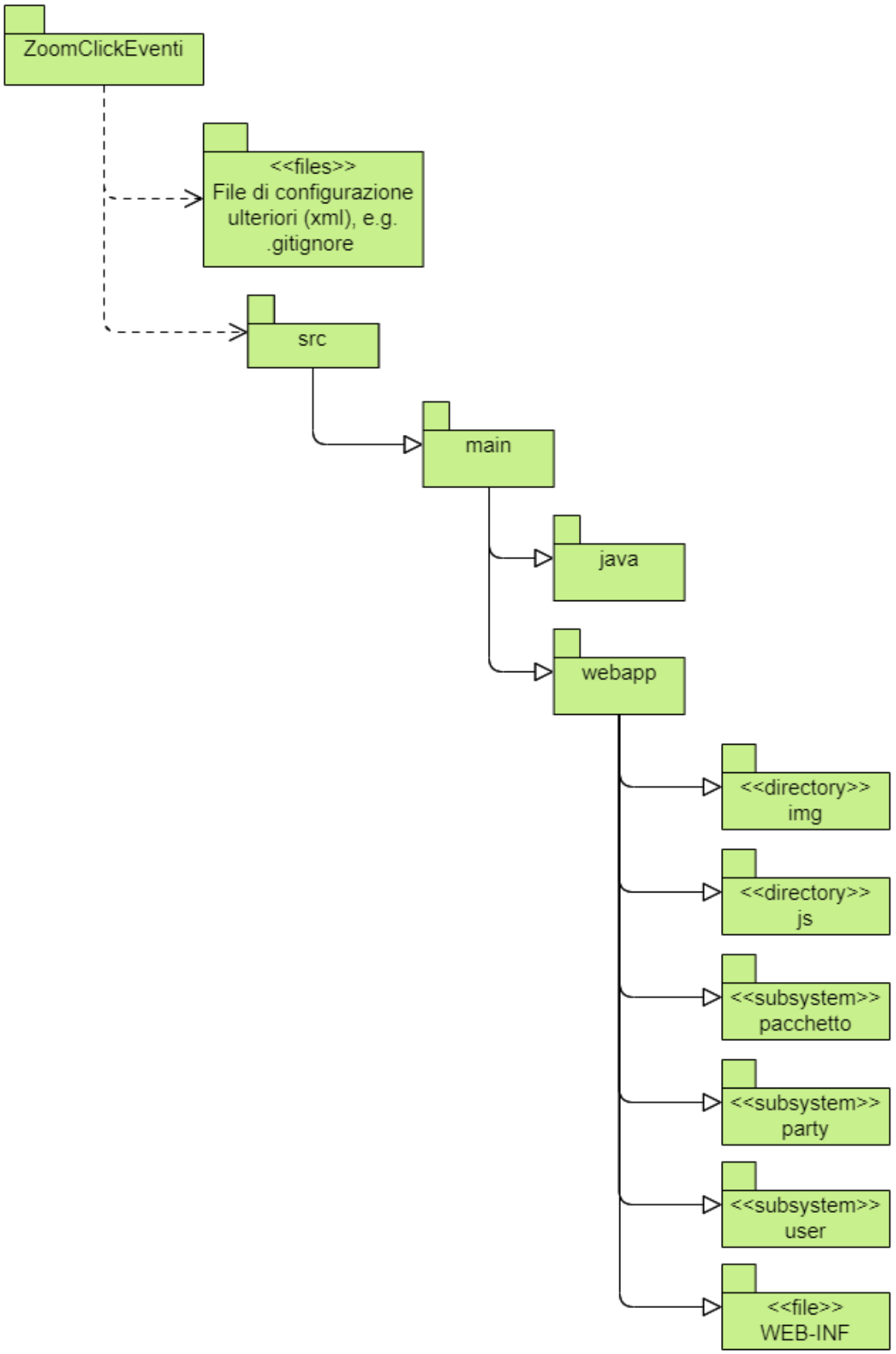
2. Organizzazione delle directory

Ogni sottosistema identificato nel SDD (sezione 2.2 decomposizione in s) verrà mappato su una directory ben precisa. Nel seguente package diagram, ogni package è una directory del progetto, mentre gli archi rappresentano la relazione “include”

- Lato Back-End

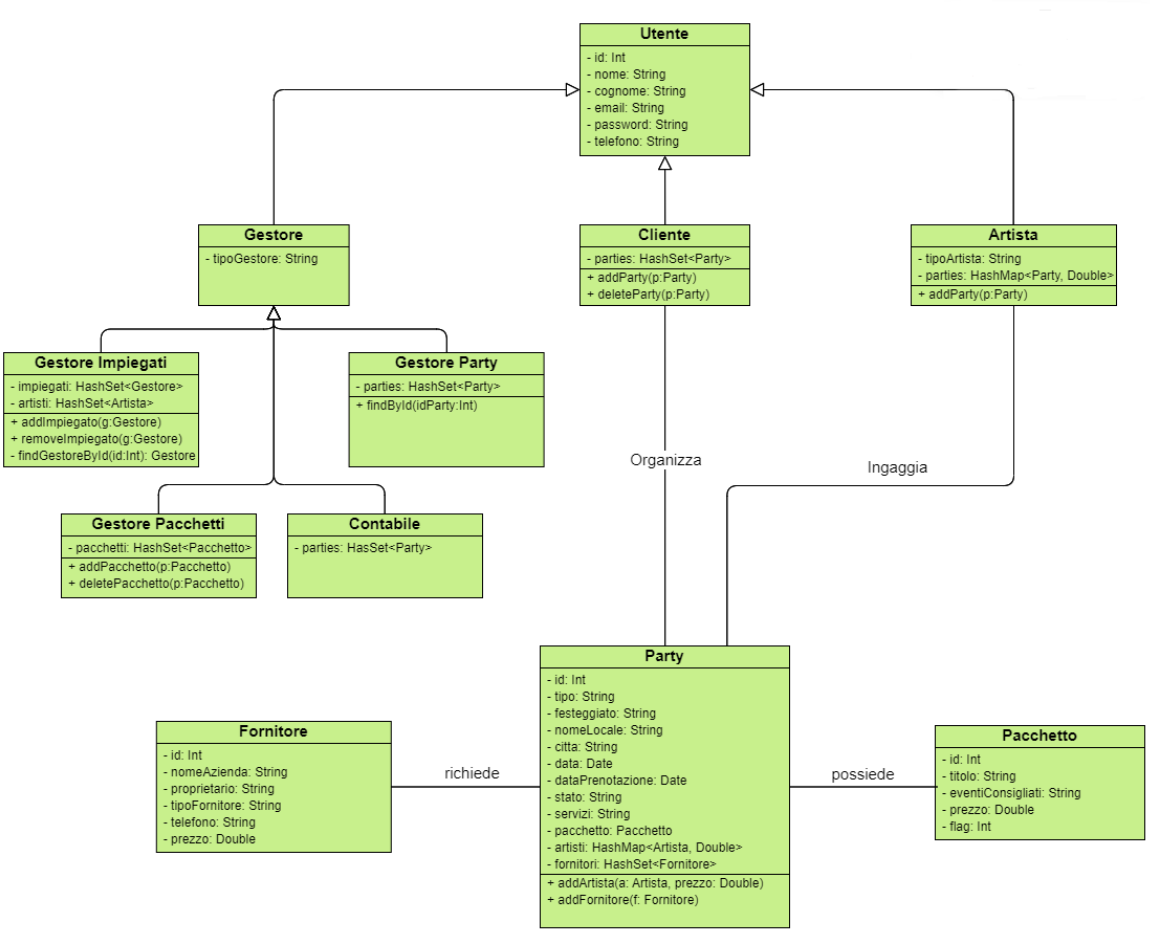


- Lato Front-End (all'interno della directory "webapp")



3. Glossario delle interfacce di classe

Il class diagram presente nel SDD (sezione ... class diagram) necessita di ulteriori raffinazioni. Ad esempio, tutte le classi identificate nel Class diagram necessitano di esporre il livello di accessibilità degli attributi e dei metodi.



Successivamente, troviamo i contratti di classi e metodi indicati attraverso OCL.

Contratti OCL

Classe Cliente:

context Cliente::addParty(p) pre:
- p != null
- not parties.contains(p)

context Cliente::addParty(p) post:
- parties.size() = parties.size() + 1

context Cliente::deleteParty(p) pre:
- p != null
- parties.contains(p)

context Cliente::deleteParty(p) post:
- parties.size() = parties.size() - 1

Classe Artista:

context Artista::addParty(p, prezzo) pre:
- p != null && prezzo!=null
- not parties.containsKey(p)

context Artista::addParty(p, prezzo) post:
- parties.size() = parties.size() + 1
- p.getArtisti().contains(this)

Classe GestoreImpiegati:

context GestoreImpiegati::addImpiegato(g) pre:
- g != null
- not impiegati.contains(g)

context GestoreImpiegati::addImpiegato(g) post:
- impiegati.size() = impiegati.size() + 1

context GestoreImpiegati::removeImpiegato(idGestore) pre:
- idGestore != null
- (findGestoreById(idGestore)) != null

context GestoreImpiegati::removeImpiegato(idGestore) post:

- impiegati.size() = impiegati.size() - 1

---

context GestoreImpiegati::findGestoreById(id) pre:
- id != null

context GestoreImpiegati::findGestoreById(id) post:
- g.getId() = id

Classe Party

context Party::addArtista(a, prezzo) pre:
- a != null
- prezzo != null
- not artisti.containsKey(a)

context Party:: addArtista(a, prezzo) post:
- artisti.size() = artisti.size + 1
- a.getParties().contains(this, prezzo)

---

context Party::addFornitore(f) pre:
- f != null
- not fornitori.containsKey(f)

context Party::addFornitore(f) post:
- fornitori.size() = fornitori.size() + 1

Classe GestorePacchetti:

context GestorePacchetti::addPacchetto(p) pre:
- p != null
- not pacchetti.contains(p)

context GestorePacchetti::addPacchetto(p) post:
- pacchetti.size() = pacchetti.size() + 1

---

context GestorePacchetti::deletePacchetto(id) pre:
- id != null
- findById(id) != null
- pacchetti.contains(p)

context GestorePacchetti::deletePacchetto(p) post:
- pacchetti.size() = pacchetti.size() - 1

context GestorePacchetti::findById(id) pre:
- id != null

context GestorePacchetti::findById(id) post:
- p.getId() = id

Classe UserManager:

context UserManager::insertUser(nome, cognome, email, password, telefono) pre:
- nome != null
- cognome != null
- email != null
- password != null
- telefono != null

context UserManager::insertUser(user_data) post:
- utente.getId() != null
- utente.getNome() = nome
- utente.getCognome() = cognome
- utente.getEmail() = email
- utente.getPassword() = password
- utente.getTelefono() = telefono

context UserManager::insertGestore(utente, tipoGestore) pre:
- utente != null
- tipoGestore != null
-

context UserManager::insertGestore(utente, tipoGestore) post:
- gestore.getId() != null
- gestore.getTipoGestore = tipoGestore
- gestore.getNome() = utente.getNome()
- gestore.getCognome() = utente.getCognome()
- gestore.getEmail() = utente.getEmail()
- gestore.getPassword() = utente.getPassword()
- gestore.getTelefono() = utente.getTelefono()

context UserManager::insertContabile(gestore) pre:
- gestore != null

context UserManager::insertContabile(gestore) post:

- contabile.getId() = gestore.getId()
- contabile.getNome() = gestore.getNome()
- contabile.getCognome() = gestore.getCognome()
- contabile.getEmail() = gestore.getEmail()
- contabile.getPassword() = gestore.getPassword()
- contabile.getTelefono() = gestore.getTelefono()
- contabile.getTipoGestore() = gestore.getTipoGestore()

---

context UserManager::insertGestoreParty(gestore) pre:
- gestore != null

context UserManager::insertGestoreParty(gestore) post:
- gestoreParty.getId() = gestore.getId()
- gestoreParty.getNome() = gestore.getNome()
- gestoreParty.getCognome() = gestore.getCognome()
- gestoreParty.getEmail() = gestore.getEmail()
- gestoreParty.getPassword() = gestore.getPassword()
- gestoreParty.getTelefono() = gestore.getTelefono()
- gestoreParty.getTipoGestore() = gestore.getTipoGestore()

---

context UserManager::insertGestoreImpiegati(gestore) pre:
- gestore != null

context UserManager::insertGestoreImpiegati(gestore) post:
- gestoreImpiegati.getId() = gestore.getId()
- gestoreImpiegati.getNome() = gestore.getNome()
- gestoreImpiegati.getCognome() = gestore.getCognome()
- gestoreImpiegati.getEmail() = gestore.getEmail()
- gestoreImpiegati.getPassword() = gestore.getPassword()
- gestoreImpiegati.getTelefono() = gestore.getTelefono()
- gestoreImpiegati.getTipoGestore() = gestore.getTipoGestore()

---

context UserManager::insertGestorePacchetti(gestore) pre:
- gestore != null

context UserManager::insertGestorePacchetti(gestore) post:
- gestorePacchetti.getId() = gestore.getId()
- gestorePacchetti.getNome() = gestore.getNome()
- gestorePacchetti.getCognome() = gestore.getCognome()
- gestorePacchetti.getEmail() = gestore.getEmail()
- gestorePacchetti.getPassword() = gestore.getPassword()
- gestorePacchetti.getTelefono() = gestore.getTelefono()

- gestorePacchetti.getTipoGestore() = gestore.getTipoGestore()

context UserManager::login(email, password) pre:
- email != null
- password != null

context UserManager::login(email, password) post:
- utente.getEmail() = email
- utente.getPassword() = password
- utente.getId() != null
- utente.getNome() != null
- utente.getCognome() != null
- utente.getTelefono() != null

context UserManager::updateUser(u) pre:
- u != null

context UserManager::updateUser(u) post:
- true

context UserManager::checkIdByEmail(email) pre:
- email != null

context UserManager::checkIdByEmail(email) post:
- true

context UserManager::retrieveAllEmployee() pre:
- true

context UserManager::retrieveAllEmployee post:
- hashSet != null

context UserManager::retrieveAllArtisti() pre:
- true

context UserManager::retrieveAllArtisti() post:
- hashSet != null

context UserManager::deleteUser(idEmployee) pre:
- idEmployee != null

context UserManager::deleteUser(idEmployee) post:
- true

context UserManager::isCliente(utente) pre:
- utente != null

context UserManager::isCliente(utente) post:
- cliente.getId()=utente.getId()
- cliente.getNome() = utente.getNome()
- cliente.getCognome() = utente.getCognome()
- cliente.getEmail() = utente.getEmail()
- cliente.getPassword() = utente.getPassword()
- cliente.getTelefono() = utente.getTelefono()

---

context UserManager::insertCliente(utente) pre:
- utente!= null

context UserManager::insertCliente(utente) post:
- true

---

context UserManager::isArtista(utente) pre:
- utente != null

context UserManager::isArtista(utente) post:
- artista.getId()=utente.getId()
- artista.getNome() = utente.getNome()
- artista.getCognome() = utente.getCognome()
- artista.getEmail() = utente.getEmail()
- artista.getPassword() = utente.getPassword()
- artista.getTelefono() = utente.getTelefono()

---

context UserManager::isGestore(utente) pre:
- utente != null

context UserManager::isArtista(utente) post:
- gestore.getId()=utente.getId()
- gestore.getNome() = utente.getNome()
- gestore.getCognome() = utente.getCognome()
- gestore.getEmail() = utente.getEmail()
- gestore.getPassword() = utente.getPassword()
- gestore.getTelefono() = utente.getTelefono()
- gestore.getTipoGestore() != null

Classe PacchettoManager:

context PacchettoManager::insertPacchetto(p) pre: - p != null context PacchettoManager::insertPacchetto(p) post: - p.getId() != null
context PacchettoManager::retrieveAll() pre: - true context PacchettoManager::retrieveAll() post: - hashSet != null
context PacchettoManager::deletePacchetto(id) pre: - id != null context PacchettoManager::deletePacchetto(id) post: - true
context PacchettoManager::findPacchettoById(id) pre: - id != null context PacchettoManager::findPacchettoById(id) post: - pacchetto.getId() != null - pacchetto.getTitolo() != null - pacchetto.getEventiConsigliati() != null - pacchetto.getPrezzo() != null - pacchetto.getFlag() != null

Classe PartyManager:

context PartyManager::deleteParty(id) pre: - id != null context PartyManager::deleteParty(p) post: - true
context PartyManager::retrieveAllParty() pre: - true context PartyManager::retrieveAllParty() post: - hashSet != null
context PartyManager::retrievePartyByIdArtista(idArtista) pre: - idArtista != null context PartyManager::retrievePartyByIdArtista(idArtista) post: - collection != null
context PartyManager::prenotaParty(p) pre: - p != null context PartyManager::prenotaParty(p) post: - p.getId() != null
context PartyManager::rifiutaParty(idParty) pre: - idParty != null context PartyManager::rifiutaParty(idParty) post: - true
context PartyManager::confermaParty(p) pre: - p != null context PartyManager::confermaParty(p) post: - true
context PartyManager::findPartyByIdCliente(idCliente) pre: - idCliente != null context PartyManager::findPartyByIdCliente(idCliente) post: - hashSet != null

context PartyManager::findArtistaByIdParty(idParty) pre:
- idParty != null

context PartyManager::findArtistaByIdParty(idParty) post:
- collection!= null

context PartyManager::findFornitoreByIdParty(idParty) pre:
- idParty != null

context PartyManager::findFornitoreByIdParty(idParty) post:
- collection != null

context PartyManager::retrieveAllFornitori() pre:
- true

context PartyManager::retrieveAllFornitori() post:
- collection != null