

# **Progetto Ingegneria del Software 2008-09**

## ***“New-Totosì”***

**Documento di Object Design**

Team Leader:  
Francesco Lauria 26334  
Team:  
Donato Sabia 17104  
Antonio Genovese 21368  
Savino Sasso 21731

Coordinatori del progetto:

Nome	Matricola
Francesco Lauria	26334
Savino Sasso	21731

Partecipanti al progetto:

Nome	Matricola
Antonio Genovese	21368
Donato Sabia	17104

Revision History

Data	Versione	Descrizione	Autore
08-02-2009	1.0	Elaborazione del documento	Team
12-02-2009	1.1	Prima Revisione globale	Team

**Documento di OBJECT DESIGN**  
***Progetto di un portale per una società***  
***di servizi dedicati alle scommesse sportive on-line***

<b>1.</b>	<b>Introduzione</b>	<b>2</b>
1.1	Object design trade-off	2
1.1.1	Comprensibilità vs tempo	2
1.1.2	Sicurezza vs efficienza	2
1.2	Interface documentation guidelines	2
1.3	Definizioni, acronimi ed abbreviazioni	4
1.4	Riferimenti	4
<b>2.</b>	<b>Package</b>	<b>5</b>
2.1	Package Totosì	5
2.1.1	Gestione Sessione Lavoro	6
2.1.2	Gestione Scommessa	6
2.1.3	Gestione Conto Personale	6
2.1.4	Gestione Scommettitore	7
2.1.5	Gestione Evento	7
2.1.6	Gestione Problemi	7
<b>3.</b>	<b>Interfaccia delle Classi</b>	<b>8</b>
3.1	Gestione Sessione Lavoro	8
3.2	Gestione Scommessa	12
3.3	Gestione Conto Personale	18
3.4	Gestione Scommettitore	23
3.5	Gestione Evento	28
3.6	Gestione Problemi	33

# 1. Introduzione

## 1.1 Object design trade-off

Le scelte progettuali imposte ci hanno spinto a decidere di utilizzare componenti già esistenti per gestire alcune funzionalità del sistema. La quantità di lavoro che si prospetta e la delicatezza dei dati che vengono trattati è tale da doversi affidare ad un DBMS particolarmente solido ed affidabile e ad un linguaggio per il trasferimento dati che provveda ad una adeguata sicurezza.

La scelta è ricaduta sul componente open source JDBC, per quanto riguarda il DBMS, e il protocollo TLS per la crittografia dei dati.

### 1.1.1 Comprensibilità vs tempo

La comprensibilità del codice è un aspetto molto importante soprattutto per la fase di testing. Ogni classe e metodo deve essere facilmente interpretabile anche da chi non ha collaborato al progetto. Per mantenere questa comprensibilità abbiamo deciso di fornire all'interno del codice una descrizione dettagliata, sotto forma di commento, a supporto di ciascuna classe e metodo; per le classi o i metodi particolarmente complessi e delicati il commento del codice sarà supportato da una documentazione che semplifica ulteriormente il lavoro di comprensione. Ovviamente questa caratteristica aggiungerà un incremento di tempo allo sviluppo del nostro progetto.

### 1.1.2 Sicurezza vs efficienza

Come anticipato nell'SDD il nostro progetto necessita di particolari attenzioni alla sicurezza per quanto riguarda le operazioni economiche. Quindi oltre alla login con password dedicata dovrà essere integrato un protocollo crittografico per gestire al meglio le transazioni monetarie che dovranno essere parte integrante del sistema.

Non tutte le richieste da parte dei client verranno criptate, ma come precedentemente anticipato, solo quelle riguardanti la Gestione del Conto dello Scommettitore.

In questo modo si otterrà un buon compromesso tra efficienza e sicurezza rallentando il sistema ove necessario per assicurare una trasparenza dati assoluta.

## 1.2 Interface documentation linee guida

Gli sviluppatori dovranno seguire alcune linee guida per la scrittura del codice, linee guida specificate anche nel quality plan:

- Stile di programmazione

- Lo spazio di indentazione delle classi deve essere di 4 spazi, oppure di un tab (correttamente impostato a 4 spazi)

- Naming Convention

- I nomi delle classi devono cominciare con una lettera maiuscola, e anche le parole seguenti all'interno del nome devono cominciare con una lettera maiuscola. I nomi delle Classi devono fornire informazioni sul loro scopo.  
*Es. GestioneScommettitore --*
- I nomi delle servlet utilizzate dovranno iniziare con "Servlet" per distinguerle dalle altre classi presenti.
- I nomi dei metodi devono cominciare con una lettera minuscola, e le parole seguenti con la lettera maiuscola. Il nome del metodo tipicamente consiste di un verbo che identifica una azione, seguito dal nome di un oggetto.  
*Es. modificaScommettitore()*
- I nomi dei metodi per l'accesso e la modifica delle variabili dovranno essere del tipo `getNomeVariabile()`, `setNomeVariabile()`.
- I nomi delle variabili devono cominciare con una lettera minuscola, e le parole seguenti con la lettera maiuscola. *Es. numeroConto*
- I nomi delle costanti devono essere scritti a lettere maiuscole, e le parole separate da un underscore.

#### - Documentazione

- Ogni classe deve avere una breve spiegazione dello scopo della classe, dopo le istruzioni di import. Il prologo deve indicare oltre allo scopo,

l'autore, la versione ( e altre informazioni aggiuntive )utilizzando il tag JavaDoc appropriato

```
/**
 * sommario dello scopo della classe.<p>
 *
 * @author nome dell'autore
 * @version numero di versione della classe
 */
```

- La descrizione del metodo deve apparire prima di ogni dichiarazione di metodo, e deve descriverne lo scopo. Deve includere anche informazioni sugli argomenti, sul valore di ritorno, e se applicabile, sulle eccezioni, utilizzando i tags JavaDoc

```
/**
 * descrizione dello scopo del metodo
 *
 * @param param1 – descrizione del parametro 1
 * @return descrizione del valore ritorno
 */
```

- Le variabili devono essere documentate solo se il loro scopo non è immediatamente intuibile. Possono essere documentate o seguendo le convenzioni JavaDoc, oppure con commenti inline.
- Ogni pagina jsp deve iniziare con una direttiva, in cui vengono definiti gli attributi di tale pagina. *Es* `<% @ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql. *, java.util. *" errorPage="" %>`
- Tutte le pagine, JSP e HTML; devono contenere il prologo  
*Es.* `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">`

### 1.3 Definizioni, acronimi ed abbreviazioni

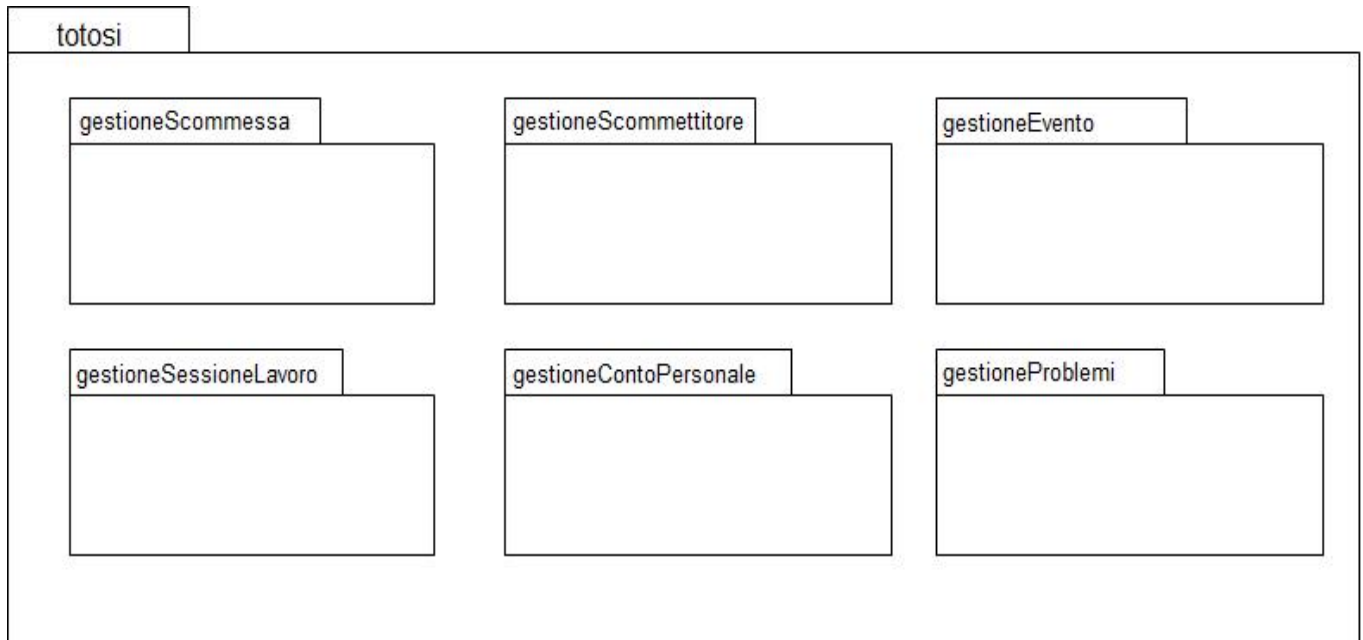
SDD : **S**ystem **D**esign **D**ocument  
DBMS: **D**ata**B**ase **M**anagement **S**ystem  
JDBC : **J**ava **D**ata**B**ase **C**onnectivity  
TLS: **T**ransport **L**ayer **S**ecurity

### 1.4 Riferimenti

SDD - Totosì

## 2. Packages

### Totosi

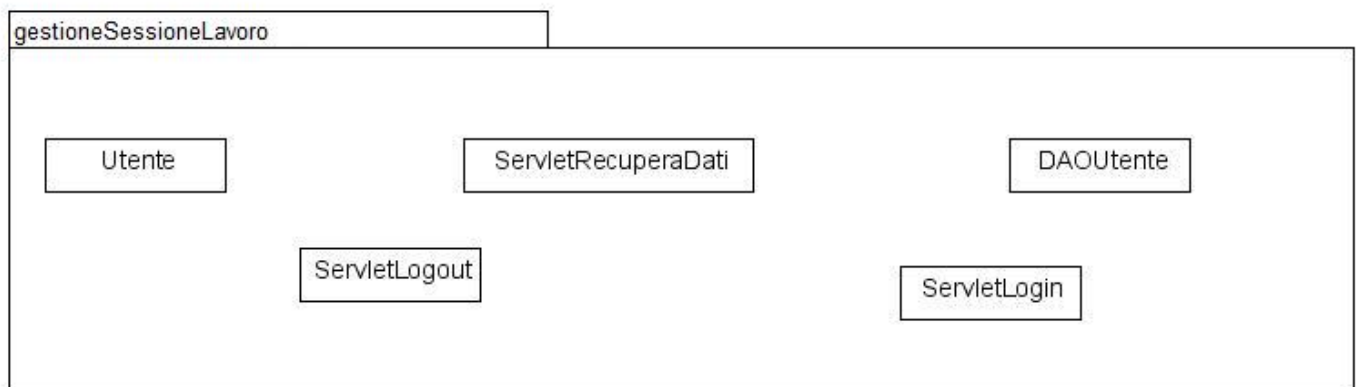


Per visionare le **servlet** contenute all'interno di ogni package rimandiamo al documento correlato SDD- tosi. Infatti i package individuati rispecchiano la divisione in sottosistemi individuata durante l'attività di System Desing.

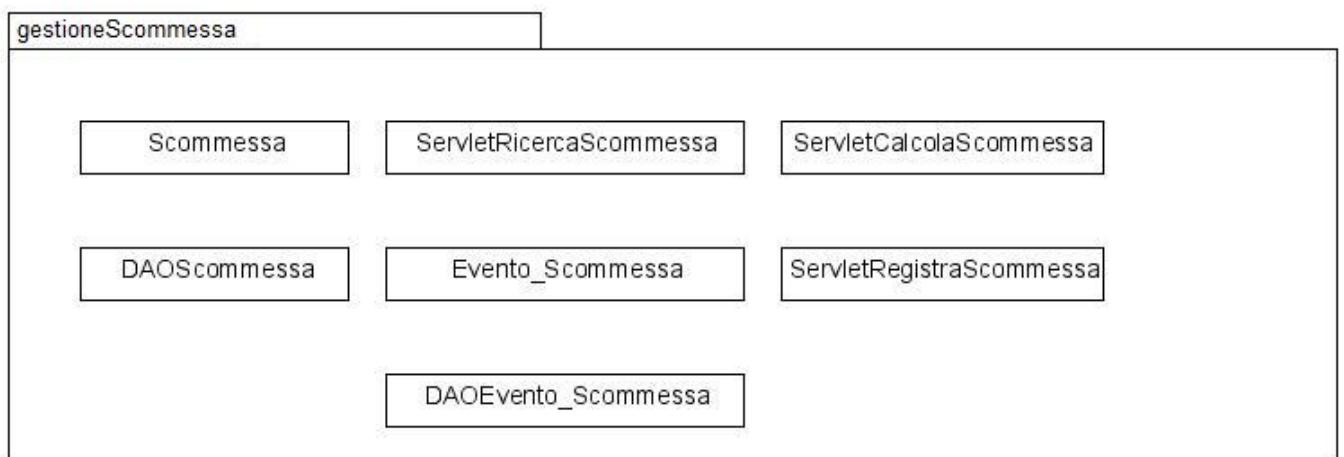
### 2.1 Package tosi

Il package tosi contiene sotto package che a loro volta inglobano le servlet atte alla gestione delle richieste utente provenienti da HTTP. Le classi contenute nel package svolgono il ruolo di gestore logico del sistema. Nel seguito riportiamo solo una panoramica delle classi contenute in ogni pacchetto.

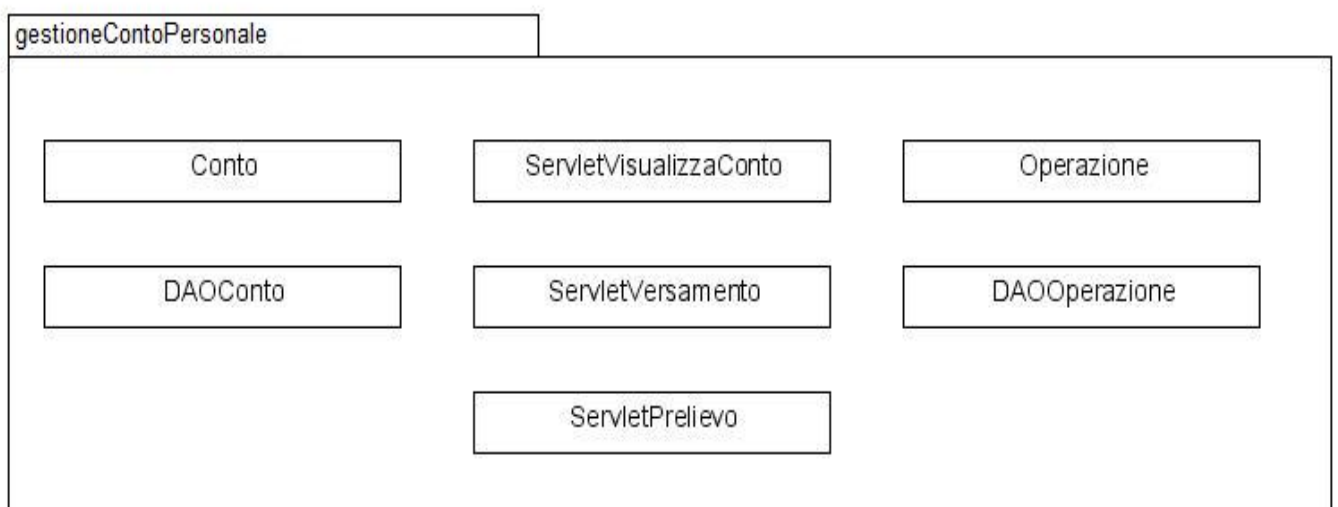
### 2.1.1 Gestione Sessione Lavoro



### 2.1.2 Gestione Scommessa

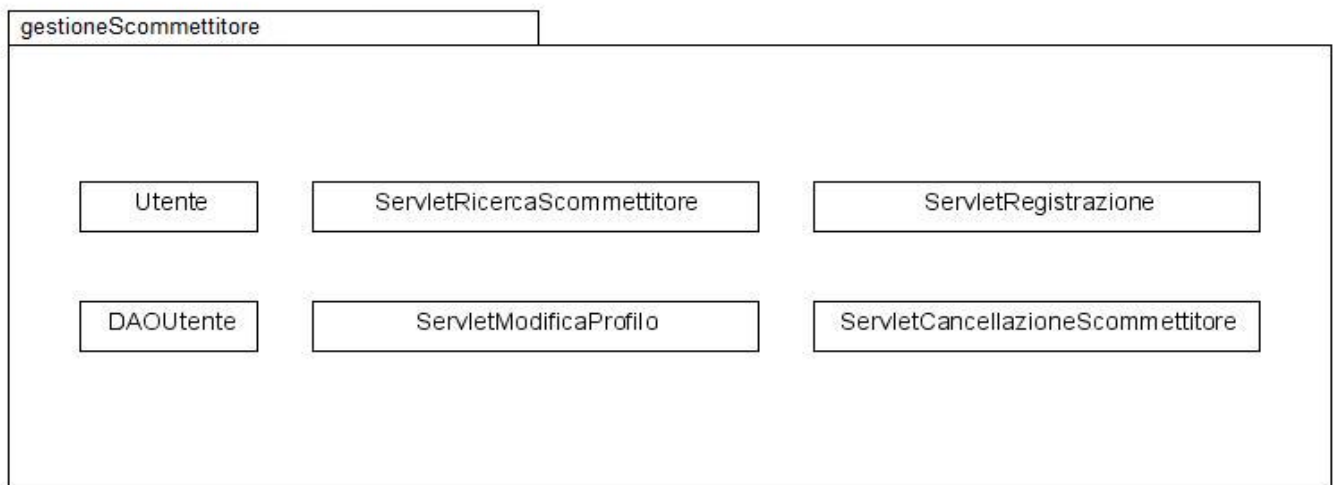


### 2.1.3 Gestione Conto Personale

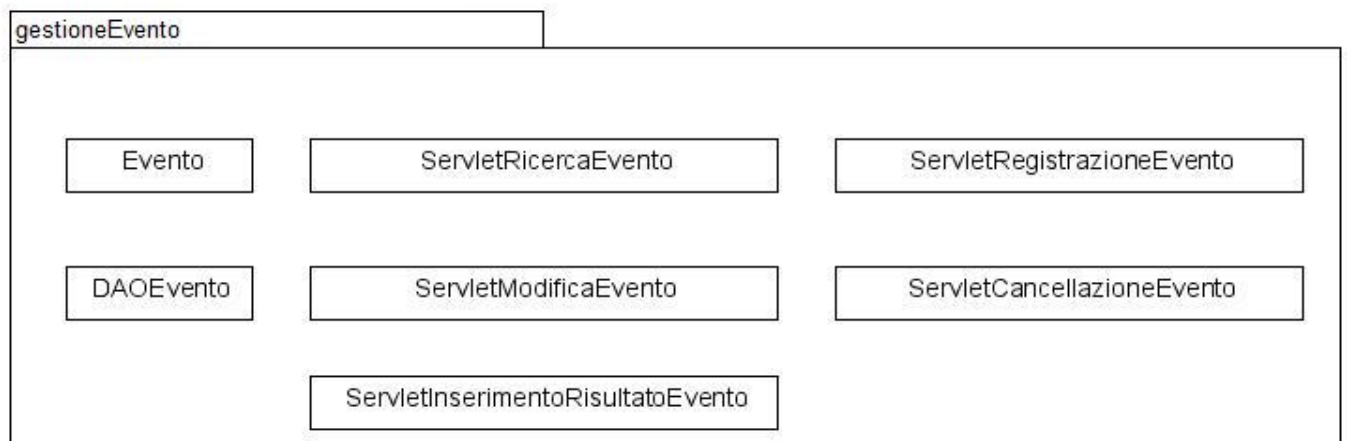




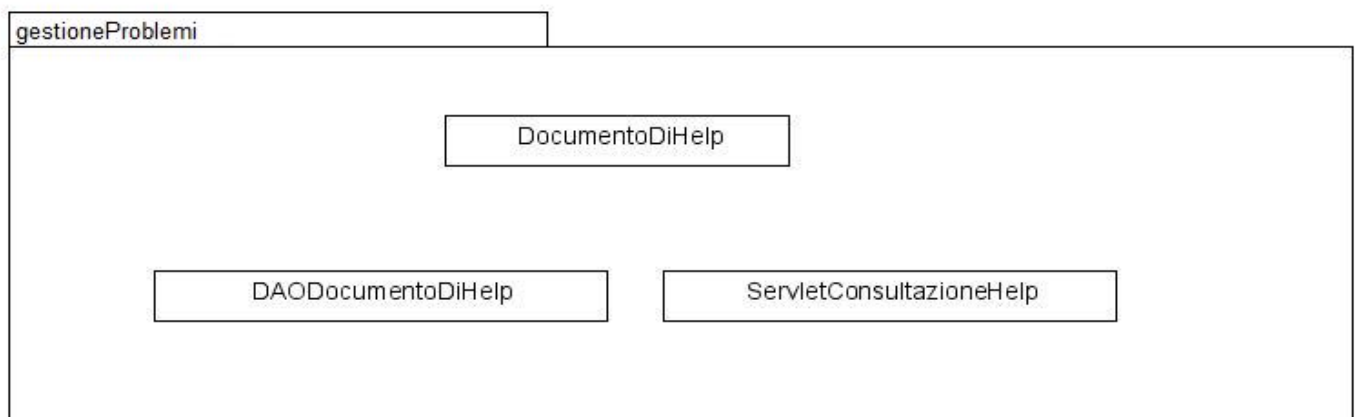
### 2.1.4 Gestione dello Scommettitore



### 2.1.5 Gestione Evento

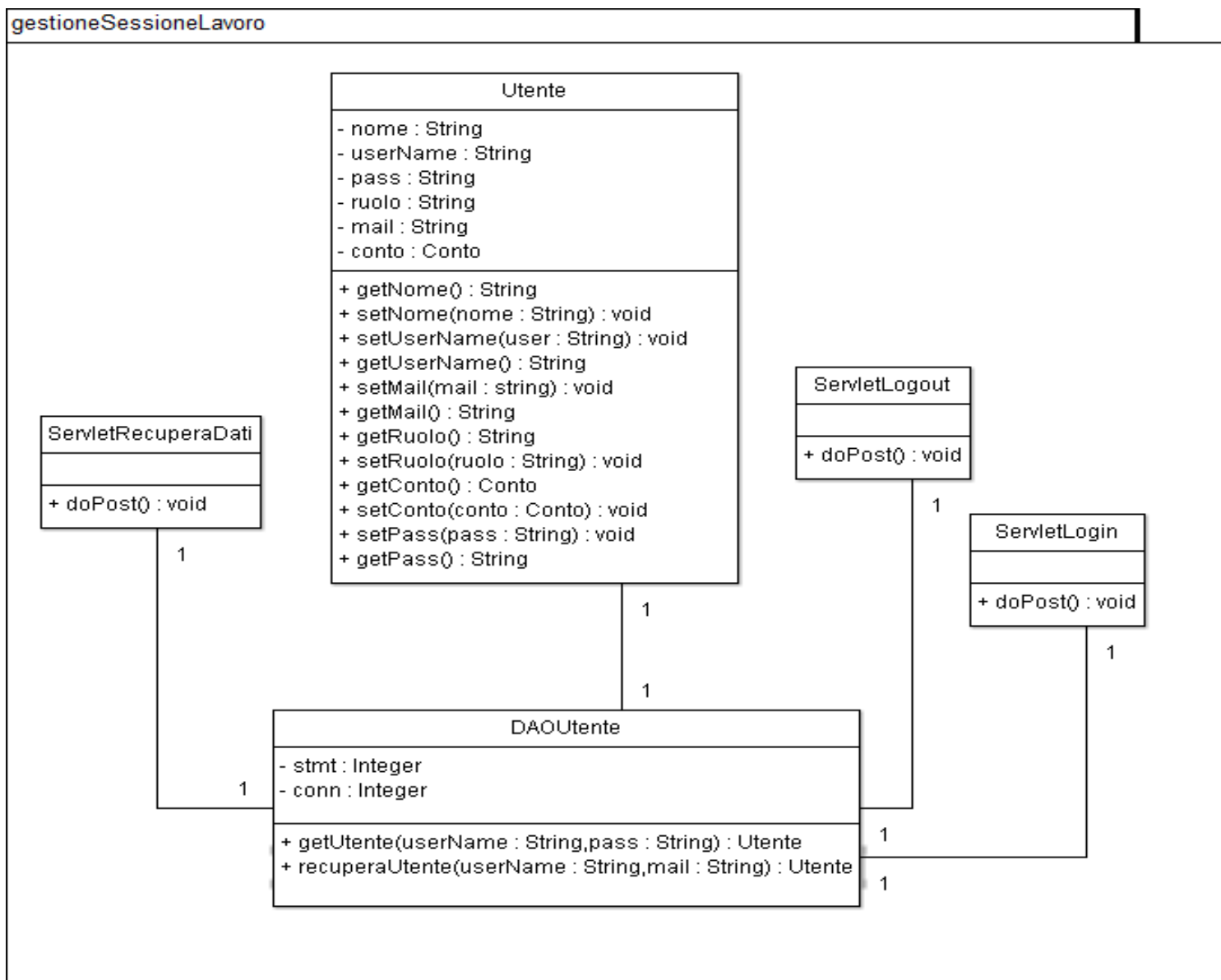


### 2.1.6 Gestione Problemi



## 3. Interfaccia delle Classi

### 3.1 Gestione Sessione Lavoro



Classe	Utente	Versione: 1.0
		Autore: Antonio Genovese
Descrizione	Classe rappresenta l'utente registrato.	
Dipendenze	Non ha dipendenze.	
Attributi		
Nome	Accesso	Descrizione
nome	Privato	Nome dell'Utente.
userName	Privato	Pseudonimo dell'utente registrato.

pass	Privato	Password di accesso.		
ruolo	Privato	Il ruolo dell'Utente che può essere 'Amministratore' oppure 'Scommettitore'		
mail	Privato	L'indirizzo email dell'Utente.		
conto	Privato	Il Conto associato all'utente se questo è uno scommettitore altrimenti sarà null.		
Metodi				
Signature	setUserName(userName : String)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire l'userName dell'Utente.			
Signature	getUserName()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce l'userName dell'utente.			
Signature	getNome()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce il nome anagrafico dell'utente.			
Signature	setNome (nome: String)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna il nome dell'Utente.			
Signature	setPass (pass : String)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta la password dell'Utente.			
Signature	getPass ()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce la password crittografata.			
Signature	setMail (mail : String)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna l'indirizzo email dell'Utente.			
Signature	getMail ()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce l'indirizzo email dell'Utente.			
Signature	setConto (conto : Conto)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna un conto all'Utente se questo è 'Scommettitore'			
Signature	getRuolo ()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce il ruolo relativo all' Utente .			
Signature	setRuolo (ruolo : String)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna il ruolo all'Utente.			

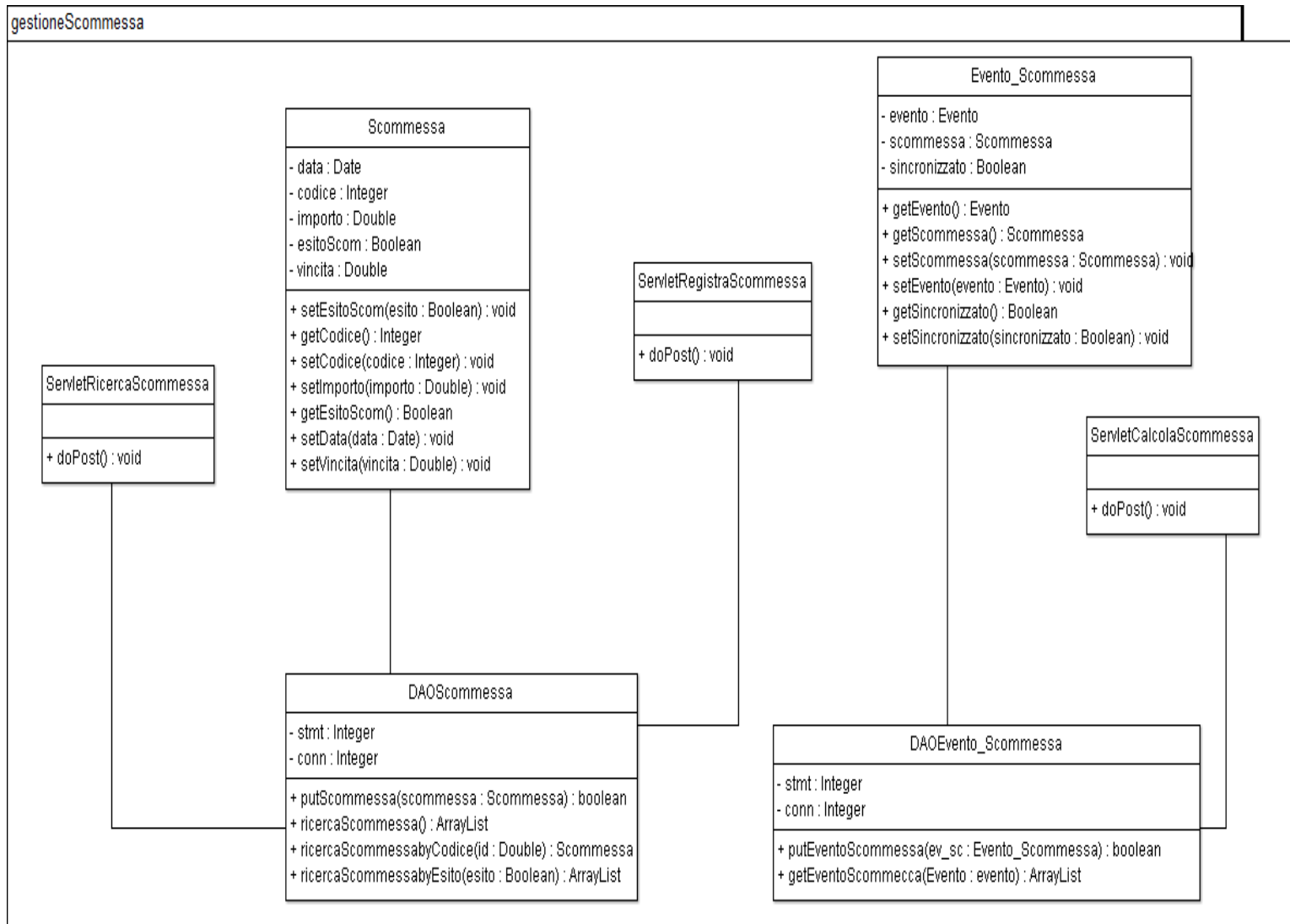
Classe	DAOUtente		Versione: 1.0	
			Autore: Antonio Genovese	
Descrizione	La classe DAOUtente si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di un Utente.			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso	Descrizione		
stmt	Privato	Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce		
conn	Privato	Oggetto di tipo Connection che gestisce una connessione con uno specifico database		
Metodi				
Signature	getUtente(userName : String , pass : String)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce un utente dal DBMS in base all'userName e alla password.			SQLException
Signature	recuperaUtente(userName : String, mail : String)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce un utente dal DBMS in base all'userName e alla mail (nel caso di recupero dati del profilo utente).			SQLException

Classe	ServletRecuperaDati	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di recupero della password dal profilo dell'utente			
Dipendenze	Dipende da DAOUtente			
Attributi				
Nome	Accesso	Descrizione		
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletLogin		Versione: 1.0		
			Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di Login, per autenticarsi al sistema.				
Dipendenze	Dipende da DAOUtente				
Attributi					
Nome	Accesso		Descrizione		
Metodi					
Signature	doPost()		Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva				ServletException, IOException

Classe	ServletLogout	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di Logout, per uscire dalla sessione autenticata.			
Dipendenze	Dipende da DAOUtente			
Attributi				
Nome	Accesso	Descrizione		
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

### 3.2 Gestione Scommessa



Classe	Scommessa		Versione: 1.0	
			Autore: Antonio Genovese	
Descrizione	Classe rappresenta le scommesse effettuate.			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso	Descrizione		
Data	Privato	Data di registrazione scommessa.		
Codice	Privato	Codice univoco identificativo della scommessa.		
Importo	Privato	Importo totale della scommessa.		
EsitoScom	Privato	Esito della scommessa, indica se la scommessa è vincente o meno.		
Vincita	Privato	Totale importo della vincita eventuale.		
Metodi				
Signature	setEsitoScom(esito : Boolean)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire l'esito della scommessa, 'true' se la scommessa è vincente.			
Signature	getCodice ()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce il codice univoco della scommessa.			
Signature	getEsitoScom()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce l'esito della scommessa, restituisce 'true' se la scommessa è vincente.			
Signature	setCodice (codice: Integer)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna un codice univoco alla scommessa.			
Signature	setData (data : Date)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta la data della scommessa.			
Signature	setImporto (importo : Double)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna l'importo speso per la scommessa.			
Signature	setVincita (vincita : Double)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna l'importo che eventualmente frutterà l'esito positivo della scommessa.			

Classe	Evento_Scommessa		Versione: 1.0	
			Autore: Antonio Genovese	
Descrizione	Classe rappresenta la classe relazionale tra Scommessa e Evento individua gli eventi connessi alle scommesse effettuate.			
Dipendenze	Dipende dalla classe Evento e da quella Scommessa.			
Attributi				
Nome	Accesso	Descrizione		
Evento	Privato	Indica l'evento su cui si è scommesso.		
Scommessa	Privato	Indica la scommessa che si èeffettuata.		
Sincronizzato	Privato	Indica se il risultato dell'evento è stato sincronizzato con la scommessa		
Metodi				
Signature	setSincronizzato(sincronizzato : Boolean)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire la Sincronizzazzine, 'true' se la scommessa è sincronizzata con il risultato dell'evento			
Signature	getSincronizzato()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce 'true' se la scommessa è sincronizzata.			
Signature	getScommessa()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce la scommessa effettuata.			
Signature	setScommessa (scommessa: Scommessa)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta la scommessa effettuata.			
Signature	setEvento (evento : Evento)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta l'evento su cui si è scommesso.			
Signature	getEvento ()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce l'evento su cui si è scommesso.			



Classe	DAOScommessa	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	La classe DAOScommessa si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di una Scommessa			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso	Descrizione		
stmt	Privato	Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce		
conn	Privato	Oggetto di tipo Connection che gestisce una connessione con uno specifico database		
Metodi				
Signature	putScommessa(scommessa : Scommessa)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire una scommessa nel DBMS e restituisce 'true' se l'operazione va a buon fine.			SQLException
Signature	ricercaScommessa()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce tutte le scommesse registrate nel DBMS.			SQLException
Signature	ricercaScommessabyCodice(id : Integer)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce la scommessa con quel codice prelevandola dal DBMS.			SQLException
Signature	ricercaScommessabyEsito(esito : Boolean)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce tutte le scommesse in base al codice prendendole dal DBMS.			SQLException

Classe	DAOEvento_Scommessa	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	La classe DAOEvento_Scommessa si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di una istanza della classe Evento_Scommessa			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso	Descrizione		
stmt	Privato	Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce		
conn	Privato	Oggetto di tipo Connection che gestisce una connessione con uno specifico database		
Metodi				
Signature	putEvento_Scommessa(ev_sc : Evento_Scommessa)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire un Evento_Scommessa nel DBMS e restituisce 'true' se l'operazione va a buon fine.			SQLException
Signature	getEvento_Scommessa(event o: Evento)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce tutte le scommesse effettuate in base all'evento.			SQLException

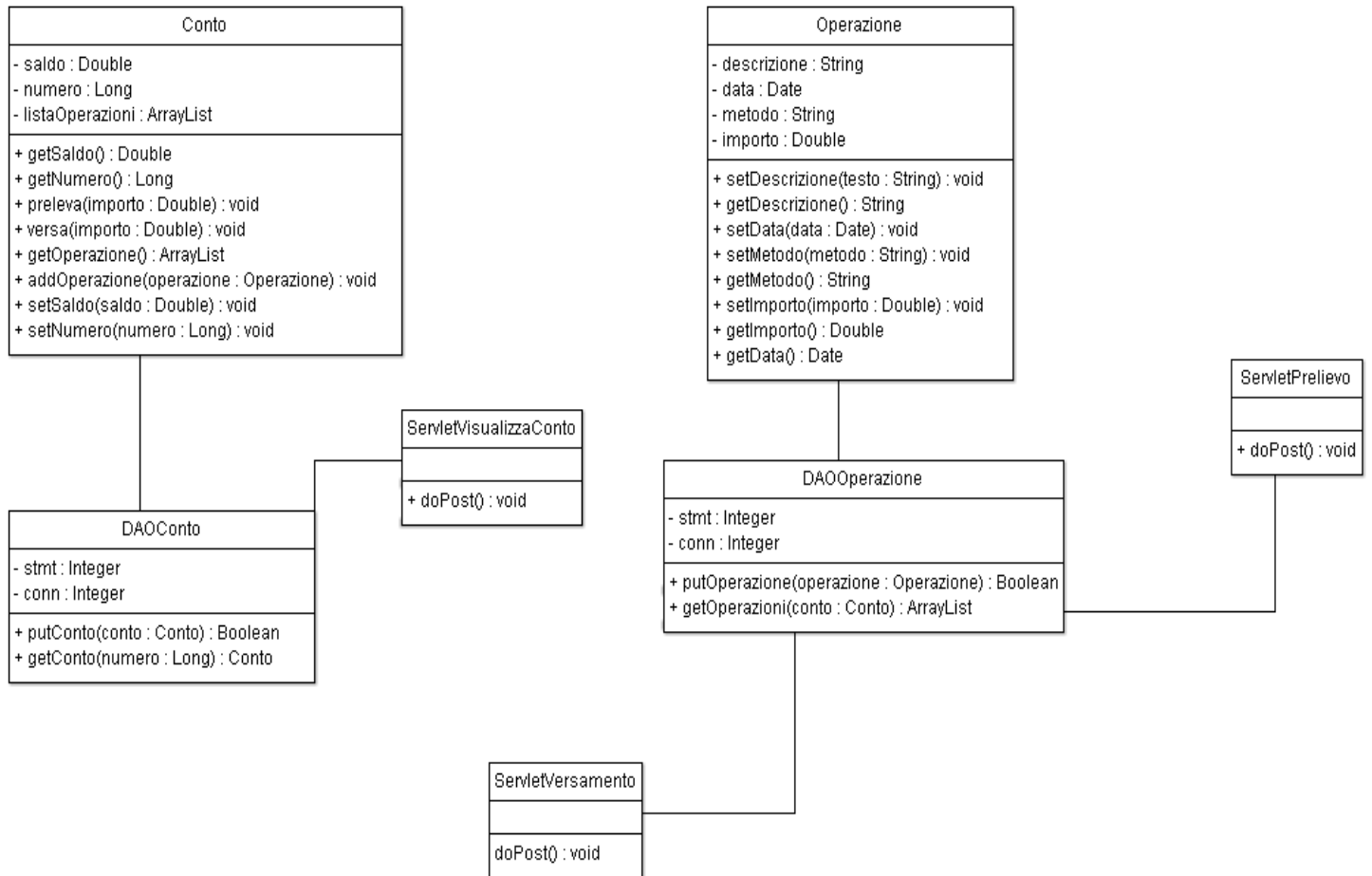
Classe	ServletRegistraScommessa	Versione: 1.0			
		Autore: Antonio Genovese			
Descrizione	Servlet che permette l'operazione di registrazione della Scommessa				
Dipendenze	Dipende da DAOScommessa				
Attributi					
Nome	Accesso		Descrizione		
Metodi					
Signature	doPost()		Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException	

Classe	ServletRicercaScommessa	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di ricerca della Scommessa			
Dipendenze	Dipende da DAOScommessa			
Attributi				
Nome	Accesso	Descrizione		
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletCalcolaScommessa		Versione: 1.0		
			Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di calcolo delle scommesse vincenti				
Dipendenze	Dipende da DAOEvento_Scommessa				
Attributi					
Nome	Accesso		Descrizione		
Metodi					
Signature	doPost()		Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException	

### 3.3 Gestione Conto Personale

gestioneContoPersonale



Classe	Conto	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	Classe rappresenta il conto personale dello scommettitore			
Dipendenze				
Attributi				
Nome	Accesso	Descrizione		
saldo	Privato	Saldo del conto personale.		
numero	Privato	Numero univoco identificativo del conto personale.		
listaOperazioni	Privato	Lista delle operazioni effettuate sul conto.		
Metodi				
Signature	preleva (saldo : Long)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di aggiornare il saldo in seguito a un prelievo.			
Signature	getNumero ()	Accesso	Pubblico	Eccezioni
Descrizione	Consente di sapere il numero del conto.			
Signature	getSaldo ()	Accesso	Pubblico	Eccezioni
Descrizione	Consente di ottenere il saldo del conto.			
Signature	versa (saldo : Long)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di aggiornare il saldo in seguito a un versamento.			
Signature	getOperazione()	Accesso	Pubblico	Eccezioni
Descrizione	Torna la lista delle operazioni associate con il Conto			
Signature	addOperazione(operazione : Operazione)	Accesso	Pubblico	Eccezioni
Descrizione	Inserisci un'operazione nella lista.			
Signature	setSaldo(saldo : Double)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta il saldo del Conto.			
Signature	setNumero(numero : Long)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta il numero del Conto.			

Classe	Operazione	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	Classe rappresenta l'operazione generica (Prelievo o Versamento) che uno scommettitore può effettuare.			
Dipendenze				
Attributi				
Nome	Accesso	Descrizione		
descrizione	Privato	Breve descrizione relativo al tipo di operazione .		
data	Privato	Data effettuazione dell'operazione.		
metodo	Privato	Breve descrizione sulla metodologia di versamento o prelievo.		
importo	Privato	Importo dell'operazione.		
Metodi				
Signature	getDescizione ()	Accesso	Pubblico	Eccezioni
Descrizione	Consente di ottenere la descrizione dell'operazione.			
Signature	setDescrizione (testo : String)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire una descrizione all'operazione.			
Signature	setData (data : Date)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire la data dell'operazione.			
Signature	getData ()	Accesso	Pubblico	Eccezioni
Descrizione	Consente di ottenere la data dell'operazione.			
Signature	setMetodo (metodo : String)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire la metodologia all'operazione.			
Signature	getMetodo ()	Accesso	Pubblico	Eccezioni
Descrizione	Consente di ottenere informazioni sul metodo di versamento o prelievo.			

Classe	DAOConto		Versione: 1.0		
			Autore: Antonio Genovese		
Descrizione	La classe DAOConto si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di un Conto.				
Dipendenze	Non ha dipendenze.				
Attributi					
Nome	Accesso		Descrizione		
stmt	Privato		Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce		
conn	Privato		Oggetto di tipo Connection che gestisce una connessione con uno specifico database		
Metodi					
Signature	putConto (conto : Conto)		Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire un Conto nel DBMS e restituisce 'true' se l'operazione va a buon fine.				SQLException
Signature	getConto(numero : Long)		Accesso	Pubblico	Eccezioni
Descrizione	Restituisce il Conto in base al numero.				SQLException

Classe	DAOOperazione		Versione: 1.0		
			Autore: Antonio Genovese		
Descrizione	La classe DAOOperazione si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di una Operazione.				
Dipendenze	Non ha dipendenze.				
Attributi					
Nome	Accesso		Descrizione		
stmt	Privato		Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce		
conn	Privato		Oggetto di tipo Connection che gestisce una connessione con uno specifico database		
Metodi					
Signature	putOperazione (operazione : Operazione)		Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire una Operazione nel DBMS e restituisce 'true' se l'operazione va a buon fine.				SQLException
Signature	getOperazioni(conto : Conto)		Accesso	Pubblico	Eccezioni
Descrizione	Restituisce la lista di tutte le operazioni relative a un conto.				SQLException

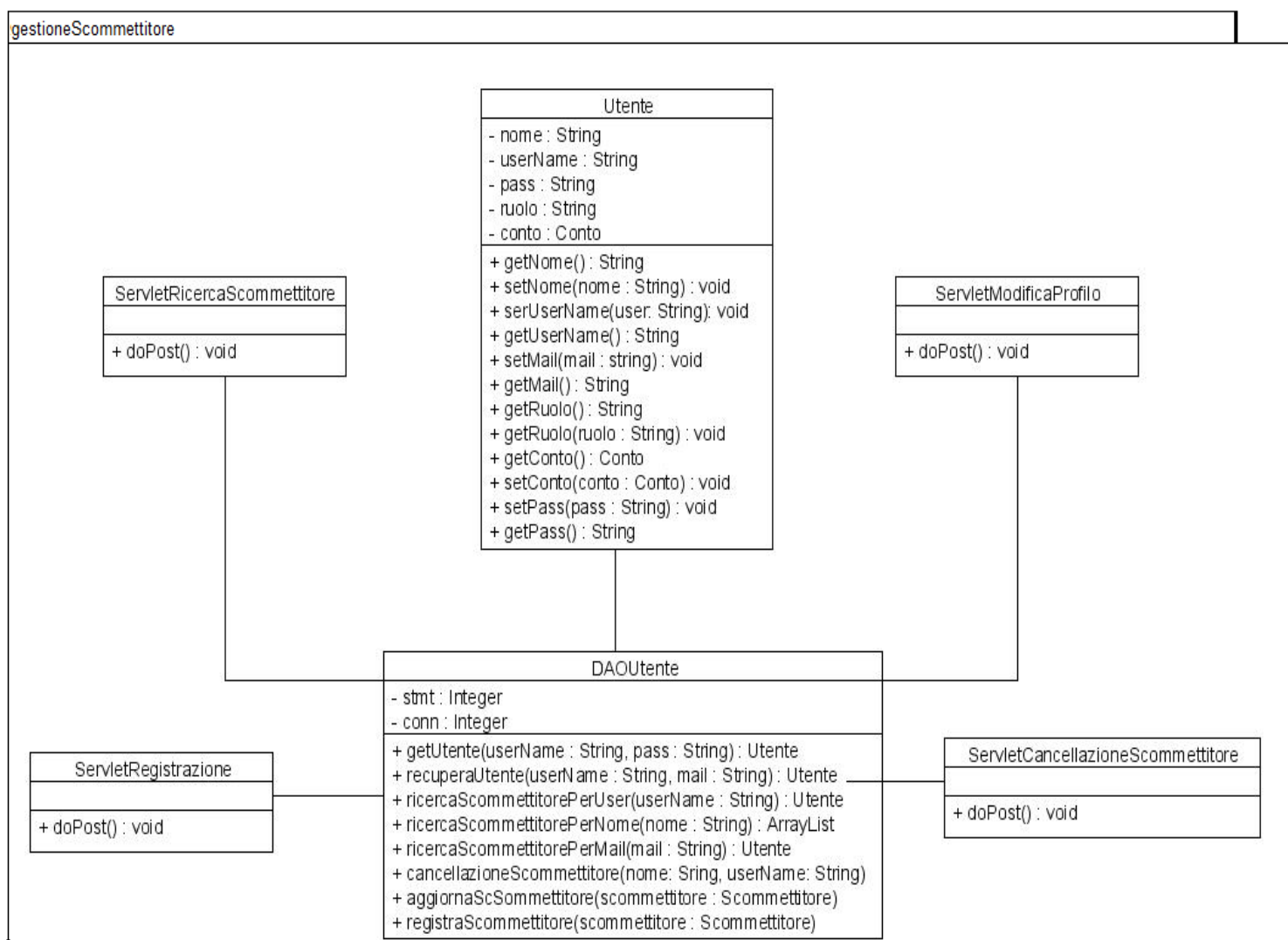
Classe	ServletVisualizzaConto		Versione: 1.0		
			Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di visualizzazione dei dati del Conto				
Dipendenze	Dipende da DAOConto				
Attributi					
Nome	Accesso		Descrizione		
Metodi					
Signature	doPost()		Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException	

Classe	ServletPrelievo		Versione: 1.0		
			Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di prelievo sul conto personale.				
Dipendenze	Dipende da DAOOperazione				
Attributi					
Nome	Accesso		Descrizione		
Metodi					
Signature	doPost()		Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException	

Classe	ServletVersamento		Versione: 1.0		
			Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione versamento sul conto personale.				
Dipendenze	Dipende da DAOOperazione				
Attributi					
Nome		Accesso	Descrizione		
Metodi					
Signature	doPost()		Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException	



### 3.4 Gestione dello Scommettitore



Classe	Utente	Versione: 1.0
		Autore: Donato Sabia
Descrizione	Classe che rappresenta l'utente registrato.	
Dipendenze	Non ha dipendenze.	
Attributi		
Nome	Accesso	Descrizione
nome	Privato	Nome dell'Utente.
userName	Privato	Pseudonimo dell'utente registrato.
pass	Privato	Password di accesso.
ruolo	Privato	Il ruolo dell'Utente che può essere 'Amministratore' oppure 'Scommettitore'
mail	Privato	L'indirizzo e-mail dell'Utente.
conto	Privato	Il Conto associato all'utente se questo è uno scommettitore altrimenti sarà null.

Metodi				
<b>Signature</b>	setUserName(userName : String)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Consente di inserire l'userName dell'Utente.			
<b>Signature</b>	getUserName()	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Restituisce l'userName dell'utente.			
<b>Signature</b>	getNome()	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Restituisce il nome anagrafico dell'utente.			
<b>Signature</b>	setNome (nome: String)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Assegna il nome dell'Utente.			
<b>Signature</b>	setPass (pass : String)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Imposta la password dell'Utente.			
<b>Signature</b>	getPass ()	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Restituisce la password crittografata.			
<b>Signature</b>	setMail (mail : String)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Assegna l'indirizzo email dell'Utente.			
<b>Signature</b>	getMail ()	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Restituisce l'indirizzo email dell'Utente.			
<b>Signature</b>	setConto (conto : Conto)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Assegna un conto all'Utente se questo è 'Scommettitore'			
<b>Signature</b>	getRuolo ()	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Restituisce il ruolo relativo all' Utente .			
<b>Signature</b>	setRuolo (ruolo : String)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Assegna il ruolo all'Utente.			

Classe	DAOUtente		Versione: 1.0	
			Autore: Donato Sabia	
Descrizione	La classe DAOUtente si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di un Utente.			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso	Descrizione		
stmt	Privato	Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce		
conn	Privato	Oggetto di tipo Connection che gestisce una connessione con uno specifico database		
Metodi				
Signature	getUtente(userName : String , pass : String)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce un utente dal DBMS in base all'userName e alla password.			SQLException
Signature	recuperaUtente(userName : String, mail : String)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce un utente dal DBMS in base all'userName e alla mail (nel caso di recupero dati del profilo utente).			SQLException
Signature	ricercaScommettitorePerUser(userName : String)	Accesso	Pubblico	Eccezioni
Descrizione	Effettua una ricerca nel DBMS tra tutti gli utenti aventi il ruolo di "Scommettitore", utilizzando come parametro di ricerca l'userName dell'utente ) e restituisce un oggetto di tipo Utente.			SQLException
Signature	ricercaScommettitorePerNome(nome : String)	Accesso	Pubblico	Eccezioni
Descrizione	Effettua una ricerca nel DBMS tra tutti gli utenti aventi il ruolo di "Scommettitore", utilizzando come parametro di ricerca il nome dell'utente ) e restituisce una lista di Utenti (il metodo restituirà una lista poiché esiste la possibilità che più utenti abbiano il medesimo nome) .			SQLException
Signature	ricercaScommettitorePerMail(mail : String)	Accesso	Pubblico	Eccezioni
Descrizione	Effettua una ricerca nel DBMS tra tutti gli utenti aventi il ruolo di "Scommettitore", utilizzando come parametro di ricerca l'e-mail dell'utente ) e restituisce un oggetto di tipo Utente.			SQLException

<b>Signature</b>	cancellazione Scommettitore (nome : String, userName : String)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Effettua una ricerca nel DBMS tra tutti gli utenti aventi il ruolo di "Scommettitore", utilizzando come parametro di ricerca il nome e l'userName e provvede all'eliminazione dello Scommettitore.			SQLException
<b>Signature</b>	aggiornaScommettitore (scommettitore : Scommettitore)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Permette di modificare il profilo dello scommettitore.			SQLException
<b>Signature</b>	registraScommettitore(scommettitore : Scommettitore)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Permette di inserire un nuovo Scommettitore all'interno del DBMS.			SQLException

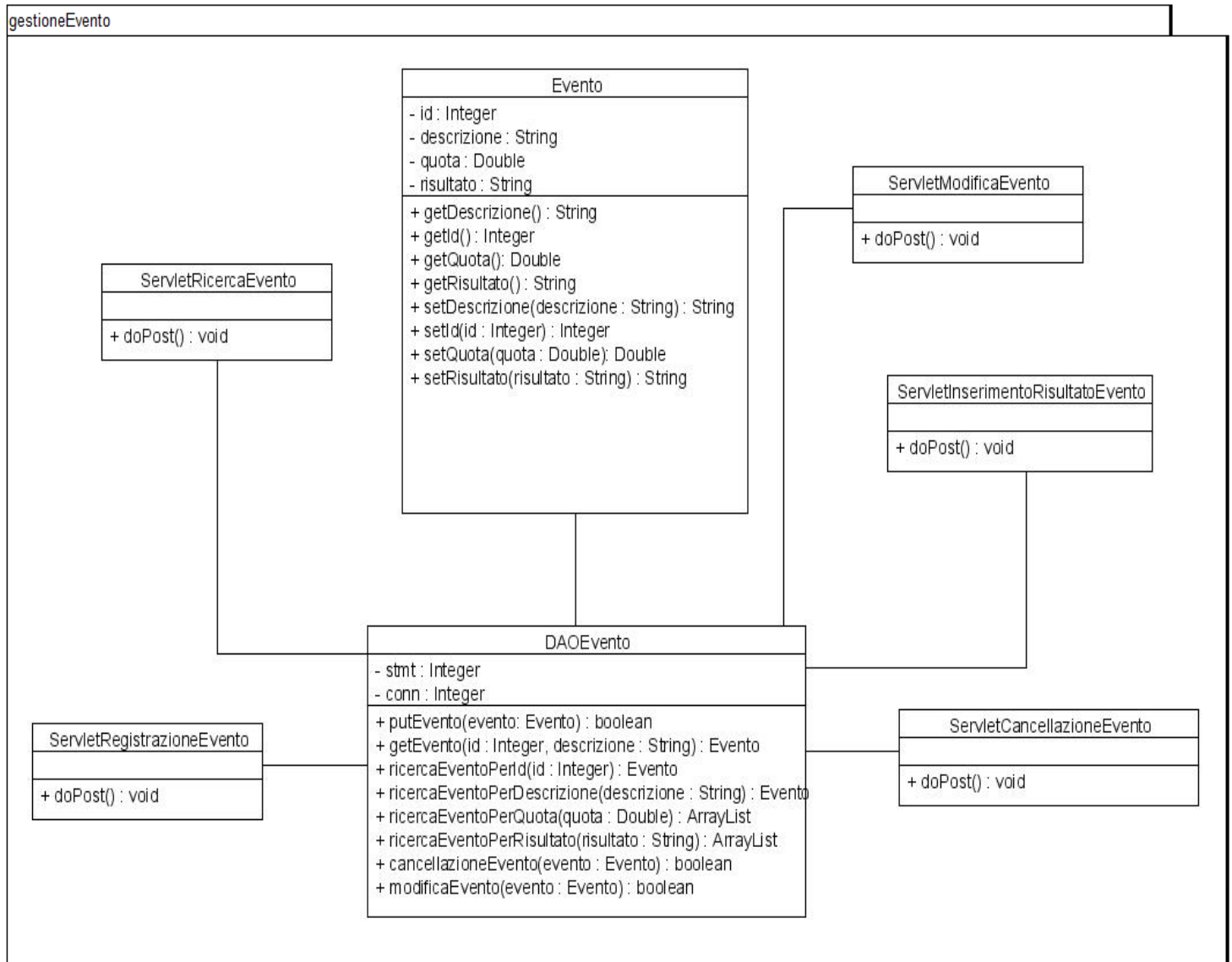
Classe	ServletRicercaScommettitore	Versione: 1.0		
		Autore: Donato Sabia		
Descrizione	Servlet che permette l'operazione di ricerca dello scommettitore mediante diversi parametri di ricerca			
Dipendenze	Dipende da DAOUtente			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletRegistrazione	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	Servlet che permette l'operazione di registrazione di un utente con il ruolo di "Scommettitore" all'interno del DBMS.			
Dipendenze	Dipende da DAOUtente			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletModificaProfilo	Versione: 1.0		
		Autore: Donato Sabia		
Descrizione	Servlet che permette di modificare i dati inerenti al profilo di un Utente.			
Dipendenze	Dipende da DAOUtente			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletCancellazioneScommettitore	Versione: 1.0		
		Autore: Donato Sabia		
Descrizione	Servlet che permette di eliminare dal DBMS un Utente il cui ruolo è quello di “Scommettitore”.			
Dipendenze	Dipende da DAOUtente			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

### 3.5 Gestione Evento



Classe	Evento	Versione: 1.0		
		Autore: Donato Sabia		
Descrizione	Classe che rappresenta l'evento che identifica la manifestazione sportiva oggetto di scommessa.			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso	Descrizione		
Id	Privato	Codice univoco identificativo dell'evento.		
Descrizione	Privato	Stringa che contiene una descrizione dell'evento.		
Quota	Privato	Valore da moltiplicare all'importo della scommessa per poter determinare l'ammontare dell'eventuale vincita.		
Risultato	Privato	Esito dell'evento.		
Metodi				
Signature	getDescription()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce la descrizione dell'evento.			
Signature	getId ()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce il codice univoco dell'evento.			
Signature	getQuota()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce il valore di tipo double che individua il moltiplicatore associato all'evento, utile al fine del calcolo dell'eventuale vincita.			
Signature	getRisultato()	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce la stringa che contiene l'esito dell'evento.			
Signature	setDescription (descrizione : String)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta la descrizione dell'evento.			
Signature	setId (id : Integer)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta il codice univoco dell'evento.			
Signature	setQuota (quota : Double)	Accesso	Pubblico	Eccezioni
Descrizione	Assegna il valore di tipo double che individua il moltiplicatore associato all'evento, utile al fine del calcolo dell'eventuale vincita.			
Signature	setRisultato (risultato : String)	Accesso	Pubblico	Eccezioni
Descrizione	Imposta l'esito dell'evento.			

Classe	DAOEvento		Versione: 1.0	
			Autore: Donato Sabia	
Descrizione	La classe DAOEvento si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di un Evento.			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso	Descrizione		
stmt	Privato	Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce		
conn	Privato	Oggetto di tipo Connection che gestisce una connessione con uno specifico database		
Metodi				
Signature	putEvento(evento : Evento)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di inserire un evento nel DBMS e restituisce 'true' se l'operazione va a buon fine.			SQLException
Signature	getEvento(Id : Integer, descrizione : String)	Accesso	Pubblico	Eccezioni
Descrizione	Restituisce un evento dal DBMS in base all'Id e alla descrizione dell'evento.			SQLException
Signature	ricercaEventoPerId( id : Integer)	Accesso	Pubblico	Eccezioni
Descrizione	Effettua una ricerca nel DBMS tra tutti gli eventi, utilizzando come parametro di ricerca l'Id dell'evento e restituisce un oggetto di tipo Evento.			SQLException
Signature	ricercaEventoPerDescrizione( descrizione : String)	Accesso	Pubblico	Eccezioni
Descrizione	Effettua una ricerca nel DBMS tra tutti gli eventi, utilizzando come parametro di ricerca la descrizione dell'evento e restituisce un oggetto di tipo Evento.			SQLException
Signature	ricercaEventoPerQuota( quota : Double)	Accesso	Pubblico	Eccezioni
Descrizione	Effettua una ricerca nel DBMS tra tutti gli eventi, utilizzando come parametro di ricerca la quota dell'evento e restituisce una Lista di eventi.			SQLException
Signature	ricercaEventoPerRisultato( risultato : String)	Accesso	Pubblico	Eccezioni
Descrizione	Effettua una ricerca nel DBMS tra tutti gli eventi, utilizzando come parametro di ricerca il risultato dell'evento e restituisce una Lista di eventi.			SQLException



<b>Signature</b>	cancellazioneEvento( evento : Evento)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Permette di eliminare un evento dalla base di dati e restituisce true se l'eliminazione è avvenuto con successo.			SQLException
<b>Signature</b>	modificaEvento( evento : Evento)	<b>Accesso</b>	Pubblico	<b>Eccezioni</b>
<b>Descrizione</b>	Permette di modificare le proprietà di un evento.			SQLException

Classe	ServletRicercaEvento	Versione: 1.0		
		Autore: Donato Sabia		
Descrizione	Servlet che permette l'operazione di ricerca dell'Evento.			
Dipendenze	Dipende da DAOEvento			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

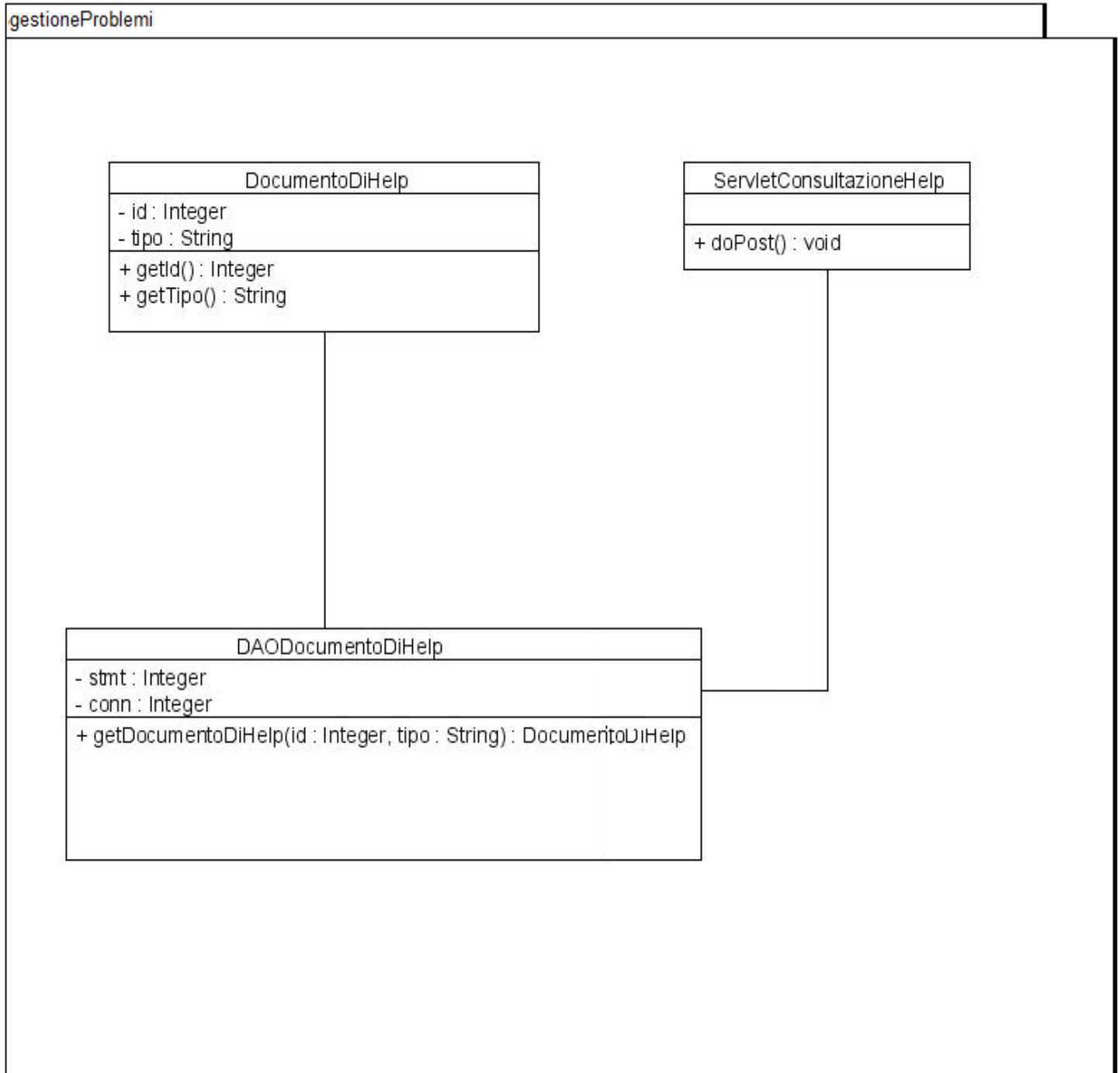
Classe	ServletRegistrazioneEvento	Versione: 1.0		
		Autore: Donato Sabia		
Descrizione	Servlet che permette l'operazione di registrazione dell'Evento.			
Dipendenze	Dipende da DAOEvento			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletModificaEvento		Versione: 1.0	
			Autore: Antonio Genovese	
Descrizione	Servlet che permette di modificare le proprietà relative ad un Evento.			
Dipendenze	Dipende da DAOEvento			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletInserimentoRisultatoEvento		Versione: 1.0	
			Autore: Donato Sabia	
Descrizione	Servlet che permette di inserire il risultato di un Evento.			
Dipendenze	Dipende da DAOEvento			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

Classe	ServletCancellazioneEvento	Versione: 1.0		
		Autore: Donato Sabia		
Descrizione	Servlet che permette di eseguire l'operazione di rimozione dell'Evento.			
Dipendenze	Dipende da DAOEvento			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni
Descrizione	È il metodo che viene eseguito quando viene generata una POST da parte della relativa pagina JSP che l'attiva			ServletException, IOException

### 3.6 Gestione Problemi



<b>Classe</b>	DocumentoDiHelp	<b>Versione:</b> 1.0
		<b>Autore:</b> Donato Sabia
<b>Descrizione</b>	Classe che rappresenta il documento di help messo a disposizione dell'utente per la risoluzione di problemi e per facilitare l'utilizzo del sistema.	
<b>Dipendenze</b>	Non ha dipendenze.	

Attributi					
Nome		Accesso	Descrizione		
Id		Privato	Codice univoco identificativo del documento di help.		
Tipo		Privato	Stringa che descrive il tipo del documento di help.		
Metodi					
Signature	getId ()		Accesso	Pubblico	Eccezioni
Descrizione	Restituisce il codice univoco del documento di help.				
Signature	getTipo()		Accesso	Pubblico	Eccezioni
Descrizione	Restituisce la stringa che indica il tipo di documento di help.				

Classe	DAODocumentoDiHelp		Versione: 1.0	
			Autore: Donato Sabia	
Descrizione	La classe DAODocumentoDiHelp si occupa di gestire le connessioni con il Database e permette,quindi, di caricare dalla persistenza i dati di un Documento di Help.			
Dipendenze	Non ha dipendenze.			
Attributi				
Nome	Accesso		Descrizione	
stmt	Privato		Oggetto di tipo Statement usato per eseguire un comando SQL e per restituire i risultati che produce	
conn	Privato		Oggetto di tipo Connection che gestisce una connessione con uno specifico database	
Metodi				
Signature	getDocumentoDiHelp (id : Integer, tipo: String)	Accesso	Pubblico	Eccezioni
Descrizione	Consente di prelevare un documento di help dal DBMS in base al id e al tipo.Tale metodo restituisce 'true' se l'operazione è andata a buon fine.			SQLException

Classe	ServletConsultazioneHelp	Versione: 1.0		
		Autore: Antonio Genovese		
Descrizione	Servlet che permette di eseguire l'operazione di consultazione di un documento di help.			
Dipendenze	Dipende da DAOEvento			
Attributi				
Metodi				
Signature	doPost()	Accesso	Pubblico	Eccezioni