Piano di collaudo PRApp Server

Sommario

[Test di Context 2](#_Toc508143538)

[createContext(user:WUtente):void 2](#_Toc508143539)

[getContext():Context 2](#_Toc508143540)

[deleteContext():void 2](#_Toc508143541)

[isValid():boolean 2](#_Toc508143542)

[getUtente():WUtente 2](#_Toc508143543)

[Test di Hash 2](#_Toc508143544)

[getSingleton(): Hash 2](#_Toc508143545)

[calculateHash(str:String):String 2](#_Toc508143546)

[evalutateHash(hashA:String, hashB:String):boolean 2](#_Toc508143547)

[generate Salt(length:uint):String 3](#_Toc508143548)

[Test di Utente 3](#_Toc508143549)

[getInstance():Utente 3](#_Toc508143550)

[registrazione(wrapper:WUtente, username:String, password:String) : void 3](#_Toc508143551)

[login(username:String, password:String) : boolean 3](#_Toc508143552)

[creaStaff(nome:String, codiceAccesso:String): WStaff 3](#_Toc508143553)

[accediStaff(wrapper:WStaff, codiceAccesso:String) : boolean 3](#_Toc508143554)

[getListaStaff() : List<WStaff> 4](#_Toc508143555)

[getListaStaffMembri() : List<WStaff> 4](#_Toc508143556)

[Test di Membro 4](#_Toc508143557)

[getInstance() : Membro 4](#_Toc508143558)

[getListaUtenti(wrapper:WStaff) : List<WUtente> 4](#_Toc508143559)

[getDirittiUtenti(wrapper:WStaff) : Map<WUtente, List<Diritti>> 4](#_Toc508143560)

[getListaEventi(wrapper:WStaff) : List<WEvento> 5](#_Toc508143561)

[getTipiPrevendita (wrapper: WEvento) : List<WTipoPrevendita> 5](#_Toc508143562)

[getListaClienti (wrapper:WStaff) : List<WCliente> 5](#_Toc508143563)

# Test di Context

## createContext(user:WUtente):void

Consente di creare un contesto di sessione.

* Exception: la sessione non può essere utilizzata (non attiva).
* InvalidArgumentException: user nullo oppure non valido.

**Verifica:** controllo delle variabili di sessione, controllo dei dati di login. Test di InvalidArgumentException.

## getContext():Context

Restituisce un contesto: potrebbe essere valido oppure no.

* Exception: la sessione non può essere utilizzata (non attiva).

**Verifica:** controllo delle variabili di sessione, controllo dei dati di login.

## deleteContext():void

Elimina il contesto.

* Exception: la sessione non può essere utilizzata (non attiva).

**Verifica:** controllo delle variabili di sessione.

## isValid():boolean

Deve restituire true se il login è avvenuto con successo.

**Verifica:** controllo delle variabili di sessione, controllo dei dati di login.

## getUtente():WUtente

Deve restituire un WUtente valido se isValid() true.

**Verifica:** controllo delle variabili di sessione, controllo dei dati di login, controllo isValid.

# Test di Hash

## getSingleton(): Hash

Restituisce l’oggetto singleton già istanziato.

**Verifica:** controllo null sull’oggetto.

## hashPassword (password:String):String

Calcola un hash utilizzando l’algoritmo definito in “algorithm:int”.

* InvalidArgumentException: str nulla oppure non di tipo stringa.

**Verifica:** controllo utilizzando le librerie definite da PHP.

## evalutatePassword (password:String, hash:String):boolean

Consente di comparare due hash in modo “sicuro”.

* InvalidArgumentException: hashA e/o hashB nulla oppure non di tipo stringa.

**Verifica:** controllo utilizzando le librerie di PHP.

## generate Salt(length:uint):String

Consente di generare un salt di lunghezza preferita.

**Verifica:** controllo sulla lunghezza della stringa rispetto a length.

# Test di Utente

## getInstance():Utente

Restituisce un’istanza valida di Utente.

* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** controllo null sull’oggetto. Simulazione di guasto per PDOException.

## registrazione(wrapper:WUtente, username:String, password:String) : void

Si compone di una operazione: registrazione sul database se l’username non è già stato usato.

* NotAvailableOperationException:Si è loggati nel sistema.
* InsertException: username già utilizzato.
* InvalidArgumentException: parametri nulli o non validi.
* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** controllo dei parametri (null, e non validi), controllo username e password vuoti, registrazione ripetuta per verificare InsertException. Login e verifica di NotAvailableOperationException.

## login(username:String, password:String) : boolean

Restituisce vero se il login è stato effettuato con successo. Il contesto si aggiorna di conseguenza.

* NotAvailableOperationException:Si è loggati nel sistema.
* InvalidArgumentException: parametri nulli o non validi.
* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifca:** controllo dei parametri (null, e non validi), controllo username e password vuoti. Verifica del risultato con password sbagliata. Verifica del contesto dopo login di successo.

## creaStaff(nome:String, codiceAccesso:String): WStaff

Si compone di due operazioni: creazione dello staff e assegnazione del creatore come membro e amministratore. L’operazione è ATOMICA.

* NotAvailableOperationException: non si è loggati nel sistema.
* InvalidArgumentException: parametri nulli o non validi.
* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** controllo dei parametri (null, e non validi), controllo nome e codice accesso vuoti. Verificare che nella creazione compaia anche l’assegnazione di amministratore e di membro.

## accediStaff(wrapper:WStaff, codiceAccesso:String) : boolean

Se il codice di accesso è corretto e l’utente non è già inscritto nello staff restituisce true dopo aver inserito l’utente nella lista dei membri.

* NotAvailableOperationException:Non si è loggati nel sistema.
* InsertException: Iscrizione già avvenuta.
* InvalidArgumentException: parametri nulli o non validi.
* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** controllo dei parametri (null, e non validi), controllo codice accesso vuoto. Verifica InsertException facendo due volte l’iscrizione.

## getListaStaff() : List<WStaff>

Restituisce la lista degli staff registrati nel database.

* NotAvailableOperationException:Non si è loggati nel sistema.
* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** confronto con il database per la lista. Non loggare per verificare NotAvailableOperationException.

## getListaStaffMembri() : List<WStaff>

Restituisce la lista degli staff registrati nel database in cui l’utente è membro.

* NotAvailableOperationException:Non si è loggati nel sistema.
* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** confronto con il database per la lista. Non loggare per verificare NotAvailableOperationException.

# Test di Membro

## getInstance(): Membro

Restituisce un’istanza valida di Membro.

* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** controllo null sull’oggetto. Simulazione di guasto per PDOException.

## getListaUtenti(wrapper:WStaff) : List<WUtente>

Restituisce la lista degli utenti presenti nello staff.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sugli oggetti di mappa. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro. Non loggare per verificare NotAvailableOperationException.

## getDirittiUtenti(wrapper:WStaff) : Map<WUtente, List<Diritti>>

Restituisce tutti i diritti dei membri dello staff desiderato.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro. Non loggare per verificare NotAvailableOperationException.

## getListaEventi(wrapper:WStaff) : List<WEvento>

Restituisce tutti gli eventi dello staff desiderato.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro. Non loggare per verificare NotAvailableOperationException.

## getTipiPrevendita (wrapper:WEvento) : List<WTipoPrevendita>

Restituisce tutti i tipi di prevendita disponibili per un evento dello staff desiderato.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro. Non loggare per verificare NotAvailableOperationException.

## getListaClienti (wrapper:WStaff) : List<WCliente>

Restituisce tutti i clienti dello staff desiderato.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro. Non loggare per verificare NotAvailableOperationException.

# Test di Cassiere

## timbraEntrata(wrapper:WPrevendita) : WEntrata

Timbra una prevendita: dopo aver posto il timbro, la prevendita risulta utilizzata.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è cassiere.
* InvalidArgumentException: parametri nulli o non validi.
* InsertException: prevendita già utilizzata oppure non valida.

**Verifica:**controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException. Timbrare due volte la prevendita e inserire prevendita non valida per InsertException.

## getDatiCliente(wrapper:WPrevendita) : WCliente

Restituisce i dati di un cliente associato alla prevendita.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è cassiere.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista.Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException.

## getStatisticheCliente(wrapper:WStaff) : List<WStatitsticheCassiere>

Restituisce le statistiche del cassiere. (Entrate effettuate per evento).

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è cassiere.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException.

## getEntrateSvolte(): List<WEntrata>

Restituisce le entrate svolte dal cassiere.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è cassiere.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista.Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException.

## getPrevenditeEvento(wrapper:WEvento) : List<WPrevendita>

Restituisce le prevendite di un evento, in modo da procedere con la ricerca manuale.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è cassiere.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException.

# Test di PR

## getInstance():PR

Restituisce un’istanza valida di PR.

* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** controllo null sull’oggetto. Simulazione di guasto per PDOException.

## aggiungiCliente(wrapper:WCliente) : WCliente

Aggiunge un cliente per lo staff. Restituisce il wrapper completo.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è PR.
* InvalidArgumentException: parametri nulli o non validi.
* InsertException: cliente già inserito.

**Verifica:** controllo null sull’oggetto. Confronto con il database per il wrapper. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException. Inserire due volte il cliente per InsertException.

## aggiungiPrevendita(prev:WPrevendita) : WPrevendita

Crea una prevendita per un cliente. Restituisce il wrapper completo.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è PR.
* InvalidArgumentException: parametri nulli o non validi.
* InsertException: il cliente ha già comprato una prevendita per questo evento.

**Verifica:** controllo null sull’oggetto. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException. Inserire due volte la prevendita e inserire una prevendita non validaper InsertException.

## modificaPrevendita(prev:WPrevendita) : void

Modifica una prevendita già venduta. Si possono modificare lo stato e il tipo di prevendita.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è PR.
* InvalidArgumentException: parametri nulli o non validi.
* InsertException: la prevendita non è valida per la modifica.

**Verifica:** controllo null sull’oggetto. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException. Inserire una prevendita non valida per InsertException.

## getPrevendite(filtro:StatoPrevendita) : List<WPrevendita>

Restituisce le prevendite create dal PR con un filtro (non nullo) specifico.

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è PR.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException.

## getStatistichePR (staff:WStaff) : List<WStatitstichePR>

Restituisce le statistiche del PR. (Prevendite vendute e ricavo effettuato).

* PDOException: problemi del database (errore di connessione, errore nel database).
* NotAvailableOperationException:Non si è loggati nel sistema.
* AuthorizationException: Non si è membro dello staff citato oppure non si è PR.
* InvalidArgumentException: parametri nulli o non validi.

**Verifica:** controllo null sull’oggetto. Confronto con il database per la lista. Controllo dei parametri (null, e non validi). Verifica AuthorizationException cercando di accedere a staff non membro e di provare ad utilizzare il metodo senza avere i diritti necessari. Non loggare per verificare NotAvailableOperationException.

# Test di Amministratore

## getInstance(): Amministratore

Restituisce un’istanza valida di Amministratore.

* PDOException: problemi del database (errore di connessione, errore nel database).

**Verifica:** controllo null sull’oggetto. Simulazione di guasto per PDOException.