

Design del Sistema

In questo documento verrà presentata la descrizione della fase di progetto. La prima sezione riguarda le tecnologie usate per lo sviluppo dell'applicazione, quindi sarà illustrata l'architettura specifica. Sono definite in seguito le classi di progetto, i loro sottosistemi e componenti, partendo dalle classi di analisi e traducendole in classi di design con corrispondenti raffinazioni e aggiunte. Inoltre verrà descritta l'interazione di particolari casi d'uso attraverso diagrammi di sequenza, i quali offriranno un ulteriore approfondimento per spiegarne al meglio lo svolgimento.

Indice

1. Progetto Architettuale
 - Comunicazione Fisica
 - Comunicazione Software
2. Classi di Design
 - Modello dei Package
 - Modello delle Classi
 - Package Entità del Sistema
 - Package Entità del Dominio
 - Package Boundary
 - Package Controller
 - Package Data Access Object
 - Package Servlet

3. Diagrammi di sequenza

- Log-In
- Registrazione Utente Semplice
- Registrazione Utente Venditore
- Ricerca Utente Venditore
- Modifica Dati Scheda
- Inserimento Valutazione
- Inserimento Offerta
- Proposta Inserimento Vino
- Rimozione Valutazione
- Convalida Inserimento Vino

Progetto Architettuale

Comunicazione Fisica

Siccome Vivino è una applicazione Web, si è deciso di adottare una architettura di tipo 3-tier per la comunicazione. Sarà formata da tre "strati", il primo definisce la presentazione del sistema all'utente dove egli potrà interagirci. Il secondo è dove risiede la logica del business, con cui vengono eseguite le interazioni con l'utente e le operazioni interne dell'applicazione, infine il terzo strato riguarda la gestione dei dati (ricezione e invio) che Vivino conserva in un database PostgreSQL.

Le richieste che gli utenti invieranno verso il server di Vivino, ospitato presso una infrastruttura Cloud (IaaS) di Amazon, saranno veicolate via internet.

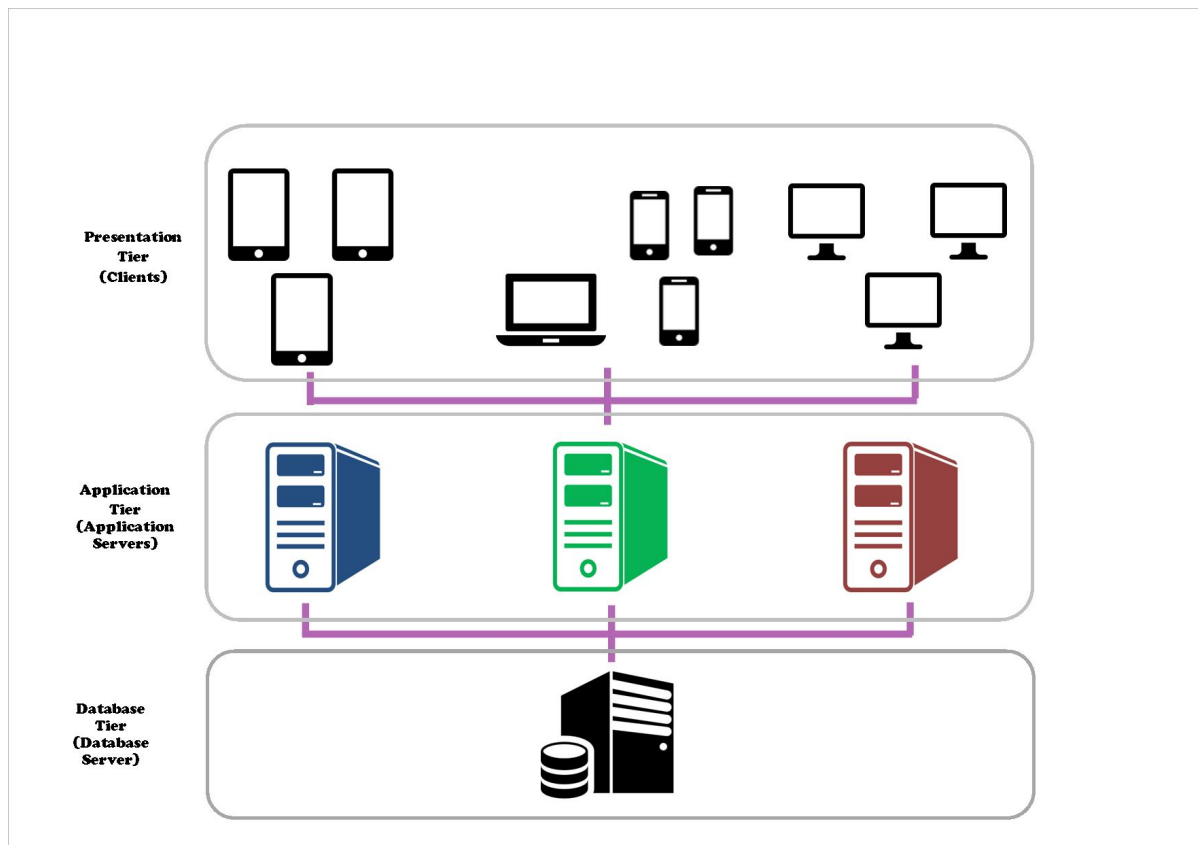


Figura 1: *Architettura 3-tier*

Comunicazione Software

La comunicazione software utilizzerà una architettura di tipo client/server composta da:

- Un Applicativo scritto in Java EE
- PostgreSQL, un RDBMS open source.
- Hibernate

Servlet

L'uso delle Servlet permette la creazione di pagine dinamiche con cui gestire l'interazione con l'utente tramite l'elaborazione delle richieste HTTP. La scelta di questa tecnologia è stata favorita dall'indipendenza delle Servlet dal server su cui operano. Nell'architettura di Vivino, ad ogni classe del package "Controller" sarà associata una Servlet che fungerà da ponte con le classi del package "Boundary".

Data Access Object e Hibernate

Il Data Access Object (DAO) è un pattern architetturale per la gestione della persistenza. Verranno fornite delle classi DAO per ogni entità dei packages "Entità del Sistema" e "Entità del Dominio" con i relativi metodi, che rappresenta una relazione di un database. I vantaggi dell'uso delle classi DAO sono quelli di impedire l'accesso diretto al database, creando maggiore livello di astrazione e fornendo una più facile manutenibilità. Il middleware Hibernate verrà utilizzato per astrarre la comunicazione tra l'applicazione e il database manager, rendendo l'applicativo database-indipendente. Verrà inoltre sfruttato l'ORM (Object-relational mapping), un servizio fornito da Hibernate che permette di effettuare una mappatura tra le classi java e le tabelle del database.

Le classi DAO comunicheranno col database tramite Hibernate.

Nell'architettura verrà utilizzata un'interazione triangolare tra i controller, le classi del DAO e le entità che, svolgendo il ruolo di DTO si occuperanno di trasferire i dati al DAO secondo le indicazioni dei relativi controller. Il DAO utilizzerà i dati ricevuti dalle entità per eseguire le operazioni comunicategli dai controller.

Nel diagramma del design delle classi verranno omesse, per chiarezza espositiva, le relazioni tra i controller e le entità, essendo le stesse descritte nel diagramma di analisi delle classi.

Classi di Design

L'elenco delle Classi di Design

Classi

Identificativo	Nome Elemento
CDD_01	Modello dei Package
CDD_02	Modello delle Classi
CDD_03	Package Entità del Sistema
CDD_04	Package Entità del Dominio
CDD_05	Package Boundary
CDD_06	Package Controller
CDD_07	Package Data Access Object
CDD_08	Package Servlet

CDD_01 - Modello dei Package

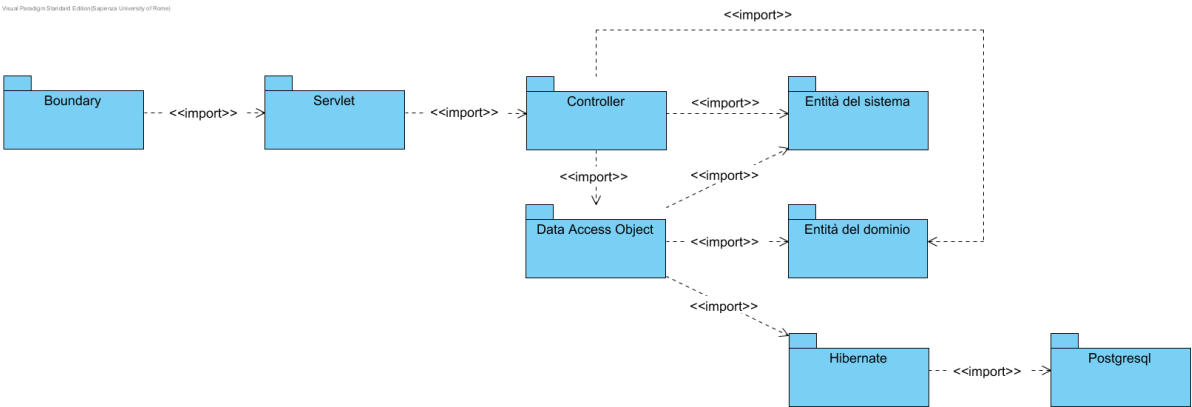


Figura 2: *Modello dei Package*

CDD_02 - Modello delle Classi

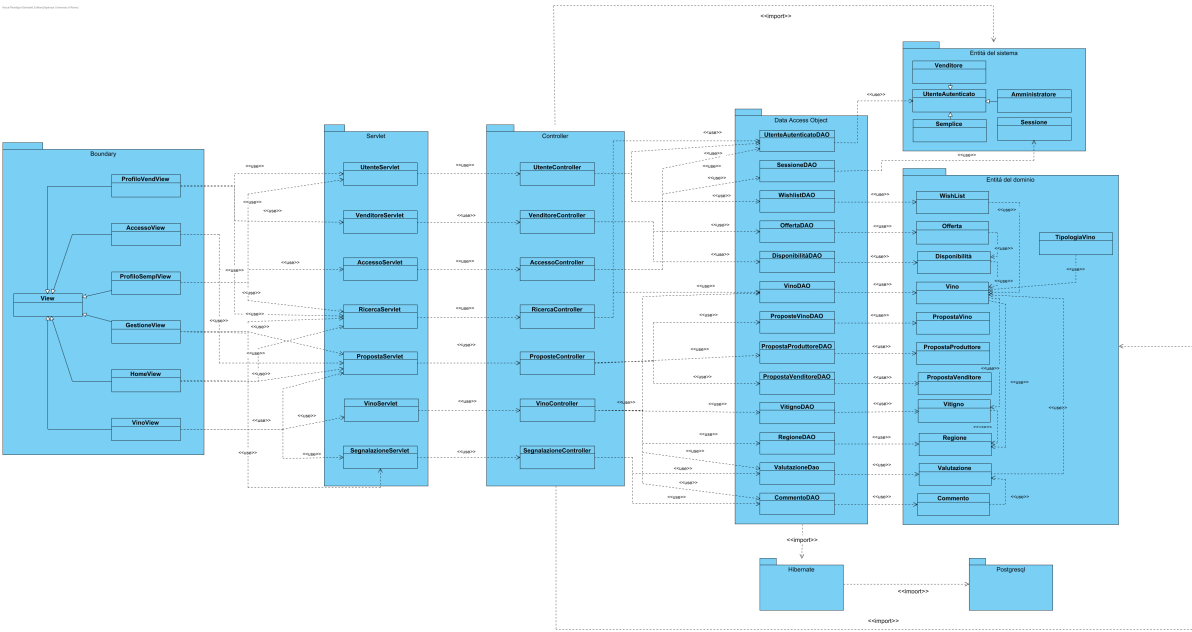


Figura 3: *Modello delle Classi*

CDD_03 - Package Entità del Sistema

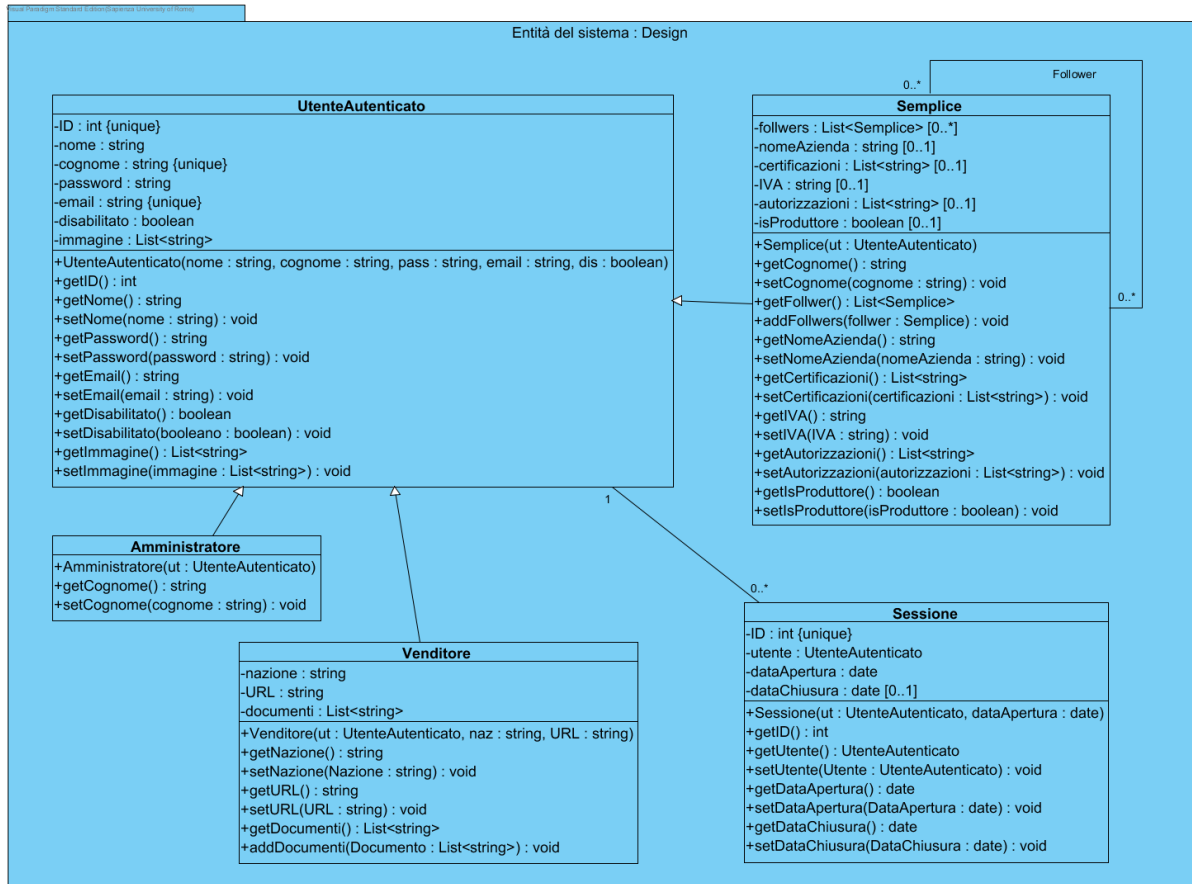


Figura 4: Package Entità del Sistema

CDD_04 - Package Entità del Dominio

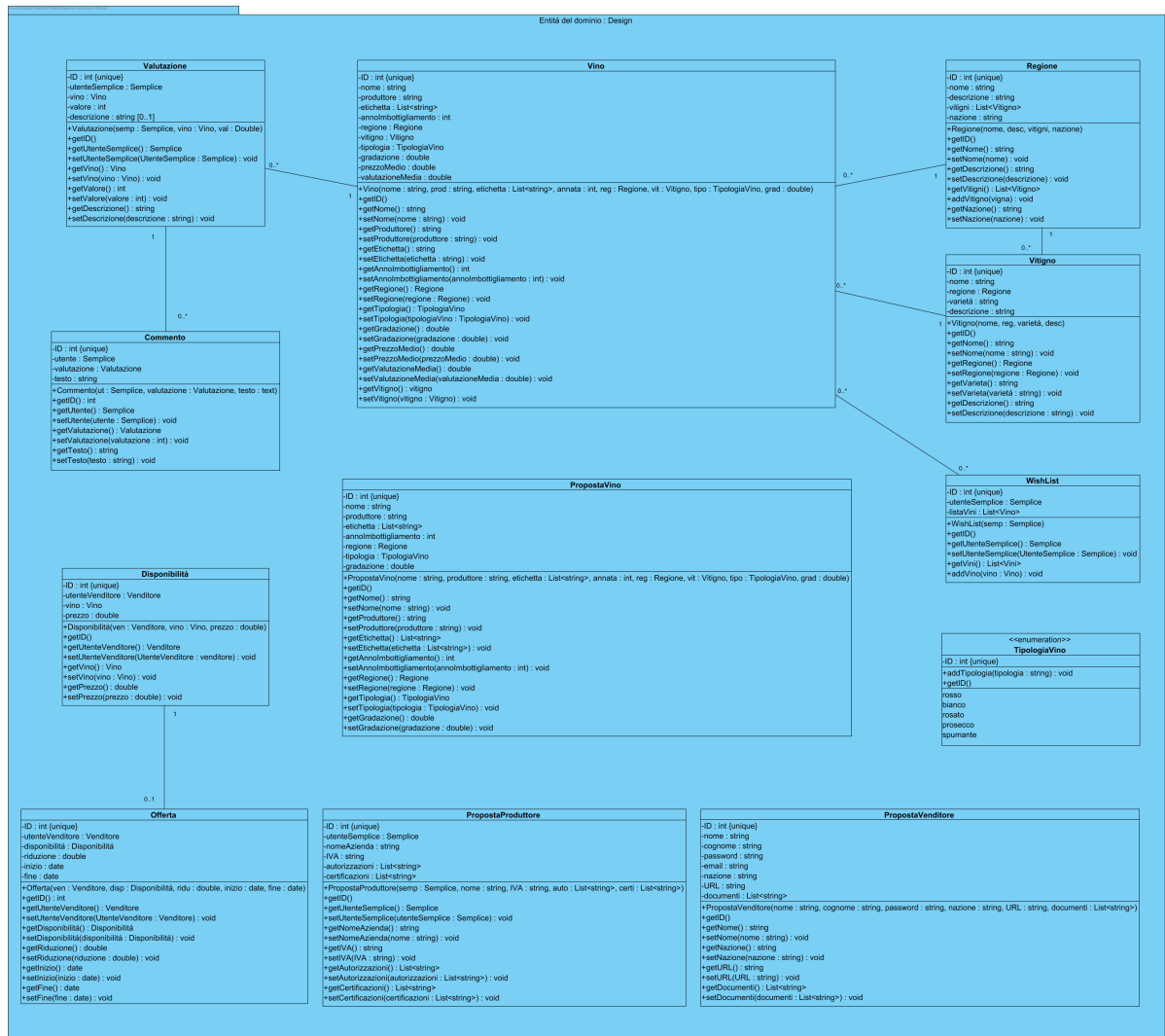


Figura 5: Package Entità del Dominio

CDD_05 - Package Boundary

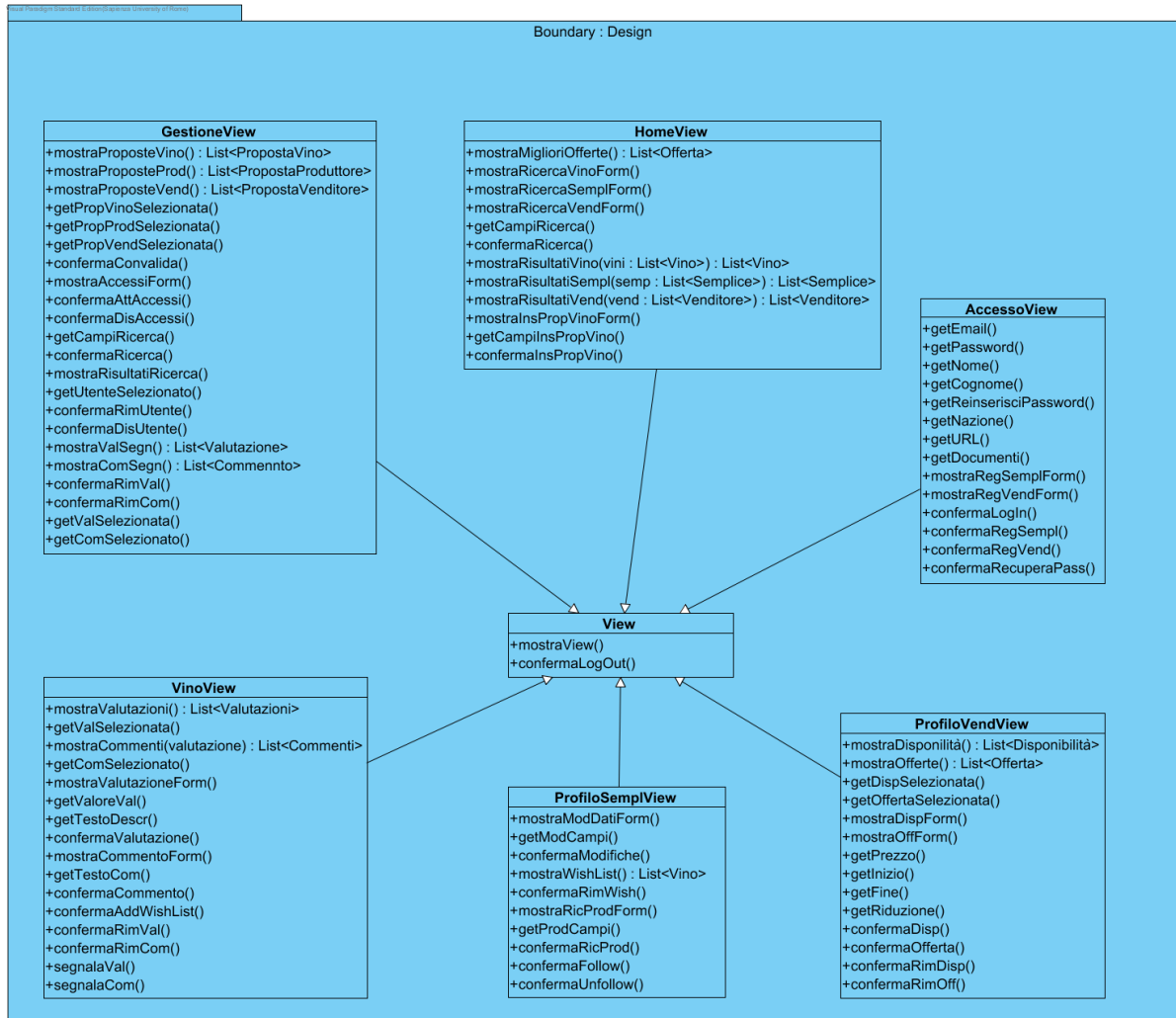


Figura 6: Package Boundary

CDD_06 - Package Controller

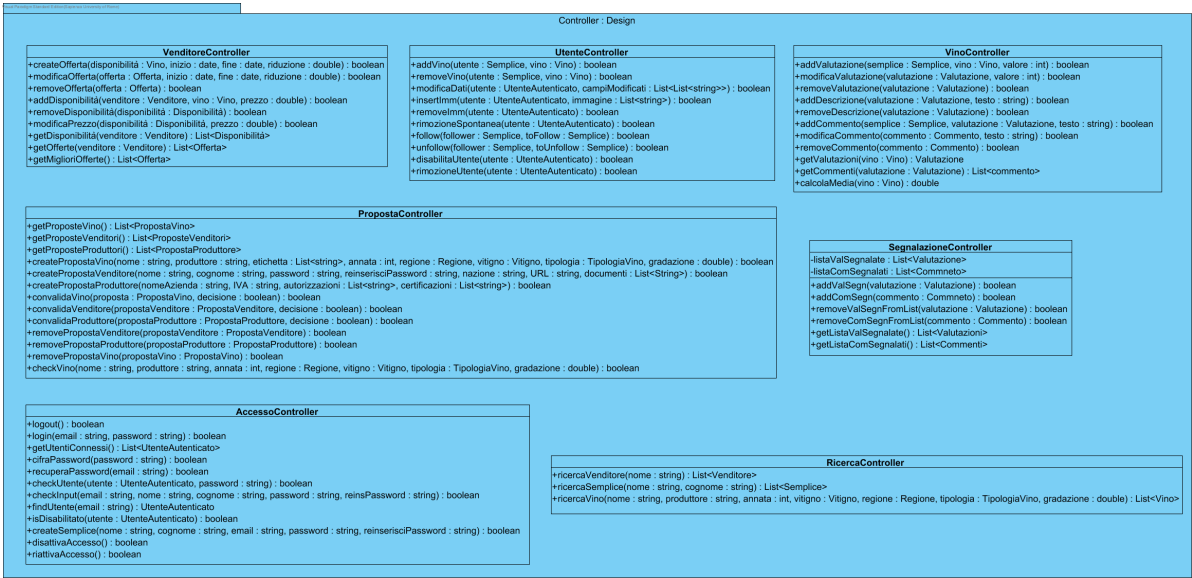


Figura 7: Package Controller

CDD_07 - Package Data Access Object

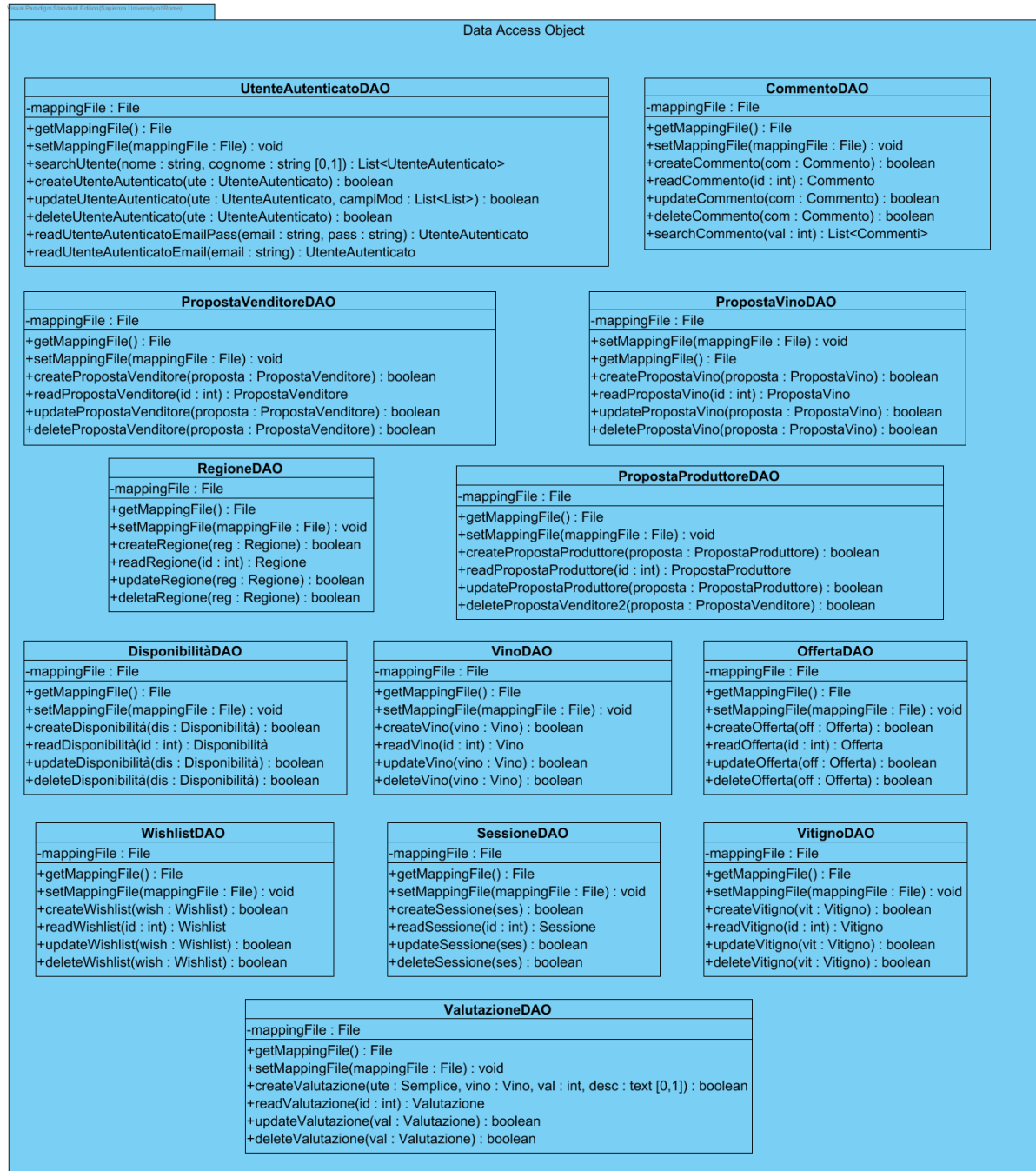


Figura 8: Package Data Access Object

CDD_08 - Package Servlet

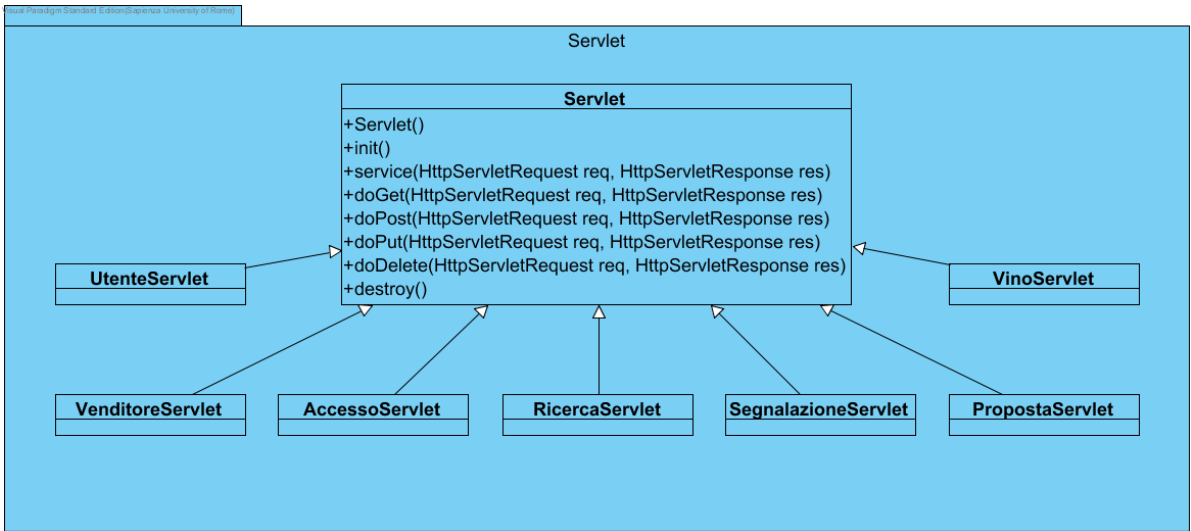


Figura 9: Package Servlet

Diagrammi di sequenza

Come già abbiamo anticipato nella descrizione dell'architettura software, la comunicazione fra le classi DAO e il database passerà tramite il middleware Hibernate. Visto che la fase di comunicazione è relativamente standard, mostreremo alcuni diagrammi di sequenza di esempio in modo da presentare il flusso di comunicazione fra le classi DAO e Hibernate.

Di seguito la tabella con la lista dei diagrammi di sequenza.

Diagrammi di Sequenza

Identificativo	Casi d'uso di riferimento
SSD_01	Log-In
SSD_02	Registrazione Utente Semplice
SSD_03	Registrazione Utente Venditore
SSD_04	Ricerca Utente Venditore
SSD_05	Modifica Dati Scheda
SSD_06	Inserimento Valutazione
SSD_07	Inserimento Offerta
SSD_08	Proposta Inserimento Vino
SSD_09	Rimozione Valutazione
SSD_10	Convalida Inserimento Vino

SSD_01 - Log-In

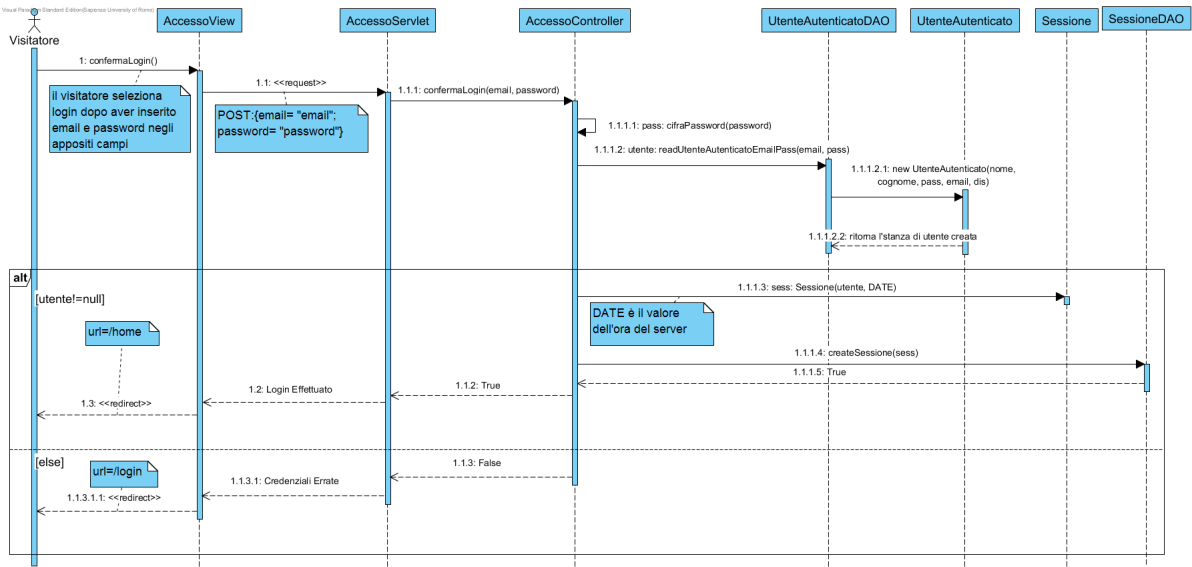


Figura 10: Log-In

SSD_02 - Registrazione Utente Semplice

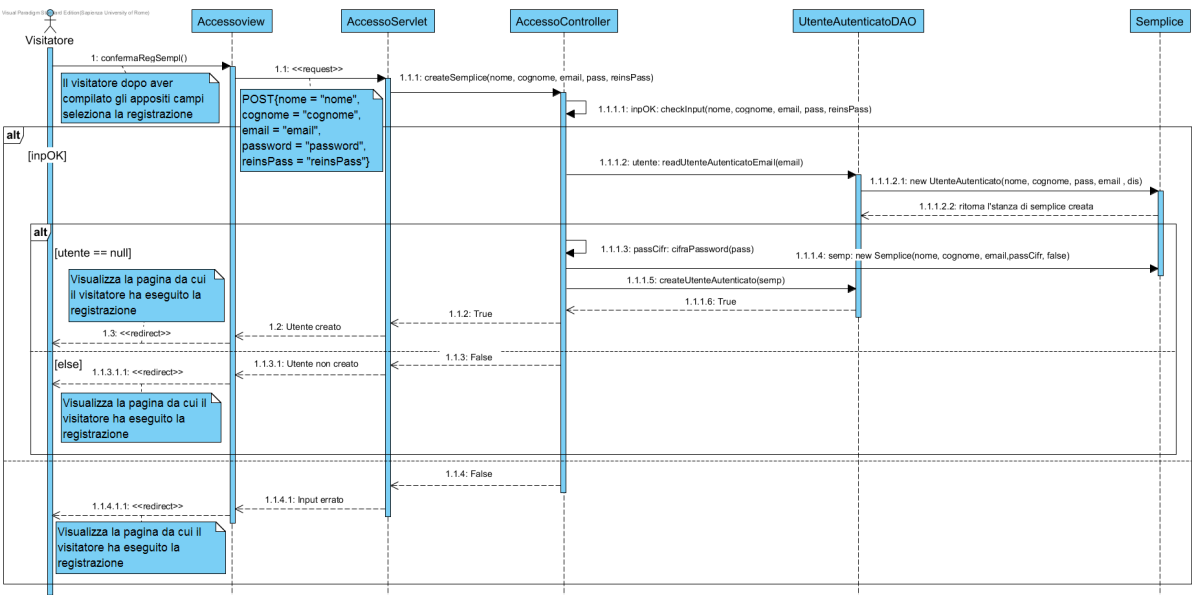


Figura 11: Registrazione Utente Semplice

SSD_03 - Registrazione Utente Venditore

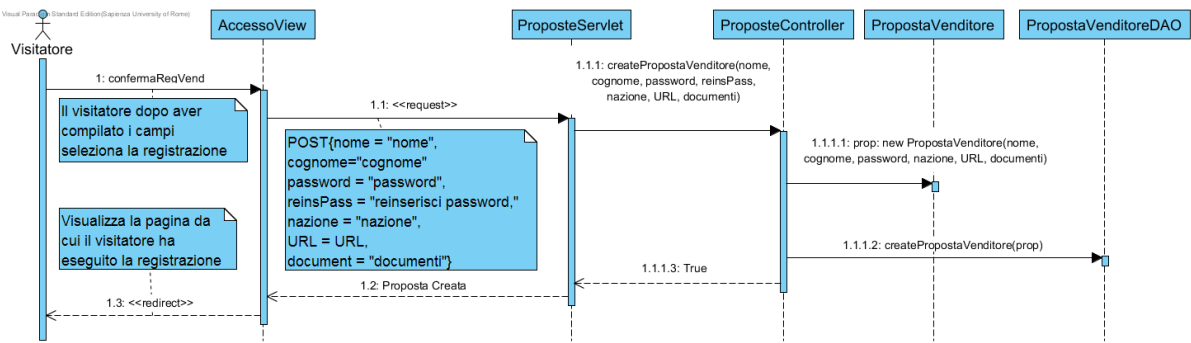


Figura 12: Registrazione Utente Venditore

SSD_04 - Ricerca Utente Venditore

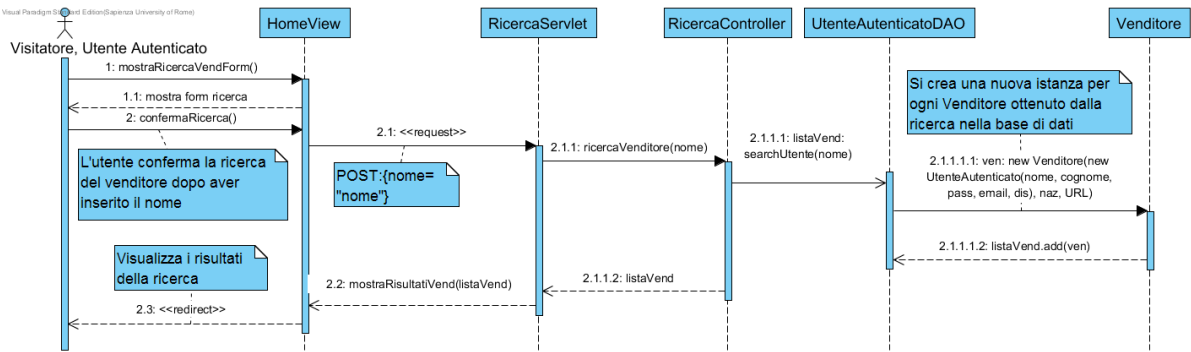


Figura 13: Ricerca Utente Venditore

SSD_05 - Modifica Dati Scheda Utente Semplice

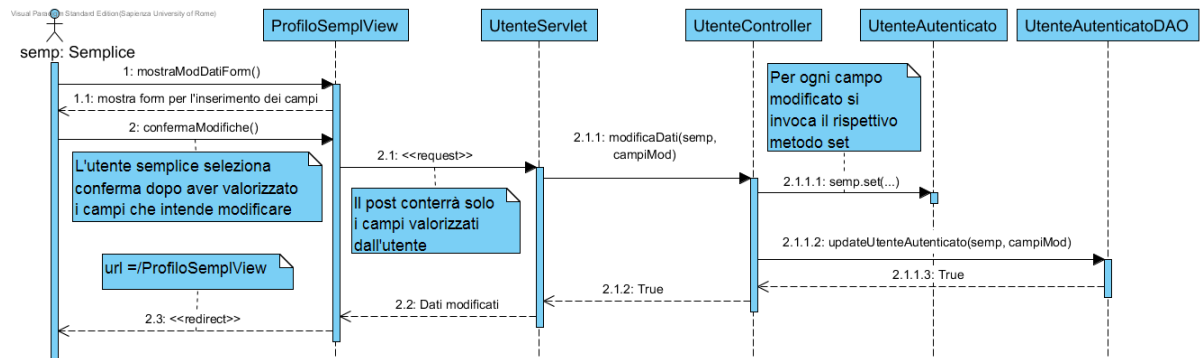


Figura 14: *Modifica Dati Scheda Utente Semplice*

SSD_06 - Inserimento Valutazione

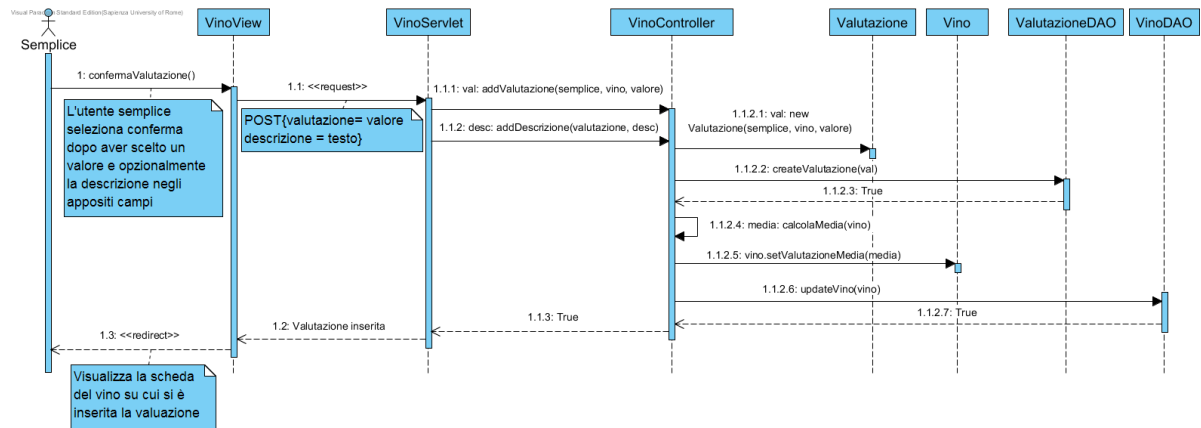


Figura 15: *Inserimento Valutazione*

SSD_07 - Inserimento Offerta

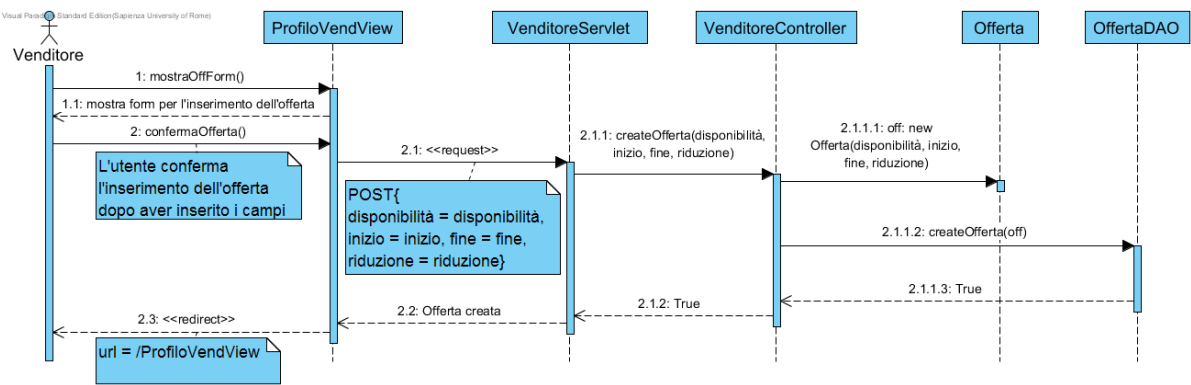


Figura 16: *Inserimento Offerta*

SSD_08 - Proposta Inserimento Vino

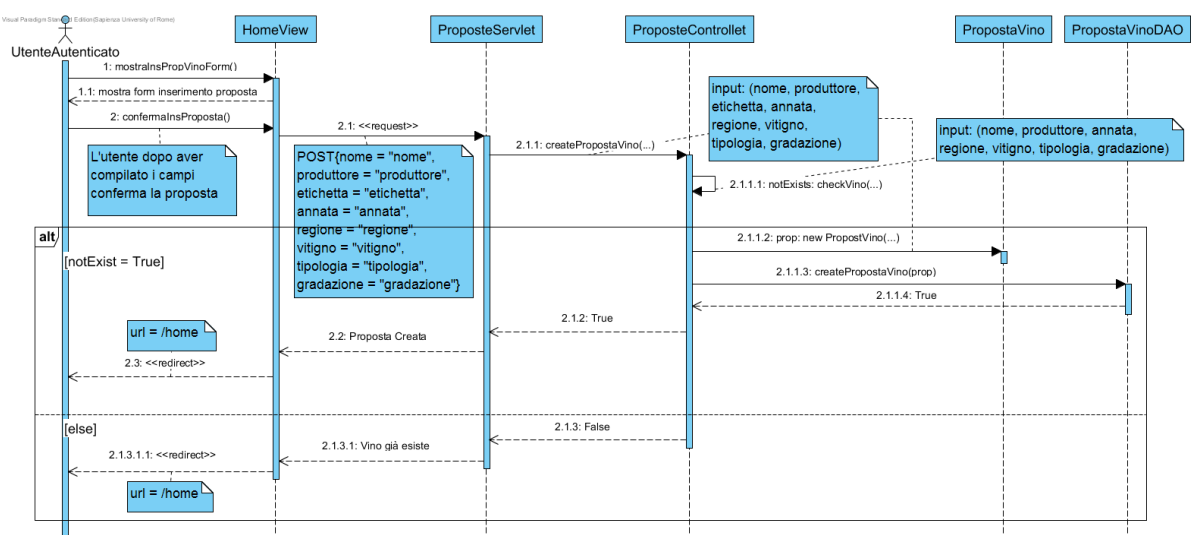


Figura 17: *Proposta Inserimento Vino*

SSD_09 - Rimozione Valutazione

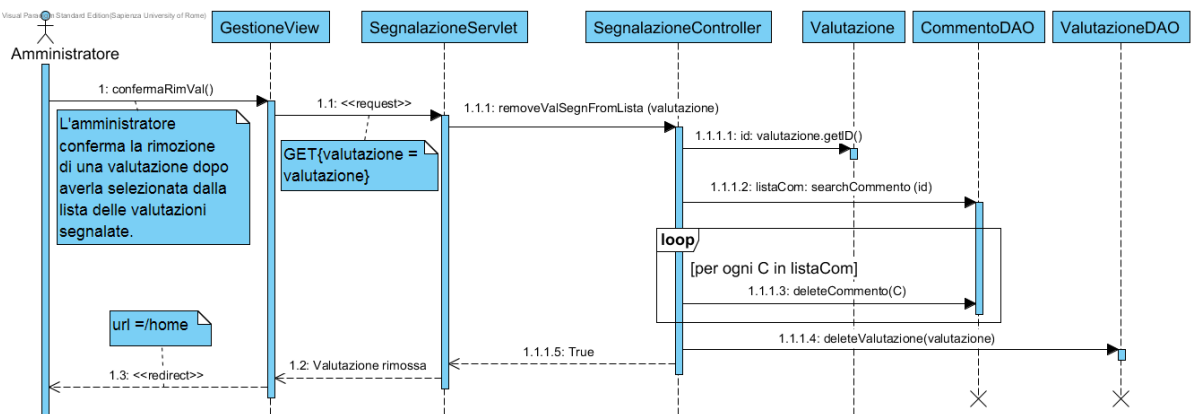


Figura 18: Rimozione Valutazione

SSD_10 - Convalida Inserimento Vino

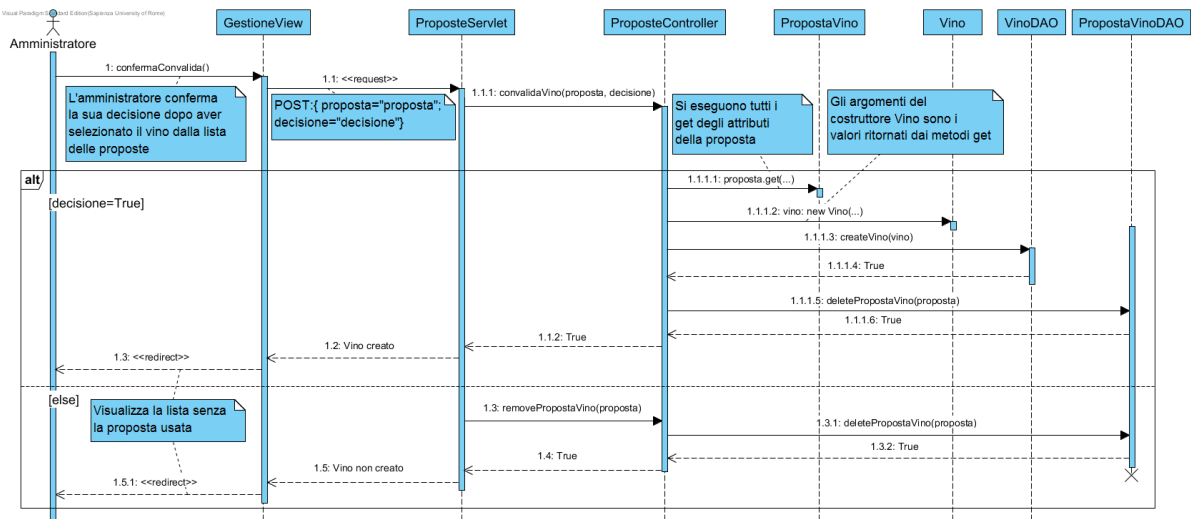


Figura 19: Convalida Inserimento Vino

Esempi di interazione con Hibernate

Si presuppone che gli oggetti DAO abbiano accesso a tutti i contenuti pubblici del package org.hibernate e che dispongano di file XML che definiscono la mappatura delle classi java a cui si riferiscono sulle tabelle del database (file ORM) che utilizzerà Hibernate previa configurazione.

Esempio Hibernate 1

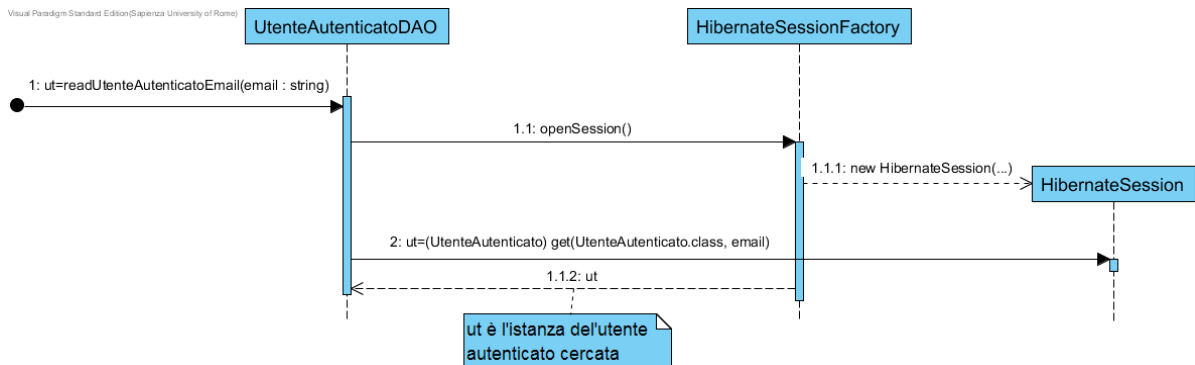


Figura 20: Operazione Ricerca Hibernate

Esempio Hibernate 2

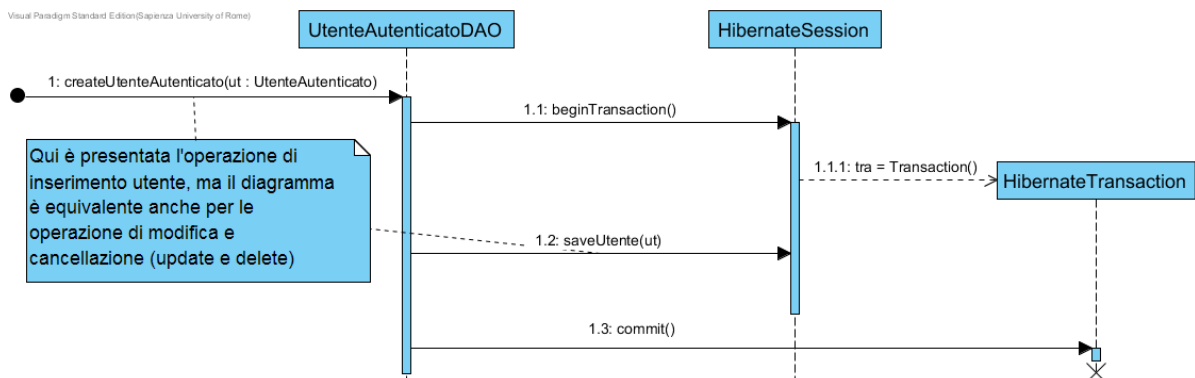


Figura 21: Transazione Hibernate

Revisioni

Numero	Descrizione	Data
1	Creazione e stesura iniziale del documento	11/05/2017
2	Aggiunta dei diagrammi di classi	20/05/2017
3	Aggiunta dei diagrammi di sequenza	28/05/2017
4	Revisione e aggiornamento	30/05/2017
5	Revisione e completamento	05/06/2017