



UNIVERSITÀ DEGLI STUDI DI SALERNO  
CORSO DI LAUREA IN INFORMATICA  
INGEGNERIA DEL SOFTWARE  
ORION  
SYSTEM DESIGN DOCUMENT

Docente: Andrea De Lucia

Studente: Michelangelo Esposito – 0512104784



## INDICE

<b>1.</b>	<b>INTRODUZIONE .....</b>	<b>3</b>
1.1	OBIETTIVI DI DESIGN .....	3
1.2	DEFINIZIONI, ACRONIMI ED ABBREVIAZIONI .....	3
<b>2.</b>	<b>ARCHITETTURA ATTUALE.....</b>	<b>3</b>
<b>3.</b>	<b>ARCHITETTURA PROPOSTA.....</b>	<b>4</b>
3.1	PANORAMICA .....	4
3.2	DECOMPOSIZIONE IN SOTTOSISTEMI.....	4
3.3	MAPPING HARDWARE/SOFTWARE.....	5
3.4	GESTIONE DEI DATI PERSISTENTI .....	6
3.4.1	PANORAMICA.....	6
3.4.2	SCHEMA DEL DATABASE.....	6
3.5	CONTROLLO DEGLI ACCESSI E SICUREZZA .....	10
3.6	CONTROLLO GLOBALE DEL SOFTWARE.....	11
3.7	CASI LIMITE.....	11
3.7.1	AVVIO ED ARRESTO .....	11
3.7.2	ECCEZIONI.....	12
<b>4.</b>	<b>SERVIZI DEI SOTTOSISTEMI.....</b>	<b>13</b>
4.1	SERVIZI ORIONSERVER .....	13
4.1.1	GESTIONEUTENZA .....	13
4.1.2	GESTIONEPRENOTAZIONI .....	13
4.1.3	GESTIONEMESSAGGI.....	13
4.2	SERVIZI SERVERINSERZIONI .....	14
4.2.1	GESTIONEINSERZIONI.....	14
4.3	SERVIZI DATABASEPROXY .....	14

# **1. INTRODUZIONE**

Questo documento si occupa di descrivere il design di Orion, in termini di architettura software utilizzata, componenti hardware di appoggio, gestione dei dati persistenti, sicurezza e controllo del software

Il focus principale del sistema è quello di fornire un servizio che sia facile da utilizzare, efficiente, mobile e disponibile ad un ampio ventaglio multiculturale di utenti.

## **1.1 OBIETTIVI DI DESIGN**

Per garantire una buona qualità del servizio, sono stati individuati degli obiettivi di design che Orion dovrà rispettare per assicurare una solida navigazione ai suoi utenti:

- **DG 01: Affidabilità e disponibilità:**  
Orion dovrà essere in grado di operare h24 7/7; i malfunzionamenti dovranno essere rari e non superare una certa soglia (dettagli in RAD sezione 3.3.2);
- **DG 02: Efficienza:**  
La navigazione su Orion dovrà essere veloce e responsiva; ciò richiede bassi ritardi, veloci operazioni di ricerca ed una buona scalabilità rispetto al numero di utenti connessi ed al numero di inserzioni presenti.
- **DG 03: Usabilità:**  
Orion dovrà garantire un'esperienza quanto più "user-friendly" possibile, permettendo agli utenti di esaminare il catalogo inserzioni in maniera facile e rendendo l'interfaccia sempre accessibile.

## **1.2 DEFINIZIONI, ACRONIMI ED ABBREVIAZIONI**

**DAO:** Data Access Object;

**DBMS:** DataBase Management System;

**HTTP:** HyperText Transfer Protocol;

**JDBC:** Java DataBase Connectivity;

**JSP:** Java Server Page;

**RAD:** Requirements Analysis Document;

**SQL:** Structured Query Language;

# **2. ARCHITETTURA ATTUALE**

Attualmente le prenotazioni vengono effettuate in una delle sedi italiane di Orion e non è in servizio nessun sistema digitale; in ogni sede, Orion offre i suoi servizi per mezzo di cataloghi che i clienti possono sfogliare prendendo un appuntamento con uno dei consulenti immobiliari;

La decisione di ampliare i propri servizi sfruttando il mercato digitale nasce da una costante richiesta da parte dei clienti italiani e di utenti esteri interessati, i quali sono interessati ad usufruire dei servizi comodamente da casa, sfruttando i propri dispositivi.

## 3. ARCHITETTURA PROPOSTA

### 3.1 PANORAMICA

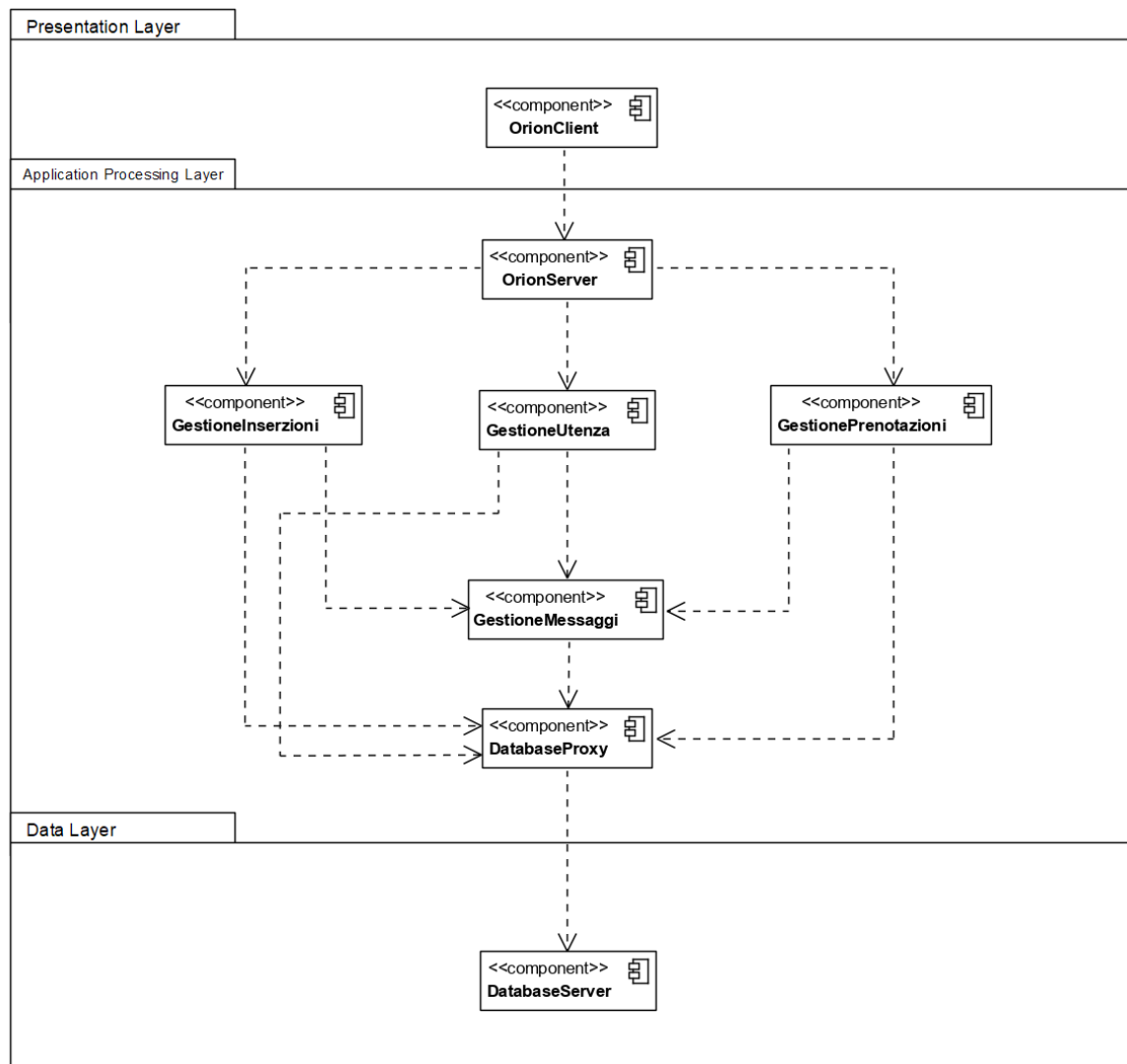
L'architettura del sistema sarà basata sul pattern MVC seguendo la divisione delle varie componenti in:

- Model: fornisce l'accesso ai dati persistenti;
- View: si occupa di gestire l'interfaccia utente basandosi sulla costruzione di pagine web dinamiche;
- Controller: gestisce le richieste dell'utente, si interfaccia al Model utilizzando metodi appositi e reindirizza il controllo alle pagine web dinamiche.

### 3.2 DECOMPOSIZIONE IN SOTTOSISTEMI

La decomposizione è stata ideata sulla base di una divisione delle funzionalità in tre strati:

- **Presentation Layer**: gestisce l'interazione con l'utente e l'invio di informazioni all' Application Processing Layer;
- **Application Processing Layer**: coordinato dalla componente OrionServer, si occupa della gestione e dell'inoltro delle richieste dell'utente in arrivo dal Presentation Layer;
- **Data Layer**: gestisce i dati persistenti del sistema;

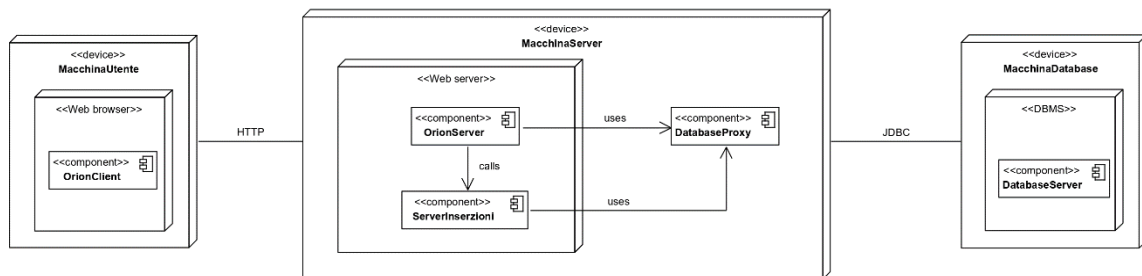


OrionClient fornisce un front-end agli utenti, permettendogli, mediante l'interfaccia grafica, di inizializzare casi d'uso come RegistrazioneCliente, AvviaPrenotazione, InserisciInserzione, ...  
 OrionServer e le relative componenti sottostanti si occupano della realizzazione della logica applicativa;

Per evitare un elevato accoppiamento tra il DatabaseServer e le componenti della logica applicativa, viene introdotta un'ulteriore componente, DatabaseProxy, la quale fornisce i servizi di accesso alle componenti interessate ed è responsabile delle interrogazioni al DatabaseServer; in questo modo, a fronte di eventuali modifiche ai meccanismi di gestione dei dati persistenti, l'unica componente da adattare ai cambiamenti sarà DatabaseProxy.

### 3.3 MAPPING HARDWARE/SOFTWARE

Dal punto di vista hardware, viene fatta una distinzione fra tre dispositivi differenti: la MacchinaUtente, su cui viene processata l'interfaccia, la MacchinaServer che gestisce la logica applicativa e le connessioni al database e la MacchinaDatabase che si occupa della gestione dei dati persistenti.



Sulla MacchinaUtente viene eseguito OrionClient, in un browser Web, desktop o mobile che soddisfi i requisiti specificati nel RAD sezione 3.3.4. OrionClient comunica con la MacchinaServer attraverso il protocollo HTTP.

Sulla MacchinaServer, il carico maggiore è previsto per la componente GestioneInserzioni, quindi, al fine di garantire una buona scalabilità rispetto al numero di inserzioni presenti e bassi tempi di risposta sulle operazioni di ricerca, viene identificato un'ulteriore componente, ServerInserzioni, dedicata alla gestione di tali operazioni; i sottosistemi, OrionServer e ServerInserzioni sono assegnati a diversi processi e sono coordinati da un server Web;

OrionServer incapsula le componenti: GestioneUtenza, GestionePrenotazioni e GestioneMessaggi; ServerInserzioni racchiude GestioneInserzioni; DatabaseProxy si occupa di stabilire la connessione alla MacchinaDatabase, utilizzando JDBC; per gestire in modo ottimale le risorse, verrà implementata una ConnectionPool per il riutilizzo delle connessioni.

Per l'implementazione delle componenti sulla MacchinaServer verrà utilizzato il framework Java Enterprise Edition, rispettando l'architettura MVC:

- **Model:** un insieme di JavaBeans costituirà le classi che rappresentano i dati persistenti ed un insieme di DAO fornirà i meccanismi di interrogazione verso il Database.
- **View:** per generare pagine web in modo dinamico e personalizzare l'interfaccia utente verranno utilizzate le JSP.
- **Controller:** le richieste dell'utente saranno gestite da un insieme di Servlet Java.

## 3.4 GESTIONE DEI DATI PERSISTENTI

### 3.4.1 PANORAMICA

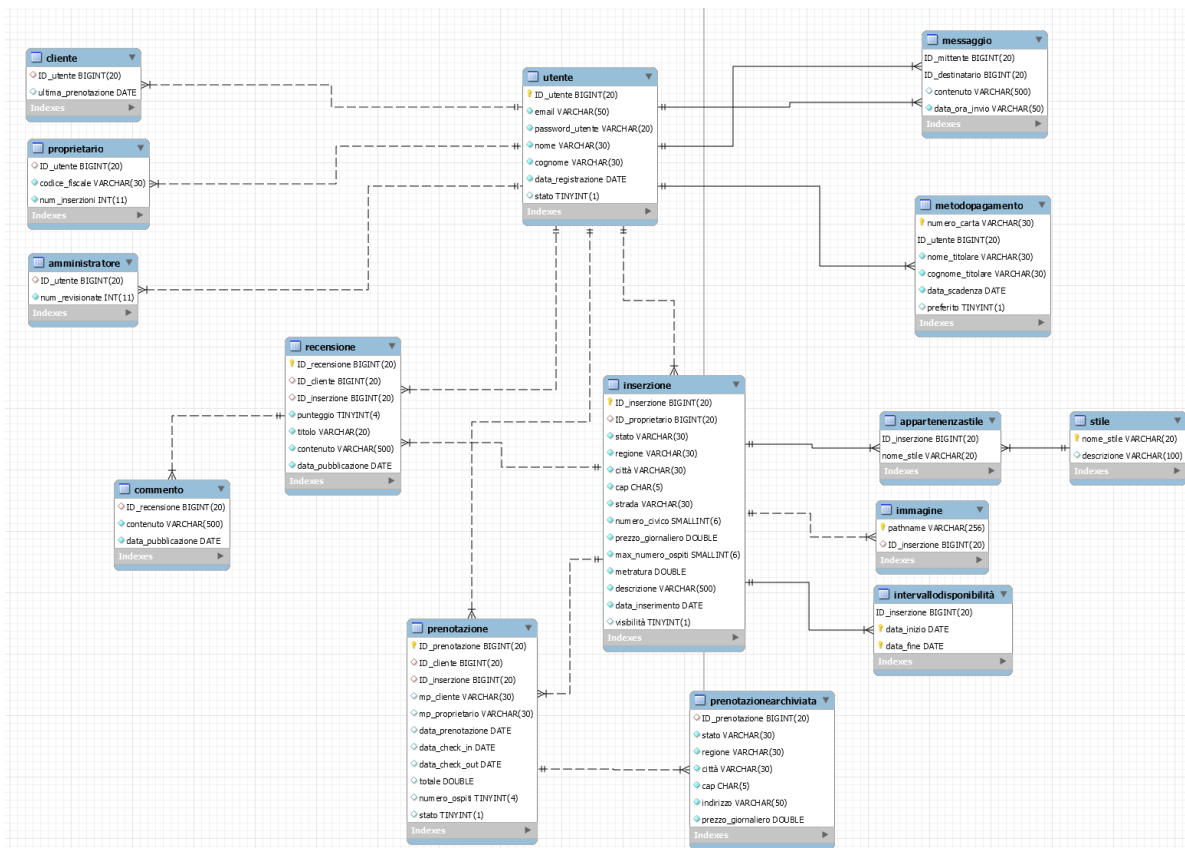
Gli oggetti da gestire in modo persistente possono essere raggruppati in tre categorie:

- Utenza e messaggistica: racchiude le credenziali degli utenti che hanno effettuato la registrazione, le informazioni aggiuntive per clienti, proprietari ed amministratore e l'archivio dei messaggi inviati nella chat di sistema;
- Inserzioni e relative recensioni: racchiude l'insieme delle inserzioni pubblicate su Orion, con rispettive immagini, stili, intervalli di disponibilità, recensioni e commenti;
- Contabilità: categoria che comprende le prenotazioni effettuate dai clienti, le prenotazioni archiviate, utilizzate per tenere traccia degli attributi di inserzioni che hanno subito modifiche o sono state rimosse ed i metodi di pagamento.

Per assicurare la sicurezza delle credenziali utente, gestire complesse operazioni di ricerca, inevitabilmente concorrenti, e prevedendo un elevato numero di inserzioni e prenotazioni archiviate presenti, verrà adottato un database relazionale per gestire la persistenza dei dati.

Il DBMS di riferimento sarà MySQL, in esecuzione sulla MacchinaDatabase.

### 3.4.2 SCHEMA DEL DATABASE



**Tabelle:**

<b>Tabella</b>	Utente
<b>Attributi e tipi di dato</b>	ID_utente: BIGINT(20) e-mail: VARCHAR(50) password_utente: VARCHAR(20) nome: VARCHAR(30) cognome: VARCHAR(30) data_registrazione: DATE stato : TINYINT(1)
<b>Chiavi</b>	Primaria: ID_utente

<b>Tabella</b>	Amministratore
<b>Attributi e tipi di dato</b>	ID_Utente: BIGINT(20) numero_inserzioni_revisionate: INT
<b>Chiavi</b>	Primaria: ID_utente Esterna: ID_utente

<b>Tabella</b>	Cliente
<b>Attributi e tipi di dato</b>	ID_Utente: BIGINT(20) data_ultima_prenotazione : DATE
<b>Chiavi</b>	Primaria: ID_utente Esterna: ID_utente

<b>Tabella</b>	Proprietario
<b>Attributi e tipi di dato</b>	ID_Utente: BIGINT(20) codice_fiscale: VARCHAR(30) numero_inserzioni_inserite: INT
<b>Chiavi</b>	Primaria: ID_utente Esterna: ID_utente (Utente)

<b>Tabella</b>	Messaggio
<b>Attributi e tipi di dato</b>	ID_mittente ID_destinatario contenuto data_ora_invio
<b>Chiavi</b>	Primaria: (ID_mittente, ID_destinatario) Esterna: ID_mittente (Utente) Esterna: ID_destinatario (Utente)

<b>Tabella</b>	Inserzione
<b>Attributi e tipi di dato</b>	ID_inserzione : BIGINT(20) ID_proprietario: BIGINT(20) stato: VARCHAR(30) regione: VARCHAR(30) città: VARCHAR(30) CAP: CHAR(5) strada: VARCHAR(30) numero_civico: SMALLINT(6) prezzo_giornaliero: DOUBLE massimo_numero Ospiti: SMALLINT(6) metratura: DOUBLE descrizione: VARCHAR(500) data_inserimento: DATE

	visibilità: TINYINT(1)
<b>Chiavi</b>	Primaria: ID_inserzione Esterna: ID_Proprietario (Utente)

<b>Tabella</b>	Immagine
<b>Attributi e tipi di dato</b>	pathname: VARCHAR(256) ID_inserzione: BIGINT(20)
<b>Chiavi</b>	Primaria: pathname Esterna: ID_inserzione (Inserzione)

<b>Tabella</b>	Stile
<b>Attributi e tipi di dato</b>	nome_stile: VARCHAR(20) descrizione: VARCHAR(100)
<b>Chiavi</b>	Primaria: nome_stile

<b>Tabella</b>	AppartenenzaStile
<b>Attributi e tipi di dato</b>	ID_inserzione: BIGINT(20) nome_stile: VARCHAR(20)
<b>Chiavi</b>	Primaria: (ID_inserzione, nome_stile) Esterna: ID_inserzione (Inserzione) Esterna: nome_stile (Stile)

<b>Tabella</b>	IntervalloDisponibilità
<b>Attributi e tipi di dato</b>	data_inizio: DATE data_fine: DATE
<b>Chiavi</b>	Primaria: (data_inizio, data_fine)

<b>Tabella</b>	Recensione
<b>Attributi e tipi di dato</b>	ID_recensione: BIGINT(20) ID_cliente: BIGINT(20) ID_inserzione: BIGINT(20) punteggio: TINYINT(4) titolo: VARCHAR(20) contenuto: VARCHAR(500) data_pubblicazione: DATE
<b>Chiavi</b>	Primaria: ID_recensione Esterna: ID_Cliente (Utente) Esterna: ID_inserzione (Inserzione)

<b>Tabella</b>	Commento
<b>Attributi e tipi di dato</b>	ID_recensione: BIGINT(20) contenuto: VARCHAR(500) data_pubblicazione: DATE
<b>Chiavi</b>	Primaria: ID_recensione Esterna: ID_recensione (Recensione)

<b>Tabella</b>	MetodoPagamento
<b>Attributi e tipi di dato</b>	numero_carta: VARCHAR(30) ID_utente: BIGINT(20) nome_titolare: VARCHAR(30) cognome_titolare: VARCHAR(30) data_scadenza: DATE preferito: TINYINT(1)



<b>Chiavi</b>	Primaria: numero_carta, ID_utente
---------------	-----------------------------------

<b>Tabella</b>	Prenotazione
<b>Attributi e tipi di dato</b>	ID_prenotazione: BIGINT(20) ID_inserzione: BIGINT(20) metodo_pagamento_cliente: VARCHAR(30) metodo_pagamento_proprietario: VARCHAR(30) data_prenotazione: DATE data_check_in: DATE data_check_out: DATE totale: DOUBLE numero_ospiti: INT stato: TINYINT(1)
<b>Chiavi</b>	Primaria: ID_prenotazione Esterna: ID_inserzione (Inserzione) Esterna: metodo_pagamento_cliente (Utente) Esterna: metodo_pagamento_proprietario (Utente)

<b>Tabella</b>	PrenotazioneArchiviata
<b>Attributi e tipi di dato</b>	ID_prenotazione: BIGINT(20) stato: VARCHAR(30) regione: VARCHAR(30) città: VARCHAR(30) CAP: CHAR(5) indirizzo: VARCHAR(50) prezzo_giornaliero: DOUBLE
<b>Chiavi</b>	Primaria: ID_prenotazione Esterna: ID_prenotazione (Prenotazione)

### 3.5 CONTROLLO DEGLI ACCESSI E SICUREZZA

Di seguito è riportata la matrice degli accessi che specifica le operazioni che ogni attore può eseguire sugli oggetti persistenti; la matrice è stata divisa per attore per facilitarne la lettura:

#### Utente:

Oggetto	Operazioni consentite
Inserzione	Ricerca inserzione.

#### Cliente:

Oggetto	Operazioni consentite
UtenteRegistrato	Modifica credenziali.
Messaggio	Invio messaggio.
Inserzione	Ricerca inserzione.
Recensione	Inserimento recensione. Rimozione recensione.
Prenotazione	Avvio prenotazione. Annullamento prenotazione. Completamento prenotazione.
PrenotazioneArchiviata	Recupero storico prenotazioni cliente.
MetodoPagamento	Inserimento metodo pagamento. Modifica metodo pagamento. Rimozione metodo pagamento.

#### Proprietario:

Oggetto	Operazioni consentite
UtenteRegistrato	Modifica credenziali.
Messaggio	Invio messaggio.
Inserzione	Inserimento inserzione. Modifica inserzione. Rimozione inserzione. Ricerca inserzione.
Commento	Inserimento commento.
PrenotazioneArchiviata	Recupero storico prenotazioni inserzione.
MetodoPagamento	Inserimento metodo pagamento. Modifica metodo pagamento. Rimozione metodo pagamento.

#### Amministratore:

Oggetto	Operazioni consentite
UtenteRegistrato	Modifica credenziali. Sospensione account. Rimozione account.
Messaggio	Invio messaggio.
Inserzione	Revisione inserzione. Rimozione inserzione. Ricerca inserzione.
PrenotazioneArchiviata	Recupero storico prenotazioni cliente. Recupero storico prenotazioni inserzione.
Stile	Inserimento stile. Modifica stile. Rimozione stile.

### 3.6 CONTROLLO GLOBALE DEL SOFTWARE

Il flusso di controllo si presenta come ibrido tra un meccanismo basato su eventi ed uno basato sul multithreading: il server Web contiene una componente che resta in attesa di una richiesta da parte del browser Web e, nel caso dovesse riceverne una, questa viene elaborata e successivamente reindirizzata alla JSP o Servlet appropriata. Per garantire la gestione parallela di diverse richieste, la Servlet istanzia un nuovo thread per ogni richiesta.

Al fine di assicurare consistenza per l'accesso ai dati durante la gestione concorrente delle richieste, verranno adottate le seguenti precauzioni:

- I dati associati a ciascuna richiesta (come i valori inseriti all'interno di un form) dovranno essere tenuti in variabili locali all'interno degli oggetti di confine, prima di, eventualmente, assicurarne la persistenza;
- Gli oggetti entità non dovranno fornire accesso diretto ai propri attributi; tutti gli accessi e le modifiche verranno effettuati attraverso metodi dedicati;
- I metodi che accedono ai dati persistenti dovranno essere sincronizzati, per garantire l'accesso ad un singolo thread per volta ed evitare race conditions.

### 3.7 CASI LIMITE

#### 3.7.1 AVVIO ED ARRESTO

Di seguito sono stati individuati due ulteriori casi d'uso, a carico dell'amministratore, per specificare le modalità di avvio ed arresto delle componenti server; per quanto riguarda la componente DataBaseServer, questa è considerata come "off-the-shelf" e trattata in maniera indipendente;

ID	UC_B1
Caso d'uso	AvviaComponentiServer
Attori partecipanti	Avviato da: Amministratore.
Condizioni di entrata	L'Amministratore effettua l'accesso alla macchina server.
Flusso degli eventi	1. L'Amministratore esegue il comando startServer sulla console di sistema. 2. Le macro-componenti, OrionServer, ServerInserzioni e DatabaseProxy vengono avviate.
Condizioni di uscita	Il web server è pronto a ricevere richieste dai browser degli utenti.
Eccezioni	2.1 Il server, dopo l'ultimo accesso, non è stato arrestato correttamente e l'Amministratore verifica l'integrità dei dati, tramite l'immissione del comando checkDataIntegrity. (Caso d'uso "VerificaIntegritàDati" – UC_E1).

ID	UC_B2
Caso d'uso	ArrestaComponentiServer
Attori partecipanti	Avviato da: Amministratore.
Condizioni di entrata	Il server è in esecuzione.
Flusso degli eventi	1. L'Amministratore esegue il comando stopServer sulla console di sistema. 2. Le macro-componenti OrionServer, ServerInserzioni e DatabaseProxy vengono arrestate.
Condizioni di uscita	Non è più possibile ricevere richieste da parte di un browser fino al riavvio.

### 3.7.2 ECCEZIONI

I principali tipi di errore a cui Orion potrebbe essere soggetto sono:

- Un errore di rete, a causa del quale una o più connessioni tra un browser Web ed il server Web di Orion vengono interrotte;
- Un errore di rete, a causa del quale una o più connessioni tra DatabaseProxy e DatabaseServer vengono interrotte e non è possibile effettuare operazioni sui dati persistenti.
- Un errore server, a causa del quale una delle componenti sulla MacchinaServer viene arrestata in modo anomalo.

Per gestire gli errori di rete tra un browser Web ed il server Web è previsto un messaggio di errore che informa gli utenti dell'accaduto; gli eventuali dati inseriti in un form, come le credenziali di registrazione o i parametri di ricerca verranno scartati e dovranno essere reinseriti in seguito all'eventuale riconnessione; in modo analogo, nel caso di errori di comunicazione tra il server ed il database remoto, non si assicura la memorizzazione dei dati.

Per gestire arresti anomali delle componenti server, viene introdotto un ulteriore caso d'uso, "VerificaIntegritàDati", per verificare lo stato degli ultimi dati trasmessi alla MacchinaDatabase:

ID	UC E1
Caso d'uso	VerificaIntegritàDati
Attori partecipanti	Avviato da: Amministratore.
Condizioni di entrata	L'Amministratore ha effettuato il ripristino del server dopo un arresto anomalo.
Flusso degli eventi	1. La componente DatabaseProxy prova a richiedere un riscontro sulle ultime operazioni effettuate al DatabaseServer inviando delle query di recupero dati.
Condizioni di uscita	L'Amministratore viene notificato.

## **4. SERVIZI DEI SOTTOSISTEMI**

### **4.1 SERVIZI ORIONSERVER**

#### **4.1.1 GESTIONEUTENZA**

Sottosistema che si occupa di gestire le operazioni relative agli utenti;

- Autenticazione: consente ad un utente di effettuare l'accesso e di autenticarsi su Orion;
- Autorizzazione: si occupa di verificare i permessi degli utenti, secondo la matrice degli accessi;
- Registrazione come cliente: consente ad un utente di effettuare la registrazione come cliente;
- Registrazione come proprietario: consente ad un utente di effettuare la registrazione come proprietario;
- Ripristino password: consente ad un utente registrato di ripristinare la propria password, in seguito a problemi di accesso;
- Sospensione account: consente all'amministratore di modificare lo stato di un utente impedendogli di effettuare l'accesso;
- Rimozione account: consente di rimuovere un account definitivamente;
- Gestione credenziali: consente ad un utente registrato di modificare le proprie credenziali.

#### **4.1.2 GESTIONEPRENOTAZIONI**

I servizi offerti riguardano:

- Avvio prenotazione: aggiorna la disponibilità associata all'inserzione di riferimento ed avvia il processo di prenotazione, notificando i partecipanti;
- Annullamento prenotazione: aggiorna la disponibilità associata all'inserzione di riferimento e annulla la prenotazione, notificando i partecipanti;
- Completamento prenotazione: completa il processo di prenotazione, gestisce il pagamento archivia le informazioni, e notifica i partecipanti;
- Archiviazione prenotazione: consente il salvataggio di alcune informazioni associate ad un'inserzione prenotata;
- Gestione pagamento: si occupa del prelievo del totale della prenotazione dal metodo di pagamento scelto dal cliente e dell'inoltro di tale somma sul metodo di pagamento predefinito del proprietario.
- Inserimento metodo di pagamento: consente ad un cliente o ad un proprietario di inserire un nuovo metodo di pagamento;
- Modifica metodo di pagamento: consente ad un cliente o ad un proprietario di modificare le informazioni associate ad uno dei metodi di pagamento precedentemente inseriti;
- Rimozione metodo di pagamento: consente ad un cliente o ad un proprietario di rimuovere uno dei metodi di pagamento precedentemente inseriti;
- Recupero storico prenotazioni cliente: recupera l'elenco delle prenotazioni archiviate di un particolare cliente;
- Recupero storico prenotazioni inserzione: recupera l'elenco delle prenotazioni archiviate di una particolare inserzione.

#### **4.1.3 GESTIONEMESSAGGI**

I servizi offerti riguardano:

- Invio messaggio
- Notifica: invio di una notifica ad un cliente o ad un proprietario in seguito ad un particolare evento (completamento prenotazione, rimozione di un'inserzione, ...);
- Contatto proprietario: predispone la comunicazione via chat tra un cliente interessato ed il proprietario di un'inserzione pubblicata;
- Recupero chat utente: recupera l'elenco delle chat passate di un utente.

## **4.2 SERVIZI SERVERINSERZIONI**

### **4.2.1 GESTIONEINSERZIONI**

I servizi offerti riguardano:

- Inserimento inserzione: consente di inserire un'inserzione nel catalogo;
- Modifica inserzione: consente al proprietario di modificare le informazioni associate ad un'inserzione precedentemente inserita;
- Rimozione inserzione: consente all'amministratore o al proprietario di rimuovere un'inserzione;
- Ricerca inserzione : recupera un elenco di inserzioni che rispettano determinati parametri di ricerca;
- Revisione inserzione: consente all'amministratore di approvare o rimuovere un'inserzione;
- Inserimento recensione: consente al cliente di inserire una recensione in riferimento ad un'inserzione precedentemente prenotata;
- Rimozione recensione: consente al cliente di rimuovere una recensione precedentemente inserita;
- Inserimento commento: consente al proprietario di rispondere ad una recensione riferita ad una propria inserzione.

## **4.3 SERVIZI DATABASEPROXY**

Sottosistema che fa da tramite tra il DatabaseServer e le componenti sulla MacchinaServer che sono interessate ad accedere ai dati persistenti.

I servizi offerti riguardano:

- Connessione al database: stabilisce la connessione JDBC con il database remoto;
- Salvataggio dati persistenti: fornisce il salvataggio di dati attraverso specifiche query al database;
- Aggiornamento dati persistenti: fornisce l'aggiornamento di dati attraverso specifiche query al database;
- Rimozione dati persistenti: fornisce la rimozione di dati attraverso specifiche query al database;
- Recupero dati persistenti fornisce il recupero di dati attraverso specifiche query al database.