UNIVERSITÀ DEGLI STUDI DI SALERNO

CORSO DI LAUREA IN INFORMATICA

INGEGNERIA DEL SOFTWARE

ORION

Docente: Andrea De Lucia

Studente: Michelangelo Esposito – 0512104784

INDICE

1. *INTRODUZIONE*

In questo documento viene presentato un riassunto complessivo delle fasi salienti di ogni step dello sviluppo di Orion, dalla formulazione del problema sino al rilascio della piattaforma.

Orion nasce come progetto universitario per il corso di Ingegneria del Software del Corso di Laurea in Informatica presso l'Università degli Studi di Salerno; si presenta come una piattaforma web per la prenotazione di appartamenti e l'insererimento di proprie inserzioni.

La progettazione e lo sviluppo effettuati hanno portato alla stesura di una serie di documenti nei quali sono descritti tutti gli spunti di analisi, progettazione e sviluppo.

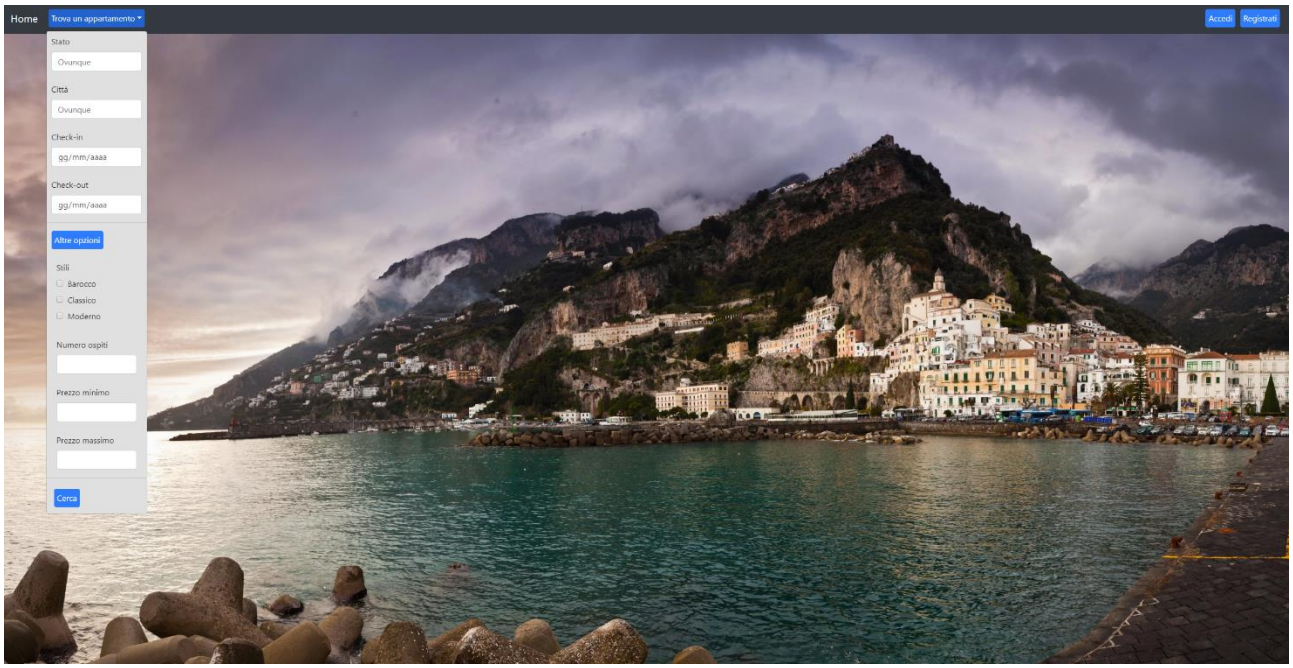
I documenti di riferimento sono:

- Problem Statement;
- Requirements Analysis;
- System Design;
- Object Design;
- Test Plan;
- Test Case Specification;
- Test Excecution Report.

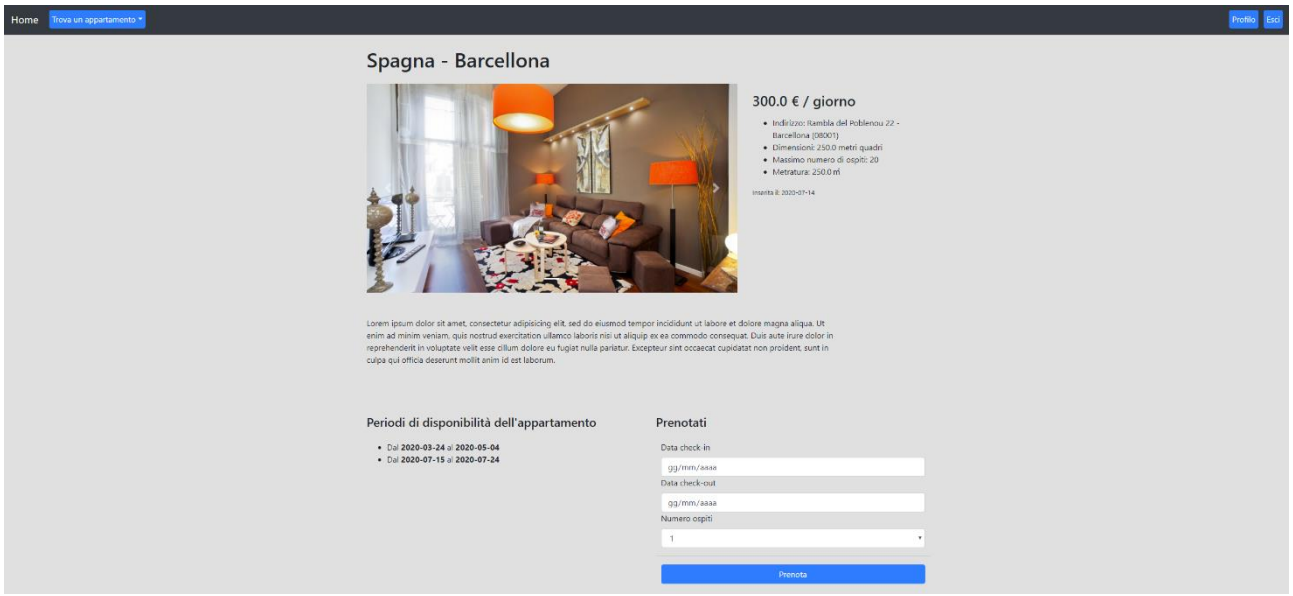
Essi sono disponibili all'indirizzo https://github.com/Espher5/IS_Orion.

2. PRESENTAZIONE DELLA PIATTAFORMA

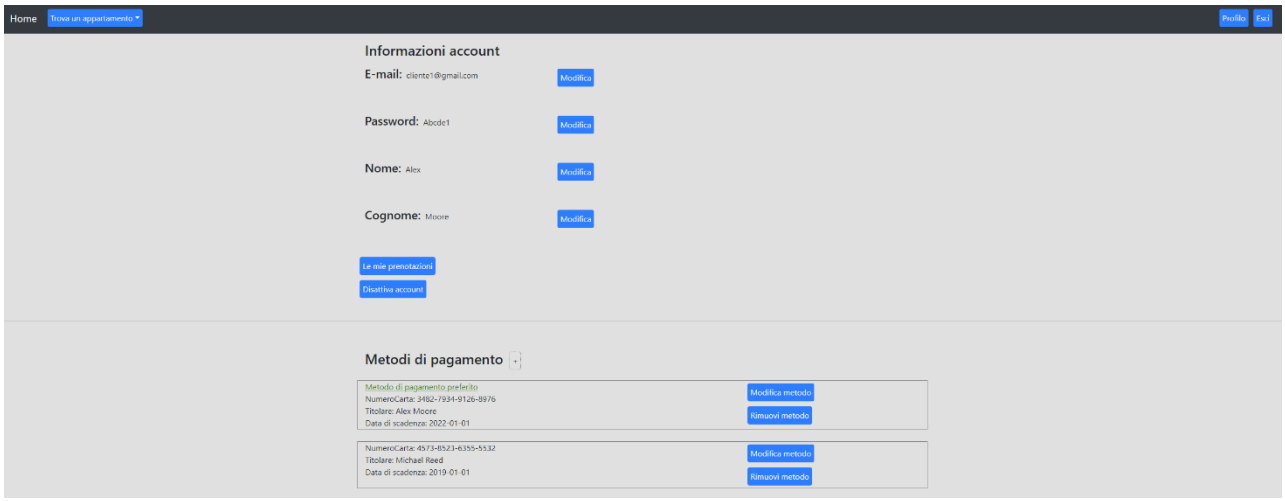
Pagina principale:



Pagina inserzione:



Profilo utente:



3. REQUIREMENTS ANALYSIS

3.1 REQUISITI FUNZIONALI

3.1.1 GESTIONE DELL'UTENZA

Utente:

ID	Requisito
FR_U1	Effettuare la registrazione come cliente.
FR_U2	Effettuare la registrazione come proprietario

UtenteRegistrato:

ID	Requisito
FR_U3	Effettuare il login alla piattaforma.
FR_U4	Recuperare la propria password.
FR_U5	Effettuare il logout dalla piattaforma.
FR_U6	Accedere alla chat di comunicazione.
FR_U7	Modificare le proprie credenziali.

Cliente e proprietario:

ID	Requisito
FR_U8	Disattivare il proprio account.

Amministratore:

ID	Requisito
FR_U9	Rimuovere l'account di un utente.
FR_U10	Sospendere l'account di un utente.
FR_U11	Riabilitare l'account di un utente.

3.1.2 GESTIONE DELLE INSERZIONI

Utente:

ID	Requisito
FR_I1	Visitare la pagina di un'inserzione.
FR_I2	Effettuare ricerche parametriche sul catalogo di inserzioni.

Cliente:

ID	Requisito
FR_I3	Scrivere una recensione.
FR_I4	Rimuovere una recensione.
FR_I5	Contattare il proprietario di un'inserzione.

Proprietario:

ID	Requisito
FR_I6	Inserire un'inserzione.
FR_I7	Modificare un'inserzione.
FR_I8	Rimuovere un'inserzione.
FR_I9	Visualizzare l'elenco delle proprie inserzioni.
FR_I10	Commentare una recensione.

Amministratore:

ID	Requisito
FR_I11	Revisionare un'inserzione.
FR_I12	Aggiungere uno stile.
FR_I13	Modificare la descrizione di uno stile.
FR_I14	Rimuovere uno stile.

3.1.3 GESTIONE DELLE PRENOTAZIONI

Cliente:

ID	Requisito
FR_P1	Visualizzare il proprio storico prenotazioni.
FR_P2	Effettuare una prenotazione.

Proprietario:

ID	Requisito
FR_P3	Visualizzare lo storico prenotazioni di una propria inserzione.

Cliente e proprietario:

ID	Requisito
FR_P4	Aggiungere un metodo di pagamento.
FR_P5	Modificare un metodo di pagamento.
FR_P6	Rimuovere un metodo di pagamento.

3.2 REQUISITI NON FUNZIONALI

3.2.1 USABILITÀ

- La piattaforma dovrà consentire di prenotare appartamenti ed inserire inserzioni in modo facile e intuitivo.

3.2.2 AFFIDABILITÀ

- La piattaforma deve essere in grado di operare h24, 7/7;
- Il numero massimo di malfunzionamenti bloccanti (nessun utente/amministratore riesce a collegarsi e/o nessuna operazione può essere eseguita) non deve superare due occorrenze annue e la durata di ciascun blocco non deve superare le tre ore, prima del ripristino del sistema. Malfunzionamenti non bloccanti non devono superare la soglia di 10 eventi nella finestra temporale di un mese di esercizio della piattaforma.

3.2.3 PERFORMANCE

- I tempi di risposta per le ricerche sul catalogo, indipendentemente dai filtri di ricerca applicati, non devono superare i 100ms;
- Al momento dell'inserimento di un'inserzione relativa ad un appartamento, un proprietario può inserire al più 20 fotografie;
- Ogni proprietario può avere al più 10 appartamenti inseriti nelle inserzioni in un dato istante.

3.2.4 IMPLEMENTAZIONE

- Gli utenti potranno accedere ad Orion utilizzando un browser web che supporti Javascript, JQuery e Bootstrap;
- Il lato server di Orion dovrà essere implementato utilizzando tecnologie Java Enterprise Edition;
- Orion dovrà supportare sistemi operativi Windows e Unix.

3.2.5 ASPETTI LEGALI

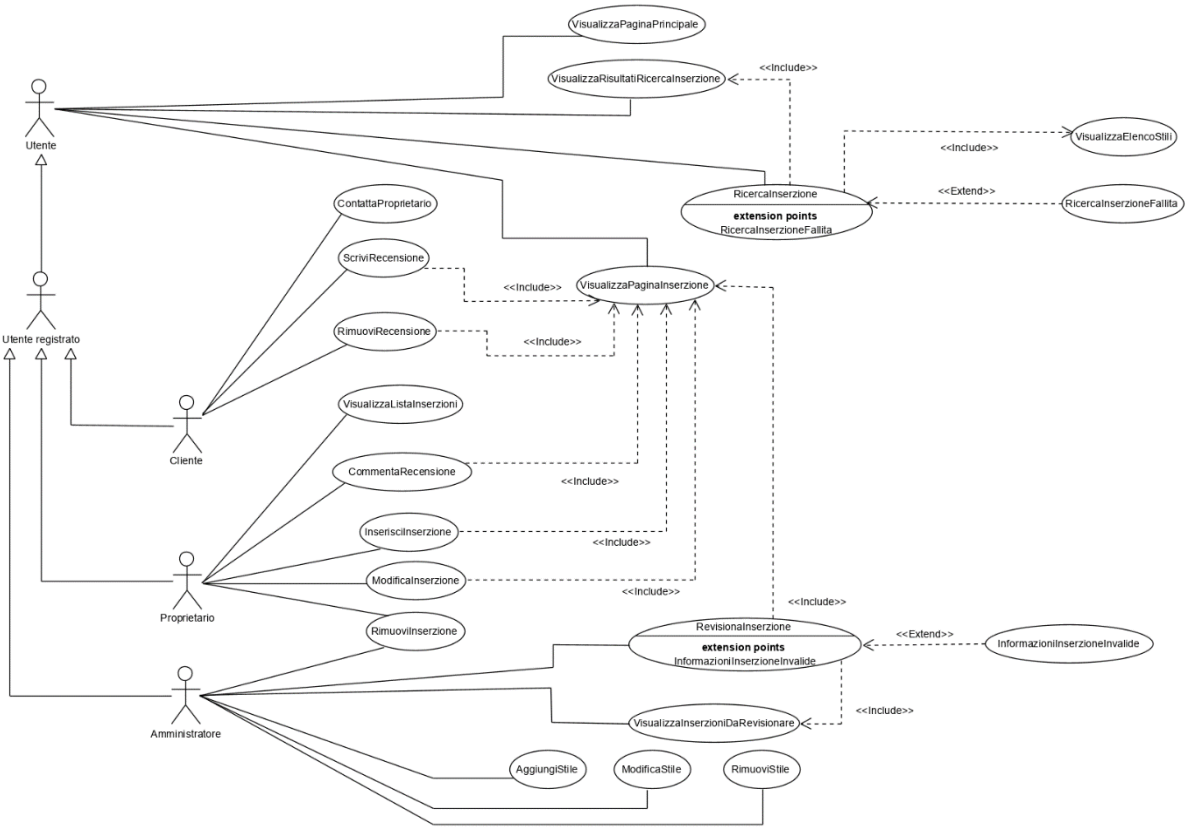
- Le informazioni relative agli utenti ed alle prenotazioni non devono essere divulgate o utilizzate a scopi di lucro;
- L'occupazione e le attività svolte all'interno degli appartamenti dovranno essere sempre conformi alle leggi locali.

3.3 ESEMPIO DI SCENARIO

Nome scenario	Registrazione, comunicazione e processo di prenotazione.
Attori partecipanti	Claudia: Cliente. Marco: Proprietario.
Flusso degli eventi	1. Claudia è una studentessa interessata ad affittare un appartamento a Barcellona dove trascorrere l'estate; un'amica le suggerisce di rivolgersi ad Orion. Claudia si collega quindi alla piattaforma attraverso il browser del suo smartphone. 2. Nell'homepage, la ragazza effettua una ricerca inserendo i parametri: - Stato: Spagna; - Città: Barcellona; - Data check-in: 24/06/2020; - Data check-out: 24/07/2020; - Fascia di prezzo (giornaliero): tra i 50 ed i 90 euro; - Numero ospiti: 1; - Stile: barocco. 3. Il sistema genera venticinque di risultati, organizzati in due pagine. 4. Sfogliando le immagini, un appartamento suscita l'attenzione di Claudia, che decide di esaminarlo più attentamente accedendo alla pagina dell'immobile. 5. Qui sono disponibili ulteriori immagini degli interni, insieme ad informazioni dettagliate sulla metratura e sul design; una planimetria dell'appartamento mostra come lo spazio è organizzato e la descrizione lasciata dal proprietario informa la ragazza sull'arredamento e sulle scelte artistiche che hanno influenzato l'arredo. 6. Claudia decide di contattare tramite il sito il proprietario, Marco, per chiedere ulteriori informazioni sullo stato strutturale dell'appartamento, essendo interessata ad eventuali malfunzionamenti, e sulle aree circostanti (servizi, eventi, …). 7. Claudia tuttavia non ha ancora effettuato la registrazione e quindi non può usufruire dei servizi di chat messi a disposizione da Orion. 8. La studentessa decide quindi di registrarsi e di fronte alla possibilità di scelta tra la registrazione come cliente e quella come proprietario, sceglie la prima. 9. Claudia compila il modulo di registrazione inserendo la propria e-mail, una password, una conferma di tale password, nome e cognome. 10. Confermando la procedura, Claudia riceve un'e-mail contenente un link di redirezione ad Orion per la conferma della registrazione. 11. A procedura completata le funzioni di chat sono state rese disponibili, quindi Claudia torna nella pagina dell'appartamento e clicca sul pulsante "Contatta proprietario". 12. Il pulsante la reindirizza nella chat del proprio profilo, dove può leggere e scrivere messaggi agli altri utenti di Orion. 13. Qui usa l'interfaccia per comporre il messaggio destinato a Marco, la cui e-mail è stata inserita automaticamente come destinatario, a seguito della redirezione, ed immettendo le informazioni a cui è interessata nel corpo del messaggio.

	14. Marco riceve il messaggio nella propria area di chat e risponde indicando che l'appartamento è stato costruito poco più di un decennio fa ed è in ottime condizioni. Inoltre, l'abitazione si trova poco distante da una stazione della metropolitana ed è quindi molto facile arrivare in centro. 15. Claudia decide di voler affittare l'appartamento e quindi mediante l'interfaccia invia una richiesta di prenotazione. 16. L'invio di tale richiesta comporta il blocco della disponibilità dell'appartamento per le successive due ore, entro le quali deve essere perfezionato il pagamento, pena la perdita della prenotazione. 17. A questo punto Marco riceve una notifica da Orion che lo informa dell'avvio della prenotazione. 18. Claudia completa il pagamento entro le due ore e la prenotazione va a termine con successo. 19. Orion informa Marco dell'esito. 20. Claudia è ora pronta a cominciare il proprio soggiorno; il calendario della disponibilità dell'appartamento viene aggiornato per rispecchiare la prenotazione effettuata. 21. Al termine del suo soggiorno, Claudia decide di lasciare una recensione positiva, dopo l'ottima esperienza, accedendo nuovamente alla pagina dell'appartamento. 22. Claudia dà alla recensione un punteggio di 5 stelle su 5 e, dopo aver inserito un titolo ed il corpo del messaggio, conferma e la pubblica. 23. Marco riceve una notifica che lo informa della presenza di una nuova recensione relativa al suo appartamento e decide di ringraziare Claudia del commento rispondendole pubblicamente.

3.4 ESEMPIO DI DIAGRAMMA DEI CASI D'USO

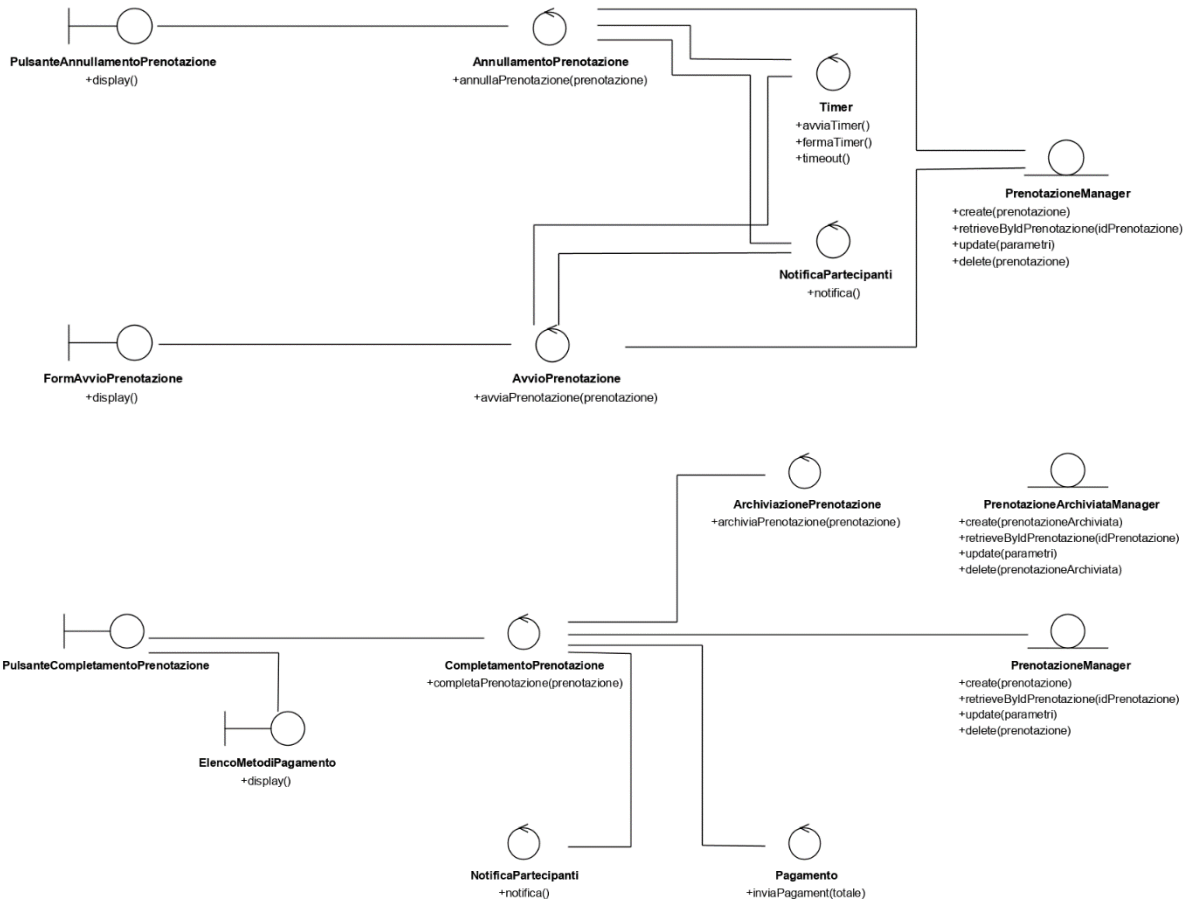


ID	UC_A6
Caso d'uso	InserisciInserzione
Attori partecipanti	Avviato da: Proprietario. Comunica con: Amministratore.
Condizioni di entrata	Il Proprietario avvia la procedura di inserimento dell'inserzione e il sistema lo reindirizza alla pagina di inserimento, dov'è presente il form da compilare.
Flusso degli eventi	1. Il Proprietario compila il form inserendo: - Stato; - Regione; - Città; - Cap; - strada - Numero civico; - Prezzo giornaliero; - Numero massimo di ospiti; - Metratura; 2. Il Proprietario prosegue la compilazione inserendo delle immagini, degli stili, una descrizione ed un calendario di disponibilità dell'appartamento. 3. Il Proprietario conferma la procedura e l'inserzione passa in stato di attesa di approvazione dell'Amministratore, il quale riceve una notifica di revisione via chat. 4. Il Proprietario viene reindirizzato alla pagina dell'inserzione per prendere visione di ciò che ha inserito (Caso d'uso "VisualizzaPaginaInserzione" – UC_A3).

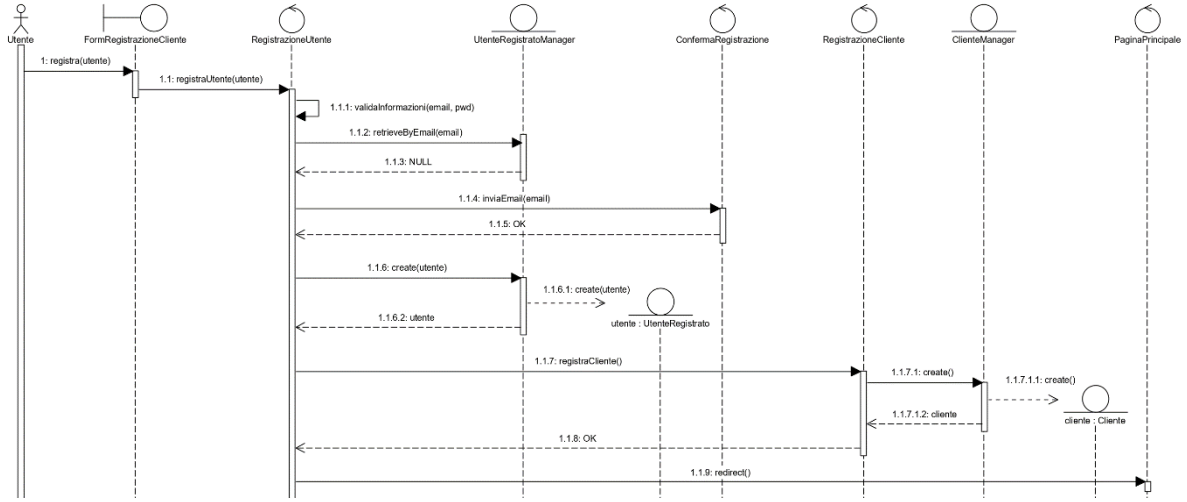
Condizioni di uscita	L'Amministratore riceve la richiesta di revisione.

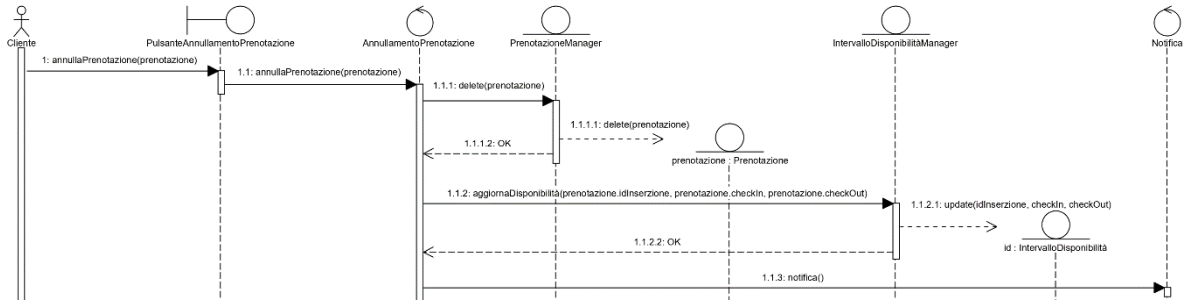
ID	UC_P5
Caso d'uso	CompletaPrenotazione
Attori partecipanti	Avviato da: Cliente. Comunica con: Proprietario.
Condizioni di entrata	Il Cliente ha avviato la procedura di prenotazione.
Flusso degli eventi	1. Il Cliente seleziona un metodo di pagamento tra quelli inseriti. 2. Il sistema fornisce un riepilogo delle informazioni dell'appartamento e del costo totale della prenotazione. 3. Il Cliente conferma la procedura ed il pagamento viene avviato.
Flusso alternativo	1.1 Il Cliente non ha precedentemente inserito alcun metodo di pagamento ed ha la possibilità di inserirne uno.
Condizioni di uscita	La procedura di pagamento viene completata correttamente.
Eccezioni	2.1 Sono passate più di due ore dall'avvio della prenotazione ed il timer è scaduto, quindi la prenotazione viene annullata (Caso d'uso "TimeOutSistema" – UC_P5.1).

3.5 ESEMPIO DI DIAGRAMMA ENTITY-BOUNDARY-CONTROL

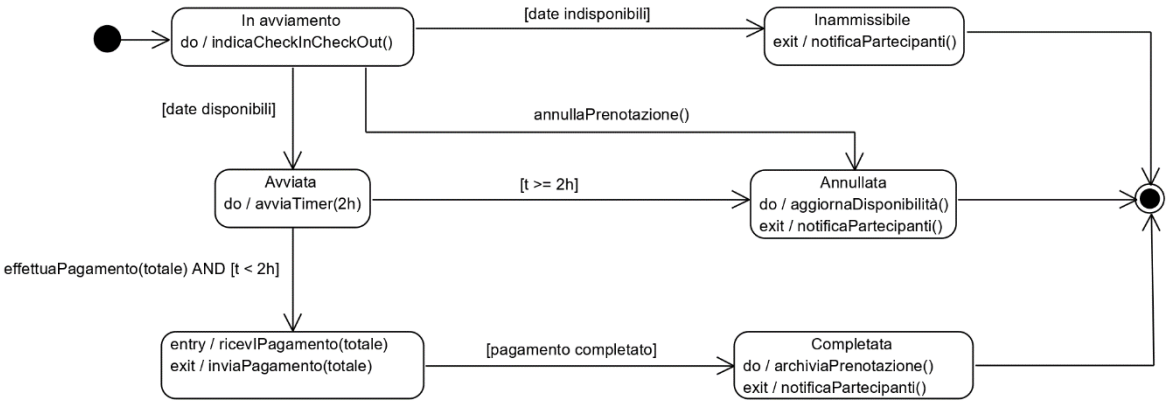


3.6 ESEMPIO DIAGRAMMI DI SEQUENZA



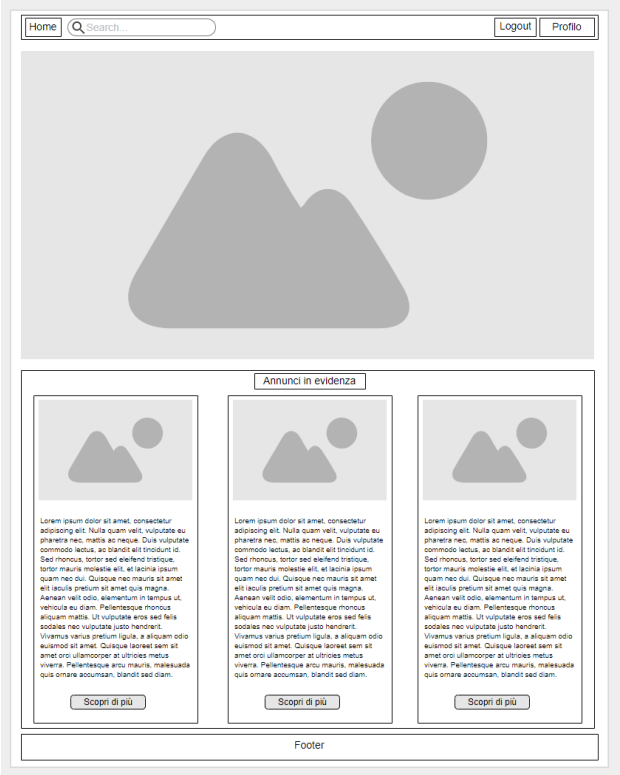


3.7 ESEMPIO DI DIAGRAMMA DI STATO

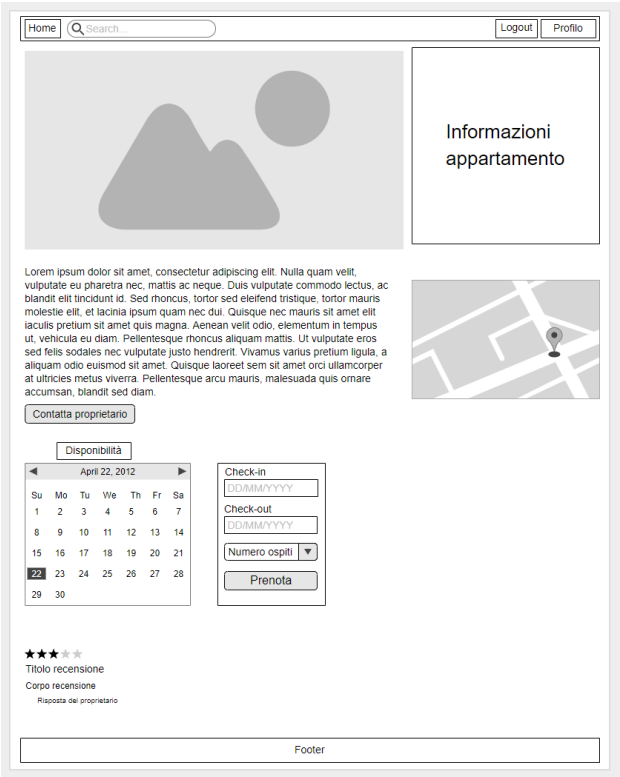


3.8 *ESEMPIO DI MOCK-UP DELL'INTERFACCIA UTENTE*

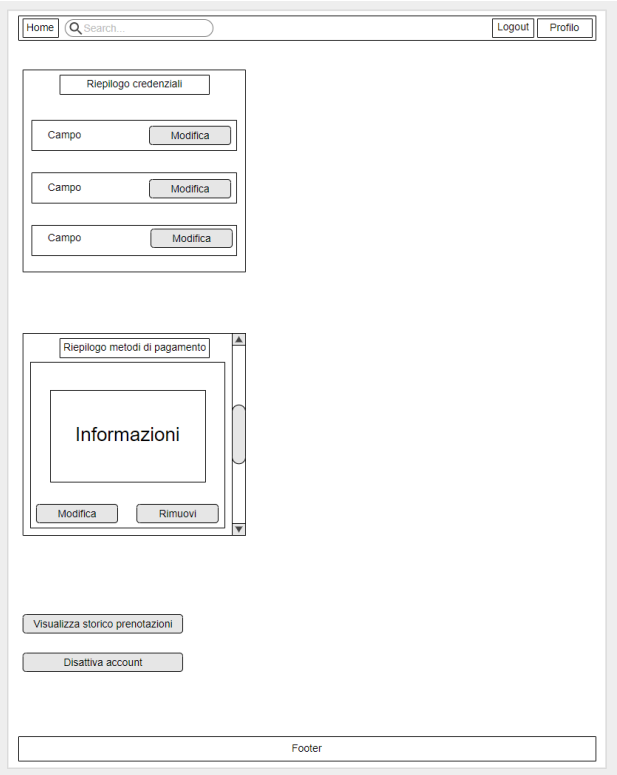
Pagina principale:



Pagina inserzione:



Profilo utente:



4. SYSTEM DESIGN

4.1 OBIETTIVI DI DESIGN

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);

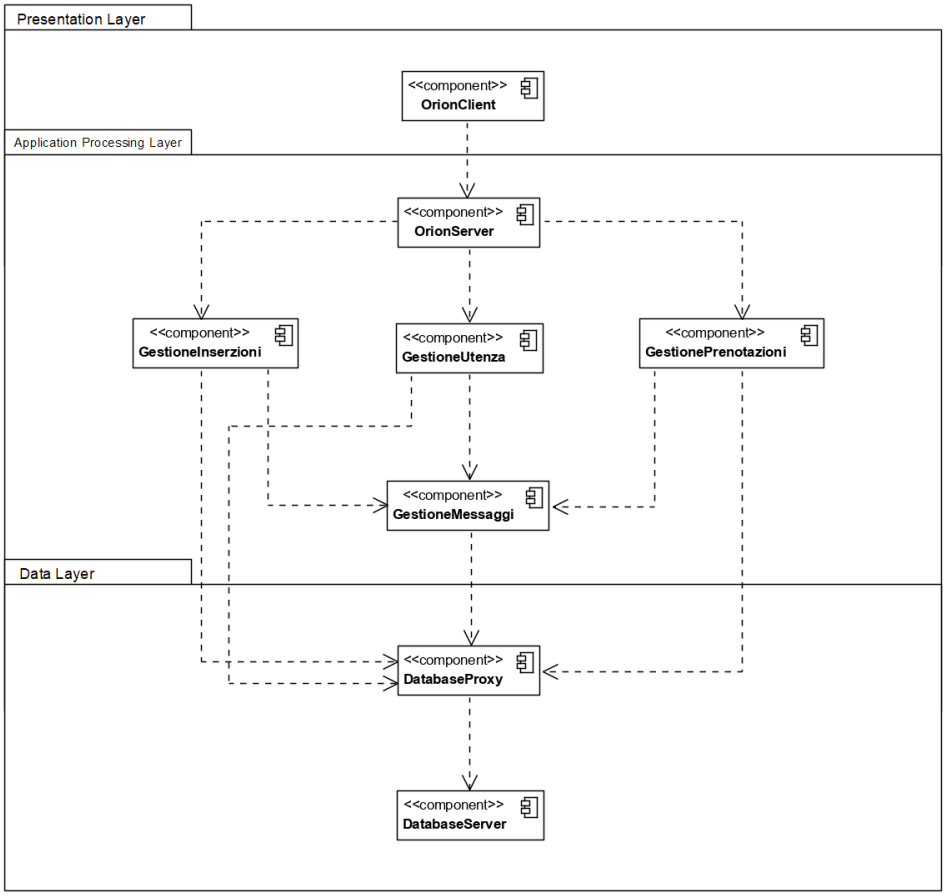
DG_02: Efficienza: la navigazione su Orion dovrà essere veloce e responsiva; ciò richiede bassi ritardi, veloci operazioni di ricerca ed una buona scalabilità rispetto al numero di utenti connessi ed al numero di inserzioni presenti.

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.

4.2 DECOMPOSIZIONE IN SOTTOSISTEMI

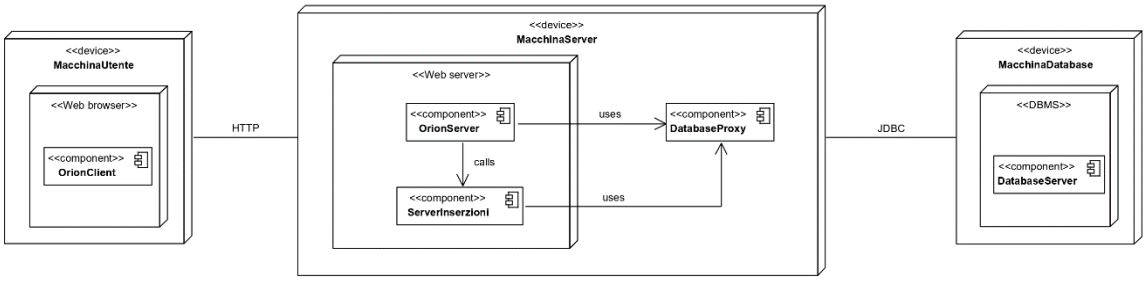
La decomposizione è stata ideata sulla base di una divisione delle funzionalità in tre strati:

- **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;



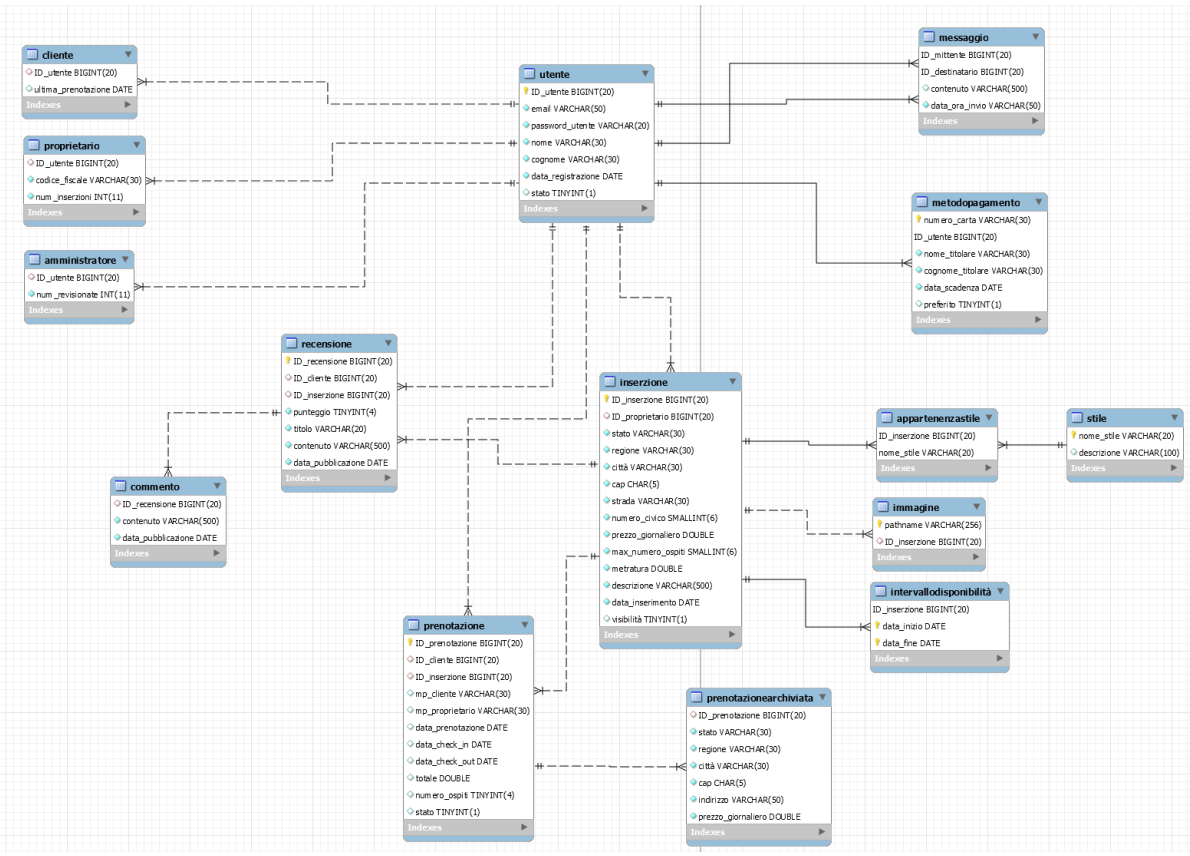
4.3 MAPPING HARDWARE / SOFTWARE

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.



4.4 GESTIONE DEI DATI PERSISTENTI

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.



4.1 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.

4.2 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 richiese, 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.

4.3 CASI LIMITE

4.3.1 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.

4.3.2 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.

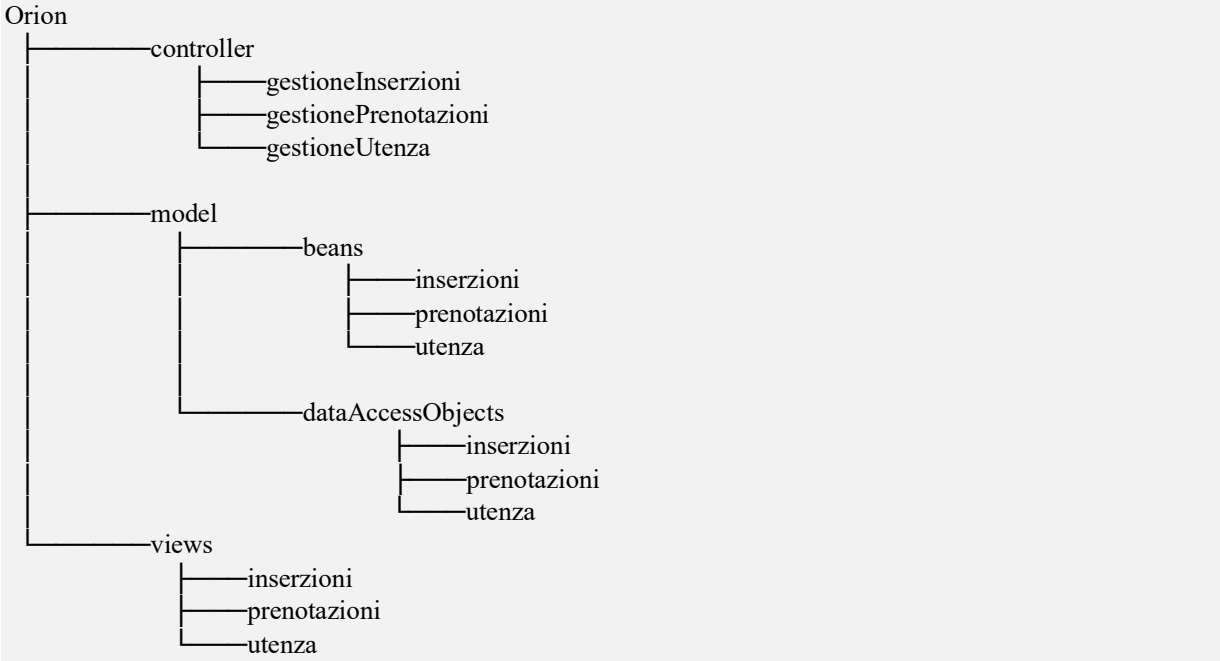
Per gestire gli errori di rete tra un browser Web ed il server Web è previsto un messaggio di errore che informa gli utenti dell'accaduto; gli eventuali dati inseriti in un form, come le credenziali di registrazione o i parametri di ricerca verranno scartati e dovranno essere reinseriti in seguito all'eventuale 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:

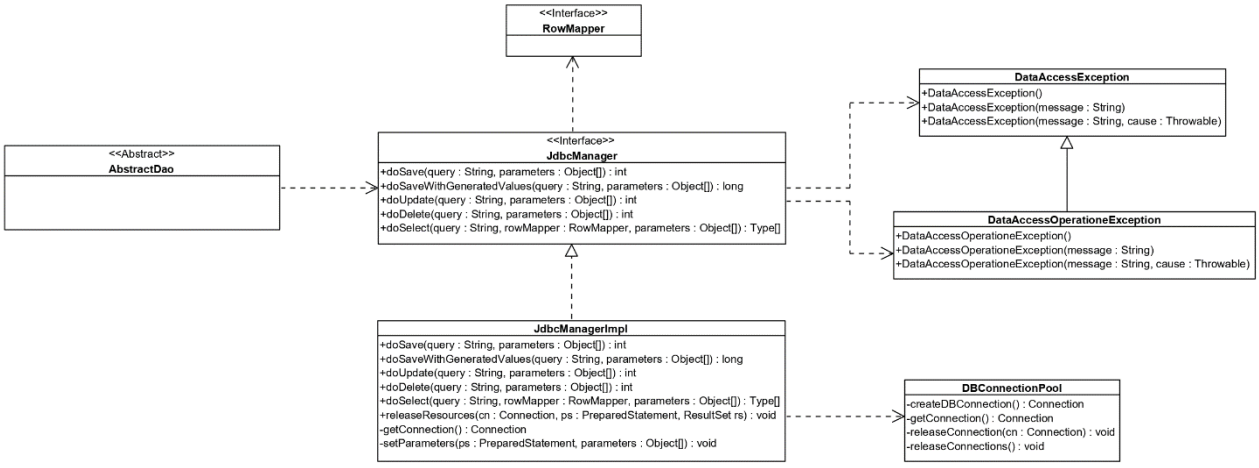
ID	UC_E1
Caso d'uso	VerificaIntegritàDati
Attori partecipanti	Avviato da: Amministratore.
Condizioni di entrata	L'Amministratore ha effettuato il ripristino del server dopo un arresto anomalo.
Flusso degli eventi	1. La componente DatabaseProxy prova a richiedere un riscontro sulle ultime operazioni effettuate al DatabaseServer inviando delle query di recupero dati.
Condizioni di uscita	L'Amministratore viene notificato.

5. OBJECT DESIGN

5.1 STRUTTURA DEI PACKAGE

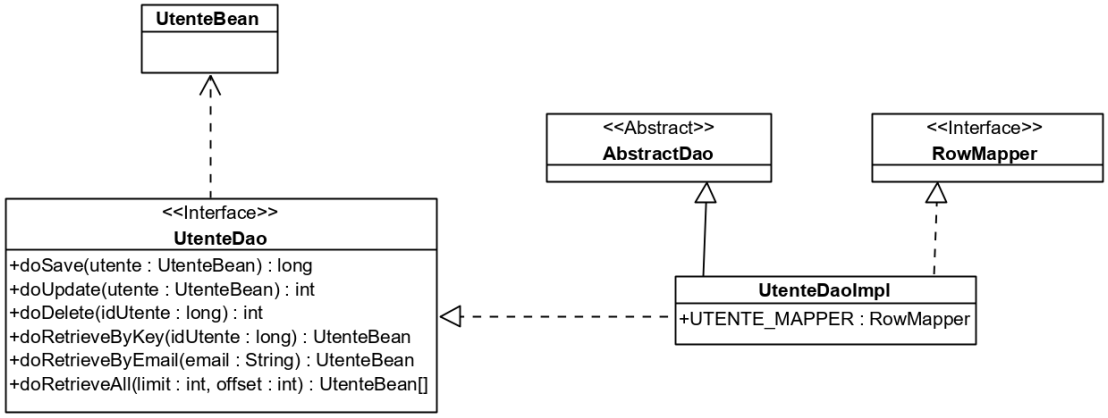


5.2 DIAGRAMMA DI CLASSE DELLE COMPONENTI DI APPOGGIO DEI DAO



5.3 DIAGRAMMA DI CLASSE DI UN DAO E RELATIVO CONTRATTO

Diagramma di classe:



Contratto:

```
context UtenteDao::doSave(ub:UtenteBean) pre:
  ub != null and
  ub.email != null and
  ub.password != null and
  ub.nome != null and
  ub.cognome != null and
  ub.stato != null and
  doRetrieveByEmail(ub.email) = null
context UtenteDao::doSave(ub:UtenteBean) post:
  result = idUtente
  doRetrieveByKey(idUtente) != null

context UtenteDao::doUpdate(ub:UtenteBean) pre:
  ub != null and
  ub.idUtente != null
  ub.email != null and
  ub.password != null and
  ub.nome != null and
  ub.cognome != null and
  ub.stato != null and
  doRetrieveByKey(ub.idUtente) != null
context UtenteDao::doUpdate(ub:UtenteBean) post:
  result = 1

context UtenteDao::doDelete(idUtente:long) pre:
  idUtente != null and
  doRetrieveByKey(idUtente) != null
context UtenteDao::doDelete(idUtente:long) post:
  doRetrieveByKey(idUtente) = null

context UtenteDao::doRetrieveByKey(idUtente:long) pre:
  idUtente != null and
  esiste UtenteBean ub nel database : ub.idUtente = Utente
context UtenteDao::doRetrieveByKey(id:long) post:
  result = ub
```

```
context UtenteDao::doRetrieveByEmail(email:String) pre:
  email != null and
  esiste UtenteBean ub nel database : ub.email = email
context UtenteDao::doRetrieveByEmail(email:String) post:
  result = ub

context UtenteDao::doRetrieveAll(limit:int, offset:int) pre:
  limit != null and
  offset != null
context UtenteDao::doRetrieveAll(limit:int, offset:int) post:
  result = i primi limit ub:UtenteBean nel database a partire da offset
```

5.4 ESEMPIO DI CLASSE CON JAVADOC

```java
/**
 * Effettua un'operazione di inseriemnto dati nel
 * database mediante query parametrica;
 *
 * @param query la query da effettuare
 * @param parameters lista di parametri da inserire nella query
 * @return un intero che rappresenta l'esito dell'operazione
 * @throws DataAccessException nel caso di errori nell'utilizzo dei parametri,
 * nella formulazione della query o nelle operazioni a carico del DBMS
 */
int doSave(String query, Object... parameters) throws DataAccessException;
```

```java
/**
 * Recupera le tuple dal database che rispettano i parametri passati
 * ed incapsula i risultati in una lista
 *
 * @param query la query da effettuare
 * @param rowMapper
 * @param parameters lista di parametri da inserire nella query
 * @return la lista recuperata o null nel caso di nessun risultato
 * @throws DataAccessException nel caso di errori nell'utilizzo dei parametri,
 * nella formulazione della query o nelle operazioni a carico del DBMS
 */
<E> List<E> doSelect(final String query, final RowMapper<E> rowMapper,
        final Object... parameters) throws DataAccessException;
```

6. TEST PLAN

6.1 COMPONENTI TESTATE

A seguito di un'analisi delle risore dedicate al processo di testing, per ognuno dei sottoinsiemi individuati, si è scelta una lista delle funzionalità da testare, in base a quelle che si ritiene verranno utilizzate più frequentemente, come operazioni di ricerca, inserimento di inserzioni e processi di prenotazione:

Gestione dell'utenza:

- Registrazione come proprietario;
- Accesso;
- Sospensione di un account;

Gestione delle inserzioni:

- Inserimento inserzione;
- Rimozione inserzione;
- Inserimento recensione;

Gestione delle prenotazioni:

- Inserimento di un metodo di pagamento;
- Rimozione di un metodo di pagamento;
- Avvio prenotazione;
- Completamento prenotazione.

6.2 APPROCCIO DI TESTING

L'approccio utilizzato per il testing di Orion è diviso in tre diverse fasi, mirate a testare il sistema partendo dalle singole funzionalità fino alla loro integrazione ed alla struttura complessiva dell'applicazione:

- **Test di unità**: durante questa fase, verranno testate le singole funzionalità .Verrà utilizzato un approccio di tipo Black-Box, ignorando l'implementazione di ciascuna unità e considerando esclusivamente il comportamento a fronte di determinati input. La strategia utilizzata per l'individuazione dei casi di test sarà di tipo Category Partition;
- **Test di integrazione**: in questa fase verranno testate le interazioni tra i diversi sottosistemi, verificando che i requisiti specificati vengano correttamente realizzati e che il sistema sia robusto; la strategia utilizzata per l'integrazione è sarà di tipo bottom-up;
- **Test di sistema**: nell'ultima fase, il sistema verrà testato nella sua interezza, attraverso l'interazione dinamica con l'applicazione, al fine di verificare l'armonia tra le varie componenti.

6.3 *ESEMPIO DI APPLICAZIONE DELLA STRATEGIA CATEGORY PARTITION*

Registrazione come cliente:

Parametro: e-mail Formato: "\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w+)+"	
Formato [FEmail]	1. Formato non rispettato [Errore] 2. Formato rispettato [FEmail_OK]
Esistenza [EEmail]	1. E-mail esistente [if FEmail_OK] [Errore] 2. E-mail insesistente [if FEmail_OK] [EEmail_OK]

Parametro: password Formato: "(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=\S+$).{6,}"	
Formato [FPwd]	1. Formato non rispettato [Errore] 2. Formato rispettato [FPwd_OK]

Parametro: conferma password Formato: "(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=\S+$).{6,}"	
Corrispondenza [CConfPwd]	1. Conferma password != password [Errore] 2. Conferma password = password [if FConfPwd_OK] [CConfPwd_OK]

Parametro: nome Formato: "[a-zA-Z]+((['. -][a-zA-Z])?[a-zA-Z]*)*"	
Formato [FNome]	1. Formato non rispettato [Errore] 2. Formato rispettato [FNome_OK]

Parametro: cognome Formato: "[a-zA-Z]+((['. -][a-zA-Z])?[a-zA-Z]*)*"	
Formato [FCognome]	1. Formato non rispettato [Errore] 2. Formato rispettato [FCognome_OK]

Casi di test:

Codice	Combinazione	Esito
TC_U1.1	FEmail_1	FAIL
TC_U1.2	FEmail_2, EEmail_1	FAIL
TC_U1.3	FEmail_2, EEmail_2, FPwd_1	FAIL
TC_U1.4	FEmail_2, EEmail_2, FPwd_2, CConfPwd_1	FAIL
TC_U1.5	FEmail_2, EEmail_2, FPwd_2, CConfPwd_2, VNome_1	FAIL
TC_U1.6	FEmail_2, EEmail_2, FPwd_2, CConfPwd_2, VNome_2, VCognome_1	FAIL
TC_U1.7	FEmail_2, EEmail_2, FPwd_2, CConfPwd_2, VNome_2, VCognome_2	PASS

7. TEST CASE SPECIFICATION

7.1 ESEMPIO DI CASI DI TEST DETTAGLIATI

Registrazione come cliente.

ID	TC_U1.1
Precondizioni	L'utente si trova in homepage, nell'area di registrazione come cliente.
Flusso degli eventi	1. L'utente riempie il form di registrazione con i seguenti valori: - E-mail: "NuovaEmailgmail.com"; - Password: "Abcde1"; - Conferma password: "Abcde1"; - Nome: "Michelangelo"; - Cognome: "Esposito"; 2. L'utente conferma la registrazione.
Oracolo	La registrazione non va a buon fine, in quanto l'e-mail inserita non è valida.

ID	TC_U1.2
Precondizioni	L'utente si trova in homepage, nell'area di registrazione come cliente.
Flusso degli eventi	1. L'utente riempie il form di registrazione con i seguenti valori: - E-mail: "EmailEsistente@gmail.com"; - Password: "Abcde1"; - Conferma password: "Abcde1"; - Nome: "Michelangelo"; - Cognome: "Esposito"; 2. L'utente conferma la registrazione.
Oracolo	La registrazione non va a buon fine, in quanto esiste un altro account che ha effettuato la registrazione con la stessa e-mail.

ID	TC_U1.3
Precondizioni	L'utente si trova in homepage, nell'area di registrazione come cliente.
Flusso degli eventi	1. L'utente riempie il form di registrazione con i seguenti valori: - E-mail: "NuovaEmail@gmail.com"; - Password: "Abcde"; - Conferma password: "Abcde"; - Nome: "Michelangelo"; - Cognome: "Esposito"; 2. L'utente conferma la registrazione.
Oracolo	La registrazione non va a buon fine, in quanto la password non rispetta il formato specificato.

ID	TC_U1.4
Precondizioni	L'utente si trova in homepage, nell'area di registrazione come cliente.
Flusso degli eventi	1. L'utente riempie il form di registrazione con i seguenti valori: - E-mail: "NuovaEmail@gmail.com"; - Password: "Abcde1"; - Conferma password: "Abcde"; - Nome: "Michelangelo"; - Cognome: "Esposito"; 2. L'utente conferma la registrazione.
Oracolo	La registrazione non va a buon fine, in quanto le password non corrispondono.

ID	TC_U1.5
Precondizioni	L'utente si trova in homepage, nell'area di registrazione come cliente.
Flusso degli eventi	1. L'utente riempie il form di registrazione con i seguenti valori: - E-mail: "NuovaEmail@gmail.com"; - Password: "Abcde1"; - Conferma password: "Abcde1"; - Nome: ""; - Cognome: "Esposito"; 2. L'utente conferma la registrazione.
Oracolo	La registrazione non va a buon fine, in quanto il campo "nome" è stato lasciato vuoto.

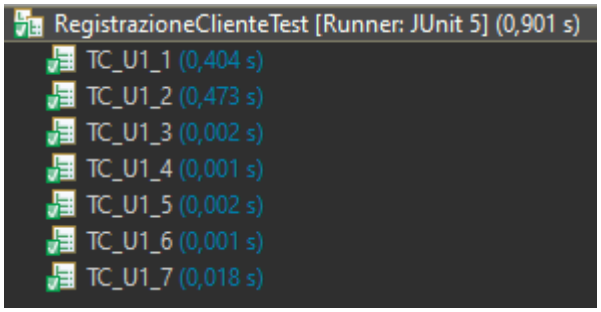
ID	TC_U1.6
Precondizioni	L'utente si trova in homepage, nell'area di registrazione come cliente.
Flusso degli eventi	1. L'utente riempie il form di registrazione con i seguenti valori: - E-mail: "NuovaEmail@gmail.com"; - Password: "Abcde1"; - Conferma password: "Abcde1"; - Nome: "Michelangelo"; - Cognome: ""; 2. L'utente conferma la registrazione.
Oracolo	La registrazione non va a buon fine, in quanto "cognome" è stato lasciato vuoto.

ID	TC_U1.7
Precondizioni	L'utente si trova in homepage, nell'area di registrazione come cliente.
Flusso degli eventi	1. L'utente riempie il form di registrazione con i seguenti valori: - E-mail: "NuovaEmail@gmail.com - Password: "Abcde1"; - Conferma password: "Abcde1"; - Nome: "Michelangelo"; - Cognome: "Esposito"; 2. L'utente conferma la registrazione.
Oracolo	La procedura di registrazione va a buon fine.

8. TEST EXECUTION REPORT

8.1 ESEMPIO DI REPORT DI CASI DI TEST

Registrazione come cliente:



ID	TC_U1.1
Data	5/02/2020
Oracolo	La registrazione non va a buon fine, in quanto l'e-mail inserita non è valida.
Eisto finale	PASSED

ID	TC_U1.2
Data	5/02/2020
Oracolo	La registrazione non va a buon fine, in quanto esiste un altro account che ha effettuato la registrazione con la stessa e-mail.
Eisto finale	PASSED

ID	TC_U1.3
Data	5/02/2020
Oracolo	La registrazione non va a buon fine, in quanto la password non rispetta il formato specificato
Eisto finale	PASSED

ID	TC_U1.4
Data	5/02/2020
Oracolo	La registrazione non va a buon fine, in quanto le password non corrispondono.
Eisto finale	PASSED

ID	TC_U1.5
Data	5/02/2020
Oracolo	La registrazione non va a buon fine, in quanto il campo "nome" è stato lasciato vuoto.
Eisto finale	PASSED

ID	TC_U1.6
Data	5/02/2020

Oracolo	La registrazione non va a buon fine, in quanto “cognome” è stato lasciato vuoto.
Eisto finale	PASSED

ID	TC_U1.7
Data	5/02/2020
Oracolo	La procedura di registrazione va a buon fine.
Eisto finale	PASSED