



# UniRent Hub

## System Design Document UniRentHub

<b>Riferimento</b>	C11_SDD_V_2.0
<b>Versione</b>	2.0
<b>Data</b>	13/12/2023
<b>Destinatario</b>	Prof.ssa Filomena Ferrucci, Prof. Fabio Palomba
<b>Presentato da</b>	C11 Fries Tech Team
<b>Approvato da</b>	Rocco Iuliano, Simone Della Porta



## Team Members

Nome	Cognome	Ruolo	Acronimo	Contatto
Rocco	Iuliano	PM	RI	<a href="mailto:r.iuliano13@studenti.unisa.it">r.iuliano13@studenti.unisa.it</a>
Simone	Della Porta	PM	SDP	<a href="mailto:s.dellaporta6@studenti.unisa.it">s.dellaporta6@studenti.unisa.it</a>
Antonio	Albanese	TM	AA	<a href="mailto:a.albanese22@studenti.unisa.it">a.albanese22@studenti.unisa.it</a>
Francesco Pio	Contaldo	TM	FPC	<a href="mailto:f.contaldo4@studenti.unisa.it">f.contaldo4@studenti.unisa.it</a>
Cristyan	Esposito	TM	CE	<a href="mailto:c.esposito175@studenti.unisa.it">c.esposito175@studenti.unisa.it</a>
Iliano	Fasolino	TM	IF	<a href="mailto:i.fasolino3@studenti.unisa.it">i.fasolino3@studenti.unisa.it</a>
Marco	Greco	TM	MG	<a href="mailto:m.greco65@studenti.unisa.it">m.greco65@studenti.unisa.it</a>
Giuseppe Pio	Sorrentino	TM	GPS	<a href="mailto:g.sorrentino101@studenti.unisa.it">g.sorrentino101@studenti.unisa.it</a>

## Revision History

Data	Versione	Descrizione	Autori
21/11/2023	0.1	Scrittura dei Design Goals	MG, IF
25/11/2023	0.2	Scrittura paragrafi 1.1, 1.2, 1.3, 1.4, 1.5	MG, CE, IF
25/11/2023	0.3	Scrittura dei Trade-off	CE, IF
25/11/2023	0.4	Scrittura paragrafi 2, 3.1	MG, CE, IF
25/11/2023	0.5	Definizione dei sottosistemi	MG, CE
26/11/2023	0.6	Creazione Diagramma Architetture dei sottosistemi	MG, CE
27/11/2023	0.7	Scrittura paragrafo 3.3, Definizione mapping Hardware e Software	CE, MG
27/11/2023	0.8	Scrittura paragrafo 3.4, 3.5	IF, GPS, CE, FPC
1/12/2023	0.9	Scrittura paragrafo 4	CE, FPC
2/12/2023	1.0	Definizione Matrice degli accessi	CE, FPC
3/12/2023	1.1	Aggiunta Boundary Condition	MG, IF, GPS, FPC
3/12/2023	1.2	Aggiunta Class Diagram e Schema ER	Tutto il team



3/12/2023	1.3	Definizione lista entità persistenti	GPS, IF, MG
3/12/2023	1.4	Definizione modello logico dei dati	GPS, AA
3/12/2023	1.5	Scrittura sezione 3.6	AA
3/12/2023	1.6	Scrittura Glossario	IF
4/12/2023	1.7	Check SDD	IF, GPS
5/12/2023	2.0	Revisione Finale	IF, GPS



## Sommario

Team Members

Revision History

1: Introduzione

1.1: Obiettivo del Sistema

1.2: Design Goals

1.2.1: Design goals

1.2.2: Trade-off

1.3: Definizioni, Acronimi e Abbreviazioni

1.4: Riferimenti

1.5: Overview

2. Architettura del Sistema Attuale

3. Architettura del Sistema Proposto

3.1: Sintesi del Sistema

3.2: Decomposizione in sottosistemi

3.2.1: Diagramma delle componenti

3.2.2: Diagramma architetturale

3.3: Mapping Hardware/Software

3.4: Gestione dei dati persistenti

3.5: Sicurezza ed Accessi

3.5.1: Matrice degli accessi

3.6: Controllo Globale del Software

3.7: Condizioni limite

4: Servizi dei sottosistemi

5: Glossario

## 1: Introduzione

### 1.1: Obiettivi del Sistema

UniRentHub si propone di agire come intermediario tra gli studenti e i locatori, focalizzandosi principalmente nel semplificare il processo di affitto e sull'incoraggiamento dell'ecosostenibilità in quanto tematica cardine del sistema.

Il sistema permette la registrazione sia degli studenti che potranno visualizzare e prendere in affitto alloggi, sia dei locatori, consentendo loro di pubblicare annunci riguardanti le loro proprietà.

Il sistema è principalmente formato da tre componenti chiave:

- La possibilità per i locatori di inserire un alloggio in affitto.
- La possibilità per gli studenti di affittare un alloggio.
- La creazione di una comunità di studenti con interessi simili, al fine di agevolare la selezione di coinquilini compatibili.

### 1.2: Design Goal

In questa sezione sono specificati tutti i design goal che il sistema punta a raggiungere. Ogni design goal è caratterizzato da:

- **Rank**
- **ID** Design Goal
- **Descrizione**
- **Categoria** (Performance, Dependability, Cost, Maintenance, End user)
- **Origine** (ID del RNF associato)

Poiché è impossibile realizzare una piattaforma "perfetta" e priva di errori, in questa sezione sono elencati anche dei trade-off che rappresentano compromessi necessari quando si prendono decisioni che coinvolgono la gestione delle risorse limitate o quando si cerca di massimizzare i benefici in situazioni dove non è possibile soddisfare completamente tutti gli obiettivi contemporaneamente.

#### 1.2.1: Design Goal

Rank	ID	Descrizione	Categoria	Origine
4	DG_3 Tempi di risposta	Il sistema deve rispondere alle richieste effettuate dall'utente in un tempo non maggiore a 10 minuti	Performance	RNF_3
12	DG_4 Sistema software ecologico	Il sistema deve promuovere l'ecosostenibilità avendo un basso impatto ambientale	Sostenibilità	RNF_6



7	DG_5 Graphic User Interface	Il sistema deve essere fornire un interfaccia grafica che facilita l'utente nella lettura e nella visualizzazione degli annunci ed informazioni	Usabilità	RNF_4
6	DG_6 Sistema Responsive	Il sistema deve funzionare su un ampia varietà di browser e su dispositivi mobile	Usabilità	RNF_5
8	DG_7 Menù contestuale	Il sistema deve essere provvisto di un menù contestuale in modo da facilitare all'utente l'effettuazione delle operazioni	Usabilità	RNF_4
9	DG_8 Nav Bar	Il sistema deve essere provvisto di una nav bar in modo da facilitare l'utente nella navigazione	Usabilità	RNF_4
10	DG_9 Utilizzo strategico dell'interfaccia	Il sistema deve essere fornito di colori scelti strategicamente per attirare l'utente	Usabilità	RNF_4
3	DG_10 Sicurezza e affidabilità	Garantire la sicurezza delle transazioni e dei dati personali degli utenti, implementando misure di sicurezza robuste e proteggendo le informazioni sensibili.	Affidabilità	RNF_2
2	DG_11 Gestione della crittografia	Il sistema deve utilizzare una buona tecnica di crittografia per garantire la sicurezza dei dati sensibili inseriti dall'utente, utilizzando protocolli di comunicazione sicuri, e garantendo la visualizzazione dei dati solo agli utenti che possono accedervi.	Sicurezza	RNF_2

11	DG_12 Vincoli di Manutenibilità	il sistema deve essere progettato per garantire semplicità ed economicità nell'esecuzione di manutenzioni ordinarie	Maintenance	N/A
4	DG_3 Tempi di risposta	Il sistema deve rispondere alle richieste effettuate dall'utente in un tempo non maggiore a 10 minuti	Performance	RNF_3
12	DG_4 Sistema software ecologico	Il sistema deve promuovere l'ecosostenibilità avendo un basso impatto ambientale	Sostenibilità	RNF_6

### 1.2.2: Trade-offs

Trade-off	Descrizione
<b>Accessibilità vs. Sicurezza dei dati</b>	Garantire un'interfaccia utente semplice e accessibile potrebbe richiedere compromessi sulla sicurezza dei dati degli utenti. Il trade-off è tra facilità d'uso e l'implementazione di misure di sicurezza robuste per proteggere le informazioni personali dei clienti.
<b>Personalizzazione vs. Privacy degli Utenti</b>	Offrire un'esperienza personalizzata sulla piattaforma potrebbe richiedere la raccolta di dati personali degli utenti. Il trade-off è tra la personalizzazione del servizio e il rispetto della privacy dei clienti.
<b>Qualità del Servizio vs. Costo</b>	Garantire un servizio di qualità potrebbe richiedere investimenti aggiuntivi, comportando di conseguenza un aumento di costi per la piattaforma. Il trade-off in questo caso è tra offrire un'esperienza singolare e avere dei prezzi accessibili.
<b>Feedback Utente vs. Protezione dei Locatori</b>	È importante rendere pubbliche le recensioni degli utenti, ma potrebbe influenzare troppo negativamente la reputazione degli host. Il trade-off è tra la trasparenza della recensione e la protezione dei locatori da recensioni ingiuste o diffamatorie.



<b>Comunità vs. Rischi di abuso</b>	Creare una piattaforma che può promuovere una comunità virtuale tra gli utenti potrebbe migliorare l'esperienza, ma potrebbe anche comportare rischi di abuso, o di altri comportamenti che potrebbero violare i TOS di UniRentHub. Il trade-off è tra la costruzione di una comunità e la gestione dei rischi associati ad essa.
<b>Flessibilità delle Prenotazioni vs. Garanzia di Entrate per i Locatori</b>	Offrire una policy di cancellazione flessibile potrebbe attirare più clienti, ma al contempo mettere in difficoltà i locatori dinanzi ad un rischio troppo elevato di perdere entrate. Il trade-off è tra la flessibilità degli ospiti (imprescindibile nel mercato odierno) e la sicurezza finanziaria dei locatori.

### 1.3: Definizioni, acronimi e abbreviazioni

- **DAO:** Data Access Object
- **DB:** Database
- **DG:** Design goal
- **ER:** Entity-Relationship
- **GUI:** Graphic User Interface
- **HTTPS:** HyperText Transfer Protocol Secure
- **HTML:** HyperText Markup Language
- **RF:** Requisito funzionale
- **RNF:** Requisito non funzionale
- **RS:** Recommendation System
- **SDD:** System Design Document
- **SQL:** Structured Query Language
- **TOS:** Terms of Service
- **TT:** Three-tier

### 1.4: Riferimenti

Libro usato come riferimento: Object-Oriented Software Engineering (Using UML, Patterns, and Java) di Bernd Bruegge & Allen H.

Di seguito una lista di riferimenti ai vari documenti utili:

- [Statement of Work](#)
- [Requisiti funzionali e non funzionali](#)
- [Requirements Analysis Document](#)
- [Matrice di tracciabilità](#)
- [Test Plan](#)
- [Test Case Specification](#)



## 1.5: Overview

Il SDD è organizzato in cinque sezioni:

- **Introduzione:** questa sezione offre una panoramica sugli obiettivi del sistema e vengono descritti i design goal in base alla loro priorità ed elencati alcuni trade-off. Vengono specificate alcune definizioni, acronimi e abbreviazioni che saranno utilizzate nel documento. Infine, vengono indicati alcuni riferimenti ad altri documenti;
- **Architettura del sistema attuale:** in questa sezione viene esaminata l'architettura software del sistema corrente, usata come riferimento per confrontare le modifiche nell'architettura del sistema proposto;
- **Architettura del sistema proposto:** qui si definisce la suddivisione del sistema in sottosistemi, il mapping Hardware/Software e il come vengono gestiti i dati persistenti;
- **Servizi dei sottosistemi:** in questa sezione è descritta la struttura dei sottosistemi, con i vari servizi che vengono forniti da ognuno e le condizioni limite del sistema;
- **Glossario:** in questa sezione sono elencati tutti i termini e definizioni presenti all'interno del documento.

## 2: Architettura del Sistema Attuale

---

Al momento, non esiste alcun software che metta a disposizione tutte le funzionalità offerte da UniRentHub. Per questo motivo, non è possibile effettuare un confronto con architetture già esistenti.

## 3: Architettura del Sistema Proposto

---

Questo paragrafo contiene la descrizione del sistema UniRentHub sotto il punto di vista architetturale e dei servizi che offre.

### 3.1: Sintesi del Sistema

UniRentHub ha come obiettivo principale assistere gli studenti universitari nel trovare un immobile che soddisfi i loro requisiti e facilitare il processo di affitto, così come aiutare i proprietari di immobili a mettere in affitto le loro proprietà. Il tutto è orientato verso un forte impegno green, offrendo la possibilità di visualizzare specialmente immobili che utilizzano fonti di energia rinnovabile.

Il software proposto si basa su un'architettura a tre livelli (three-tier) che mantiene una suddivisione in tre componenti distinti: Model, Controller e Interface.

Per quanto riguarda il livello View, la creazione dell'interfaccia utente verrà implementata attraverso l'uso di HTML, Bootstrap, JS e Python.

Le logiche di business del sistema saranno sviluppate principalmente in Python, sfruttando il framework Flask per gestire le chiamate al back-end i vari reindirizzamenti verso il front-end.

Per la gestione dei dati persistenti, faremo uso di un database MySQL, al quale accederemo direttamente tramite il framework Flask.

### 3.2: Decomposizione in sottosistemi

In questa sezione vengono descritti i vari sottosistemi individuati:

- **Gestione Utente:** sottosistema che gestisce le operazioni di registrazione e di accesso dei vari tipi di utenti (guest, studente, locatore, admin, homechecker), i loro ruoli e le loro autorizzazioni. Inoltre gestisce i vari profili utente con le informazioni personali e si occupa del recommendation system per la ricerca di coinquilini;
- **Gestione Annunci:** sottosistema che amministra gli annunci di un locatore permettendogli di creare un nuovo annuncio, di visualizzarlo, modificare dati di un'annuncio, inserire nuove date disponibili per la

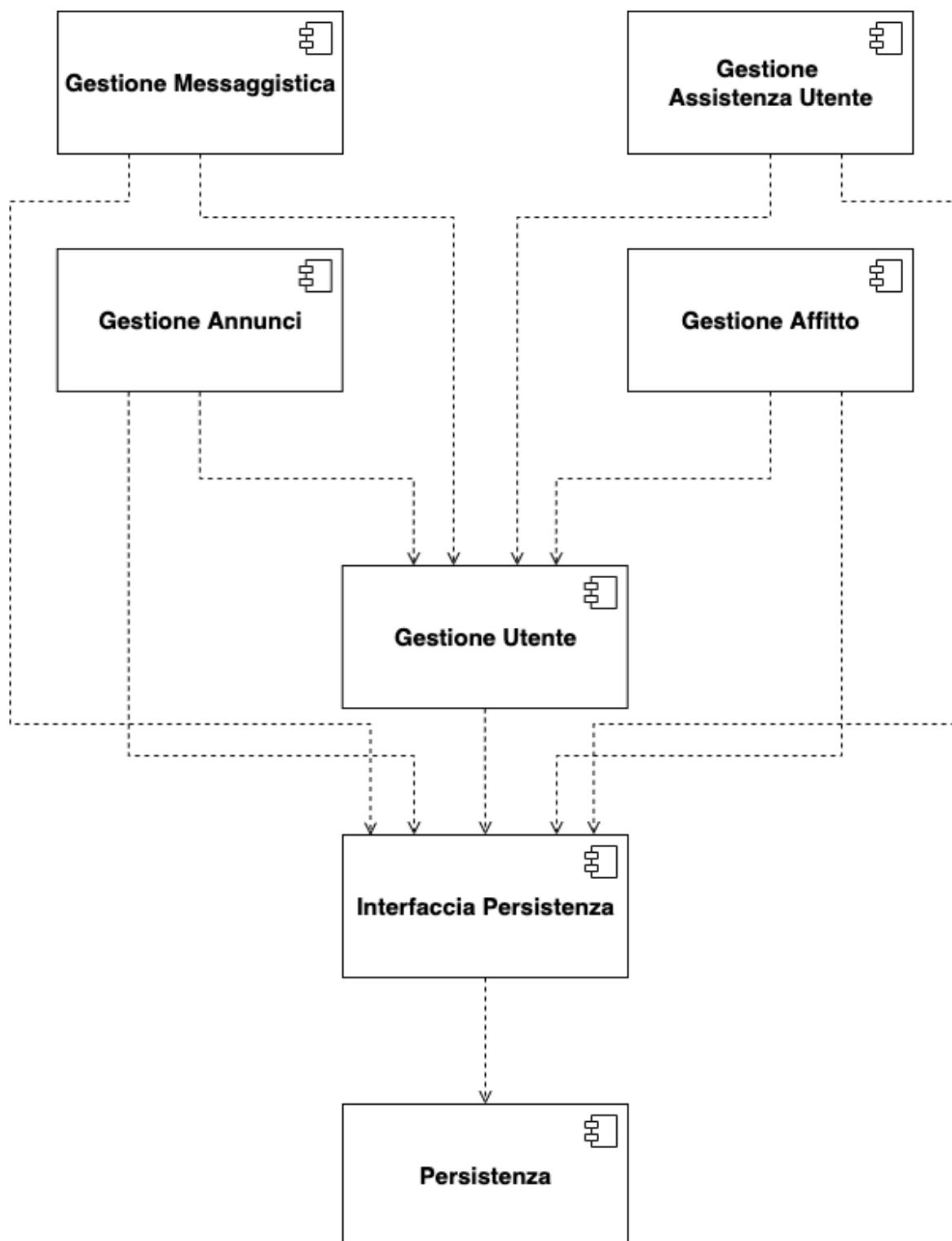


visita dell'alloggio, visualizzare lo stato di accettazione dell'homechecker e utilizzare AI per la predizione del prezzo di affitto.

Per lo studente invece, permette di poter visualizzare gli annunci di affitti con le relative informazioni, e di creare un nuovo annuncio per la ricerca di coinquilini con cui condividere un affitto.

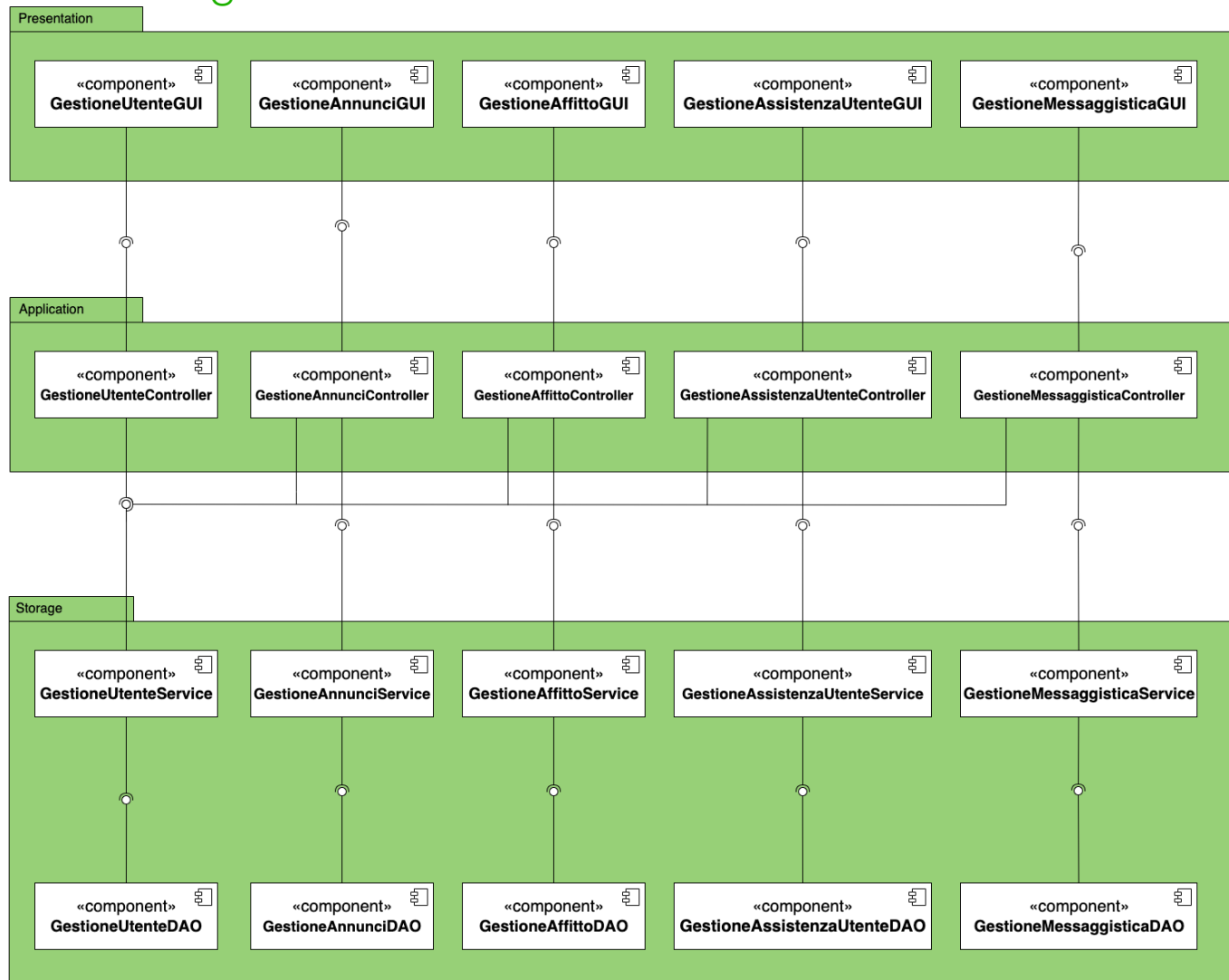
- **Gestione Affitto:** sottosistema che gestisce il meccanismo di affitto degli alloggi, permettendo agli studenti di prenotare una data per la visita di un alloggio, completare l'intera operazione con il conseguente pagamento e recensire l'alloggio che si è preso in affitto;
- **Gestione Messaggistica:** sottosistema che amministra la messaggistica per la comunicazione tra utenti, le notifiche di nuovi messaggi e l'aggiornamento sugli annunci;
- **Gestione Assistenza Utente:** sottosistema che gestisce l'assistenza all'utente tramite l'utilizzo di un chatbot fornendo un supporto nell'utilizzo corretto della piattaforma, negli spostamenti tra le varie pagine, ed un supporto nell'esecuzione di varie operazioni;
- **Interfaccia di persistenza:** interfaccia posta tra i vari sottosistemi ed il sottosistema dedicato alla persistenza;
- **Persistenza:** sottosistema che amministra la persistenza dei dati del sistema tramite l'utilizzo di una base dati.

### 3.2.1: Diagramma delle componenti



In questo diagramma sono state rappresentate tutte le dipendenze tra i vari sottosistemi.

### 3.2.2: Diagramma architetturale dei sottosistemi



In questo diagramma, invece, vengono mostrati nello specifico le componenti di ogni sottosistema:

- **GUI**, contiene le varie interfacce grafiche che verranno create e mostrate al cliente;
- **Controller**, si occupa della logica per il controllo del sistema;
- **Service**, si occupa della logica di business del sistema;
- **DAO**, si occupa di fornire accesso ai dati persistenti.

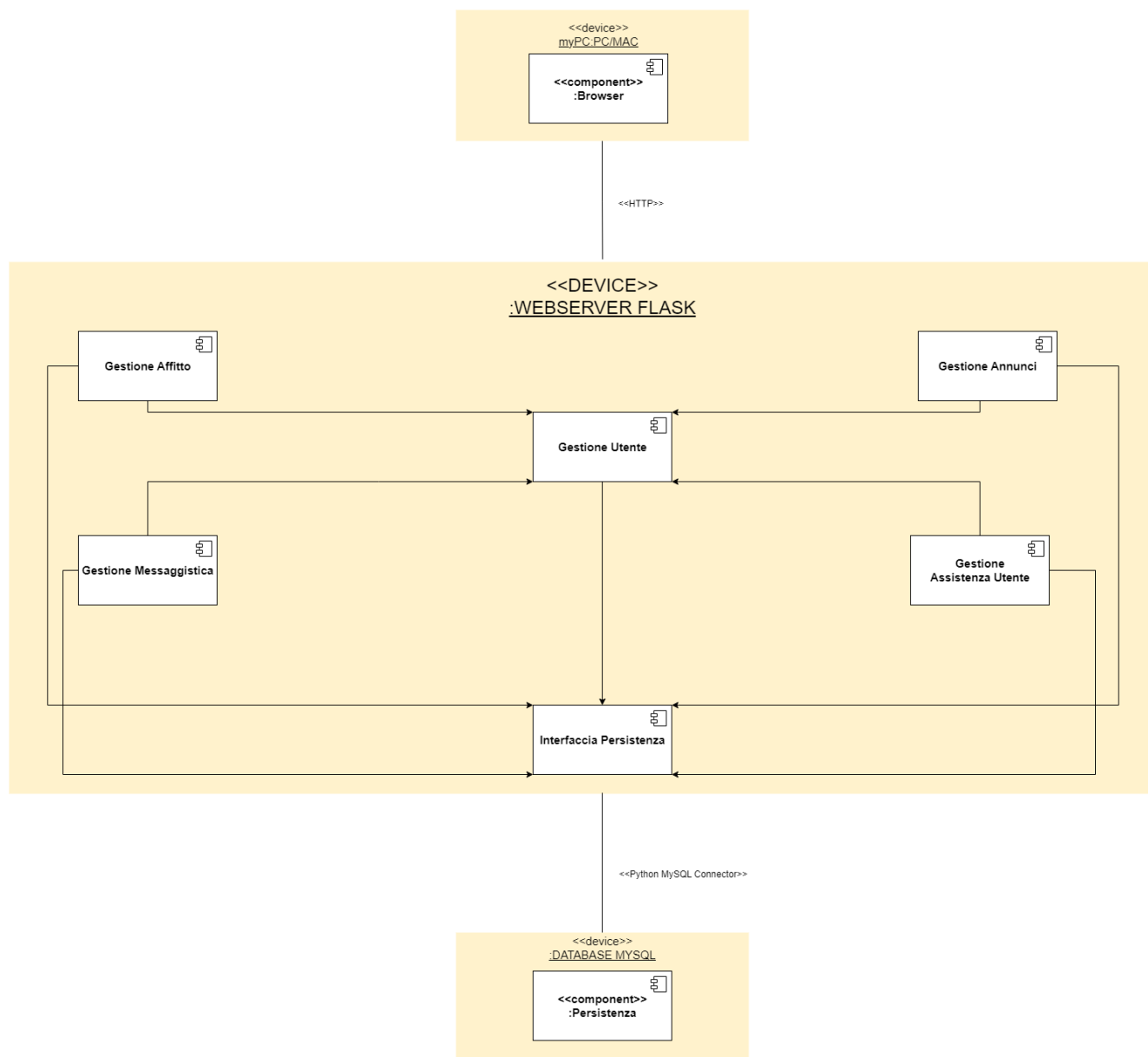
### 3.3: Mapping Hardware e Software

Il sistema utilizzerà una struttura hardware con un architettura di tipo Client-Server.

Il client è qualsiasi macchina connessa alla rete Internet con un motore di ricerca installato in modo che si possa connettere al server tramite il protocollo HTTPS, per interagire col sistema.

Il server invece è un dispositivo connesso alla rete Internet che fornisce i servizi del sistema proposto, gestisce la logica di business e la comunicazione coi vari client che richiedono dei servizi.

Infine abbiamo la gestione dei dati persistenti che viene effettuata tramite l'utilizzo del DBMS MySQL.



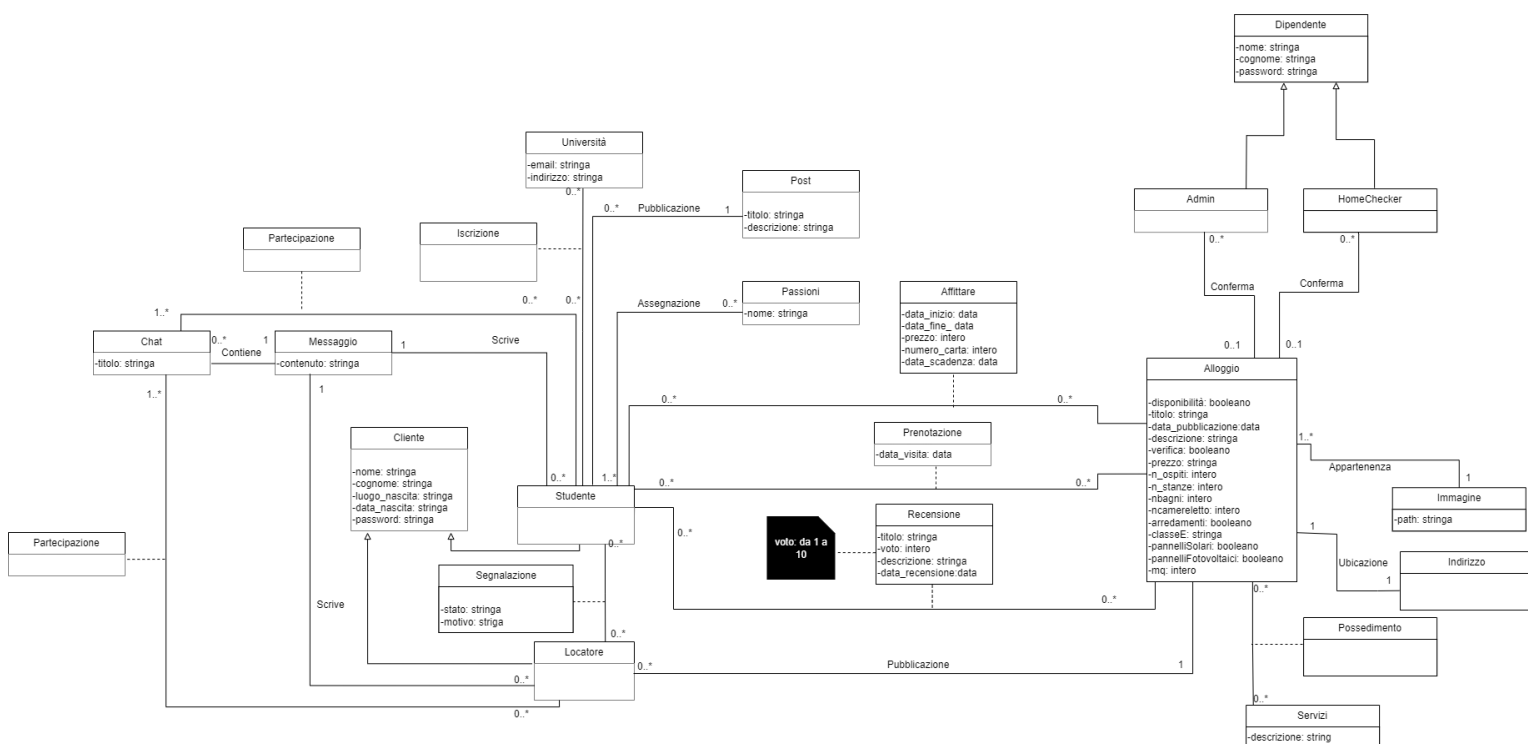
### 3.4: Gestione dei dati persistenti

Per la gestione, il salvataggio e le modifiche dei dati persistenti relativi al sistema proposto, si è scelto di utilizzare un database di tipo relazionale, per garantire una facilità di accesso, di utilizzo dei dati e una garanzia per la consistenza dei dati stessi.

Questo è possibile tramite l'uso di un DBMS, in particolare un DBMS MySQL, che ci permette di garantire:

- **Consistenza**, grazie all'utilizzo di vincoli utili per non alterare la coerenza dei dati ad ogni operazione di modifica effettuata sulla base dati;
- **Atomicità**, operazioni che possono essere effettuate per intero o non effettuate affatto in modo da garantire la consistenza dei dati.
- **Affidabilità**, grazie all'utilizzo di backup che ci permette di poter avere una copia dei dati salvati all'interno della base dati;

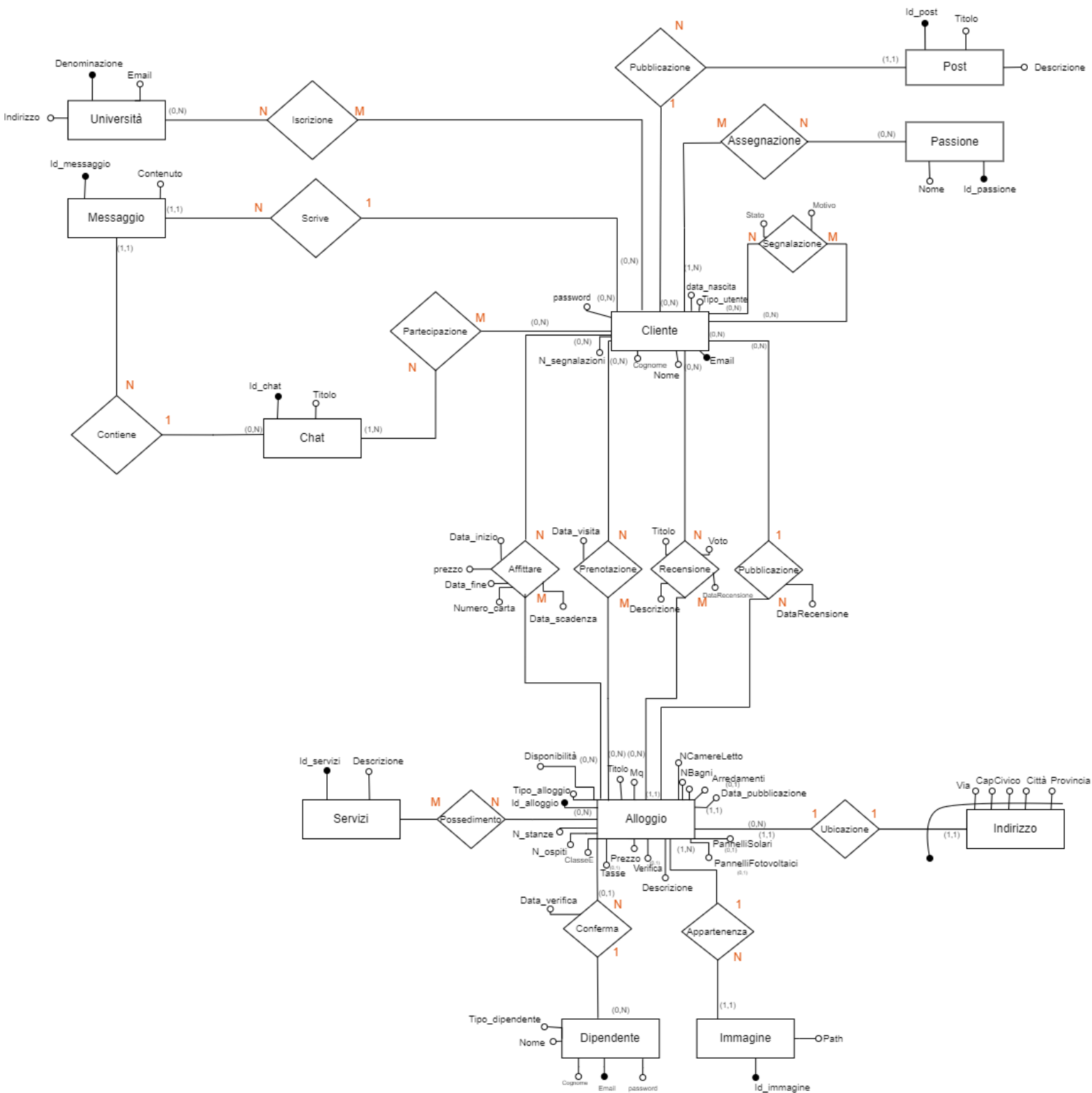
### CD: Entity Class Diagram ristrutturato



### Lista delle entità persistenti

Entità	Descrizione
<b>Cliente</b>	Rappresenta i clienti iscritti a questa piattaforma
<b>Università</b>	Rappresenta le università presenti sul territorio
<b>Iscrizione</b>	Permette di memorizzare l'università a cui è iscritto uno studente
<b>Messaggio</b>	Rappresenta il messaggio che viene inviato da un cliente
<b>Chat</b>	Rappresenta l'insieme di messaggi inviati tra due utenti
<b>Partecipazione</b>	Permette di memorizzare chi sta partecipando alla chat
<b>Passione</b>	Rappresenta gli interessi dell'utente
<b>Assegnazione</b>	Permette di assegnare più passioni a un utente e la stessa passione a più utenti
<b>Post</b>	Rappresenta un post pubblicato sulla community del sito da parte di un utente
<b>Alloggio</b>	Rappresenta la casa che verrà pubblicata sul sito
<b>Affittare</b>	Permette di memorizzare l'affitto di una casa
<b>Prenotazione</b>	Permette di memorizzare le prenotazioni di visite su una casa
<b>Recensione</b>	Permette di memorizzare le recensioni fatte da uno studente a un alloggio
<b>Segnalazione</b>	Permette di memorizzare le segnalazioni da parte degli studenti su un locatore
<b>Indirizzo</b>	Rappresenta l'indirizzo di un alloggio
<b>Immagine</b>	Comprende le foto dell'alloggio pubblicato
<b>Dipendente</b>	Rappresenta le informazioni relative ai dipendenti del sito, gli admin e gli homechecker
<b>Servizi</b>	Rappresenta i servizi inclusi nell'affitto di un determinato alloggio
<b>Possedimento</b>	Permette di collegare i servizi offerti all'alloggio

## ER: Schema ER database





## Modello Logico

**Dipendente** (Email, Nome, Cognome, Password, Tipo\_dipendente)  
**Cliente** (Email, Nome, Cognome, Tipo\_Utente, Data\_Nascita, N\_Segnalazioni, Password)  
**Università** (Denominazione, Email, Indirizzo)  
**Iscrizione** (Denominazione, *emailS*)  
**Messaggio** (Id\_messaggio, Contenuto, *Email*, *Id\_chat*)  
**Chat** (Id\_chat, Titolo)  
**Partecipazione** (Id\_chat, *EmailU1*, *EmailU2*)  
**Segnalazione** (CodiceSegnalazione, *Email*, *Email\_segnalato*, *stato*, *motivo*)  
**Post** (Id\_post, Titolo, Descrizione, *Email*)  
**Passione** (Id\_passione, Nome)  
**Assegnazione** (Id\_passione, *Email*)  
**Immagine** (Id\_immagine, *Id\_Alloggio*, path)  
**Alloggio** (Id\_Alloggio, Tipo\_alloggio, Disponibilità, Titolo, Mq, N CamereLetto, NBagni, ClasseE, Arredamenti, Data\_publicazione, PannelliSolari, PannelliFotovoltaici, Descrizione, Verifica, Prezzo, N Ospiti, N stanze, Tasse, *emailDip*, *emailLoc*)  
**Servizi** (Id\_Servizio, descrizione)  
**Possedimento** (Id\_servizio, *Id\_Alloggio*)  
**Indirizzo** (Id\_Alloggio, via, Cap, Civico, Città, Provincia)  
**Affittare** (Id\_Alloggio, *Email*, Data\_Inizio, Data\_Fine, prezzo, Numero\_carta, Data\_scadenza)  
**Prenotazione** (Id\_Alloggio, *Email*, Data\_visita)  
**Recensione** (Id\_Alloggio, *Email*, titolo, voto, descrizione, dataRecensione)

## VINCOLI

- Alla Rimozione di un Cliente vengono eliminati di conseguenza anche tutti i suoi annunci (nella community se studente, di case se locatore) e viene eliminata la sua presenza nella tabella iscrizione. Le chat rimarranno memorizzate per un breve periodo in modo da avere prove di eventuali truffe organizzate dal cliente. Rimarranno in memoria anche i dati relativi ai suoi affitti, prenotazioni e recensioni passate.
- Alla rimozione di un alloggio vengono eliminati di conseguenza il relativo indirizzo e le recensioni. Gli affitti e le prenotazioni avvenute in precedenza invece rimarranno memorizzate per un determinato periodo di tempo.
- La rimozione di un dipendente non comporta altre modifiche allo schema.
- La rimozione di una foto non comporta altre modifiche allo schema.
- La rimozione di un post non comporta altre modifiche allo schema.
- La rimozione di una passione comporta l'eliminazione nella tabella "Assegnazione" di tutti i dati correlati a quella passione.
- La rimozione di un'Università non comporta modifiche allo schema.
- La rimozione di una chat e di un messaggio non sono previste.

## 3.5: Sicurezza ed Accessi

Ogni tipo di utente avrà a disposizione diverse interfacce grafiche in modo che possa accedere alle funzionalità che rientrano nella propria categoria di utenza. Fanno eccezione l'utente guest e l'utente registrato che possono avere alcune interfacce in comune.

Admin e homechecker possono interagire con i dati del sistema solo se sono correttamente autenticati dal sistema.

L'utente guest ha la possibilità di interagire solo parzialmente con il sistema, e per poter accedere a tutte le funzionalità che quest'ultimo offre è necessaria la registrazione o l'autenticazione.



Il locatore per poter effettuare le operazioni che il sistema gli permette è obbligatoria l'autenticazione. Per potersi autenticare al sistema è necessaria un email ed una password. Vengono utilizzati due tipi di login differenti: uno per gli admin e homechecker ed un secondo per gli altri utenti registrati.

Il sistema tiene molto alla sicurezza dei dati degli utenti registrati e per questo offre protezione sui dati inseriti, in particolare password, dati utilizzati per il pagamento ed informazioni sensibili.

Per proteggere questi dati viene utilizzato l'algoritmo di cifratura RSA basato sulla complessità computazionale della fattorizzazione in numeri primi che ne garantisce la robustezza.

Inoltre, è fondamentale sottolineare l'impegno del nostro sistema nel rispettare le normative europee in materia di protezione dei dati personali, in particolare la Regolamentazione Generale sulla Protezione dei Dati (GDPR) e le normative sull'e-privacy.

In conformità con queste regolamentazioni di conseguenza garantiamo massimo rispetto per la privacy degli utenti, i quali hanno il controllo completo sui propri dati e vengono informati su come quest'ultimi vengono raccolti, utilizzati e protetti.

### 3.5.1: Matrice degli accessi

Attori	Utente Guest
Oggetti	
Gestione Utente	Registrazione
Gestione Annunci	RicercaAnnuncioAffitto VisualizzaAnnuncioAffitto CondivisioneAnnuncioAffitto
Gestione Affitto	N/A
Gestione Messaggistica	N/A
Gestione Assistenza Utente	ContattaChatBot ChiudiChatBot

Attori	Utente registrato studente	Utente registrato locatore
Oggetti		
Gestione Utente	Login Logout VisualizzaAreaUtente ModificaDatiPersonali VerificaAccount SegnalaUtente	Login Logout VisualizzaAreaUtente ModificaDatiPersonali VerificaAccount



<b>Gestione Annunci</b>	RicercaAnnuncioAffitto VisualizzaAnnuncioAffitto CreaAnnuncioCoinquilino VisualizzaAnnuncioCoinquilino ModificaAnnuncioCoinquilino InserisciAnnuncioListaPreferiti RimuoviAnnuncioListaPreferiti VisualizzaListaPreferiti RecommendationSystem CondivisioneAnnuncio RecensioneAnnuncio ModificaRecensioneAnnuncio	CreaAnnuncioAffitto VisualizzaAnuncioAffitto ModificaDatiAnnuncioAffitto InserisciDateVisitaAlloggio VisualizzaStatoAccettazioneAnnuncio Regressore CondivisioneAnnuncio
<b>Gestione Affitto</b>	PrenotaDataVisitaAlloggio VisualizzaPrenotazioneVisitaAlloggio AffittaAlloggio PagaAffittoAlloggio RecensioneAlloggio	VisualizzaPrenotazioneVisitaAlloggio
<b>Gestione Messaggistica</b>	ContattaStudiante ContattaLocatore VisualizzaChat	ContattaStudiante VisualizzaChat
<b>Gestione Assistenza Utente</b>	ContattaChatBot ChiudiChatBot	ContattaChatBot ChiudiChatBot

Attori		
Oggetti	Utente Admin	Utente Homechecker
<b>Gestione Utente</b>	Login Logout VisualizzaAreaUtente BloccaUtente RegistraHomechecker VisualizzaHomechecker EliminaHomechecke VisualizzaSegnalazioniUtente	Login Logout VisualizzaAreaUtente
<b>Gestione Annunci</b>	N/A	VisualizzaAnuncioAffitto PubblicaStatoAccettazione VisualizzaStatoAccettazione
<b>Gestione Affitto</b>	N/A	N/A
<b>Gestione Messaggistica</b>	N/A	N/A
<b>Gestione Assistenza Utente</b>	N/A	N/A

### 3.6: Controllo globale del software

Il flusso di controllo del nostro sistema è un controllo centralizzato. Più nel dettaglio è un event-driven. In questo controllo di flusso il sistema aspetta che si verifichi un evento esterno.

Quando l'evento esterno si verifica, la richiesta viene direzionata opportunamente all'oggetto interessato. Alla ricezione dei dati da parte del sistema il flusso di controllo, esso diventa di tipo procedure-driven nel quale la logica di business aspetta i dati richiesti per poterli elaborare.

### 3.7: Condizioni limite

In questo paragrafo verranno rappresentati le boundary conditions relativi all'avvio del sistema, spegnimento del sistema, arresto del sistema e errore di accesso ai dati persistenti.

#### Avvio del sistema

<b>Identificativo</b> <i>UCBC_1</i>	<i>UCBC_1 – Avvio del Sistema</i>	<i>Data</i>	<i>29/11/2023</i>
		<i>Vers.</i>	<i>1.0</i>
		<i>Autore</i>	<i>Greco Marco</i>
<b>Descrizione</b>	<i>Lo UC permette l'avvio del sistema</i>		
<b>Attore Principale</b>	<b>Nome</b> Admin		
<b>Attori Secondari</b>	<b>Nome</b> N/A		
<b>Entry Condition</b>	Il sistema è spento AND l'admin accede al server AND l'admin invia il segnale di avvio del sistema		
<b>Exit Condition</b> On success	Il sistema è avviato correttamente		
<b>Exit Condition</b> On failure	Il sistema non viene avviato correttamente		
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>			
1	Admin:	Esegue sulla macchina il comando di avvio del sistema	
2	Sistema:	Il server si collega al database	
3	Sistema:	Il server si collega ad internet	
4	Sistema:	Il sistema si mette in attesa per la ricezione di richieste	
<b>I Scenario/Flusso di eventi Alternativo: mancata connessione al database</b>			
2.1	Sistema:	Invia il segnale di eccezione di mancata connessione al database	
2.2	Admin:	Controlla il collegamento al database	
2.3	Admin:	Effettua la connessione al database	
<b>II Scenario/Flusso di eventi Alternativo: Connessione ad Internet non riuscita</b>			
3.1	Sistema:	Invia il segnale di eccezione di mancata connessione a internet	
3.2	Admin:	Controlla la connessione a internet	
3.3	Admin:	Effettua la connessione a internet	

### Spegnimento del sistema

Identificativo UCBC_2		UCBC_2 - Spegnimento del sistema	Data	29/11/2023
			Vers.	1.0
			Autore	Fasolino Iliano
Descrizione		Lo UC permette lo spegnimento del sistema		
Attore Principale		Nome Admin		
Attori secondari		Nome N/A		
Entry Condition		L'admin accede al server and il sistema è stato precedentemente avviato and il sistema non è ancora spento		
Exit Condition On success		Il sistema viene spento correttamente		
Exit Condition On failure		Il sistema non viene spento		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Admin:	Invia il segnale di spegnimento al sistema		
2	Sistema:	Controlla le connessioni attive con l'esterno, se ciò viene correttamente verificato termina l'esecuzione del sistema		
I Scenario/Flusso di eventi Alternativo: ci sono connessioni ancora attive (aspetta lo slot di tempo)				
2.1	Sistema:	Manda un segnale di eccezione segnalando all'admin che ci sono ancora connessioni attive con l'esterno		
2.2	Sistema:	Aspetta uno slot di tempo per rispondere ad eventuali richieste dall'esterno senza generare nuove connessioni se esse non sono essenziali per la risposta alle già esistenti richieste		
2.3	Sistema:	Controlla di nuovo se ci sono connessioni verso l'esterno, se questa condizione è verificata spegne il sistema		
2.4	Sistema:	Notifica l'admin dell'avvenuto spegnimento del sistema		
II Scenario/Flusso di eventi Alternativo: connessioni ancora attive (interrompe le connessioni)				
2.5	Sistema:	Interrompe le connessioni verso l'esterno		
2.6	Sistema:	Notifica l'admin dell'avvenuto spegnimento del sistema		

### Errore di accesso ai Dati Persistenti

<b>Identificativo</b> <i>UCBC_3</i>	<i>UCBC_3 - Errore nell'accesso dei dati persistenti</i>		<i>Data</i>	<i>29/11/2023</i>
			<i>Vers.</i>	<i>1.0</i>
			<i>Autore</i>	<i>Giuseppe Pio Sorrentino</i>
<b>Descrizione</b>	<i>Lo UC descrive il comportamento del sistema nel caso vi fosse un errore nell'accesso ai dati persistenti</i>			



<b>Attore Principale</b>		<b>Nome</b> Admin
<b>Attori Secondari</b>		<b>Nome</b> N/A
<b>Entry Condition</b>		Si è verificato un errore nell'accesso ai dati persistenti <b>or</b> i dati sono corrotti
<b>Exit Condition</b> On success		Il sistema riprende il suo normale funzionamento
<b>Exit Condition</b> On failure		Il sistema non riprende il suo normale funzionamento
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>		
1	Sistema:	Notifica all'admin che non è stato in grado di accedere ai dati persistenti
2	Sistema:	Smette di processare le richieste
3	Admin:	Include (UCBC_2)
4	Admin:	Risolve l'errore e ripristina l'accesso ai dati
5	Admin:	Include (UCBC_1)

### Crash del sistema

<b>Identificativo</b> <i>UCBC_4</i>	<i>UCBC_4 - Crash del sistema</i>	<i>Data</i>	<i>29/11/2023</i>
		<i>Vers.</i>	<i>1.0</i>
		<i>Autore</i>	<i>Francesco Pio Contaldo</i>
<b>Descrizione</b>	<i>Lo UC descrive il crash del sistema e ne permette la risoluzione</i>		
<b>Attore Principale</b>	<b>Nome</b> Admin		
<b>Attori Secondari</b>	<b>Nome</b> N/A		
<b>Entry Condition</b>	Il sistema è stato precedentemente avviato and il sistema va in crash		
<b>Exit Condition</b> On success	Vengono avviate procedure di gestione delle eccezioni and il sistema viene riavviato e continua la sua normale esecuzione		
<b>Exit Condition</b> On failure	Il sistema non continua la sua normale esecuzione		
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>			
1	Sistema:	Notifica l’admin dell’avvenuta eccezione	
2	Admin:	Include (UCBC_2)	
3	Admin:	Avvia le procedure di gestione degli errori	
4	Admin:	<i>Include (UCBC_1)</i>	

## 4: Servizi dei sottosistemi

Sono elencati di seguito i vari servizi erogati dai sottosistemi specificati nella sezione precedente.

### Servizi del sottosistema Gestione Utente

Servizio	Descrizione
<b>Registrazione Studente</b>	Permette la registrazione di uno studente
<b>Registrazione Locatore</b>	Permette la registrazione di un locatore
<b>Login</b>	Permette l'accesso alla piattaforma ad un utente precedentemente registrato
<b>Logout</b>	Permette la disconnessione dalla piattaforma ad un utente che precedentemente è acceduto
<b>Verifica Account</b>	Permette ad un utente registrato di poter verificare il proprio account tramite link via mail
<b>Modifica Dati Personali</b>	Permette ad un utente di poter modificare i dati personali del proprio account
<b>Segnala Utente</b>	Permette ad un utente di poter segnalare un altro utente della piattaforma
<b>Blocca Utente</b>	Permette ad un utente admin di bloccare un utente del sito impedendogli di effettuare operazioni sulla piattaforma
<b>Registra Homechecker</b>	Permette ad un utente admin di poter registrare nella piattaforma un utente homechecker
<b>Elimina Homechecker</b>	Permette ad un utente admin di poter eliminare dalla piattaforma un utente homechecker
<b>Visualizza Homechecker</b>	Permette ad un utente admin di poter visualizzare i vari homechecker registrati sulla piattaforma
<b>RecommendationSystem</b>	Permette ad un utente studente di poter usufruire di questo sistema per la ricerca di altri studenti con cui condividere l'affitto in base alle proprie passioni indicate



### Servizi del sottosistema Gestione Annunci

Servizio	Descrizione
<b>Ricerca Annuncio Affitto</b>	Permette ad un utente guest o ad un utente studente di poter ricercare gli annunci pubblicati sulla piattaforma
<b>Visualizza Annuncio Affitto</b>	Permette ad un utente guest o ad un utente studente di poter visualizzare la pagina di un annuncio con le relative informazioni
<b>Condivisione Annuncio</b>	Permette ad un utente guest, un utente studente e locatore di poter condividere la pagina di un annuncio
<b>Crea Annuncio Affitto</b>	Permette ad un utente locatore di poter pubblicare un annuncio relativo ad una sua proprietà che intende dare in affitto
<b>Modifica Dati Annuncio Affitto</b>	Permette ad un utente locatore di poter modificare i dati relativo ad un annuncio precedentemente pubblicato
<b>Inserimento Date Visita Alloggio</b>	Permette ad un utente locatore di poter inserire nuove date per la visita ad un alloggio da lui pubblicato sulla piattaforma
<b>Regressorore</b>	Permette ad utente locatore di poter ricevere una stima del valore da assegnare al suo annuncio in base ai prezzi del mercato locale
<b>Pubblica Stato Accettazione Annuncio</b>	Permette ad un utente homechecker di poter inserire lo stato di accettazione relativo ad un annuncio pubblicato da un utente locatore
<b>Visualizza Stato Accettazione Annuncio</b>	Permette ad un utente homechecker di poter visualizzare lo stato di accettazione relativo ad un annuncio. Permette ad un utente locatore di poter visualizzare lo stato di accettazione relativo ad un suo annuncio precedentemente pubblicato
<b>Crea Annuncio Coinquilino</b>	Permette ad un utente studente di poter pubblicare un annuncio sulla piattaforma indicando che è in cerca di un coinquilino
<b>Modifica Dati Annuncio Coinquilino</b>	Permette ad un utente studente di poter modificare i dati relativo ad un annuncio precedentemente pubblicato
<b>Visualizza Annuncio Coinquilino</b>	Permette ad un utente studente di poter visualizzare un annuncio di ricerca coinquilino sulla piattaforma





<b>Inserisci Annuncio Lista Preferiti</b>	Permette ad un utente studente di poter inserire un annuncio in una lista personale che può raggiungere in un secondo momento
<b>Rimuovi Annuncio Lista Preferiti</b>	Permette ad un utente studente di poter rimuovere un annuncio inserito precedentemente nella lista
<b>Visualizza Lista Preferiti</b>	Permette ad un utente studente di poter visualizzare tutti gli annunci inseriti precedentemente all'interno della lista
<b>Segnala Annuncio</b>	Permette ad un utente studente, che ha preso in affitto un alloggio, di poter effettuare una segnalazione
<b>Visualizza Segnalazioni Annunci</b>	Permette ad un utente admin di poter visualizzare tutte le segnalazioni effettuate a degli annunci pubblicati sulla piattaforma

### Servizi del sottosistema Gestione Affitto

Servizio	Descrizione
<b>Prenota Data Visita Alloggio</b>	Permette ad un utente studente di prenotare una data per visitare un alloggio pubblicato sulla piattaforma
<b>Affitta Alloggio</b>	Permette ad un utente studente di poter effettuare la pratica di locazione di un alloggio
<b>Pagamento Affitto</b>	Permette ad utente studente di poter effettuare il pagamento per completare la pratica di presa di locazione di un alloggio
<b>Recensione Alloggio</b>	Permette ad un utente studente di poter recensire un alloggio da lui preso in locazione
<b>Modifica Recensione Alloggio</b>	Permette ad un utente studente, che ha preso in affitto un alloggio, di poter effettuare delle modifiche ad una recensione pubblicata precedentemente sulla piattaforma
<b>Visualizza Prenotazione Visita Alloggio</b>	Permette ad un utente locatore di visionare le prenotazioni ricevute ad un suo alloggio pubblicato sulla piattaforma.



### Servizi del sottosistema Gestione Messaggistica

Servizio	Descrizione
<b>Contatta Studente</b>	Permette ad un utente studente, o locatore, di poter contattare un altro utente studente registrato alla piattaforma
<b>Contatta Locatore</b>	Permette ad un utente studente di poter contattare un utente locatore registrato alla piattaforma
<b>Visualizza Chat</b>	Permette ad un utente studente o locatore di poter visualizzare la chat precedentemente iniziata

### Servizi del sottosistema Gestione Assistenza Utente

Servizio	Descrizione
<b>Contatta ChatBot</b>	Permette ad un utente che ha fatto accesso alla piattaforma di poter contattare il chatbot ed effettuare richieste di assistenza per lo svolgimento di operazioni
<b>Chiudi ChatBot</b>	Permette ad un utente che ha precedentemente contattato un chatbot di chiudere l'assistenza

## 5: Glossario

---

Di seguito un'utile raccolta di termini chiave e relative definizioni per l'aiuto della comprensione di concetti specifici. Una guida rapida e chiara attraverso il linguaggio tecnico, fornendo una panoramica esaustiva e accessibile a tutti.

Ha lo scopo di fornire chiarezza ed una migliore comprensione dei termini essenziali che sono stati utilizzati all'interno del SDD.

- **DBMS: (Database Management System)** sistema software progettato per consentire la creazione, la manipolazione e l'interrogazione efficiente di database.
- **MySQL:** DBMS relazionale open source, utilizza SQL come linguaggio di query.
- **Three-Tier:** Architettura software formata da 3 stati disaccoppiati, ovvero Presentazione, Logica e Persistenza.
- **Modello ER:** Modello Entità-Relazioni, usato per rappresentare graficamente dei dati, dove si definiscono le varie relazioni tra le entità, usato per la progettazione di basi di dati.
- **Framework:** Architettura di supporto il quale può esser usato per realizzare un software sfruttando l'infrastruttura che egli possiede, aiutando il programmatore a focalizzarsi principalmente nella realizzazione del suo progetto.
- **Flask:** Framework Python-based usato per sviluppare applicazioni web.
- **HTTPS:** Protocollo di comunicazione sicura per contenuti web.