



UNIVERSITA' DEGLI STUDI DI NAPOLI PARTHENOPE

PROGETTO DI INGEGNERIA DEL SOFTWARE E INTERAZIONE UOMO-MACCHINA

“EIGHT B&B”

System Design Document



CANDIDATI:

| | |
|---------------------|------------|
| SIMONE CIOFFI | 0124002047 |
| LUIGI MALVONE | 0124002200 |
| NICOLA IMPRONTA | 0124002159 |
| MARIO GIORDANO | 0124002104 |
| GIUSEPPE CUCCURULLO | 0124002206 |

Indice

| | |
|--|----|
| 1. Introduzione: | 3 |
| 1.1. Scopo del sistema: | 3 |
| 1.2. Obiettivi di progettazione: | 3 |
| 1.3. Definizioni, acronimi e abbreviazioni: | 4 |
| 1.4. Riferimenti: | 4 |
| 1.5. Panoramica: | 5 |
| 1.5.1. Authentication: | 5 |
| 1.5.2. Booking: | 6 |
| 1.5.3. Payment: | 7 |
| 1.5.4. PrivateClient: | 8 |
| 1.5.5. PrivateManagmentDocument: | 9 |
| 1.5.6. PrivateOwner: | 10 |
| 1.5.7. Search: | 11 |
| 1.5.8. API: | 12 |
| 1.5.9. Database: | 13 |
| 2. Sistema corrente: | 14 |
| 3. Sistema proposto: | 15 |
| 3.1 Panoramica: | 15 |
| 3.2 Decomposizione del sistema: | 16 |
| 3.3 Hardware/Software mapping: | 18 |
| 3.4 Gestione dei dati persistenti: | 19 |
| 3.5 Controllo accessi e sicurezza: | 19 |
| 3.6 Decisioni sul flusso di controllo globale: | 20 |
| 3.7 Condizioni limite: | 20 |
| 4. Servizi del sottosistema: | 21 |
| 5. Glossario | 22 |

1. Introduzione:

1.1. Scopo del sistema:

Il sistema da noi proposto ha lo scopo di raccogliere una serie di strutture, come Hotels e B&B, e farle visualizzare all'utente che accede al sistema.

Queste strutture vengono visualizzate anche con una serie di immagini e con una descrizione sul tipo di struttura e dei servizi che questi ultimi offrono. In base a ciò, l'utente che accede al sistema, dopo una registrazione, ha la possibilità di prenotare una determinata struttura, per un determinato periodo di tempo.

Il sistema permette, inoltre, al proprietario della struttura, di gestire telematicamente la propria struttura, aggiungendo promozioni e modificando le informazioni della propria struttura, o adempiere alle questioni burocratiche riguardanti la struttura stessa.

1.2. Obiettivi di progettazione:

Durante la scoperta dei requisiti, abbiamo specificato i seguenti requisiti non funzionali per Eight B&B:

- **Usabilità:** Il software deve essere intuitivo da usare, con un'interfaccia semplice sia per il cliente sia per i proprietari
- **Conformità sulle linee guida:** Sistema sviluppato su Java.
- **Privacy dati:** I dati di ogni singolo utente non devono essere resi pubblici.
- **Garanzia di modifica dati:** L'utente deve avere la possibilità di visualizzare, aggiornare, inserire e rimuovere, tramite il sistema, i dati in un certo intervallo di tempo.
- **Garanzia multiutenza:** Il sistema deve garantire il corretto funzionamento del sistema stesso per un massimo di 3.500 utenti attivi contemporaneamente.

1.3. Definizioni, acronimi e abbreviazioni:

CF= acronimo che definisce l'attributo "Codice Fiscale" di Cliente e Proprietario

n.person= abbreviazione di "numero di persone", utilizzato nella sezione di prenotazione di una struttura

n.child= abbreviazione di "numero di bambini", utilizzato nella sezione di prenotazione di una struttura

num.cam.=abbreviazione di "numero camera", utilizzato nella sezione di prenotazione di una struttura

desc.= abbreviazione di "descrizione", riferendosi alla descrizione di una determinata struttura, immessa dal proprietario, e visualizzato dal cliente.

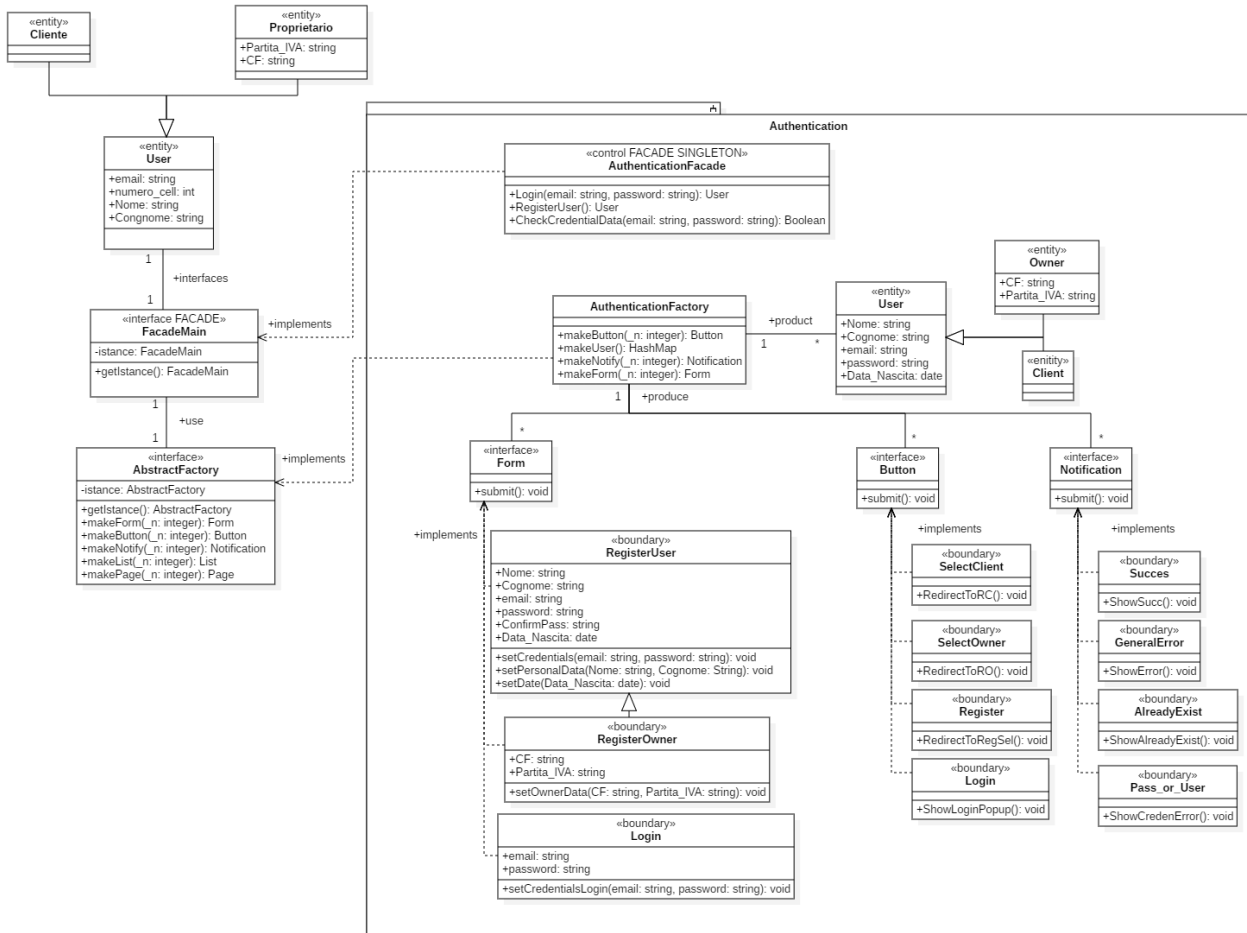
1.4. Riferimenti:

si faccia riferimento al RAD del sistema, allegato insieme a questo documento per una maggiore compressione del sistema creato.

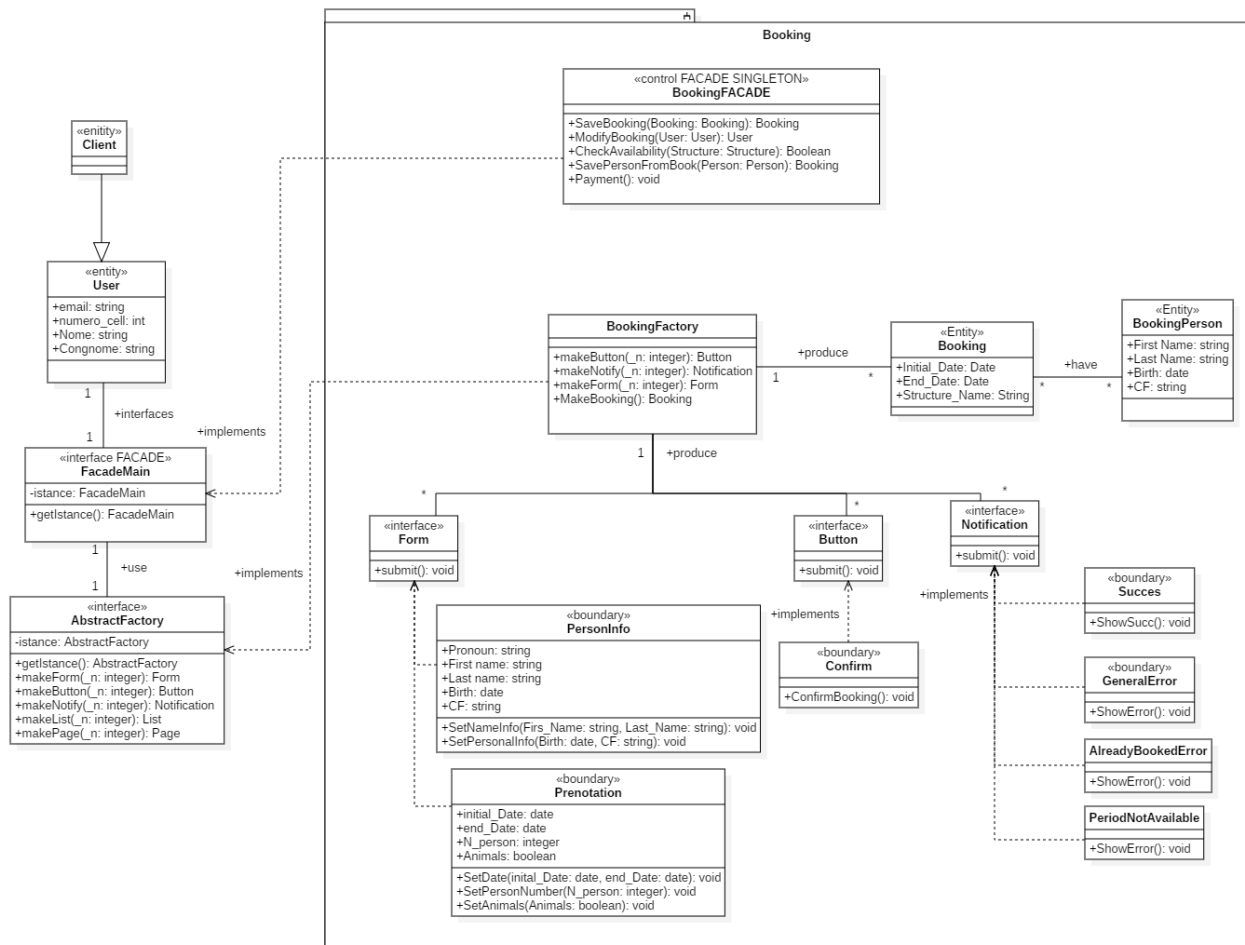
1.5. Panoramica:

Di seguito viene riportata la versione decomposta e raffinata del diagramma ad oggetti precedentemente visto nel RAD. In particolar modo si precisa che i pattern utilizzati sono: SINGLETON, FACADE, ABSTRACT FACTORY.

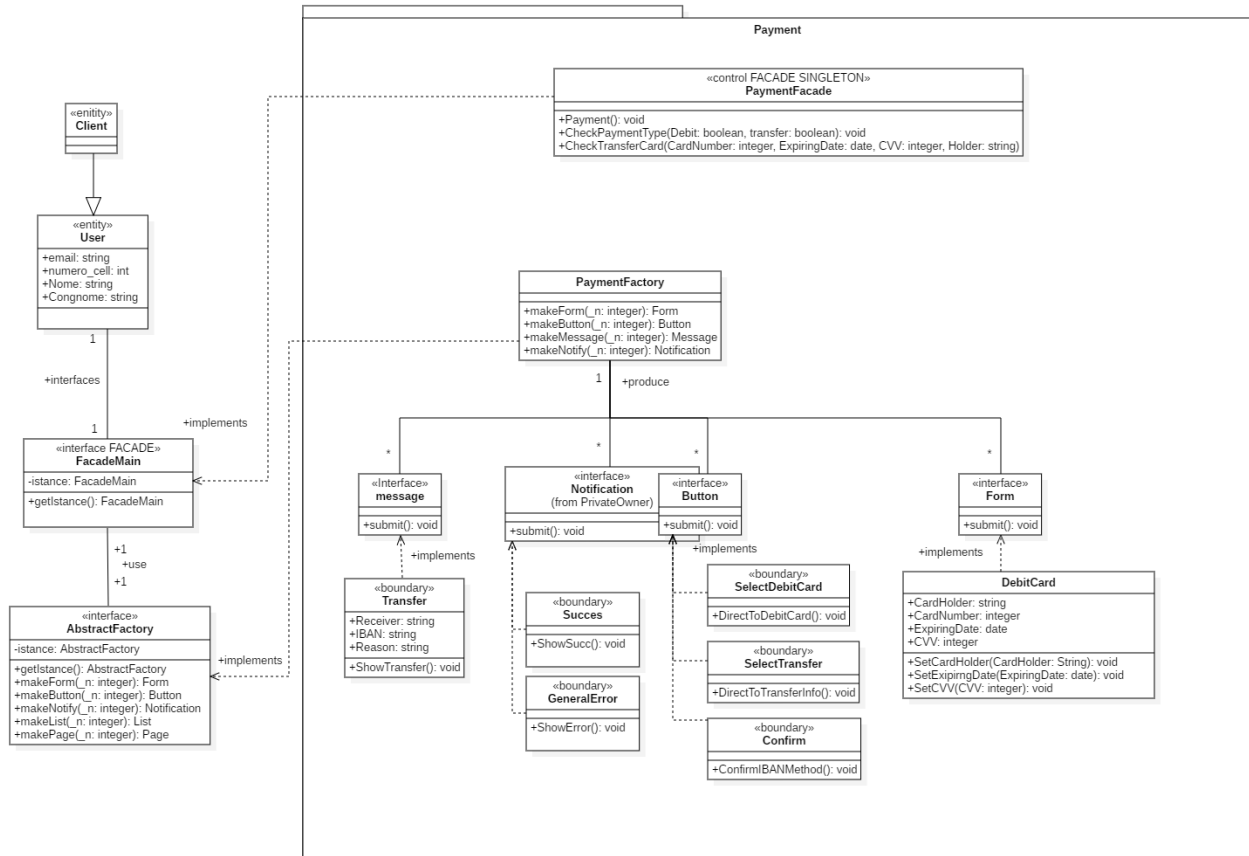
1.5.1. Authentication:



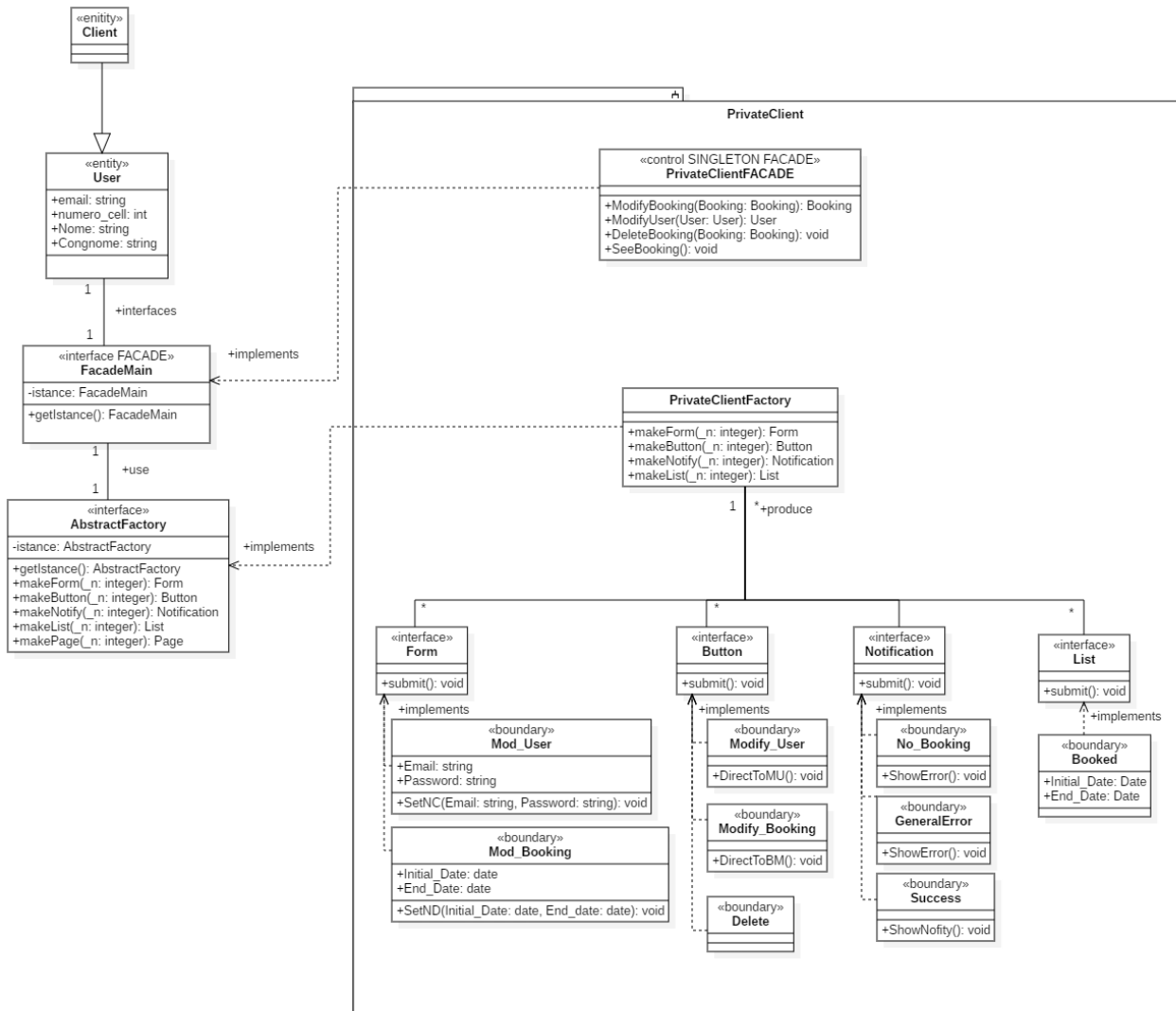
1.5.2. Booking:



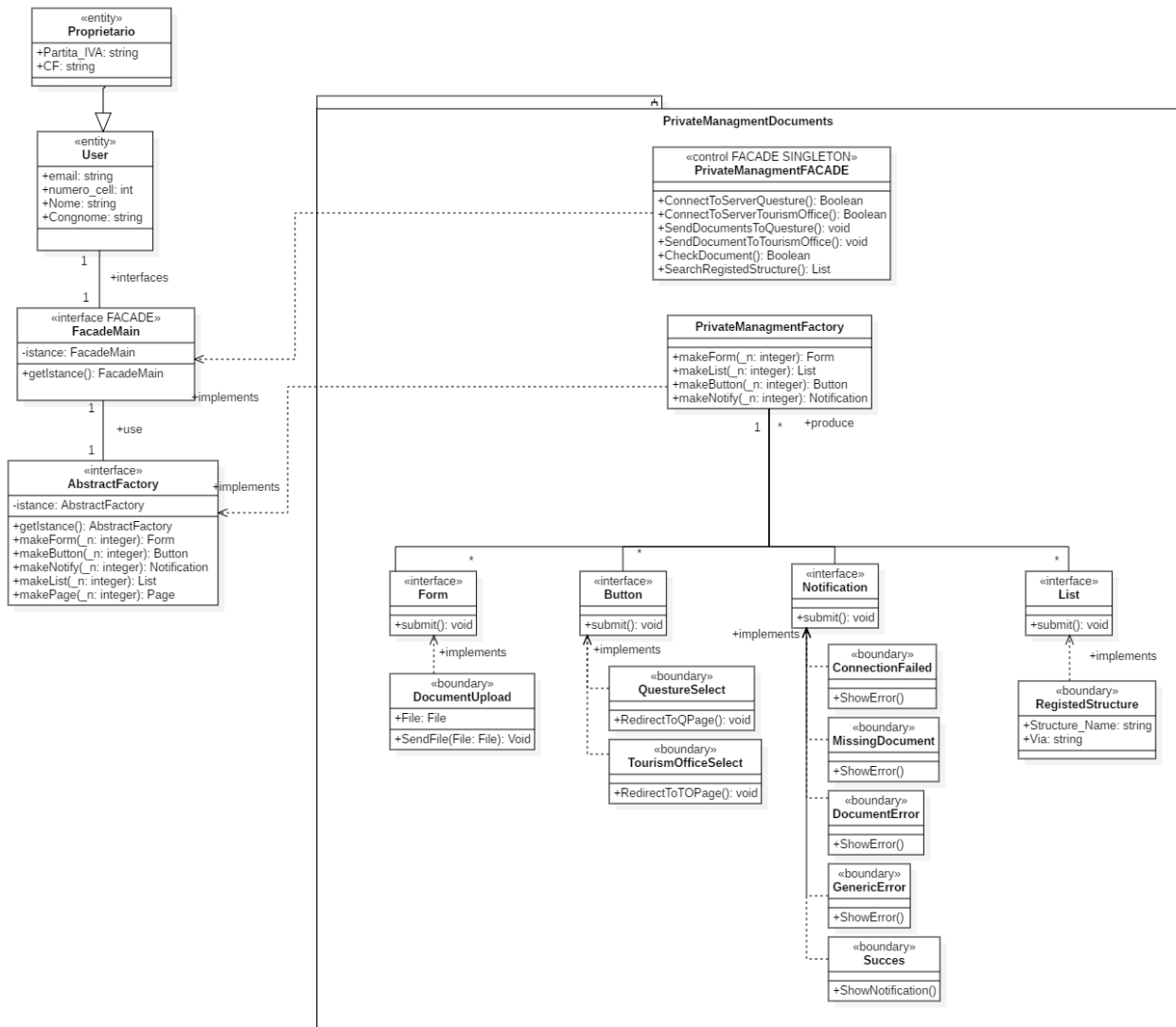
1.5.3. Payment:



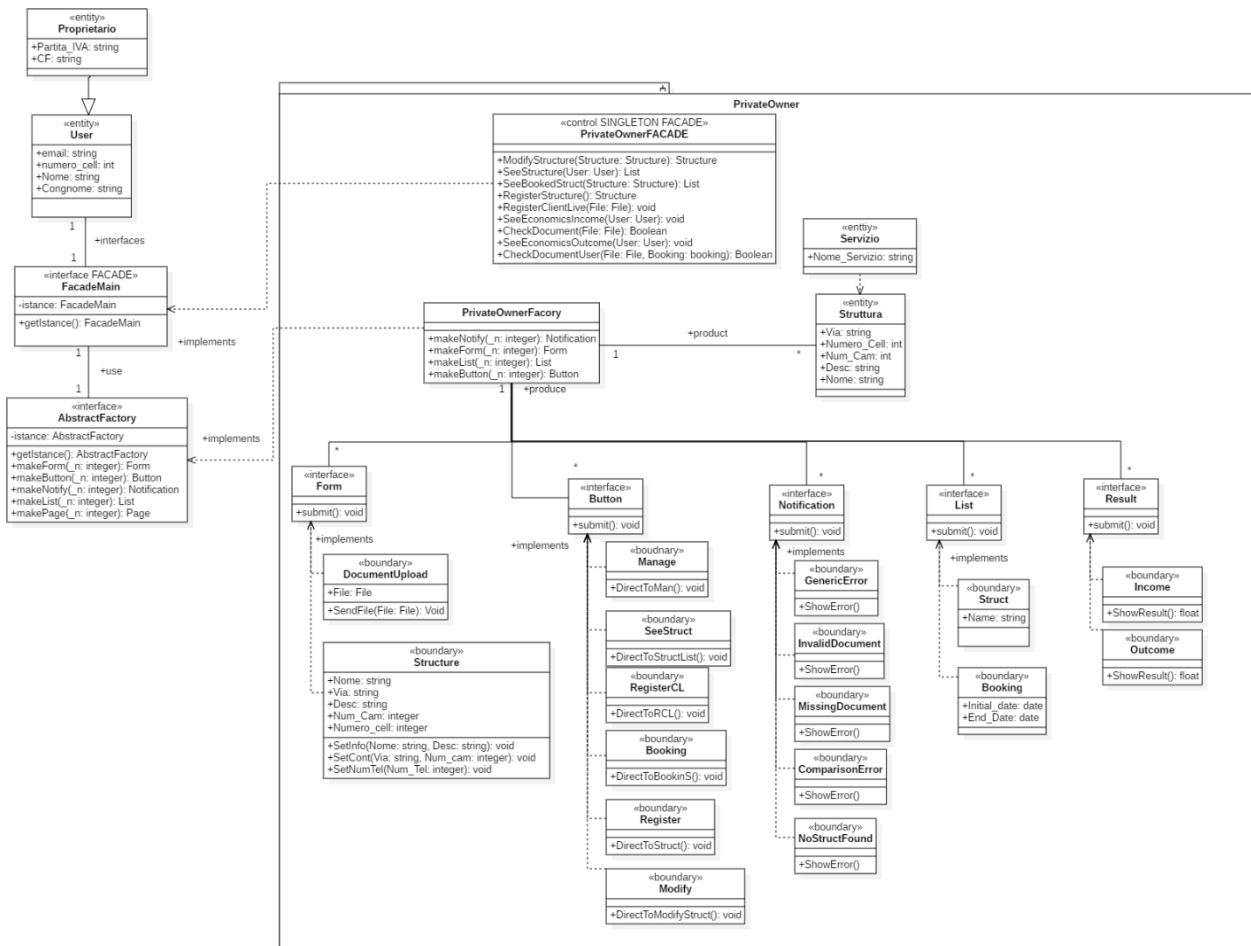
1.5.4. PrivateClient:



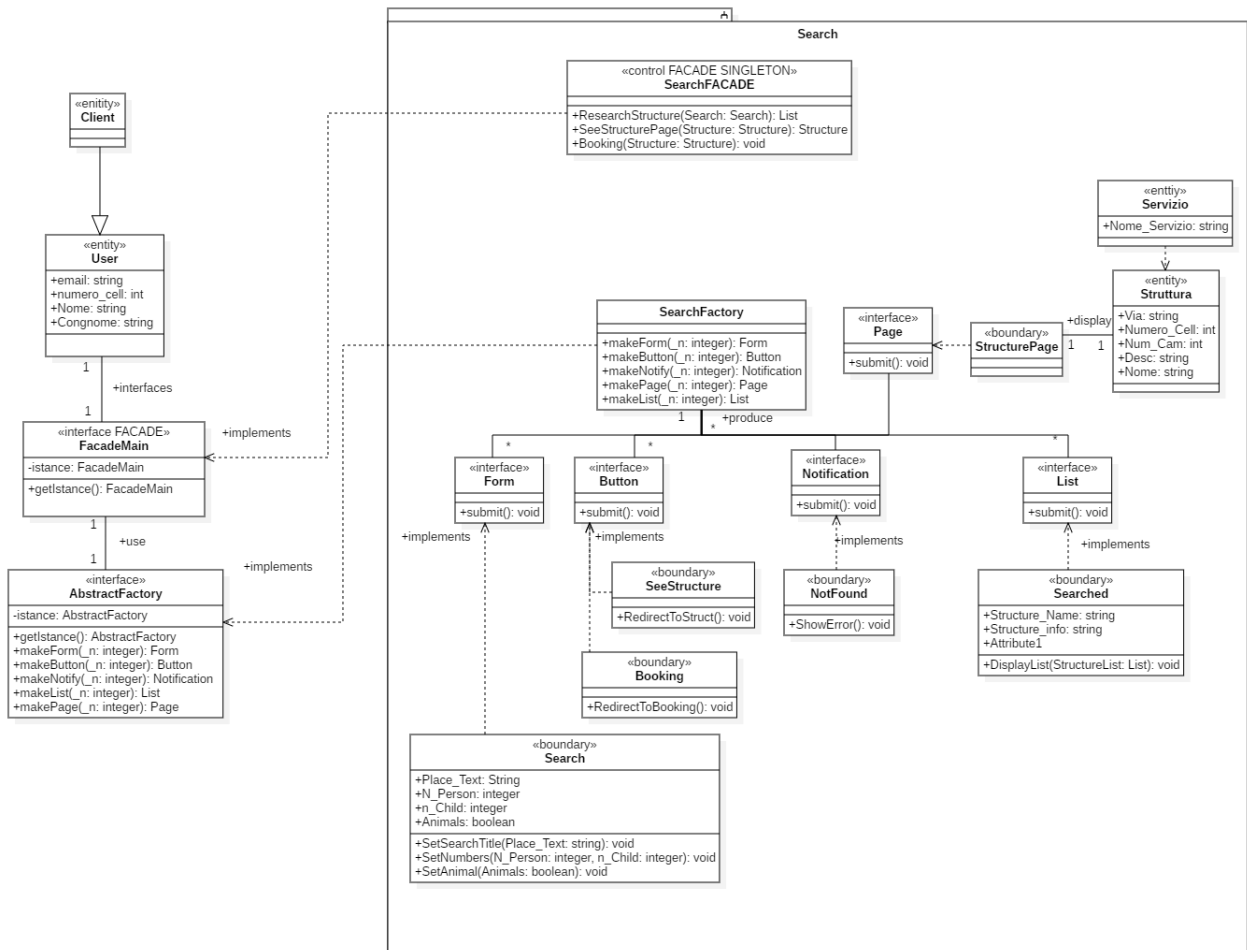
1.5.5. PrivateManagmentDocument:



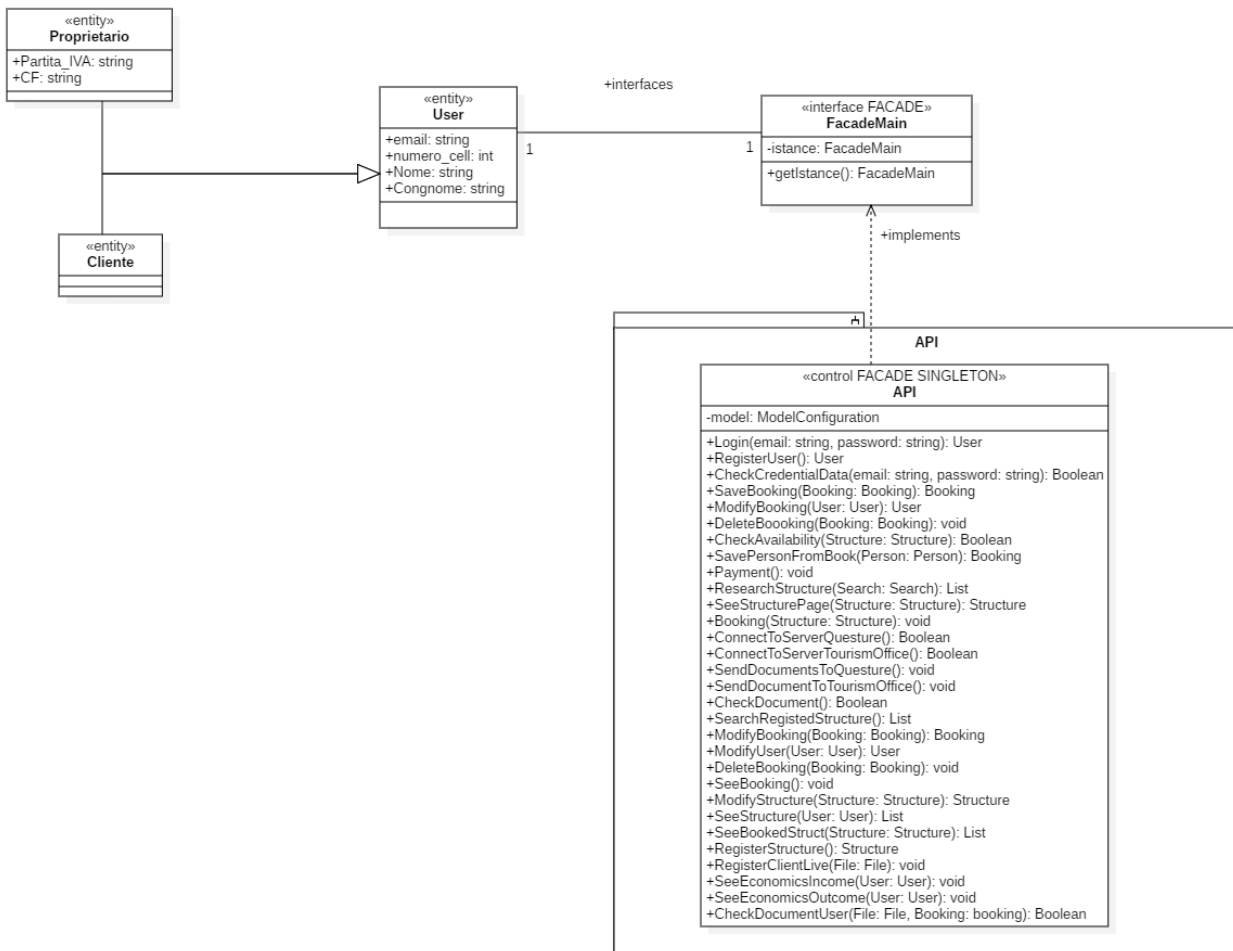
1.5.6. PrivateOwner:



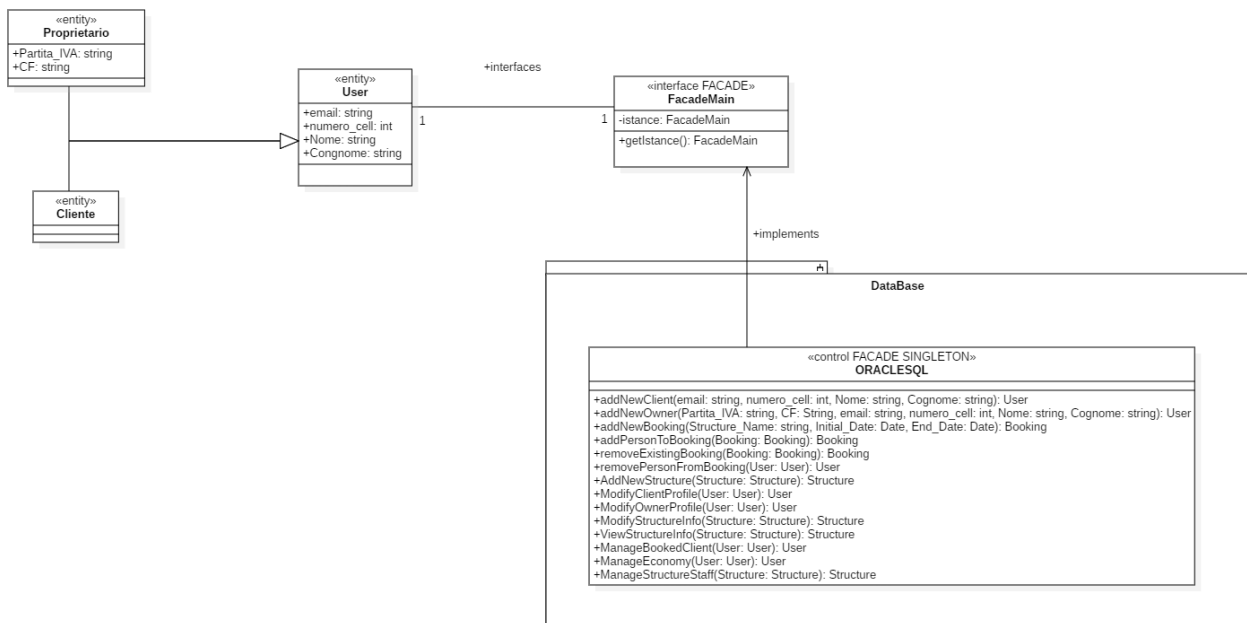
1.5.7. Search:



1.5.8. API:



1.5.9. Database:



2. Sistema corrente:

Il nostro sistema in questione è frutto di un progetto Greenfield che si ispira ad un progetto già esistente chiamato AirBnb.



AirBnb è un portale online che permette agli utenti di prenotare e visualizzare B&B in tutto il mondo. Tale software garantisce vantaggi anche ai proprietari, offrendogli la possibilità di gestire la propria struttura al meglio.

3. Sistema proposto:

3.1 Panoramica:

Il sistema proposto utilizza come stile architetturale lo stile client/server, individuando un client condiviso tra due utenti e un server che gestisce tutte le operazioni di controllo e interazione con il database.

I due utenti analizzati sono Cliente e Proprietario, ed entrambi hanno accesso a determinati sottosistemi analizzati. L'unico sottosistema in comune è quello di Authentication, ovvero quello di registrazione ed accesso al sistema.

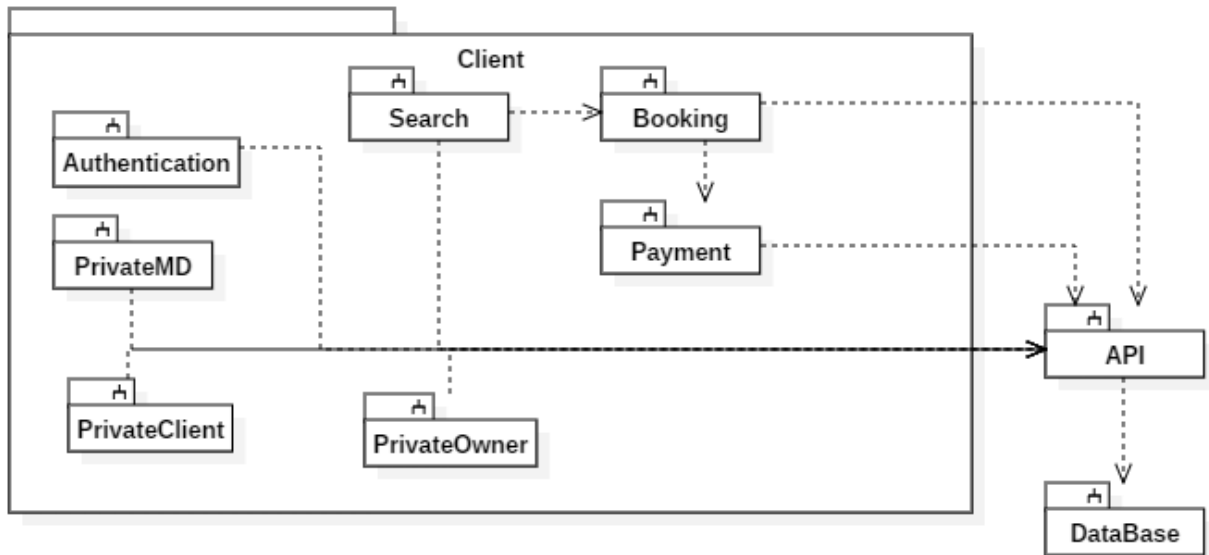
Il cliente può ricercare e visualizzare le strutture desiderate all'interno del sistema ed eventualmente prenotarsi a tali strutture, inoltre può modificare o cancellare tale prenotazione nella sua area privata.

Il proprietario invece può gestire le sue strutture al meglio, registrandole o modificandole e inoltre, nella sua area privata, può visualizzare tutte le prenotazioni effettuate in tale struttura. Può inoltre visualizzare le entrate e uscite di tutte le strutture registrate, e anche caricare e mandare a enti nazionali documenti come, documenti burocratici e tasse.

Il server dispone di un API in grado di fornire una serie di servizi richiesti dal cliente, inoltre è colui che fa da tramite tra il client e il Database.

Le ragioni per cui si è scelto uno stile architetturale Client/Server, ovvero un'applicazione di rete, piuttosto dei altri perché si trova la possibilità di gestire i vari compiti tra le varie tipologie di utenti registrati, garantendo così maggiore usabilità del sistema.

3.2 Decomposizione del sistema:



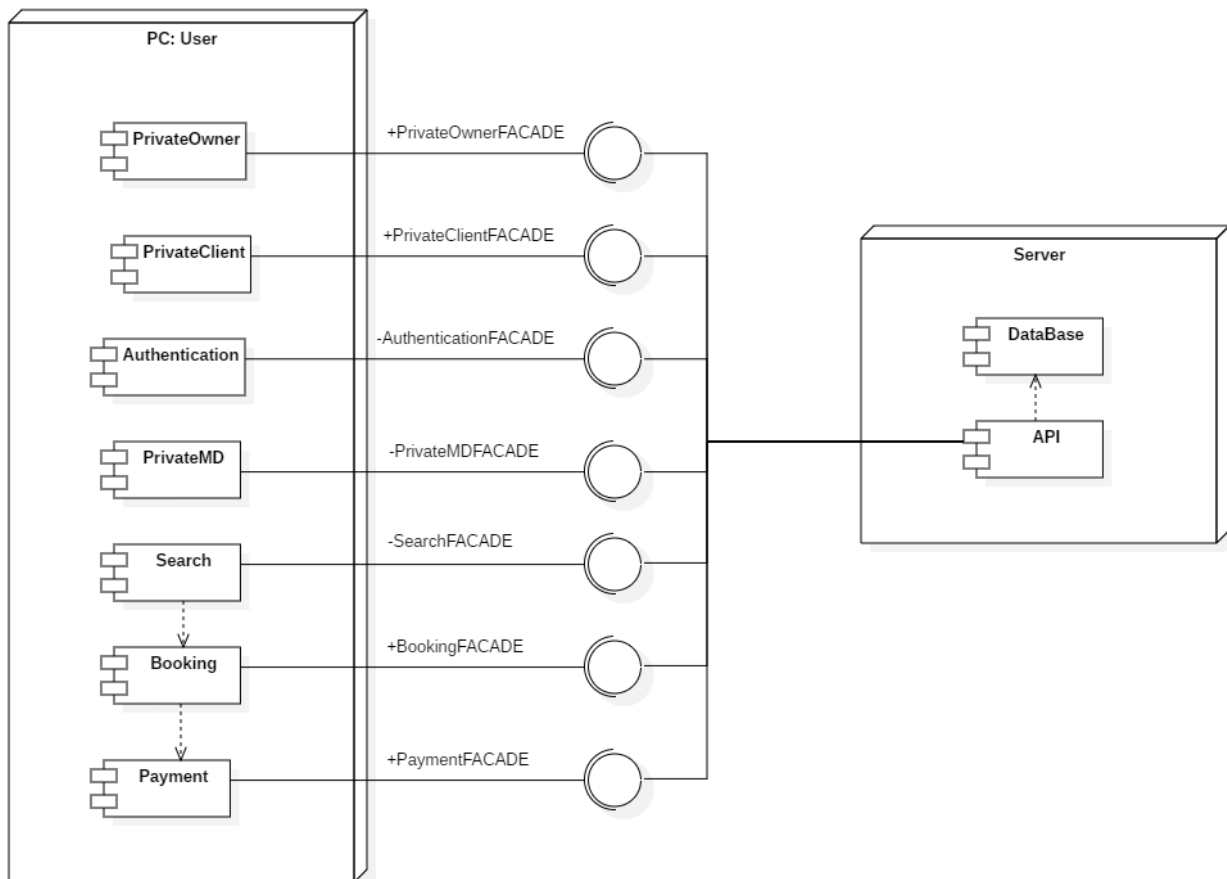
Di seguito le descrizioni di ogni sottosistema:

- Authentication: Sottosistema in cui all'interno sono gestite la fase di registrazione e login degli utenti verso il sistema.
- Booking: Sottosistema in cui all'interno sono gestite le fasi riguardante la registrazione e modifica delle prenotazioni dei B&B.
- Payment: Sottosistema in cui all'interno sono gestite le fasi riguardante i pagamenti delle prenotazioni.
- Search: Sottosistema in cui all'interno sono gestite le varie fasi per quanto riguarda la ricerca dei B&B.
- PrivateClient: Sottosistema in cui all'interno sono gestite le fasi che riguardano l'area privata del cliente.
- PrivateOwner: Sottosistema in cui all'interno sono gestite le fasi che riguardano l'area privata del proprietario, come ad esempio la gestione del proprio (o dei propri) B&B.

-
- PrivateMD: Sottosistema in cui all'interno sono gestite le varie fasi che riguardano la gestione dei documenti.
 - API: Sottosistema facente da ponte tra i client ed il server. Gestisce tutte le operazioni che il client può effettuare tramite apposita interfaccia verso il server.
 - Database: Sottosistema riguardante aggiunta, modifica e cancellazione dei dati all'interno del sistema.

3.3 Hardware/Software mapping:

Il sistema è pensato per poter gestire e garantire massima disponibilità verso gli utenti e per questo è utilizzabile da qualunque utente che abbia un account e il software installato su qualunque PC.



3.4 Gestione dei dati persistenti:

Il sistema sviluppato, essendo una piattaforma per prenotazioni di B&B, necessita di una gestione dati persistenti.

Si è scelto di utilizzare per tale scopo un Database relazionale, più precisamente un Database SQL Oracle, per memorizzare tutti i dati persistenti, quali sono: le informazioni riguardo gli utenti, le strutture registrate e le prenotazioni. Si è utilizzato come modello di database relazionale anche per creare le relazioni tra i vari dati persistenti, come le strutture e i loro servizi, le strutture e i loro proprietari, i clienti e le loro prenotazioni ed infine le prenotazioni ed ogni persona salvata.

3.5 Controllo accessi e sicurezza:

| Oggetti/Attori | Cliente | Proprietario |
|----------------|--|--|
| Authentication | Login () RegisterUser () | Login () RegisterUser () |
| Search | ResearchStructure () SeeStructurePage () Booking () | |
| Booking | SaveBooking () SavePersonFromBooking () | |
| Payment | Payment () | |
| PrivateClient | ModifyBooking () ModifyUser () DeleteBooking () SeeBooking () | |
| PrivateOwner | | RegisterStructure () RegisterClientLive () ModifyStructure () SeeStructure () SeeBookedStructure () SeeEconomicIncome () SeeEconomicOutcome () |
| PrivateMD | | ConnectToServerQuesture () ConnectToServTourismOffice () SendDocumentToQuesture () SendDocumentToTourismOffice () |

Si vuole notare che tali operazioni debbano restituire un risultato entro un tempo limite, come indicato nel RAD.

3.6 Decisioni sul flusso di controllo globale:

il server, all'interno del sistema, deve gestire in contemporanea tutte le richieste che avvengono dai vari utenti del sistema. Per questo motivo si è deciso di utilizzare un flusso di controllo Thread based, così da gestire le concorrenze di tali richieste.

Tali Thread vengono creati grazie all'utilizzo del Design Pattern FACADE, che appare interfacciarsi con le richieste del client e del Server, gestisce la creazione dei thread quando servono, così da gestire più utenti in contemporanea.

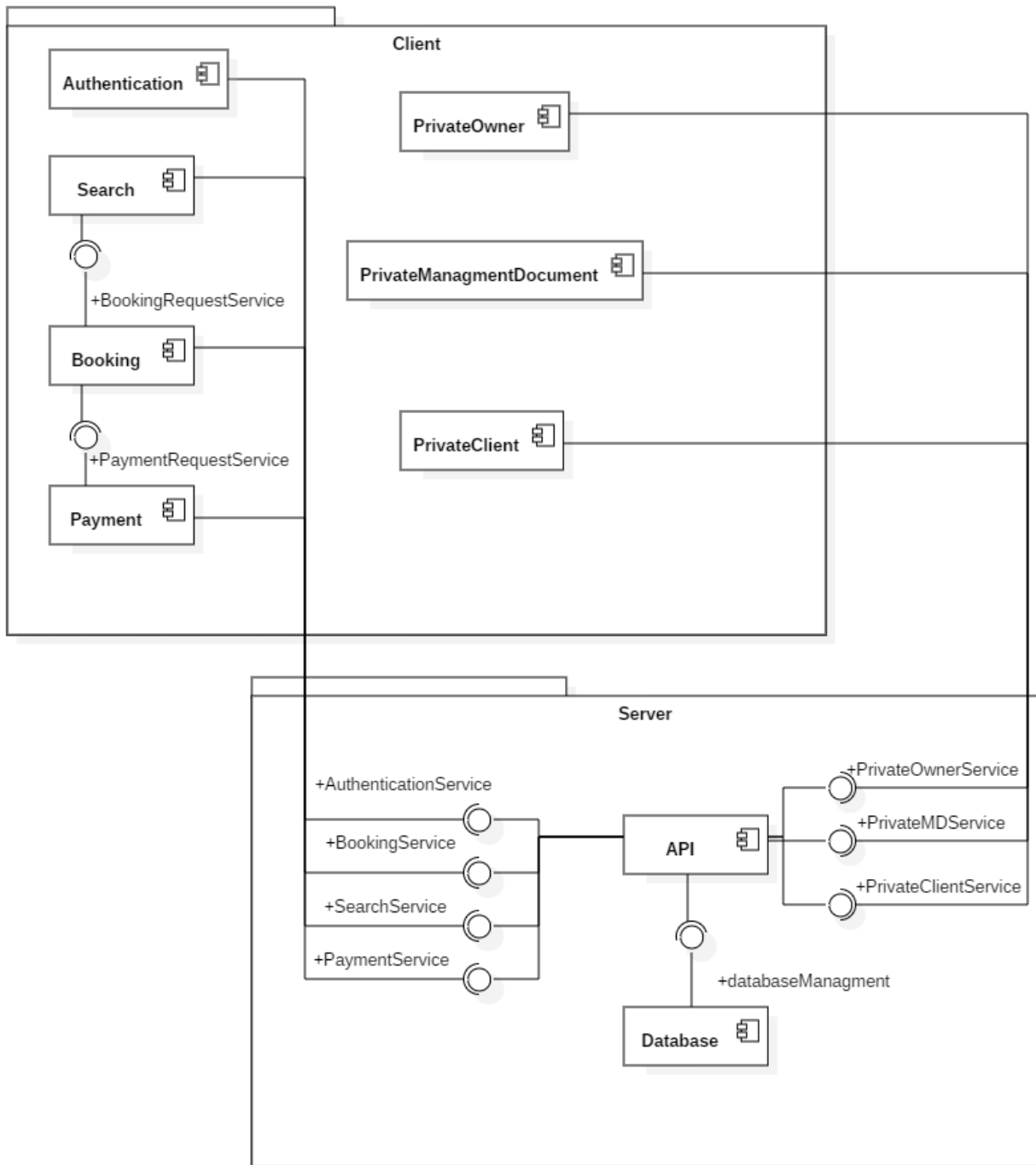
3.7 Condizioni limite:

-**Inizializzazione:** Al momento dell'inizializzazione, il sistema interroga il DB, per ricevere le relative informazioni all'utente, come i dati di Login, e le informazioni e tag usati in precedenza dall'utente in una sessione precedente. Si visualizza inoltre, la pagina iniziale che prevede la barra di ricerca dove è possibile ricercare le strutture e delle immagini su delle strutture scelte in base alle ricerche precedenti e ai tag selezionati in precedenza.

-**Terminazione:** Una volta terminata la sessione, tutte le informazioni di ricerca vengono resettate, tranne le informazioni di base dell'utente, in modo da permettere un successivo accesso, più semplice e intuitivo, conservando inoltre le ricerche effettuate nella sessione precedente.

-**Guasto:** All'accadere di un guasto o di un errore, il sistema comunica con l'utente tramite una notifica, non provocando però un reset, facilitando l'utente nella correzione di eventuali errori, e di continuare nelle sue operazioni.

4. Servizi del sottosistema



5. Glossario

API: Entità che permette ai service di interfacciarsi con il database fornendo tutte le funzioni atte a modificare l'interazione con il Database. Si è scelto di utilizzare il pattern Singleton poiché incapsulando la sua unica istanza, può mantenere un controllo stretto sulle modalità con le quali i client vi possono accedere.

AbstractFactory: Design Pattern per la creazione di oggetti in Java.

Facade: Un Design Pattern che permette attraverso un'interfaccia più semplice, l'accesso a sottosistemi che espongono interfacce più complesse.

Authentication: Form necessario per quanto riguarda l'autenticazione del proprietario/cliente.

Booking: Form necessario per quanto riguarda la prenotazione al B&B.

FacadeMain: Interfaccia principale tra i sottosistemi, l'API e il database.

Form: Interfaccia implementata da oggetti che rappresentano form.

Payment: Form necessario riguardante il pagamento nella fase di prenotazione del B&B.

PersonInfo: Form necessario per aggiornare le info personali del cliente.

Prenotation: Form necessario riguardante la prenotazione della struttura.

Private MD: Sottosistema riguardante la gestione dei documenti da parte del proprietario.

Private Owner: Sottosistema riguardante l'area privata del proprietario.

PrivateClientFacade: Applicazione del design pattern al sottosistema 'Private MD'.

RegisterOwner: Form necessario per la registrazione del proprietario.

RegisterUser: Form necessario per la registrazione del cliente.

Search: Form utile per la ricerca di strutture/abitazioni.

User: Entità generalizzata che rappresenta il client utente con i suoi attributi.