# Documentazione progetto Ingegneria del software

Sky Hotel management

Pawel Stalowczyk [1073420]

Alessandra Cortinovis [1073344]

**Software life cycle**

Il modello di processo utilizzato per la realizzazione del progetto è quello agile, più precisamente il metodo scrum a prototipazione con sviluppo incrementale.

Le attività di ingegneria dei requisiti, progettazione, design e implementazione sono alternate.

La creazione del prototipo è stata fatta ispirandosi al modello MDA, pensando prima al modello del programma che si voleva creare e successivamente implementando le varie funzionalità. Il programma finale è frutto dell’ampliamento del prototipo creato inizialmente.

**Configuration management**

Come strumento per configuration management è stato usato github per la condivisione del codice sorgente e dei file relativi al progetto, come i diagrammi e documentazione.

Le operazioni di push e pull sono state effettuate direttamente dall’IDE utilizzato, Apache NetBeans, in quanto già incorporate al suo interno.

**People management and team organization**

La squadra dedicata al progetto ha struttura semplice, è agile e autorganizzata, basata su scrum ed è composta da Stalowczyk Pawel e Cortinovis Alessandra.

I ruoli di scrum master, ovvero project manager, e di programmatore principale sono svolti da Stalowczyk e designer dell’interfaccia grafica e programmatore secondario da Cortinovis, l'unico altro membro dello scrum team.

**Software quality**

Le qualità principali che abbiamo voluto rispettare nel progetto sono state la funzionalità e la semplicità di utilizzo del programma.

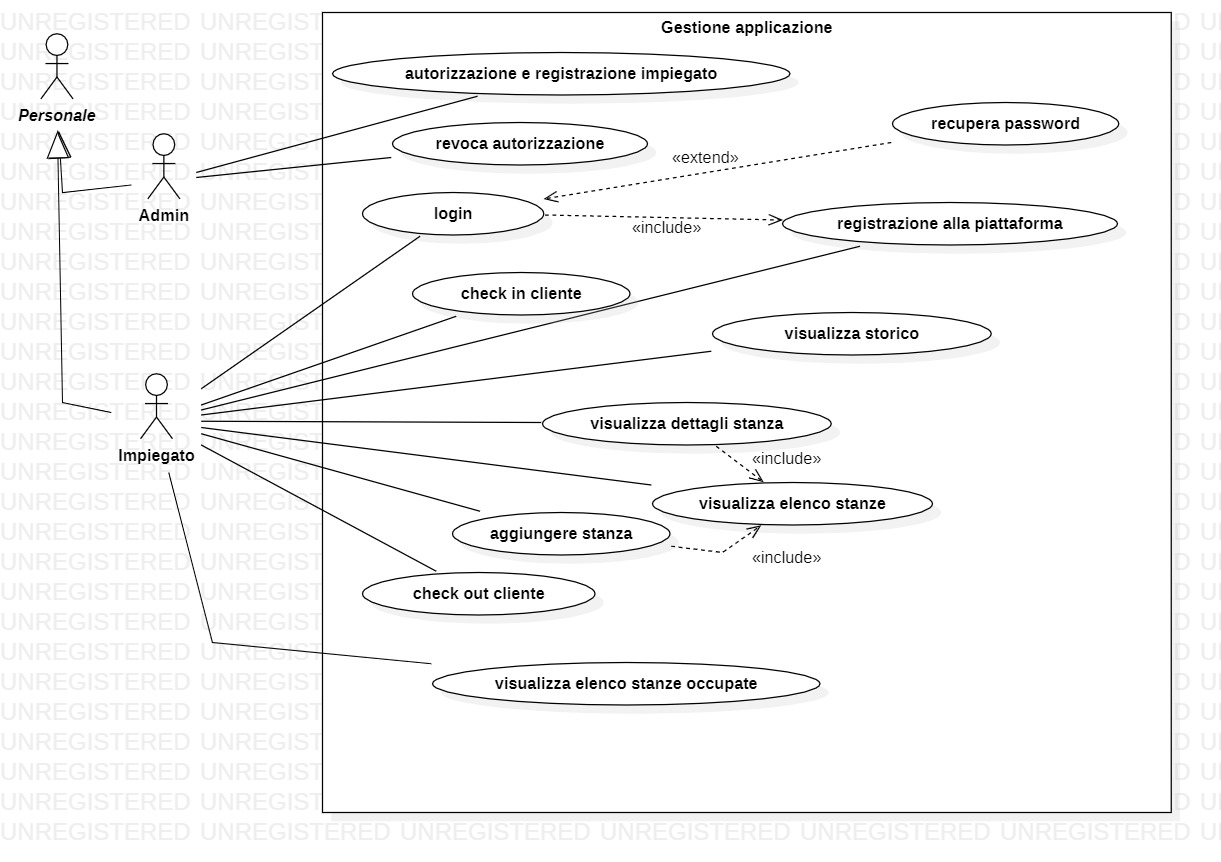
**Requirement engeneering**

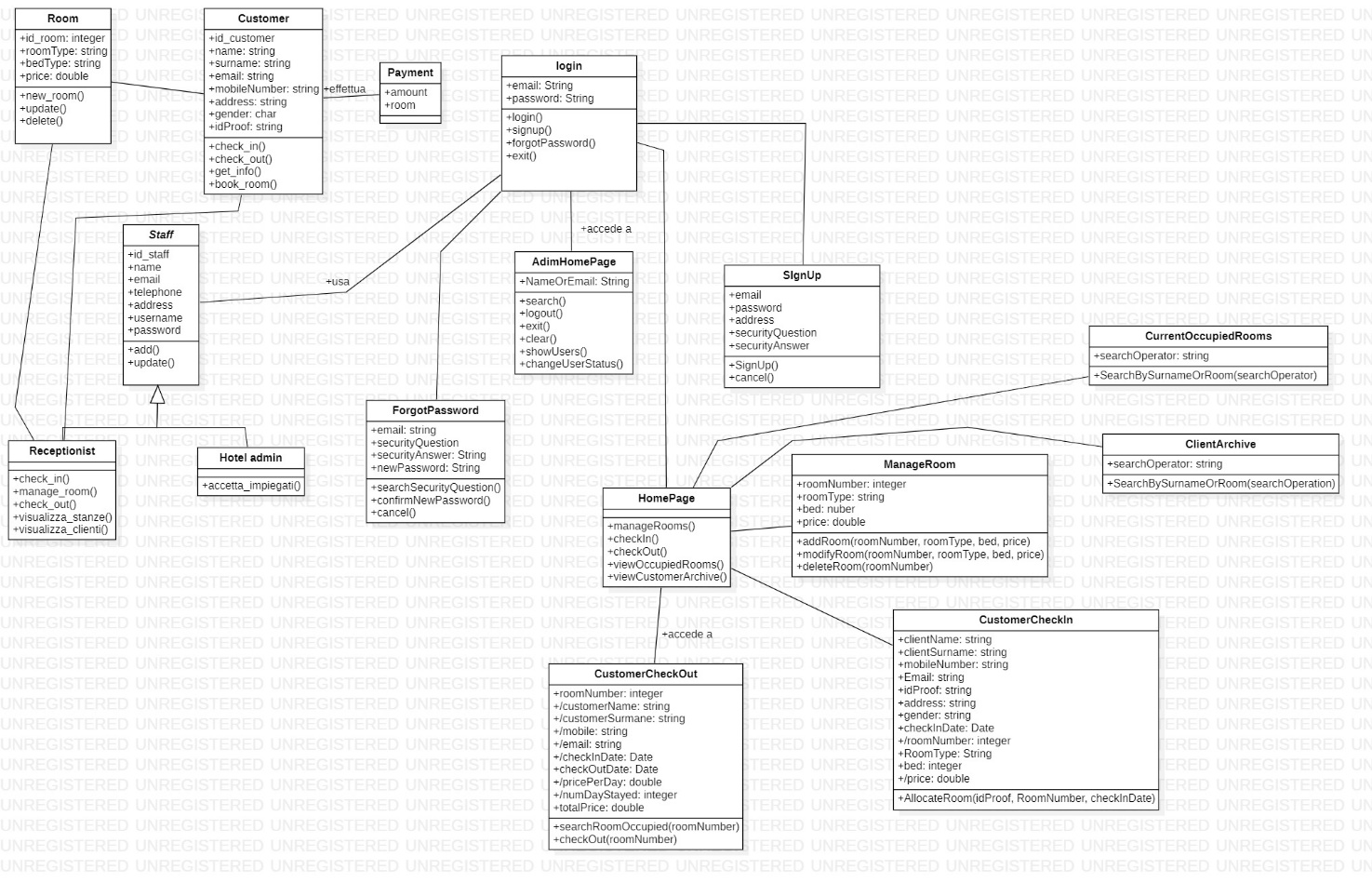
Gli standard seguiti durante le fasi del progetto sono quelli Java, in quanto il programma è scritto principalmente nel linguaggio java, e seguendo gli standard SQL per la parte di creazione delle tabelle del database.

**Modeling**

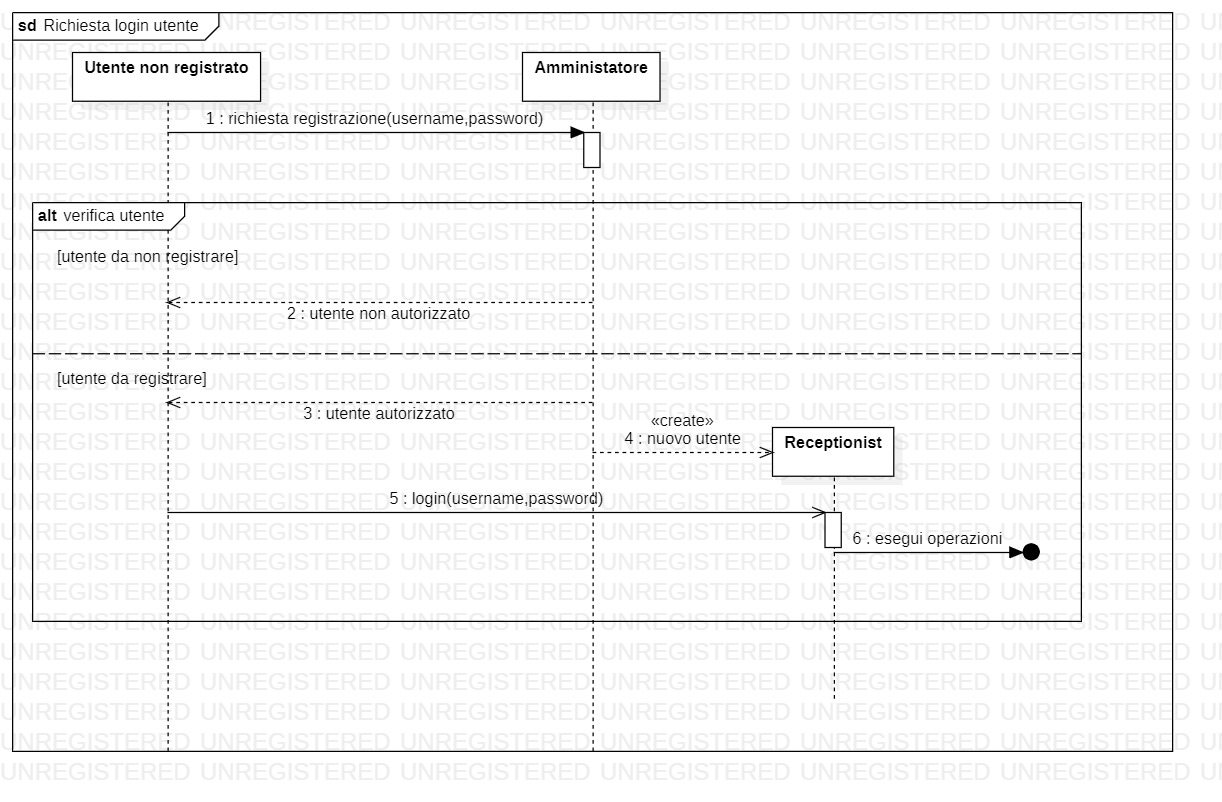
*Diagrammi realizzati:*

* Use case diagram con tutte le operazioni previste per la gestione;
* Class diagram con le classi utilizzate, specialmente quelle della UI;
* State machine con rappresentazione dello stato della camera (libera, occupata, in modifica, …);
* Sequence diagram che descrive i passi svolti dalle varie figure durante la richiesta di autorizzazione/registrazione del nuovo utente all’amministratore;
* Activity diagram che contiene tutte le operazioni e attività svolgibili dalle varie figure nel programma.

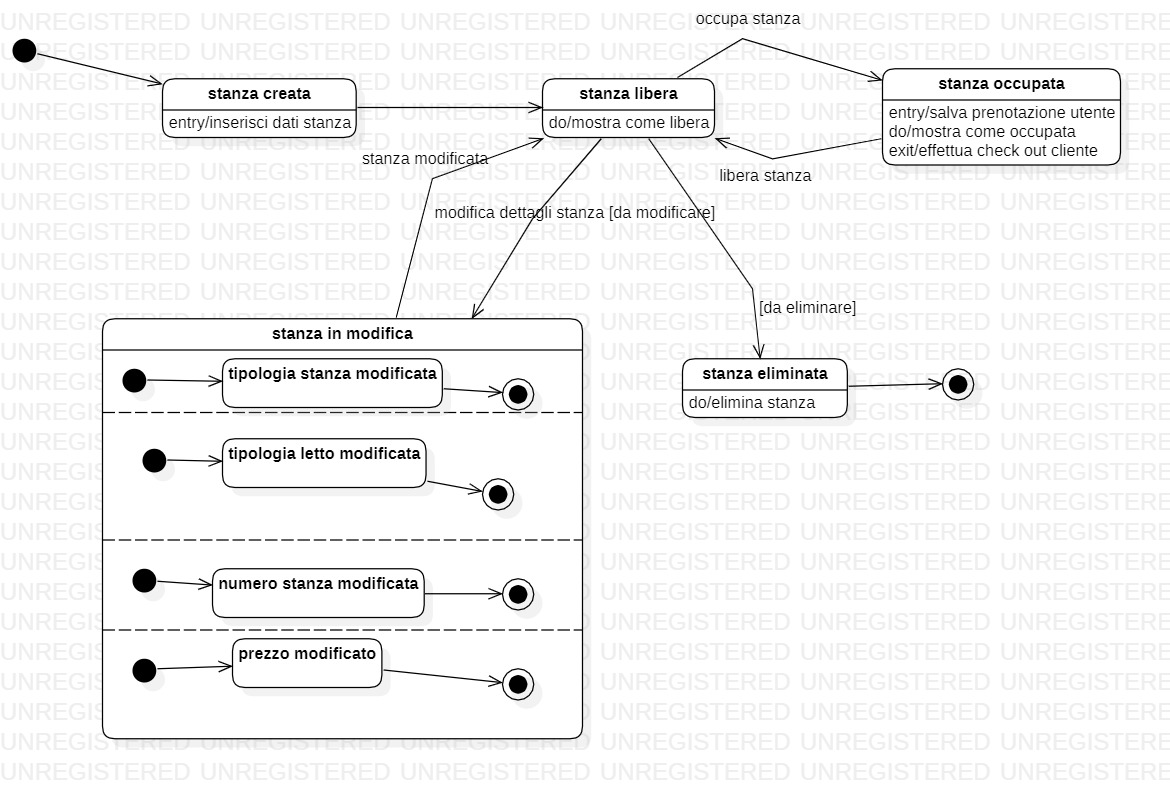
**Use case diagram**

**Class Diagram**

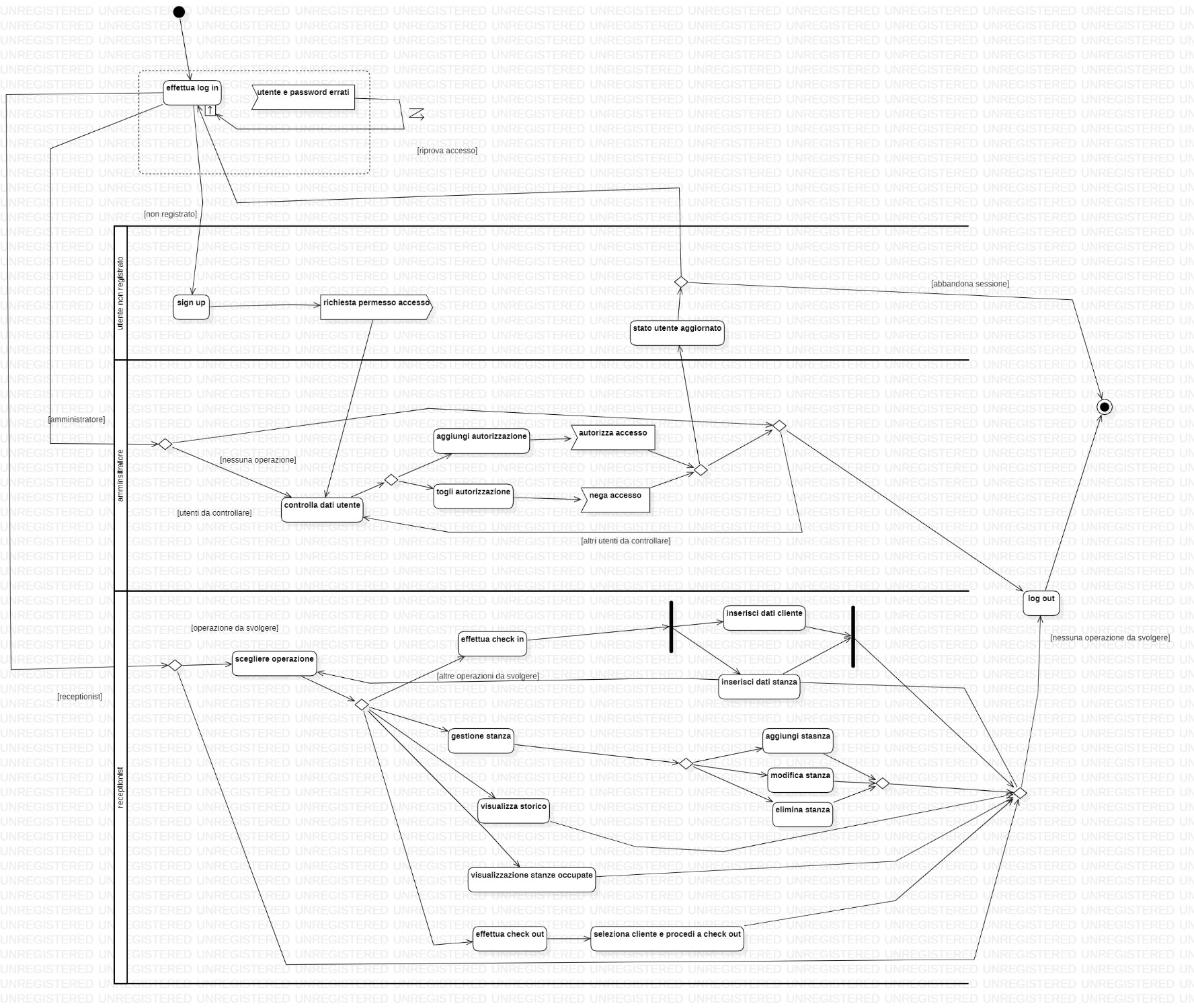
**Sequence diagram**

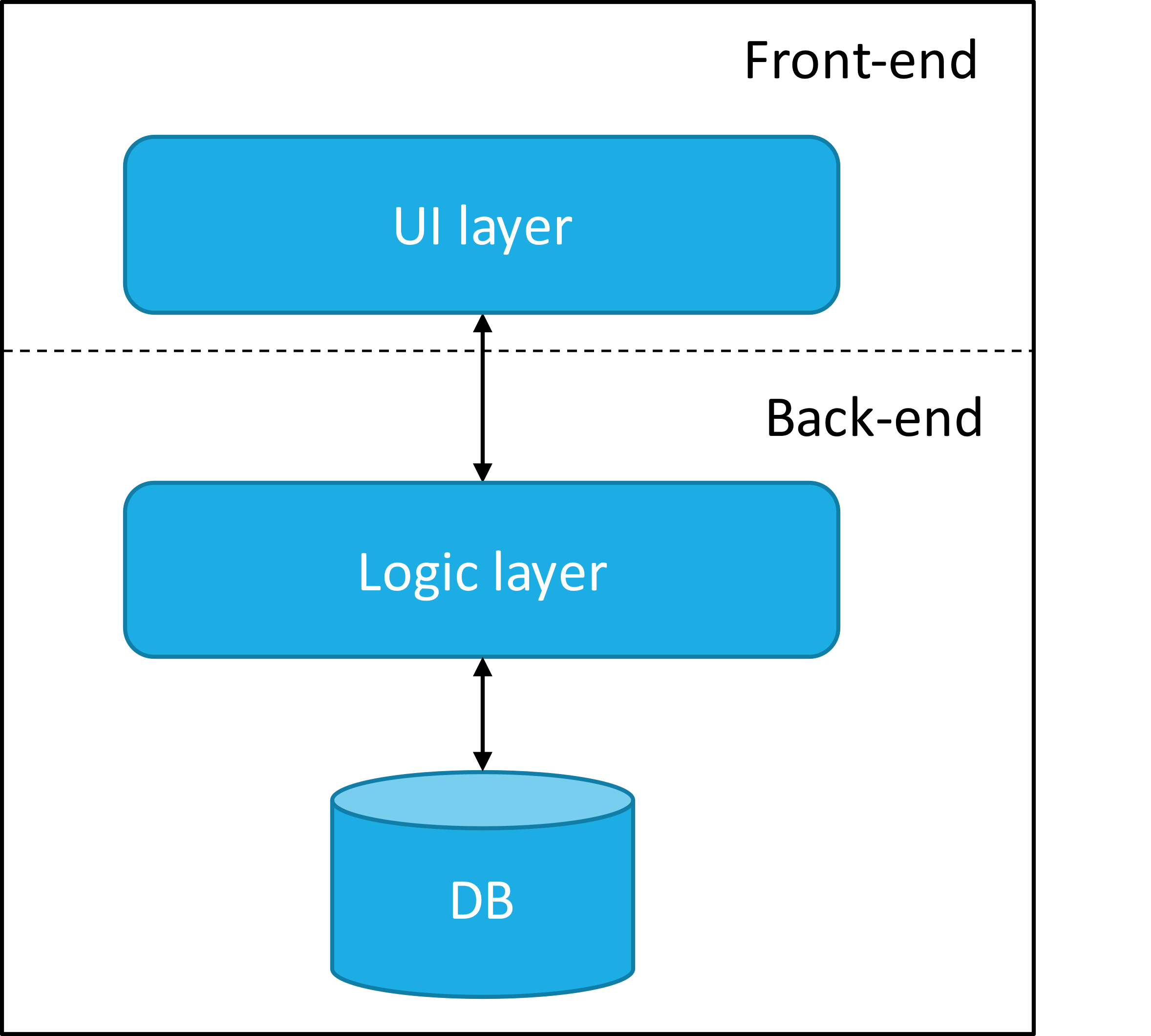
Rappresenta le operazioni effettuate quando un nuovo utente si registra al programma.

**State machine diagram**

Stato della stanza, gestita dal receprionist.

**Activity diagram**

Con tutte le operazioni effettuabili dopo il login.

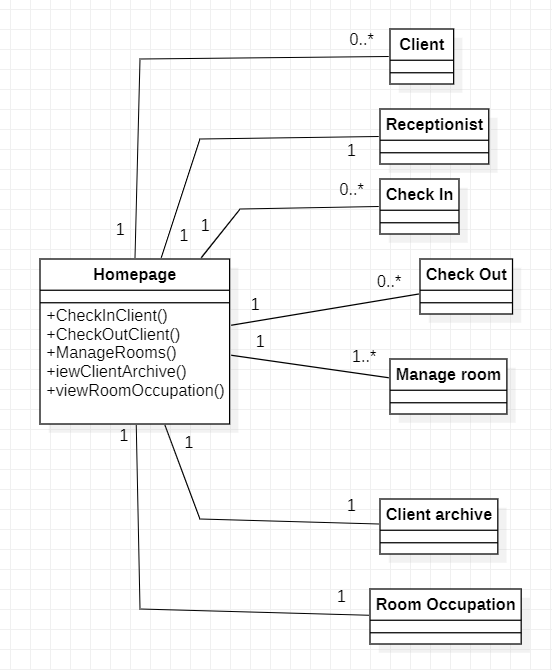
**Software architecture**

L’architettura del programma è una architettura a strati ed è divisa in due aree principali:

* area di backend, dove sono contenuti
* il DataBase, con le tabelle utente (users), stanze(room), cliente(clients);
* la parte logica con le classi e i metodi relativi alla gestione del programma;
* area di frontend, che contiene
  + la parte UI del programma.

**Software design**

Lo stile architetturale utilizzato per il progetto è il Façade pattern, che si adatta meglio alla struttura del programma.

Home page è infatti la facciata attraverso la quale il receptionist accede a tutte le operazioni eseguibili.

**Software testing**

I test sul programma sono stati fatti manualmente, siccome non è stato semplice testare la GUI (una delle probabili soluzioni sarebbe stato usare mvc disign pattern).

I test sono stati effettuati sulla compilazione dei vari form, con il controllo che

* + appaiano errori se i campi sono stati compilati non correttamente:
    - Nella schermata di Login: testato inserimento non corretto della email e della password;
  + non permette di proseguire se non sono stati compilati tutti i campi:
    - Nel form di Check In Cliente è stato testato che non faccia allocare la stanza al cliente fino a quando non sono stati inseriti tutti i campi;
  + appaiano tutti i campi richiesti nelle tabelle di visualizzazione.

**Software maintenance**

Le attività di manutenzione principali sono state soprattutto di natura correttiva, per gli errori individuati nella fase finale della programmazione e per migliorare il codice.

Una delle attività di refractoring effettuate è stato il miglioramento dei controlli di corretta compilazione dei campi nel form di check in, passando da un controllo parziale dei campi (solo quelli strettamente necessari) a uno totale, rendendo più sicuro l’inserimento e evitando un salvataggio e archiviazione parziale dei dati.