**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**

Sistema di configuration management…

Come strumento per configuration management è stato usato github per la condivisione del codice sorgente e dei file relativi al progetto. Le operazioni di push e pull sono state effettuate direttamente dall’IDE utilizzato, Apache NetBeans.

**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**

Potrebbe contenere Lista qualità che hanno guidato il progetto

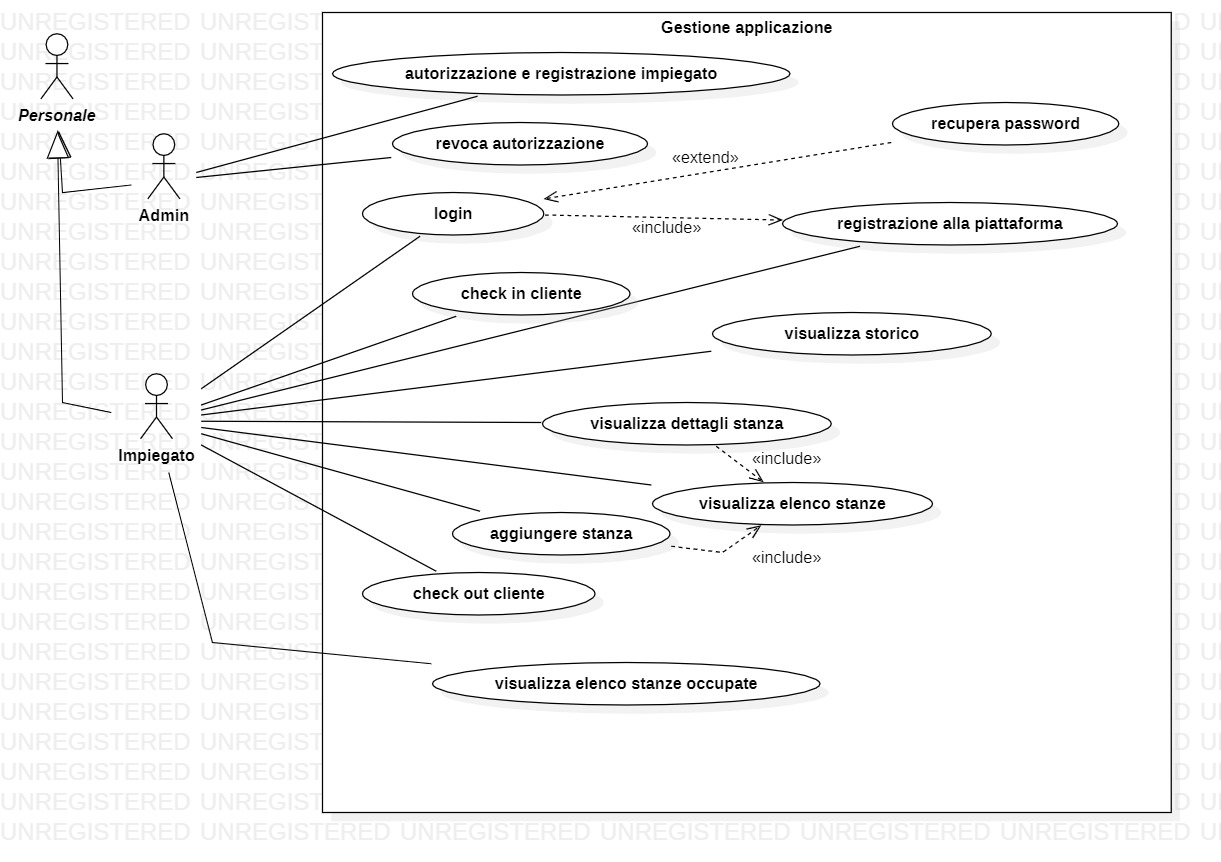
**Requirement engeneering**

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.

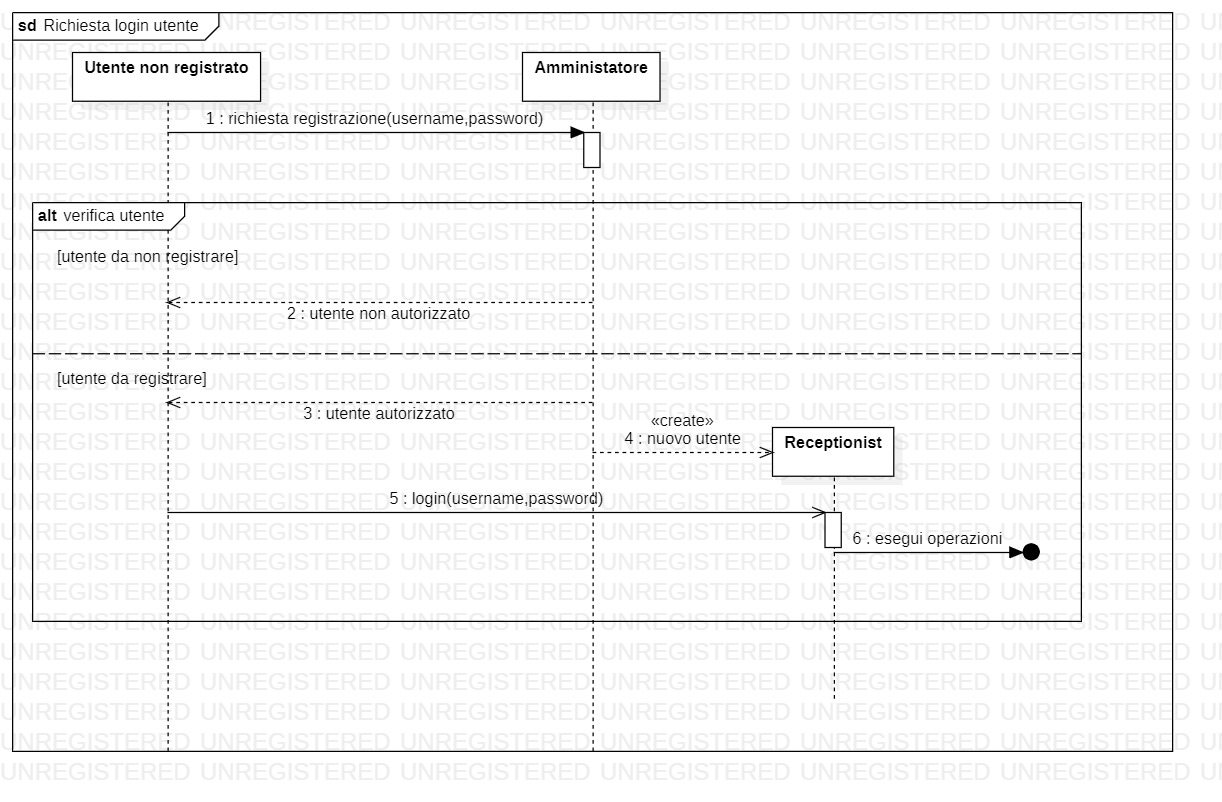
**Modeling**

**Diagrammi realizzati:**

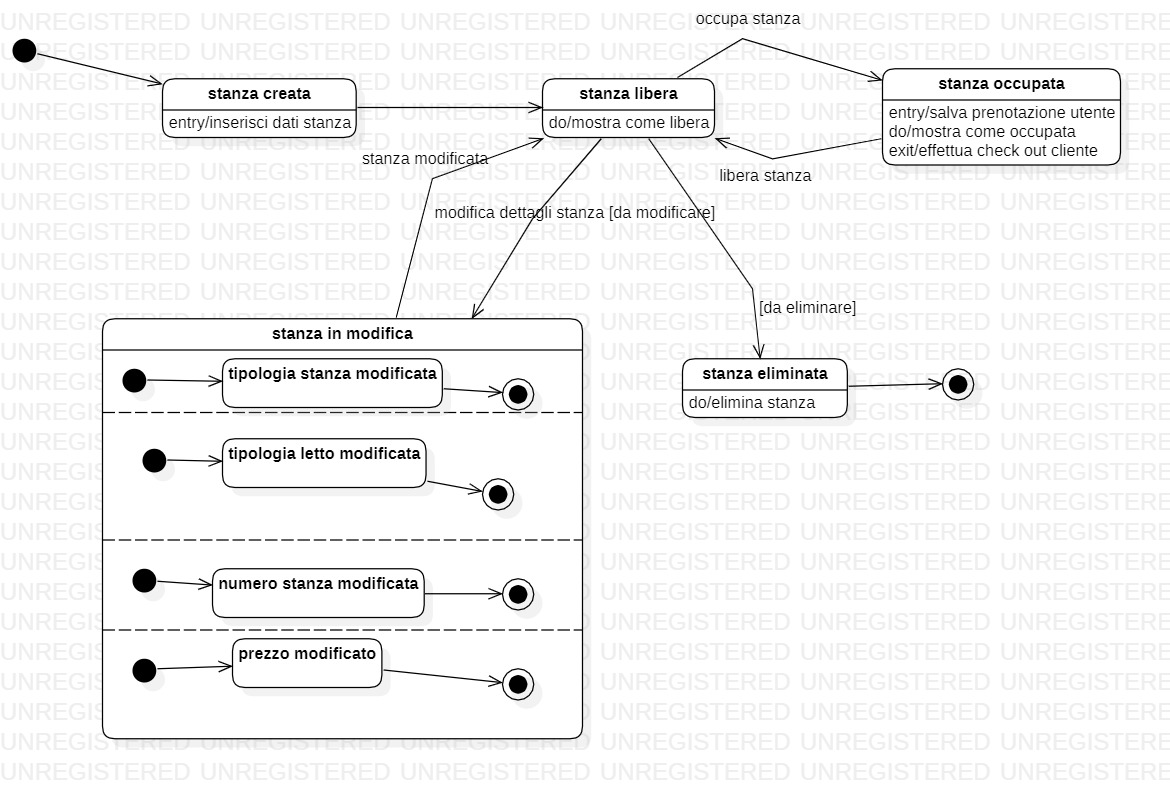
* Use case diagram con tutte le operazioni previste per la gestione;
* Class diagram
* State machine con rappresentazione dello stato della camera (libera, occupata, in modifica, …);
* Sequence diagram che descrive i passi svolti dalle varie figure durante la richiesta di autorizzazione/registrazione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

**Use case diagram**

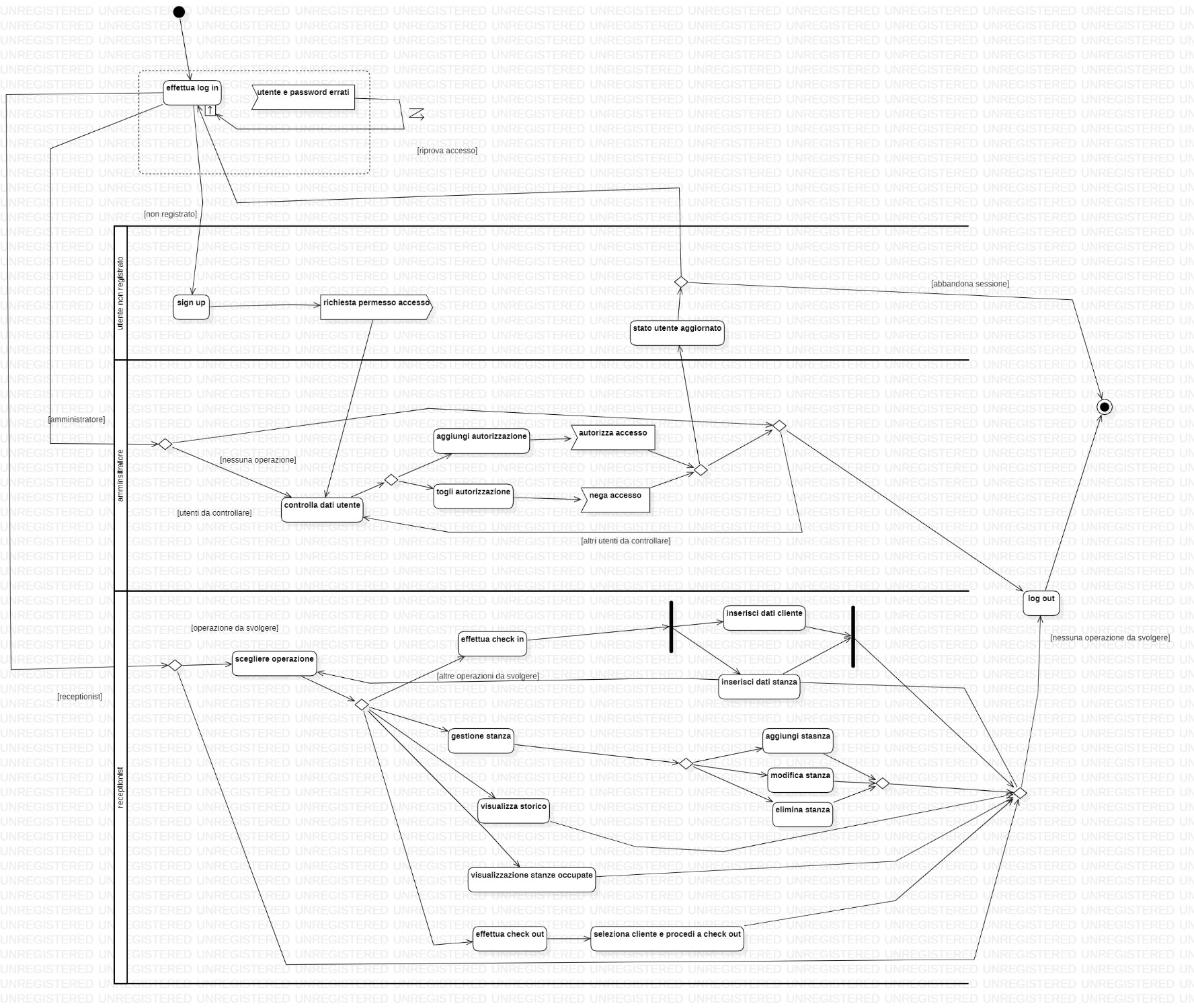
**Sequence diagram**

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**

Il programma è diviso in due aree: area di DB, dove sono contenuti le classi e i metodi relativi alla creazione e gestione delle tabelle in SQL; area grafica, che contengono le classi per la gestione dell’interfaccia grafica e esecuzione delle operazioni.

* Architectural views…

**Software design**

Desgin…

Façade …

E misurazione del codice…

**Software testing**

Test…

**Software maintenance**

Refractoring…