

## Iterazione I

### Introduzione

Durante questa prima iterazione i requisiti scelti su cui concentrarsi sono i seguenti:

-Implementare lo scenario principale di successo del caso d'uso *UC1: Effettua Prenotazione*.

- Implementare il caso d'uso *UC2: Validazione Carta Credito*.

-Implementare un caso d'uso di Start-Up necessario per gestire le esigenze di inizializzazione per questa iterazione.

Nome del caso d'uso	UC1: Effettua Prenotazione
Portata	Applicazione <i>JourneyHouse</i>
Livello	Obiettivo utente
Attore primario	Cliente
Parti interessate e interessi	<b>Cliente:</b> desidera poter prenotare la camera da lui scelta per il periodo selezionato, in modo autonomo e semplice. <b>Gestore Struttura:</b> auspica che i clienti non abbiano difficoltà nella prenotazione e possano usufruire in modo agevole del servizio di pagamento. Desidera ricevere in modo sicuro la somma di denaro e che la disponibilità delle camere venga aggiornata correttamente.
Precondizioni	Il cliente ha effettuato l'accesso.
Garanzia di successo	Il cliente sceglie la modalità di pagamento e conferma la prenotazione.
Scenario di principale di successo	1. Il cliente inserisce luogo e date del soggiorno desiderato. 2. Il sistema mostra le camere disponibili seguendo le preferenze del cliente. 3. Il cliente sceglie una fra le camere disponibili. 4. Il cliente seleziona la modalità di pagamento con carta. 5. Il sistema procede alla validazione della Carta di Credito [vedi caso d'uso Validazione Carta Credito] 6. Il sistema aggiorna la disponibilità delle camere di quella determinata struttura per il periodo selezionato e mostra l'id della prenotazione insieme ad un messaggio di successo. 7. Il sistema genera un file PDF riguardante le informazioni sulla sua prenotazione.
Estensioni	2a. Il cliente seleziona la modalità di pagamento in contanti. <ol style="list-style-type: none"><li>Il sistema aggiunge allora anche un costo di interesse.</li><li>Si procede al passo 6.</li></ol> 2a. Il pagamento non va a buon fine. <ol style="list-style-type: none"><li>Il sistema mostra un messaggio di errore e chiede all'utente di ricominciare con l'azione di effettuazione della prenotazione.</li></ol>
Requisiti speciali	Non specificato.
Elenco delle variabili tecnologiche e dei dati	Il sistema accetta qualunque tipo di carta di pagamento.
Frequenze di ripetizioni	Probabilmente alta, dipende dal numero di clienti che utilizzano <i>JourneyHouse</i> .
Varie	Non specificato.

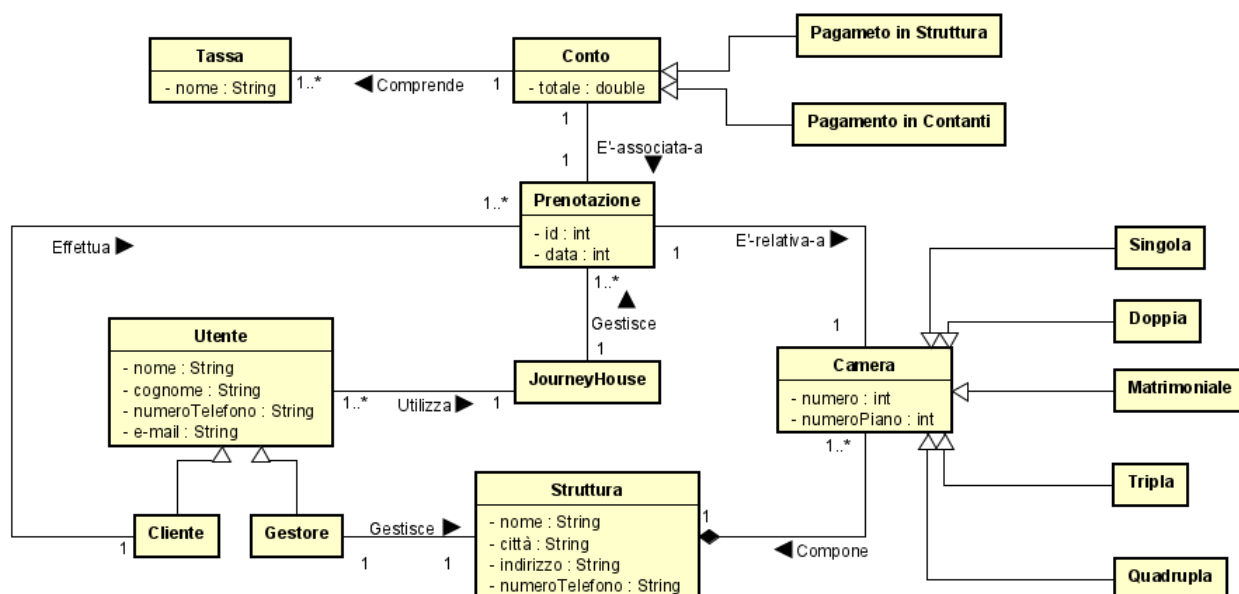
Nome del caso d'uso	UC2: Validazione Carta Credito
Portata	Applicazione <i>JourneyHouse</i>
Livello	Obiettivo utente

<b>Attore primario</b>	Cliente
<b>Parti interessate e interessi</b>	<p><b>Cliente:</b> desidera poter prenotare il suo soggiorno mediante la sua carta di credito.</p> <p><b>Gestore Struttura:</b> desidera poter essere certo che la carta di credito inserita dall'utente sia valida.</p>
<b>Precondizioni</b>	Il cliente ha effettuato l'accesso al sistema, scelto la struttura di suo gradimento e confermato di voler pagare con Carta di Credito.
<b>Garanzia di successo</b>	Il sistema verifica correttamente se la carta di credito è valida o meno.
<b>Scenario di principale di successo</b>	<p>1. Il cliente sceglie la modalità "Effettua Prenotazione".</p> <p>2. Il cliente sceglie la modalità di pagamento con carta di Credito.</p> <p>3. Il sistema verifica che la carta di credito inserita sia valida.</p> <p>4. Viene fornito il benessere e la prenotazione può concludersi.</p>
<b>Estensioni</b>	<p>4a. La carta di credito inserita non è valida.</p> <p>1. Il sistema mostra un messaggio d'errore.</p> <p>2. Viene chiesto al cliente di reinserire il numero della carta di credito.</p>
<b>Requisiti speciali</b>	Non specificato.
<b>Elenco delle variabili tecnologiche e dei dati</b>	Non specificato.
<b>Frequenze di ripetizioni</b>	Probabilmente alta, dipende dal numero di clienti che utilizzano <i>JourneyHouse</i> .
<b>Varie</b>	Non specificato.

## Analisi Orientata agli Oggetti

### Modello di Dominio

Il modello di dominio ottenuto alla fine di questa iterazione rappresenta l'insieme delle classi concettuali necessarie per il corretto funzionamento dello scenario. Inoltre, mette in evidenza le caratteristiche principali di tali classi, facendo sempre riferimento a concetti del mondo reale, e le relazioni presenti tra di loro.

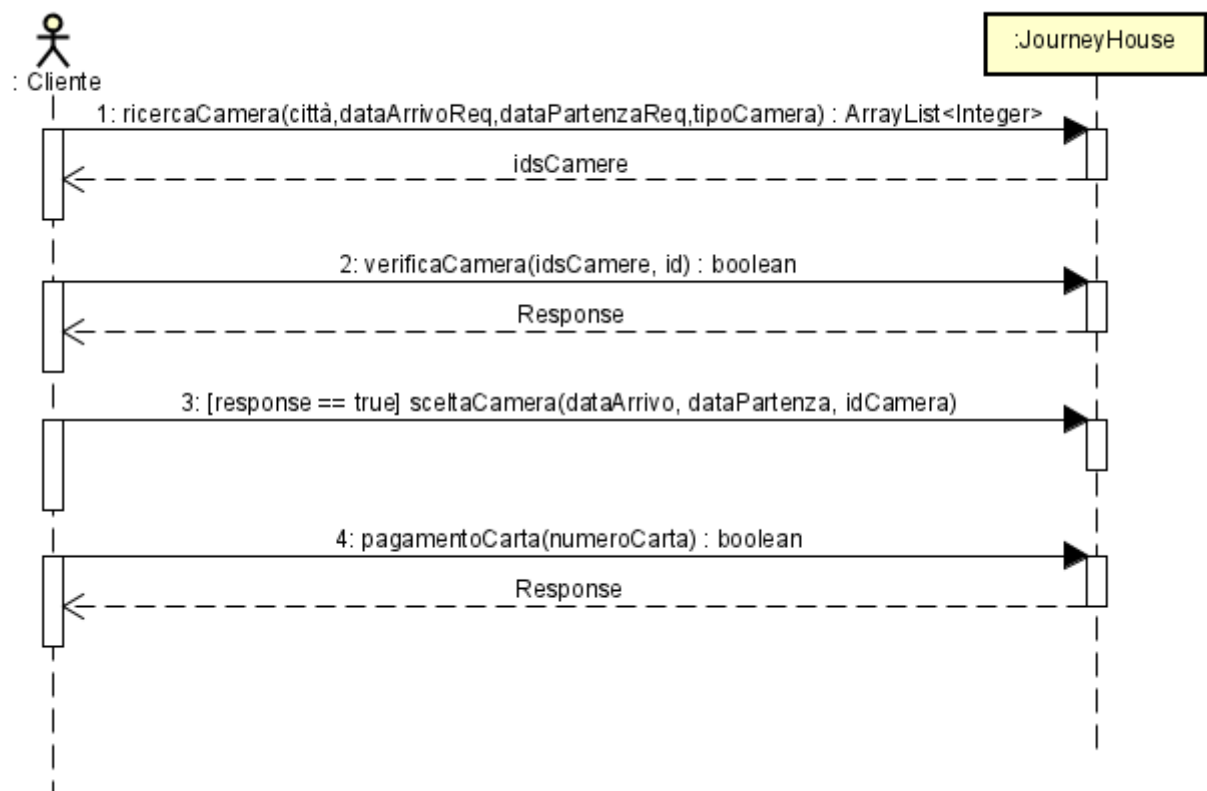


- **JourneyHouse**: sistema centrale.
- **Gestore**: gestore della struttura.
- **Cliente**: cliente del sistema che effettua la prenotazione.
- **Struttura**: è la struttura creata dal Gestore che ospiterà i vari clienti.
- **Camera**: ciò che il Cliente va a prenotare.
- **Tassa**: tipologia di tassa che viene aggiunta al totale da pagare.

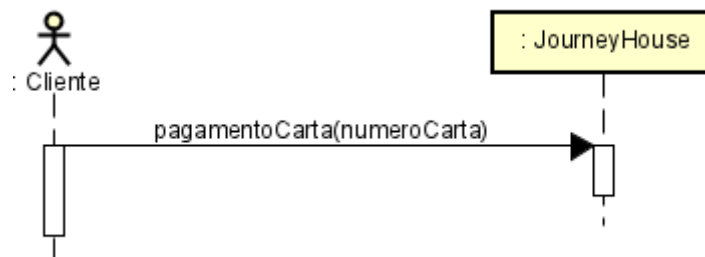
### Diagramma di sequenza di sistema

Di seguito riporto i diagrammi di sequenza di sistema (SSD) relativi ai casi d'uso in analisi. Tali diagrammi descrivono l'insieme di operazioni da effettuare, tra l'attore primario e il sistema. Esaminando il caso d'uso UC2, poiché esso è un sottostep del caso d'uso UC1, si nota che vi è una sola operazione all'interno del diagramma di sistema in quanto l'operazione di verifica è contenuta all'interno dell'operazione di pagamentoCarta.

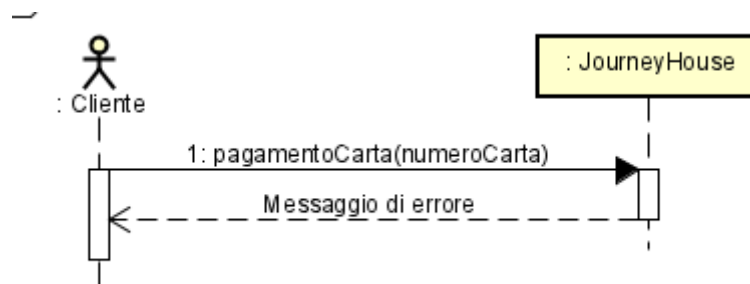
#### SSD UC1:



#### SSD UC2:



## SSD UC2 alternativo:



## Contratti delle operazioni

Di seguito riporto i contratti delle operazioni, tali contratti rappresentano in modo dettagliato quali saranno le precondizioni necessarie allo svolgimento delle operazioni dei casi d'uso e le conseguenze di tali operazioni (post-condizioni).

- **CO1: ricercaCamera (città, dataArrivo, dataPartenza, tipoStanza)**  
**Operazione:** ricercaCamera (città: String, dataArrivo: Date, dataPartenza: Date, tipoCamera: String).  
**Riferimenti:** UC1: *Effettua Prenotazione*.  
**Precondizioni:** È in corso la ricerca di una camera in base alle preferenze del cliente e la conseguente prenotazione. La data di arrivo e partenza devono essere date valide nel formato specificato. Il tipo di stanza deve essere una stringa valida corrispondente a uno dei tipi di camera disponibili.  
**Post-Condizioni:** - Sono state create tre ArrayList (idsStrutture, idsCamere, idsCamereNonDisponibili).
- **CO2: sceltaCamera (dataArrivo, dataPartenza, idCamera)**  
**Operazione:** sceltaCamera (dataArrivo: Date, dataPartenza: Date, idCamera: int).  
**Riferimenti:** UC1: *Effettua Prenotazione*.  
**Precondizioni:** È in corso la ricerca di una camera in base alle preferenze del cliente e la conseguente prenotazione.  
**Post-Condizioni:** - *clnCorso* viene aggiornato con la camera scelta dal cliente.  
- Viene avviata una Prenotazione *plnCorso*.  
- Gli attributi di *plnCorso* sono stati aggiornati.
- **CO3: pagamentoCarta(numeroCarta)**  
**Operazione:** pagamentoCarta (numeroCarta: long).  
**Riferimenti:** UC1: *Effettua Prenotazione*, UC2: *Validazione Carta Credito*.  
**Precondizioni:** È in corso una Prenotazione *plnCorso*, il cliente ha già selezionato la camera di suo gradimento.  
**Post-Condizioni:** - È stata creata una nuova istanza di Pagamento *paglnCorso*.  
- È stata creata una nuova istanza di Prenotazione *nuovaPrenotazione*.  
- Gli attributi di *nuovaPrenotazione* sono stati aggiornati con quelli di *plnCorso*.  
- *nuovaPrenotazione* è stata aggiunta all'archivio delle prenotazioni.  
- *nuovaPrenotazione* è stata aggiunta all'archivio delle prenotazioni della camera *clnCorso*.  
- È stato creato un file PDF riguardante le informazioni sulla prenotazione appena effettuata.

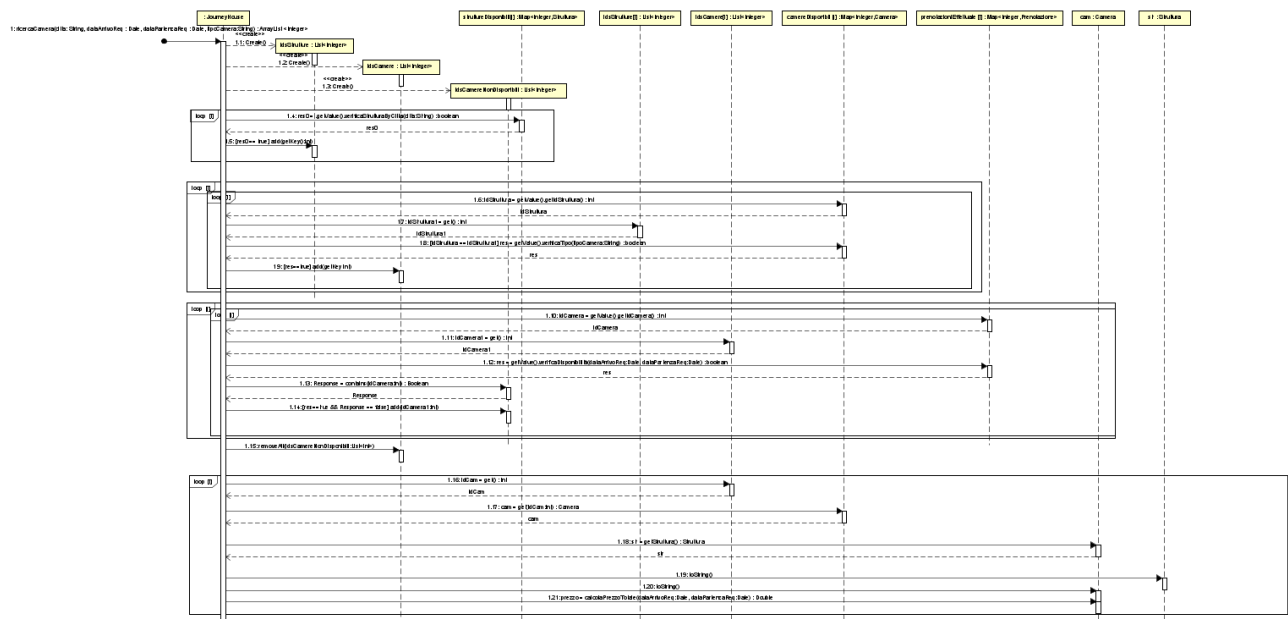
## Progettazione

### Diagrammi di Sequenza

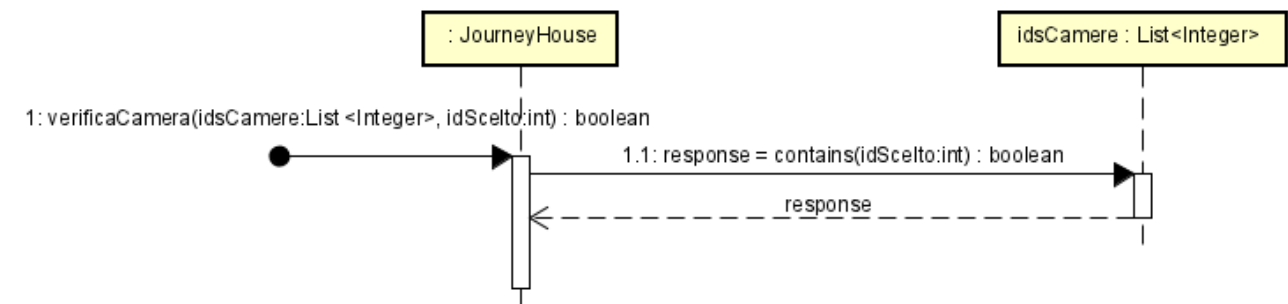
In questa sezione verranno riportati i diagrammi di sequenza, che mostrano come viene svolta una operazione e risultano essere fondamentali per l'assegnazione delle responsabilità delle classi software.

## UC1 – ricercaCamera

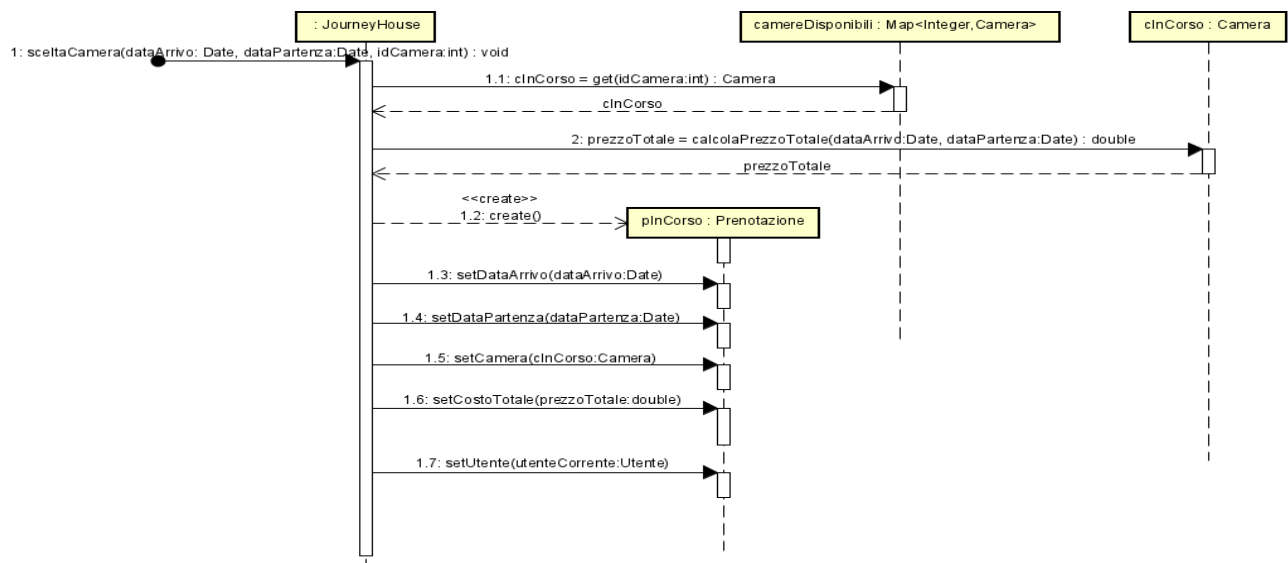
Per una migliore visione si veda l'allegato A.



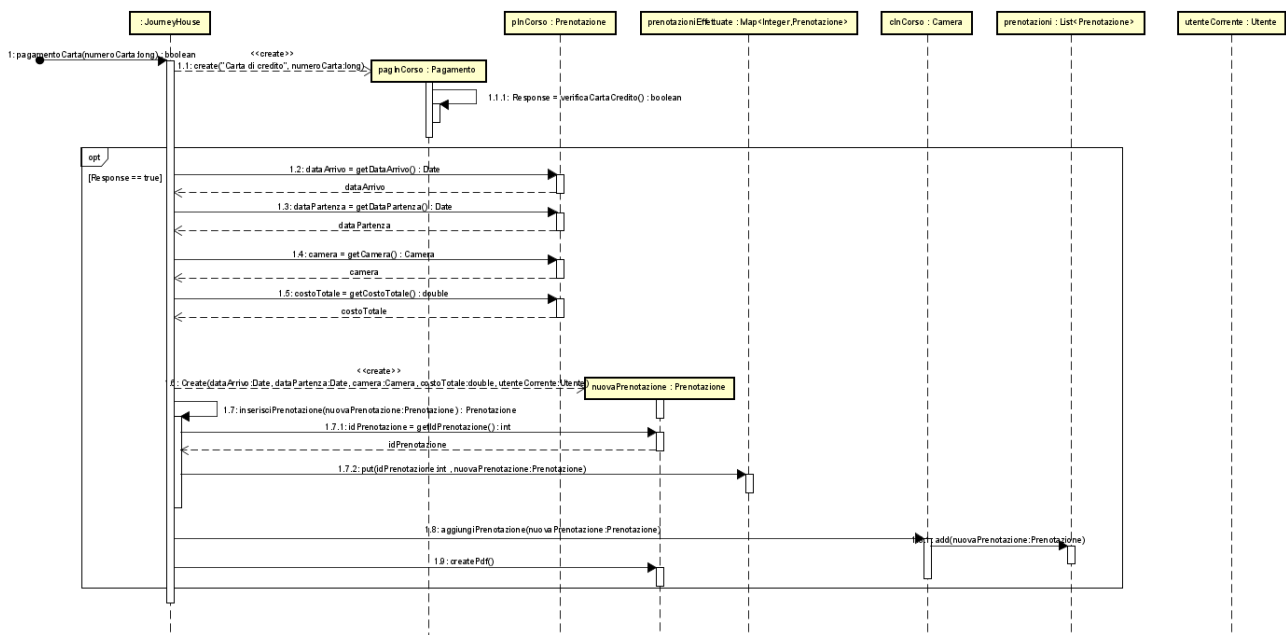
## UC1 – verificaCamera



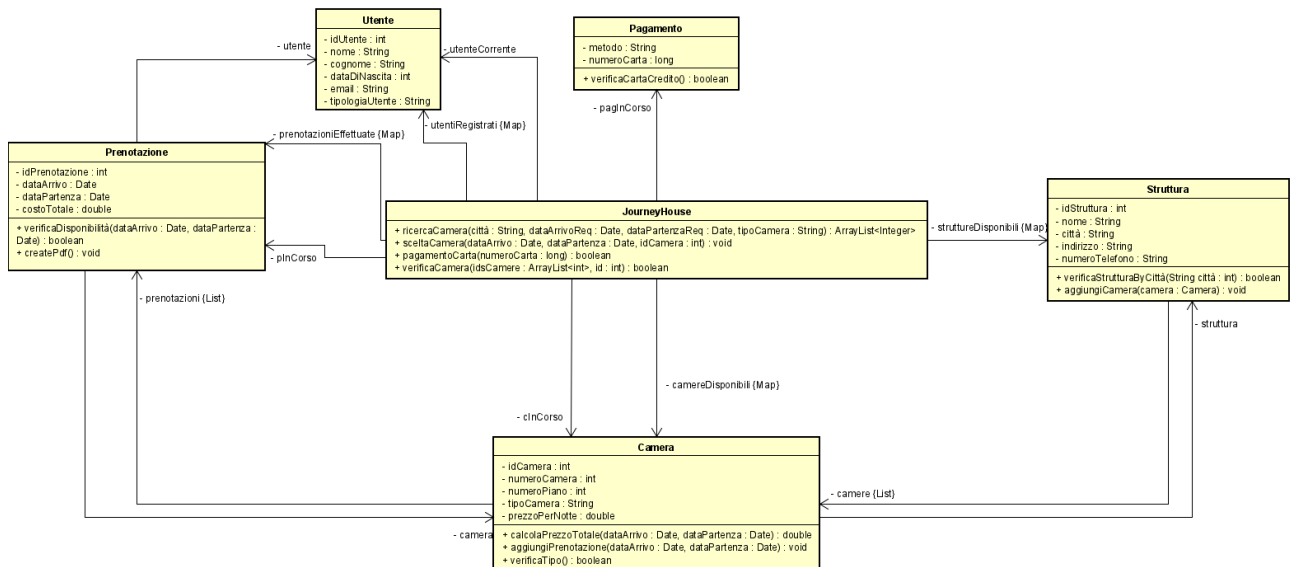
## UC1 – sceltaCamera



## UC1 – pagamentoCarta



## Diagramma delle classi



## Startup e libreria

All'interno di questa elaborazione è stato sviluppato un caso d'uso di *start-up* dove sono state create delle istanze di Utente, Struttura, Camera e prenotazione successivamente inserite all'interno delle rispettive HashMap al fine di permettere il corretto funzionamento dei casi d'uso analizzati. Inoltre, viene utilizzato una libreria customizzata con all'interno delle funzioni necessarie alla verifica del numero della carta di credito.