# Elaborazione – Iterazione 3

L’obiettivo dell’Iterazione 3 è implementare lo scenario principale di successo e tutte le estensioni individuate nel caso d’uso UC3: “Modifica/annullamento di una prenotazione” nella fase di ideazione. Un altro obiettivo è quello di implementare alcuni pattern GoF per rendere più robusta l’applicazione.

## Analisi Orientata agli Oggetti

L'analisi orientata agli oggetti si basa sulla creazione di una descrizione del dominio da un punto di vista ad oggetti. Vengono utilizzati diversi strumenti per fornire tale descrizione: Modello di Dominio, SSD, e Contratti delle operazioni.

### Modello di Dominio

Per il caso d’uso scelto UC3 sono stati identificati le seguenti classi concettuali:

* EasyRoom: rappresenta l’applicazione;
* Giocatore: utente che vuole utilizzare il Sistema per prenotare una stanza, è l’attore primario di questo caso d’uso.
* Prenotazione: entità che rappresenta l’esclusiva sull’utilizzo di una stanza in un intervallo di tempo ben definito.
* Stanza: contiene le informazioni relative ad una escape room.
* Magazzino: contiene le informazioni e tiene traccia delle richieste effettuate.
* Attrezzatura: contiene la quantità di utilities richieste.

È stato ricavato il seguente Modello di Dominio:

Immagine che contiene testo, diagramma, schermata, Piano

Descrizione generata automaticamente

### Diagramma di sequenza di sistema

Diagramma di Sequenza di Sistema (SSD) per lo scenario del caso d’uso UC3:

Immagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

In questo diagramma, il giocatore(Utente) chiede al sistema di modificare o annullare una prenotazione. Il sistema gestisce queste due operazioni in modo separato tramite i metodi modificaPrenotazione() e annullaPrenotazione(), che vengono invocati in base alla scelta dell’utente.

Il metodo modificaPrenotazione() si occupa di chiedere all’utente quale prenotazione debba essere modificata, il sistema verifica la presenza della prenotazione e, in caso di successo, chiede all’utente di reinserire i valori tipici di una prenotazione modificando la prenotazione corrente senza crearne una nuova. Sulla prenotazione modificata verranno eseguiti i controlli tipici della prenotazione, in caso di inserimento di una prenotazione non valida, il sistema ritornerà al menù e non modificherà la prenotazione scelta.

Il metodo annullaPrenotazione() si occupa di chiedere all’utente quale prenotazione debba essere annullata, il sistema verifica la presenza della prenotazione e, in caso di successo, elimina la prenotazione dal sistema. Il sistema, in base al periodo temporale in cui viene eliminata la prenotazione, calcola un valore di rimborso che può essere pari al valore totale o ad una certa percentuale a causa di una penale dovuta all’eliminazione con poco preavviso.

### 1.3.3 Contratto delle operazioni

Di seguito viene indicato il Contratto delle operazioni per l’UC2:

|  |  |
| --- | --- |
| **Operazione:** | modificaPrenotazione() |
| **Riferimenti:** | Caso d’uso UC3: Modifica/annullamento di una prenotazione. |
| **Pre-condizioni:** | - |
| **Post-condizioni:** | 1. È stata modificata l’istanza prenotazioneDaModificare di Prenotazione. 2. Gli attributi idPrenotazione, stanzaPrenotata, primoGiocatore, secondoGiocatore, dataPartita, tempoInizio, tempoFine, attrezzaturaRichiesta, costo di nuovaPrenotazione sono stati aggiornati correttamente con valori validi che rispettano le regole di dominio.   prenotazioneDaModificare è stato associato a EasyRoom. Viene inserito in una lista mantenuta in memoria in maniera persistente (file di testo). |

|  |  |
| --- | --- |
| **Operazione:** | annullaPrenotazione() |
| **Riferimenti:** | Caso d’uso UC3: Modifica/annullamento di una prenotazione. |
| **Pre-condizioni:** | - |
| **Post-condizioni:** | È stata eliminata dalla memoria e dai file l’istanza prenotazioneDaModificare di Prenotazione. |

|  |  |
| --- | --- |
| **Operazione:** | checkDurata(inizioPartita, finePartita) |
| **Riferimenti:** | Caso d’uso UC3: Modifica/annullamento di una prenotazione. |
| **Pre-condizioni:** | È in corso la modifica di una nuova prenotazione.  L’utente ha fornito degli orari di inizio e fine della partita. |
| **Post-condizioni:** | È stato restituito un Messaggio di Verifica dal Sistema. Viene restituito un booleano che da informazioni sull’esito. |

|  |  |
| --- | --- |
| **Operazione:** | checkStanzaLibera(stanzaPrenotazione, dataPrenotazione, inizioPrenotazione) |
| **Riferimenti:** | Caso d’uso UC3: Modifica/annullamento di una prenotazione. |
| **Pre-condizioni:** | È in corso l’inserimento di una nuova prenotazione.  L’utente ha immesso correttamente i dati. |
| **Post-condizioni:** | È stato restituito un Messaggio di Verifica dal Sistema. Viene restituito un booleano che da informazioni sull’esito. |

|  |  |
| --- | --- |
| **Operazione:** | prospettoPrezzo(stanzaPrenotazione, attrezzaturaRichiesta, durata) |
| **Riferimenti:** | Caso d’uso UC3: Modifica/annullamento di una prenotazione. |
| **Pre-condizioni:** | È in corso la modifica di una nuova prenotazione.  I Messaggi di Verifica hanno dato esito positivo, cioè la durata della partita è nei limiti, la stanza è disponibile e si è scelta l’attrezzatura. |
| **Post-condizioni:** | È stato calcolato un prezzo dal Sistema. Viene restituito un float che tiene conto del prezzo della stanza e dell’attrezzatura scelta. |

## 1.4 Progettazione

Di seguito sono presenti i Diagrammi di Sequenza:

modificaPrenotazione() : void

Immagine che contiene testo, diagramma, Parallelo, schermata

Descrizione generata automaticamente

Immagine che contiene testo, linea, diagramma, ricevuta

Descrizione generata automaticamente

annullaPrenotazione() : float

Immagine che contiene testo, schermata, diagramma, Parallelo

Descrizione generata automaticamente

## Modello delle classi di Progetto

Immagine che contiene testo, schermata, Parallelo, diagramma

Descrizione generata automaticamente