# Elaborazione - Iterazione 1

# Refactoring

E’ stato effettuato un refactor degli Use Case. Per scelte progettuali si è deciso di modificare “UC7: Inserisci recensione” e “UC8: Aggiungi richiesta”.

L’attuale UC7 sarà sviluppato nella seconda fase del progetto, insieme alla gestione notifiche via mail, poiché ritenuti poco importanti per lo sviluppo di questa prima fase.

L’attuale UC8 verrà invece inglobato nell’ ”UC5: Iscrizione altri servizi”.

Pertanto, in seguito a queste scelte progettuali si è scelto di modificare gli use case menzionati in precedenza in UC7: Pagamento e UC8: Rimborso.

Inoltre UC3 è stato rinominato in “Gestione prenotazioni pendenti” per una maggiore chiarezza espositiva.

Di seguito una lista riepilogativa con tutti gli use case aggiornati:

|  |  |  |
| --- | --- | --- |
| **Attore** | **Obiettivo** | **Caso d’uso** |
| **Host** | Inserire di una nuova postazione con le relative informazioni | UC1: Inserisci nuova postazione |
| **Host** | Inserire, aggiornare, rimuovere altri servizi (attrezzatura o eventi) | UC2: Inserisci altri servizi |
| **Host** | Gestire le prenotazioni pendenti: accettando o rifiutando | UC3: Gestione prenotazioni pendenti |
| **Utente** | Affittare la postazione | UC4: Affitta postazione |
| **Utente** | Iscriversi ad eventi extra- lavorativi, mensa, servizio di catering | UC5: Prenota altri servizi |
| **Utente** | Eliminazione della prenotazione di un evento o un evento | UC6: Elimina Prenotazione |
| **Utente** | Pagamento in seguito a prenotazioni | UC7: Pagamento |
| Utente | Rimborso pagamento in seguito a cancellazioni | UC8: Rimborso |

# Introduzione

# Successivamente alla fase di ideazione ha inizio la fase di elaborazione. In questa prima iterazione verranno approfonditi quei casi d’uso necessari e fondamentali per l’utilizzo di CoLinking, ovvero l’inserimento di una nuova postazione e la registrazione a questa. L’host potrà creare una **workstation** con un id univoco che sarà comporto dal nome dell’host (tipicamente il nome dell’azienda) e la sede e ogni host potrà avere associata una e soltanto una workstation. L’utente invece dopo aver selezionato l’id della postazione di suo interesse, invierà la richiesta di prenotazione al sistema, la quale rimarrà in pending.

# Pertanto nella prima iterazione verranno approfonditi i seguenti use case:

* UC 1: Gestisci nuova postazione
* UC 4: Affitta postazione

# Analisi UC1: Inserisci nuova postazione

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc1: Inserisci nuova postazione** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Host |
| **Parti interessate e interessi** | Host: vuole aggiungere una nuova postazione prenotabile nella piattaforma. |
| **Pre-condizioni** |  |
| **Garanzia di successo** | Le informazioni relative alla postazione sono state inserite nel sistema |
| **Scenario principale di successo** | 1. L’host vuole aggiungere una nuova postazione nel sistema. 2. L’host inserisce descrizione e quantità delle workstations. 3. Il sistema genera un id 4. Il sistema raccoglie le informazioni dell’host 5. Il sistema crea la postazione |
| **Estensioni** | 3a: Se il workstation id esiste già, viene lanciato un messaggio di errore e si ritorna allo stato precedente |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Legata alla quantità di host registrati nel sistema |
| **Varie** |  |

# Modello di dominio UC1

In questo caso d’uso sono state individuate le seguenti classi:

* **CoLinking:** Rappresenta il sistema
* **Host:** Utente che interagisce col sistema e crea la workstation
* **Workstation:** Postazione di lavoro prenotabile dall’utente creato dall’host

Immagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

# Diagramma di sequenza UC1

Il diagramma di sequenza di sistema relativo all’UC1 è il seguente:

Immagine che contiene testo, schermata, linea, numero

Descrizione generata automaticamente

# Contratti delle operazioni UC1

|  |  |
| --- | --- |
| **Operazione** | ***createWorkstation(id, companyName, city, address, description)*** |
| **Riferimenti** | UC1: Inserisci nuova postazione |
| **Pre-condizioni** | L’host ha effettuato l’accesso e vuole creare una workstation |
| **Post-condizioni** | E’ stata creata una nuova istanza della workstation |

# Analisi UC4: Affitta postazione

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc4: Affitta postazione** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Utente |
| **Parti interessate e interessi** | Utente: vuole affittare una nuova postazione prenotabile. |
| **Pre-condizioni** | Le informazioni relative alla workstation sono state inserite |
| **Garanzia di successo** | La prenotazione viene aggiunta nel database con successo. |
| **Scenario principale di successo** | 1. L’utente vuole affittare un servizio 2. Il sistema mostra le workstation registrate nel sistema 3. Utente inserisce workstation da prenotare e data (inserisce un range se vuole prenotare più giorni) 4. Il sistema controlla se esiste la postazione 5. L’utente prenota la postazione 6. La prenotazione sarà in stato di pending fino a quando non verrà accettata dall’host |
| **Estensioni** | 4a: Se la workstation non esiste, viene lanciato un messaggio di errore e si ritorna allo stato precedente |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Molto frequente |
| **Varie** |  |

# Modello di dominio UC4

In questo caso d’uso sono state aggiunte le seguenti classi:

* **Booking:** Rappresenta la prenotazione alla postazione di lavoro
* **User:** Utente generico che vuole usufruire del servizio e vuole prenotare una postazione di lavoro

Inoltre è stata fatta una modifica relativa alla classe Workstation, ove, sono stati aggiunti gli attributi capacityByDate la quale mapperà il numero di posti disponibili in base alla data e capacity è stata modificata in maxCapacity.

Immagine che contiene testo, diagramma, Piano, schermata

Descrizione generata automaticamente

# Diagramma di sequenza UC4

Il diagramma di sequenza di sistema relativo all’UC4 è il seguente:

Immagine che contiene testo, schermata, linea, Parallelo

Descrizione generata automaticamente

# Contratto delle operazioni UC4

|  |  |
| --- | --- |
| **Operazione** | ***showWorkstationMap()*** |
| **Riferimenti** | UC4: Affitta postazione |
| **Pre-condizioni** | L’utente vuole prenotare una workstation  L’host deve aver creato almeno una workstation nel sistema |
| **Post-condizioni** | L’utente visualizza tutte le workstation presenti nel sistema |

|  |  |
| --- | --- |
| **Operazione** | ***bookWorkstation(workstation, date)*** |
| **Riferimenti** | UC4: Affitta postazione |
| **Pre-condizioni** | L’utente vuole prenotare una workstation  L’host deve aver creato almeno una workstation nel sistema |
| **Post-condizioni** | E’ stata creata una nuova istanza di Booking |

# Progettazione Iterazione 1

Per abbassare l’accoppiamento si è deciso di usare un design pattern MVC (Model-View-Controller). Inoltre usando il pattern Creator si è deciso di delegare la creazione degli oggetti ai Controller. L’ oggetto della classe di sistema CoLinking, sono stati creati seguendo il pattern creazionale Singleton

# Diagrammi di sequenza UC1

Immagine che contiene testo, linea, ricevuta, diagramma

Descrizione generata automaticamente

# Diagrammi di sequenza UC4

Immagine che contiene testo, linea, diagramma, schermata

Descrizione generata automaticamente

# Diagrammi delle classi UC1 e UC4

Immagine che contiene testo, diagramma, Rettangolo, Piano

Descrizione generata automaticamente

# Testing

Tutti i test eseguiti e che verranno eseguiti nel futuro sono di tipo black box.

# UC1

* E’ stato settato un Host di test prima di eseguire i vari test
* E’ stato verificato che la creazione della workstation avvenisse correttamente
* E’ stato verificato che nel caso in cui la workstation esiste, viene generata un eccezione

# UC4

* E’ stato settato un User di test, una workstation di test prima di eseguire i vari test
* E’ stato verificato che la prenotazione della workstation avvenisse con successo
* E’ stato verificato che nel caso di prenotazione di una Workstation non esistente viene lanciata un eccezione
* E’ stato verificato che durante la prenotazione, nel caso di superamento della capienza massima di quella workstation viene lanciata un eccezione

# Elaborazione - Iterazione 2

# Refactoring

E’ stato effettuato un refactor sul menù. La parte di UI veniva inizialmente gestita dalla classe di sistema CoLinking, ora in seguito al refactor è stata istituita la classe **Menu** e **MenuView** dentro il package **ui,** per abbassare ulteriormente l’accoppiamento. Perciò tutte le richieste di sistema, da ora in poi, partiranno dal Menu.

# Introduzione

In questa seconda iterazione verranno approfonditi quei casi d’uso necessari per completare la fase di prenotazione avviata nella precedente iterazione ovvero la gestione delle prenotazioni pendenti e l’eliminazione di una richiesta di prenotazione o di una prenotazione già effettuata. L’host potrà accettare o rifiutare la richiesta di prenotazione in pending mandata dall’utente. L’utente potrà cancellare una prenotazione effettuata, sia nel caso che sia stata già accettata dall’host che nel caso sia ancora in pending.

# Analisi UC3: Gestione prenotazioni pendenti

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc3: Gestione prenotazioni pendenti** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Host |
| **Parti interessate e interessi** | Host: vuole confermare una prenotazione pendente. |
| **Pre-condizioni** | La richiesta di prenotazione da parte dell’utente è stata inviata con successo |
| **Garanzia di successo** | La richiesta non risulti più pendente e risulti: accettata o rifiutata |
| **Scenario principale di successo** | 1. L’host vuole accettare o rifiutare una richiesta di prenotazione pendente 2. Il sistema mostra le prenotazioni pendenti 3. L’host seleziona la l’id della richieste di prenotazione che vuole gestire 4. L’host seleziona se conferma o rifiuta la prenotazione 5. La prenotazione è stata gestita e non sarà più in stato pending |
| **Estensioni** | 4a: Se l’utente conferma, il sistema controlla se ci sono posti disponibili per quella data: se ci sono i posti conferma la prenotazione altrimenti lancia un messaggio di errore  4b: Nel caso in cui non venga accettata la richiesta , la richiesta viene rimossa dallo stato di pending e si ritorna al menù precedente. |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Legata alla quantità di utenti che effettuano prenotazioni |
| **Varie** |  |

# Diagramma di sequenza UC3

Dato che il modello di dominio non ha subito modifiche, si illustrerà il diagramma di sequenza dello use case 3.

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente

# Contratto delle operazioni UC3

|  |  |
| --- | --- |
| **Operazione** | ***showPendingBookingMap(host)*** |
| **Riferimenti** | UC3: Gestione prenotazioni pendenti |
| **Pre-condizioni** | L’host vuole confermare una prenotazione pendente  L’utente deve aver prenotato una workstation |
| **Post-condizioni** | L’host visualizza tutte le prenotazioni pendenti |

|  |  |
| --- | --- |
| **Operazione** | ***confirmBooking(booking)*** |
| **Riferimenti** | UC3: Gestione prenotazioni pendenti |
| **Pre-condizioni** | L’host vuole confermare una prenotazione pendente  L’utente deve aver prenotato una workstation |
| **Post-condizioni** | L’host conferma la prenotazione |

# Analisi UC6: Elimina prenotazione

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc6: Elimina prenotazione** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Utente |
| **Parti interessate e interessi** | Utente: vuole eliminate una prenotazione già effettuata o pendente. |
| **Pre-condizioni** | La richiesta di prenotazione da parte dell’utente è stata inviata con successo |
| **Garanzia di successo** | La richiesta o la prenotazione non risulti più presente nel sistema |
| **Scenario principale di successo** | 1. L’utente vuole eliminare una prenotazione o una richiesta di prenotazione 2. Il sistema mostra le prenotazioni effettuate e/o le prenotazioni pendenti 3. L’user sceglie l’id della prenotazione da cancellare 4. L’user seleziona se conferma o rifiuta la prenotazione 5. La richiesta di cancellazione è stata gestita |
| **Estensioni** | 4a: Se l’utente conferma, il sistema controlla se ci sono prenotazioni effettuate o pendenti corrispondenti a quell’id: se esiste conferma la rimozione altrimenti lancia un messaggio di errore  4b: Nel caso in cui si scelga di non continuare con l’eliminazione, si ritorna al menù precedente. |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Rara |
| **Varie** |  |

# Diagramma di sequenza UC6

Dato che il modello di dominio non ha subito modifiche, si illustrerà il diagramma di sequenza dello use case 6.

Immagine che contiene testo, schermata, linea, Parallelo

Descrizione generata automaticamente

# Contratto delle operazioni UC6

|  |  |
| --- | --- |
| **Operazione** | ***showBookingUser(user)*** |
| **Riferimenti** | UC6: Elimina prenotazione |
| **Pre-condizioni** | L’utente vuole eliminare una prenotazione effettuata  L’utente deve aver prenotato una workstation |
| **Post-condizioni** | L’utente visualizza tutte le prenotazioni effettuate |

|  |  |
| --- | --- |
| **Operazione** | ***deleteBooking(booking)*** |
| **Riferimenti** | UC6: Elimina prenotazione |
| **Pre-condizioni** | L’utente vuole eliminare una prenotazione effettuata  L’utente deve aver prenotato una workstation |
| **Post-condizioni** | L’utente elimina la prenotazione |

# Progettazione Iterazione 2

# Diagrammi di sequenza UC3

Immagine che contiene testo, linea, diagramma, ricevuta

Descrizione generata automaticamente

# Diagrammi di sequenza UC6

Immagine che contiene testo, ricevuta, numero, diagramma

Descrizione generata automaticamente

# Diagrammi delle classi UC3 e UC6

Immagine che contiene testo, diagramma, Piano, Rettangolo

Descrizione generata automaticamente

# Testing

# UC3

* E’ stato settato un Booking di test, un User di test, e una Workstation prima di eseguire i vari test
* E’ stato verificato il corretto funzionamento della conferma delle prenotazioni e la diminuzione della capienza massima
* E’ stato verificato che in fase di conferma delle prenotazioni, nel caso di superamento della capienza massima di quella workstation viene lanciata un eccezione
* E’ stato verificato che in caso di non conferma della prenotazione, questa non sia più pendente

# UC6

* E’ stato settato un User di test, una Workstation di test e sono stati aggiunti nei rispettivi database una prenotazione in fase di pending e una prenotazione confermata
* E’ stato verificata l’eliminazione di una prenotazione e il successivo incremento della capienza disponibile
* E’ stato verificata l’eliminazione di una richiesta di prenotazione e il successivo incremento della capienza disponibile
* E’ stato verificato il caso in cui la prenotazione non è stata trovata e viene lanciata un eccezione
* E’ stato verificato il caso in cui si decide di non confermare l’eliminazione e si arresterà il processo di eliminazione ripristinando lo stato precedente

# Elaborazione - Iterazione 3

# Refactoring

E’ stato effettuato un refactor su workstation. E’ stato aggiunto l’attributo ‘price’ e i metodi relativi per la gestione su Workstation e WorkstationController. Un altro refactor è stato effettuato sulla creazione dell’Utente: ogni utente avrà associata una CreditCard con numero di carta, un cvv e data di scadenza.

# Introduzione

In questa terza iterazione verranno approfonditi quei casi d’uso necessari per il pagamento e il rimborso.

Verrà effettuata una richiesta HTTP verso il server che gestirà i pagamenti. In questa prima fase il server è stato creato e simulato utilizzando un container docker, nella prossima fase verrà sostituito dal partner esterno che gestirà tutta la parte dei pagamenti.

In seguito alla richiesta di prenotazione di una workstation, per completare il processo, verrà effettuata una richiesta verso il server con gli estremi per il pagamento nel body (numero di carta, cvv, scadenza).

In seguito a una richiesta di cancellazione di una prenotazione, verrà effettuata una richiesta rimborso effettuando una richiesta HTTP verso il server che gestisce i pagamenti. Il processo di cancellazione sarà completato in seguito al rimborso.

# Analisi UC7: Pagamento

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc7: Pagamento** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Utente |
| **Parti interessate e interessi** | Utente: vuole pagare per la workstation prenotata |
| **Pre-condizioni** | La richiesta di prenotazione da parte dell’utente è stata inviata con successo |
| **Garanzia di successo** | Conferma da parte del server di gestione dei pagamenti |
| **Scenario principale di successo** | 1. L’utente vuole pagare in seguito alla propria richiesta di prenotazione 2. L’utente manda i propri dati al sistema 3. Il sistema recupera le informazioni sulla carta di credito associata all’utente 4. Il sistema crea la richiesta 5. Il sistema invia la richiesta 6. Il sistema restituisce la risposta all’utente |
| **Estensioni** |  |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Molto frequente |
| **Varie** |  |

# Modello di dominio UC7

E’ stato aggiunto l’attributo price su Workstation, l’attributo creditCard e la classe CreditCard

Immagine che contiene testo, diagramma, schermata, Piano

Descrizione generata automaticamente

# Diagramma di sequenza UC7

Immagine che contiene testo, linea, schermata, diagramma

Descrizione generata automaticamente

# Contratto delle operazioni UC7

|  |  |
| --- | --- |
| **Operazione** | ***makePayment(user)*** |
| **Riferimenti** | UC7: Pagamento |
| **Pre-condizioni** | L’utente vuole effettuare un pagamento e concludere la sua richiesta di prenotazione  L’utente deve aver effettuato una richiesta di prenotazione di una workstation |
| **Post-condizioni** | L’utente riceve una risposta dal server con indicato che il pagamento è andato a buon fine |

# Analisi UC8: Rimborso

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc8: Rimborso** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Utente |
| **Parti interessate e interessi** | Utente: vuole il rimborso per la workstation prenotata |
| **Pre-condizioni** | La richiesta di prenotazione da parte dell’utente è stata inviata o gestita con successo |
| **Garanzia di successo** | Conferma da parte del server di gestione dei pagamenti |
| **Scenario principale di successo** | 1. L’utente vuole essere rimborsato in seguito alla propria richiesta di cancellazione 2. Il sistema crea la richiesta 3. Il sistema invia la richiesta 4. Il sistema restituisce la risposta all’utente |
| **Estensioni** |  |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Raro |
| **Varie** |  |

# Diagramma di sequenza UC8

Il modello di dominio è rimasto invariato.

Immagine che contiene linea, testo, schermata, diagramma

Descrizione generata automaticamente

# Contratto delle operazioni UC8

|  |  |
| --- | --- |
| **Operazione** | ***requestRefund()*** |
| **Riferimenti** | UC8: Rimborso |
| **Pre-condizioni** | L’utente vuole richiedere un rimborso e concludere la sua richiesta di eliminazione  L’utente deve aver effettuato una prenotazione di una workstation |
| **Post-condizioni** | L’utente riceve una risposta dal server con indicato che il rimborso è andato a buon fine |

# Progettazione Iterazione 3

# Diagrammi di sequenza UC7

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente

# Diagrammi di sequenza UC8

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente

# Diagramma delle classi UC7 e UC8

Immagine che contiene testo, diagramma, Piano, Rettangolo

Descrizione generata automaticamente

# Testing

In questa iterazione, per questi due Use Case, la fase di test è stata effettuata da terminale, per verificare la disponibilità e il funzionamento del server.

Sono state provate delle richieste API POST e GET verso il server, utilizzando il comando curl, per testare la risposta di questo in caso di successo o errore

# Elaborazione - Iterazione 4

# Introduzione

In questa quarta iterazione verranno approfonditi quei casi d’uso necessari per la creazione di attività e la prenotazione di queste.

L’host sceglie se offrire il pernottamento, il pranzo, o eventi di teambuilding e per ognuno di questi setterà un prezzo.

# Analisi UC2: Inserisci altri servizi

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc2: Inserisci altri servizi** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Host |
| **Parti interessate e interessi** | Host: vuole inserire dei servizi aggiuntivi |
| **Pre-condizioni** | Deve esistere una workstation |
| **Garanzia di successo** | Le informazioni relative ai servizi sono state inserite nel sistema |
| **Scenario principale di successo** | 1. L’host vuole aggiungere altri servizi da poter affittare alla propria workstation 2. Il sistema controlla se è associata una workstation per quell’host 3. L’host sceglie se vuole aggiungere il pernottamento ed eventualmente setta il prezzo 4. L’host sceglie se vuole aggiungere il pranzo ed eventualmente setta il prezzo 5. L’host sceglie se vuole aggiungere l’evento di team building ed eventualmente setta il prezzo 6. Il sistema controlla se esiste già un attività con lo stesso id 7. Viene settata l’attività |
| **Estensioni** | 2a. Se non esiste una workstation associata, si ritorna allo stato precedente  6a. Se esiste già un attività con lo stesso id viene generato un errore |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Molto frequente |
| **Varie** |  |

# Modello di dominio UC2

E’ stata creata la classe Activity

Immagine che contiene testo, diagramma, Piano, schermata

Descrizione generata automaticamente

# Diagramma di sequenza UC2

Immagine che contiene schermata, testo, linea, Rettangolo

Descrizione generata automaticamente

# Contratto delle operazioni UC2

|  |  |
| --- | --- |
| **Operazione** | ***setActivity()*** |
| **Riferimenti** | UC2: Inserisci altri servizi |
| **Pre-condizioni** | L’host vuole inserire altri servizi associati alla workstation  L’host deve aver creato una workstation |
| **Post-condizioni** | E’ stata creata una nuova istanza di Activity |

# Analisi UC5: Prenota altri servizi

|  |  |
| --- | --- |
| **Nome del caso d’uso** | **Uc5: Prenota altri servizi** |
| **Portata** | Applicazione CoLinking |
| **Livello** | Obiettivo utente |
| **Attore primario** | Utente |
| **Parti interessate e interessi** | Utente: vuole prenotare uno dei servizi aggiuntivi associati alla workstation prenotata |
| **Pre-condizioni** | Deve esistere una workstation  Deve esistere una prenotazione o una richiesta di prenotazione  L’host deve aver creato i servizi per la sua workstation |
| **Garanzia di successo** | La prenotazione viene aggiunta nel database con successo. |
| **Scenario principale di successo** | 1. L’utente vuole prenotare dei servizi associati alla workstation prenotata 2. Il sistema mostra le prenotazioni effettuata 3. L’utente sceglie la prenotazione della workstation per cui prenotare altri servizi 4. Il sistema mostra i servizi associati a quella workstation 5. L’utente sceglie se implementare servizi previsti per quella workstation 6. L’utente sceglie quanti giorni usufruirà di questi servizi 7. Il sistema calcola il prezzo totale dei servizi scelti 8. Crea una nuova istanza di booking 9. Ritorna all’utente il prezzo totale dei servizi 10. L’utente effettua il pagamento dei servizi |
| **Estensioni** | 2a. Se non esiste una prenotazione, si ritorna allo stato precedente |
| **Requisiti speciali** |  |
| **Elenco delle varianti tecnologiche e dei dati** |  |
| **Frequenza di ripetizioni** | Molto frequente |
| **Varie** |  |

# Modello di dominio UC5

E’ stato aggiunto l’attributo activity alla classe Booking.

Immagine che contiene testo, diagramma, Piano, schermata

Descrizione generata automaticamente

# Diagramma di sequenza UC5

Immagine che contiene testo, schermata, linea, Parallelo

Descrizione generata automaticamente

# Contratto delle operazioni UC5

|  |  |
| --- | --- |
| **Operazione** | ***showBookingUser()*** |
| **Riferimenti** | UC5: Prenota altri servizi |
| **Pre-condizioni** | L’utente vuole prenotare i servizi associati alla workstation  L’utente deve aver inviato almeno una richiesta di prenotazione per una workstation |
| **Post-condizioni** | Mostra all’utente le prenotazioni |

|  |  |
| --- | --- |
| **Operazione** | ***showActivities()*** |
| **Riferimenti** | UC5: Prenota altri servizi |
| **Pre-condizioni** | L’host deve aver associato delle attività a quella workstation  L’utente vuole prenotare i servizi associati alla workstation  L’utente deve aver inviato almeno una richiesta di prenotazione per una workstation |
| **Post-condizioni** | Mostra all’utente le attività |

|  |  |
| --- | --- |
| **Operazione** | ***bookActivity()*** |
| **Riferimenti** | UC5: Prenota altri servizi |
| **Pre-condizioni** | L’host deve aver associato delle attività a quella workstation  L’utente vuole prenotare i servizi associati alla workstation  L’utente deve aver inviato almeno una richiesta di prenotazione per una workstation |
| **Post-condizioni** | E’ stata creata una nuova istanza di Booking |

# Progettazione Iterazione 4

# Diagrammi di sequenza UC2

Immagine che contiene testo, ricevuta, linea, Parallelo

Descrizione generata automaticamente

# Diagrammi di sequenza UC5

Immagine che contiene testo, Parallelo, diagramma, numero

Descrizione generata automaticamente

# Diagramma delle classi UC2 e UC5

Immagine che contiene testo, diagramma, Rettangolo, Piano

Descrizione generata automaticamente

# Refactoring

E’ stato effettuato un refactor riguardante la cancellazione di una prenotazione. Dopo aver effettuato una cancellazione di una prenotazione per una workstation verranno cancellati anche i servizi associati che erano stati prenotati

# Testing

# UC2

* E’ stato settato un Host di test prima di eseguire i vari test
* E’ stata verificata la corretta creazione delle attività
* E’ stato verificato il caso in cui si provi a creare un attività già esistente, verrà lanciata un eccezione

# UC5

* E’ stato settato un Activity di test ed è stato aggiunto nel proprio database prima di eseguire i vari test
* E’ stata verificata la corretta gestione della richiesta di prenotazione soltanto per il pernottamento
* E’ stata verificata la corretta gestione della richiesta di prenotazione soltanto per il pranzo
* E’ stata verificata la corretta gestione della richiesta di prenotazione soltanto per l’evento di team building
* E’ stata verificata la corretta gestione della richiesta di prenotazione soltanto per tutte le attività disponibili