Università degli Studi di Catania

Dipartimento di Ingegneria Elettrica Elettronica e Informatica

Corso di Laurea Magistrale in Ingegneria Informatica (LM-32)

A.A. 2019 – 2020

Ingegneria del Software

Finch Airlines

Studenti:

Cavallaro Salvatore 1000008690

Dilonardo Matteo 1000014339

Patanè Rosario 1000013474

Prefazione

Il presente documento ha l'obiettivo di descrivere in modo completo l'intero processo di sviluppo del progetto software Finch Airlines, relativo al corso di Ingegneria del Software. Le pagine seguenti dettagliano le singole fasi del processo di sviluppo presentando i risultati finali delle singole elaborazioni. Il progetto è inoltre stato realizzato, da un punto di vista software, mediante IDE Eclipse con linguaggio di programmazione Java. La redazione dei diagrammi di Analisi e Progettazione è stata eseguita impiegando il software Astah. Infine, si è prevista anche l'esecuzione delle fasi di testing per il codice prodotto mediante il tool jUnit 5.

Indice

1. Ideazione e analisi dei requisiti

Introduzione

Nella fase di ideazione si effettua uno studio per la realizzazione di un punto di partenza del progetto da realizzare. Tale passaggio prevede lo sviluppo degli elaborati utili a comprendere la disciplina dei requisiti, comprendente quindi i seguenti documenti: Modello dei Casi d'Uso, Documento di Visione, Specifiche supplementari, Regole di Business e Glossario.

Requisiti

Il CTO di una compagnia aerea richiede l'implementazione di un software per la gestione automatizzata della prenotazione di voli da parte dei suoi clienti. Il sistema deve prevedere la possibilità di prenotazione di un volo da parte del cliente e la gestione della prenotazione stessa. Il sistema deve anche prevedere la gestione da parte di un amministratore. In modo specifico:

- L'amministratore deve potersi autenticare nel sistema.
- L'amministratore deve poter gestire i voli, aggiungendoli, modificandoli ed eliminandoli.
- L'amministratore deve poter gestire le tratte (aeroporto partenza – destinazione).
- L'amministratore deve poter gestire i programmi fedeltà disponibili.
- Il cliente deve potersi registrare nel sistema.
- Il cliente deve potersi autenticare nel sistema.
- Il cliente deve poter cercare un volo selezionando la tratta e la data. Il sistema effettua la ricerca dei voli disponibili e li comunica al cliente:
    a. Se almeno un volo è disponibile il cliente visualizza le informazioni.
    b. Se nessun volo è disponibile il sistema restituisce un messaggio informativo.

  Nota: il volo ricercato può essere di sola andata o di Andata e Ritorno.

- Nel caso a. il sistema visualizza a schermo le informazioni del volo quali orario, numero di volo e prezzo e offre la possibilità di prenotarlo.
- Il cliente deve poter prenotare il volo inserendo nome, cognome, e-mail e documento identificativo (es. CI, Passaporto, …).
- Il cliente deve poter effettuare il pagamento del volo mediante metodi differenti, quindi, il sistema invia una e-mail riepilogativa.
- Il cliente all'atto del pagamento riceve dei punti per il programma fedeltà da poter usare per ottenere uno sconto su altre prenotazioni.
- Il cliente può modificare la propria prenotazione, anche aggiungendo servizi (es. prenotazione posto, bagaglio in stiva, …).
- Il cliente deve poter effettuare il check-in online e stampare la carta di imbarco.
- Il cliente deve poter cancellare una prenotazione già effettuata.

Obiettivi e casi d'uso

Attore	Obiettivo	Caso d'uso
Cliente	Ricercare un volo inserendo l'aeroporto di partenza, quello di destinazione e la data	UC1: Ricerca volo
Cliente	Selezionare il volo desiderato, inserire, rimuovere, modificare i dati personali e i servizi aggiuntivi	UC2: Gestisci prenotazione (CRUD, Ricerca prenotazione)
Cliente	Pagare la prenotazione effettuata (con eventuali servizi aggiuntivi e sconto)	UC3: Effettua pagamento
Cliente	Ricercare la prenotazione, effettuare il check-in online e stampare la carta d'imbarco	UC4: Effettua check-in

Cliente	Registrare il nuovo cliente inserendo i suoi dati	UC5: Registra Cliente
Cliente	Autenticare il cliente al sistema	UC6: Autentica Cliente
Amministratore	Autenticare l'amministratore al sistema	UC7: Autentica Amministratore
Amministratore	Selezionare il volo, inserire, rimuovere, modificare i dati relativi	UC8: Gestisci volo (CRUD, Ricerca volo)
Amministratore	Selezionare il programma fedeltà, inserire, rimuovere, modificare il programma	UC9: Gestisci programma fedeltà (CRUD, Ricerca programma fedeltà)

Modello dei casi d'uso

UC1: Ricerca volo

Nome del caso d'uso	**UC1: Ricerca volo**
Portata	Applicazione Finch Airlines
Livello	Obiettivo utente
Attore primario	Cliente
Parti interessate e Interessi	- Cliente: vuole ricercare un volo e ottenere informazioni rapide sulle possibili scelte - Compagnia aerea: vuole offrire un servizio per presentare in modo semplice i voli disponibili
Pre-condizioni	Il cliente è autenticato
Garanzia di successo	La ricerca restituisce una lista di voli disponibili per la tratta e la data indicate dal cliente

Scenario principale di successo	1. Il cliente vuole effettuare la ricerca di un volo 2. Il cliente inserisce le informazioni del volo desiderato quali aeroporto di partenza, aeroporto di destinazione e tipo di viaggio (solo andata, A/R, multi-tratta) 3. Il cliente avvia la ricerca 4. Il sistema visualizza la lista con i voli trovati per le date e le tratte selezionate
Estensioni	*a. In qualsiasi momento il sistema fallisce 1. Il sistema avvisa il cliente dell'errore 2. Il sistema ritorna alla schermata di ricerca iniziale 1b. Il cliente non è registrato 1. Il sistema avvisa il cliente 2. Il cliente procede alla registrazione 3. Il cliente risulta autenticato 2b. Il volo da ricercare è di sola andata 1. Il cliente seleziona l'opzione 2. Il cliente inserisce la data di andata 2c. Il volo da ricercare è di A/R 1. Il cliente seleziona l'opzione 2. Il cliente inserisce le date di andata e di ritorno 2d. Il volo da ricercare è multi-tratta 1. Il cliente seleziona l'opzione 2. Il cliente inserisce le tratte 3. Il cliente inserisce le date 4b. Non esistono voli che soddisfano i criteri di ricerca 1. Il sistema avvisa il cliente 2. Il sistema ritorna alla schermata di ricerca iniziale
Requisiti speciali	Non presenti
Elenco delle varianti tecnologiche e dei dati	

Frequenza di ripetizioni	Frequenza elevata, legata al numero di clienti che intendono cercare un volo e al numero di ricerche per cliente
Varie	

UC2: Gestisci prenotazione

Nome del caso d'uso	**UC2: Gestisci prenotazione**
Portata	Applicazione Finch Airlines
Livello	Obiettivo utente
Attore primario	Cliente
Parti interessate e Interessi	- Cliente: vuole gestire una prenotazione creandola, modificando i dati di una effettuata in precedenza o cancellandola.
Pre-condizioni	Il cliente ha effettuato la ricerca di un volo
Garanzia di successo	Il sistema ha effettuato il salvataggio delle informazioni della prenotazione
Scenario principale di successo	1. Il cliente vuole effettuare una prenotazione e seleziona i voli desiderati tra quelli proposti dal sistema 2. Il cliente seleziona un posto e il tipo desiderato per il singolo volo 3. Il cliente seleziona il tipo di bagaglio desiderato per il singolo volo 4. Il sistema visualizza un riepilogo dei dati della prenotazione 5. Il cliente conferma la prenotazione 6. Il sistema spedisce una e-mail riepilogativa dell'ordine
Estensioni	*a. In qualsiasi momento il sistema fallisce 1. Il sistema avvisa il cliente dell'errore 2. Il sistema ritorna alla schermata di ricerca iniziale

1b. Il cliente vuole modificare una prenotazione esistente

1. Il cliente inserisce il numero di prenotazione.
2. Il sistema ricerca la prenotazione e la mostra al cliente

   2a. Il sistema non trova la prenotazione

      1. Il sistema avvisa il cliente
      2. Il cliente inserisce un nuovo numero di prenotazione

3. Il cliente seleziona l'operazione da svolgere sulla prenotazione:

   3a. Il cliente seleziona la modifica dei dati anagrafici

      1. Il cliente modifica i dati (nome, cognome, documento identificativo)
      2. Il cliente conferma la modifica dei dati
      3. Il sistema conferma la modifica effettuata

   3b. Il cliente seleziona l'acquisto di un posto a sedere

      1. Il cliente seleziona il posto tra quelli ancora disponibili
      2. Il sistema aggiorna il totale corrente
      3. Il cliente conferma la scelta

   3c. Il cliente seleziona l'acquisto di un bagaglio

      1. Il cliente sceglie la quantità di bagagli da aggiungere alla prenotazione
      2. Il sistema aggiorna il totale corrente
      3. Il cliente conferma la scelta

4. Il cliente conferma la modifica della prenotazione

	5. Il sistema mostra il totale e procede con l'eventuale pagamento 1c. Il cliente vuole annullare una prenotazione esistente 1. Il cliente inserisce il numero di prenotazione 2. Il sistema ricerca la prenotazione e la mostra al cliente 2a. Il sistema non trova la prenotazione 1. Il sistema avvisa il cliente 2. Il cliente inserisce un nuovo numero di prenotazione 3. Il cliente effettua la rimozione della prenotazione 4. Il sistema effettua lo storno 1d. Il cliente vuole cercare e visualizzare una prenotazione 1. Il cliente inserisce il numero di prenotazione 2. Il sistema ricerca la prenotazione e la mostra al cliente 2a. Il sistema non trova la prenotazione 1. Il sistema avvisa il cliente 2. Il cliente inserisce un nuovo numero di prenotazione 3. Il cliente visualizza la prenotazione
Requisiti speciali	Non presenti
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Frequenza elevata nella creazione delle nuove prenotazioni
Varie	

UC3: Effettua pagamento

1. Il cliente procede con il pagamento di una prenotazione effettuata

2. Il cliente sceglie il metodo di pagamento e viene mostrato il totale della prenotazione
3. Il cliente seleziona un eventuale sconto mediante punti accumulati con programma fedeltà
4. Il sistema mostra il totale da pagare, eventualmente scontato
5. Il cliente effettua il pagamento
6. Il sistema conferma l'avvenuto pagamento della prenotazione
7. Il sistema accredita i punti del programma fedeltà, se quest'ultimo è presente, e li mostra al cliente

Scenari Alternativi:

1a. Il cliente procede con il pagamento dei servizi aggiuntivi su prenotazione effettuata in precedenza

2a. Il cliente sceglie di pagare con carta di credito/prepagata

2b. Il cliente sceglie di pagare con PayPal

5a. Il cliente paga con carta di credito/prepagata

1. Il sistema reindirizza al servizio di pagamento esterno
2. Il sistema riceve la conferma di avvenuto pagamento dal servizio esterno

5b. Il cliente paga con PayPal

1. Il sistema reindirizza al servizio di pagamento esterno
2. Il sistema riceve la conferma di avvenuto pagamento dal servizio esterno

UC4: Effettua check-in

1. Il sistema ricerca tutte le prenotazioni associate al Cliente
2. Il cliente seleziona la prenotazione
3. Il cliente inserisce le informazioni richieste (volo) e conferma i dati inseriti
4. Il sistema restituisce la carta d'imbarco al cliente per la prenotazione selezionata

Scenari Alternativi:

3a. Il check-in è stato già effettuato

1. Il sistema avvisa il cliente
2. Il sistema restituisce la carta d'imbarco al cliente per la prenotazione selezionata

3b. Sono mancanti più di 7 giorni alla data di partenza del volo

1. Il sistema avvisa il cliente

3c. Il volo è stato già effettuato

1. Il sistema avvisa il cliente

3d. Il pagamento non è ancora stato effettuato

UC5: Registra Cliente

1. Il cliente vuole registrarsi al sistema di prenotazioni di voli della compagnia Finch Airlines
2. Il sistema mostra la pagina di registrazione
3. Il cliente inserisce i propri dati quali nome, cognome, e-mail e password
4. Il sistema verifica i dati inseriti
5. Il cliente conferma i dati inseriti
6. Il sistema notifica l'avvenuta registrazione

Scenario alternativo:

4a. I dati inseriti non sono corretti

1. Il sistema avvisa il cliente
2. Il cliente reinserisce i dati

UC6: Autentica Cliente

1. Il cliente vuole accedere al sistema mediante autenticazione
2. Il cliente inserisce e-mail e password
3. Il sistema verifica i dati di autenticazione
4. Il sistema avvisa il cliente dell'avvenuta autenticazione

Scenario alternativo:

3a. I dati di autenticazione inseriti dal cliente non sono corretti

1. Il sistema avvisa il cliente:

1a. Il cliente non è registrato

1. Il cliente procede con la registrazione

1b. La password inserita non è corretta

1. Il cliente inserisce la password corretta

UC7: Autentica Amministratore

1. L’amministratore vuole accedere al sistema mediante autenticazione
2. L’amministratore inserisce e-mail e password
3. Il sistema verifica i dati di autenticazione
4. Il sistema avvisa l’amministratore dell’avvenuta autenticazione

Scenario alternativo:

3a. I dati di autenticazione inseriti dall’amministratore non sono corretti

1. L’amministratore inserisce i dati corretti

UC8: Gestisci volo, CRUD

1. L’amministratore vuole inserire un nuovo volo nel sistema (singolo o periodico)
2. L’amministratore inserisce la tratta del volo
3. L’amministratore inserisce la descrizione del volo con i relativi dati: il codice volo e la tratta
4. L’amministratore inserisce il prezzo, la data e la descrizione del volo (multipli se il volo è periodico)
5. Il sistema restituisce un riepilogo del/dei volo/i creato/i
6. L’amministratore conferma i dati inseriti
7. Il sistema avvisa l’amministratore dell’avvenuto inserimento del volo

Scenari alternativi:

1a. L’amministratore vuole ricercare un volo già presente nel sistema

1. L’amministratore inserisce il numero del volo

   1a. Il volo non è presente

   1. Il sistema avvisa l’amministratore
   2. L’amministratore inserisce un nuovo numero di volo
2. Il sistema mostra il volo e le informazioni correlate

1b. L’amministratore vuole modificare un volo già presente nel sistema

1. L’amministratore inserisce i dati da modificare
2. L’amministratore conferma i dati inseriti
3. Il sistema avvisa dell’avvenuta modifica del volo

1c. L’amministratore vuole eliminare un volo già presente nel sistema

1. L’amministratore conferma la richiesta di eliminazione del volo
2. Il sistema avvisa dell’avvenuta eliminazione del volo

UC9: Gestisci programma fedeltà (CRUD, Ricerca programma fedeltà)

1. L’amministratore vuole inserire un nuovo programma fedeltà nel sistema
2. L’amministratore inserisce i dati del programma fedeltà: nome programma, coefficiente per la generazione dei punti sulla base del prezzo
3. L’amministratore associa il nuovo programma fedeltà ad un volo tramite il relativo codice
4. Il sistema avvisa l’amministratore dell’avvenuto inserimento del programma fedeltà

Scenari alternativi:

1a. L’amministratore vuole ricercare un programma fedeltà esistente nel sistema

1. L’amministratore inserisce il nome del programma fedeltà
2. L’amministratore conferma i dati inseriti
3. Il sistema trova il programma fedeltà e lo mostra all’amministratore

   3a. Il sistema non trova il programma fedeltà

      1. Il sistema avvisa l’amministratore

1b. L’amministratore vuole modificare un programma fedeltà esistente nel sistema

1. L’amministratore ricerca il programma fedeltà
2. L’amministratore modifica i dati del programma fedeltà
3. L’amministratore conferma i dati inseriti
4. Il sistema avvisa l’amministratore dell’avvenuta modifica

1c. L’amministratore vuole eliminare un programma fedeltà esistente nel sistema

1. L’amministratore ricerca il programma fedeltà
2. L’amministratore elimina il programma fedeltà
3. Il sistema avvisa l’amministratore dell’avvenuta eliminazione

Documento di Visione

Contestualmente alla stesura del documento di ideazione è stata redatta una prima versione del documento di Visione (vedi appendice A).

Regole di business

ID	Regola	Modificabilità	Sorgente
R1	È possibile effettuare il check-in online solo da 7 giorni prima del volo	Bassa, il limite temporale potrebbe variare in specifici momenti dell'anno	Politica interna della compagnia
R2	La cancellazione della prenotazione è effettuabile entro la data di apertura del check-in online	Nulla	Politica interna della compagnia
R3	La penale di cancellazione della prenotazione prevede: • 30% del prezzo della prenotazione sino a 30 giorni lavorativi prima della partenza • 50% del prezzo della prenotazione da 29 a 10 giorni lavorativi prima della partenza • 100% del prezzo della prenotazione dopo tali termini	Bassa, gli importi possono variare al più una volta l'anno	Politica interna della compagnia
R4	I punti fedeltà sono impiegati come sconto sul prezzo della prenotazione secondo determinati programmi	Media, i programmi possono variare più volte l'anno	Politica interna della compagnia

Specifiche Supplementari

Usabilità

- L’interfaccia grafica deve essere intuitiva e semplice anche per un utente non esperto.

Affidabilità

- Il software prevede un backup periodico del database

Vincoli hardware e software

- Per l’esecuzione è necessaria la Java Virtual machine (JVM) versione 1.8
- Il software necessita di una connessione a internet sempre attiva per la comunicazione tramite e-mail

Vincoli di sviluppo del software

- Il software è interamente scritto in Java e viene utilizzato un database per mantenere i dati dei voli e dei clienti

Aspetti legali

- Finch Airlines verrà rilasciato sotto licenza proprietaria

Glossario

- **Tratta**: percorso da un aeroporto ad un altro senza distinzione del verso di percorrenza
- **Volo:** tratta percorsa con un verso caratterizzata da una data e un orario
- **Prenotazione:** insieme dei dati selezionati dal cliente che comprende informazioni del volo e del cliente stesso
- **Cliente:** utente del sistema registrato e che accede per l’acquisto o la gestione di prenotazioni
- **Amministratore:** utente del sistema che accede per la gestione dello stesso
- **Programma fedeltà:** politica di sconto applicata sulla prenotazione di un cliente secondo differenti programmi prestabiliti
- **Check-in:** operazione di verifica dei dati del cliente per una data prenotazione e rilascio della relativa carta d’imbarco in forma digitale
- **Pagamento:** operazione di acquisto della prenotazione effettuata da parte di un cliente con differenti modalità

- **Carta d'imbarco:** documento contenente i dati della prenotazione di un cliente relativi ad uno specifico volo
- **Volo periodico:** è un volo che si può ripetere periodicamente
- **Prezzo multiplo:** prezzo associato ad un volo periodico, un prezzo per ogni volo
- **Coefficiente per la generazione dei punti:** è un numero usato dal programma fedeltà per ricavare i punti sulla base del prezzo tramite una formula.

2. Analisi Orientata agli Oggetti

Introduzione

L'approccio utilizzato per lo sviluppo del presente progetto software ha seguito i principi dettati dal processo di tipo evolutivo iterativo incrementale con *Unified Processing*. In particolare, lo sviluppo è stato articolato in 4 differenti elaborazioni, le quali hanno consentito via via di aggiungere nuove funzionalità al sistema partendo da un nucleo di base in grado di eseguire semplici operazioni. La singola iterazione ha quindi previsto l'analisi dei requisiti e la progettazione/implementazione dei nuovi singoli casi d'uso e/o la modifica di quelli fatti in precedenza, in conformità alle variazioni che il software ha subito nel processo di sviluppo.

In particolare, all'interno delle singole iterazioni si ha:

- **Iterazione 1**
    - Implementazione dei soli scenari di successo del caso d'uso *UC1: Ricerca volo* e del caso d'uso *UC2: Gestisci prenotazione*
    - Allestimento di un caso d'uso di start up al fine di testare gli elementi prodotti nella corrente iterazione

- **Iterazione 2**
    - Implementazione del caso d'uso UC3: Effettua pagamento utilizzato per le operazioni di pagamento di un volo appena prenotato dal Cliente. In particolare, in questa iterazione non si considera l'applicazione dello sconto mediante punti fedeltà accumulati. Si modellano inoltre gli scenari alternativi 4a) e 4b).
    - Implementazione del caso d'uso UC4: Effettua check-in utilizzato per le operazioni di check-in online dopo che è stata effettuata una prenotazione da parte del Cliente.

- **Iterazione 3**

- Implementazione del caso d'uso UC8: Gestisci volo, utilizzato per la gestione di tutte le operazioni (CRUD) relative ad un volo da parte dell'amministratore del sistema. In particolare, si prende in considerazione lo scenario di inserimento di un nuovo volo.
- Implementazione del caso d'uso UC9: Gestisci programma fedeltà, utilizzato dall'amministratore per inserire nel sistema differenti tipologie di programmi fedeltà mediante punti accumulati dal cliente durante l'acquisto, utilizzabili per ottenere uno sconto su acquisti successivi. In particolare, si prende in considerazione lo scenario di inserimento di un nuovo programma fedeltà.

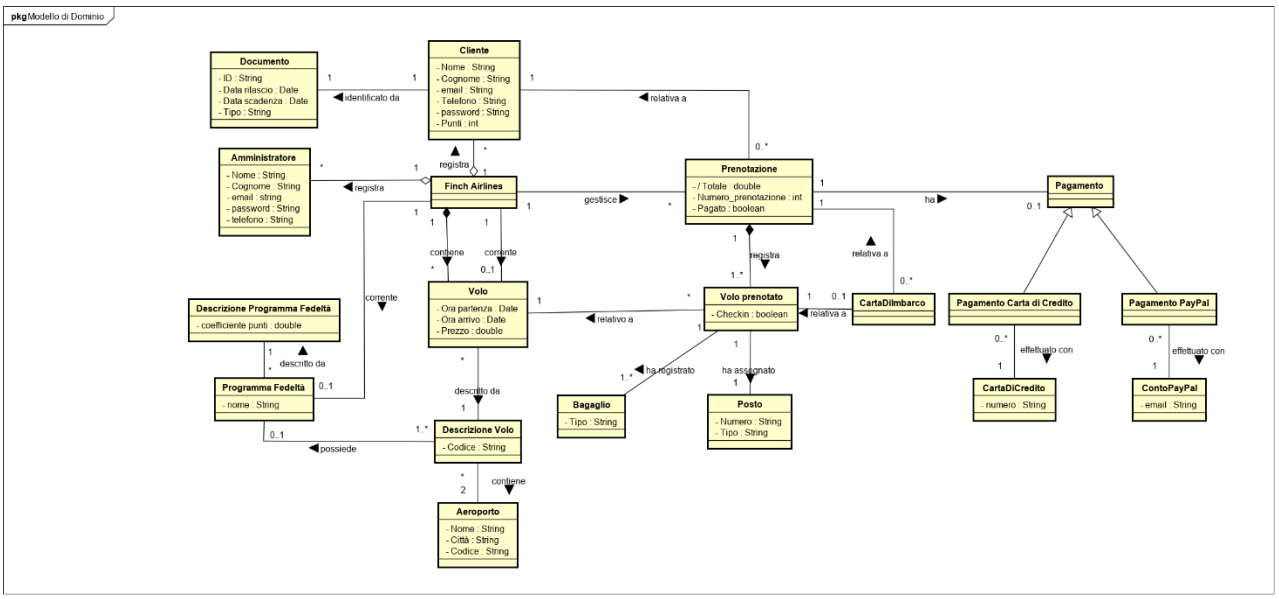
- **Iterazione 4**
  - Implementazione del caso d'uso UC6: Autentica Cliente, utilizzato per l'autenticazione del cliente al sistema e richiesta prima di eseguire ogni operazione a lui pertinente. Inoltre, si prende in considerazione lo scenario principale del caso d'uso.
  - Implementazione del caso d'uso UC7: Autentica Amministratore, utilizzato dall'amministratore per l'autenticazione, necessaria per poter eseguire le operazioni a lui pertinenti. Si prende in considerazione lo scenario principale del caso d'uso.
  - Aggiornamento del caso d'uso UC1: Ricerca Volo, viene rimossa l'operazione di autenticazione da parte del cliente perché è presente all'interno del caso d'uso UC6.
  - Aggiornamento del caso d'uso UC3: Effettua Pagamento, viene modificato per includere la funzionalità di assegnazione punti del programma fedeltà all'atto dell'acquisto.

Una delle prime operazioni eseguite per ogni iterazione è stata l'analisi dei requisiti seguendo una descrizione del dominio basata su un approccio di tipo OO. In particolare, in fase di analisi sono stati realizzati il Modello di Dominio, i Diagrammi di Sequenza di Sistema (SSD) e i Contratti delle Operazioni.

Modello di Dominio

La realizzazione del Modello di Dominio consente, attraverso un diagramma, di mettere in evidenza quelle che sono classi che costituiscono il sistema in termini di *concetti*. Ne vengono quindi messe in risalto le

relazioni, attraverso le *associazioni* e le relative proprietà, in termini di *attributi*. Il Modello di Dominio complessivo del sistema è il seguente:



Il Modello di Dominio è costituito dalle seguenti classi concettuali:

- **Finch Airlines**: rappresenta il sistema
- **Cliente**: rappresenta il cliente che vuole effettuare una prenotazione
- **Documento**: rappresenta il documento identificativo utilizzato dal Cliente per prenotare
- **Prenotazione**: rappresenta la prenotazione contenente l'insieme dei dati dei singoli voli che la costituiscono
- **Volo prenotato**: rappresenta il volo associato ad una particolare prenotazione e l'insieme dei servizi ad esso correlati
- **Posto**: rappresenta il posto associato ad un volo e le relative informazioni descrittive
- **Bagaglio**: rappresenta il bagaglio associato ad un volo e le relative informazioni descrittive
- **Volo**: rappresenta un volo operato dalla compagnia aerea
- **Descrizione Volo**: rappresenta la descrizione associata ad un particolare volo
- **Aeroporto**: rappresenta un aeroporto e le relative informazioni descrittive
- **Pagamento**: classe che generalizza i metodi di Pagamento utilizzabili
- **Pagamento Carta di Credito**: sottoclasse per il pagamento mediante Carta di Credito
- **Pagamento PayPal**: sottoclasse per il pagamento mediante conto PayPal
- **Carta di Credito:** classe contenente le informazioni relative ad una carta di credito
- **Conto PayPal:** classe contente le informazioni relative ad un conto PayPal
- **Carta di Imbarco:** rappresenta la carta di imbarco associata ad un volo acquistato da un cliente in una data prenotazione

- **Amministratore**: classe che rappresenta l'amministratore del sistema che ne svolge le operazioni di gestione.
- **Programma Fedeltà:** classe che modellizza un dato programma fedeltà associato ad una descrizione di un volo
- **Descrizione Programma Fedeltà:** classe che contiene le informazioni descrittive di un dato programma fedeltà

SSD e Contratti delle Operazioni

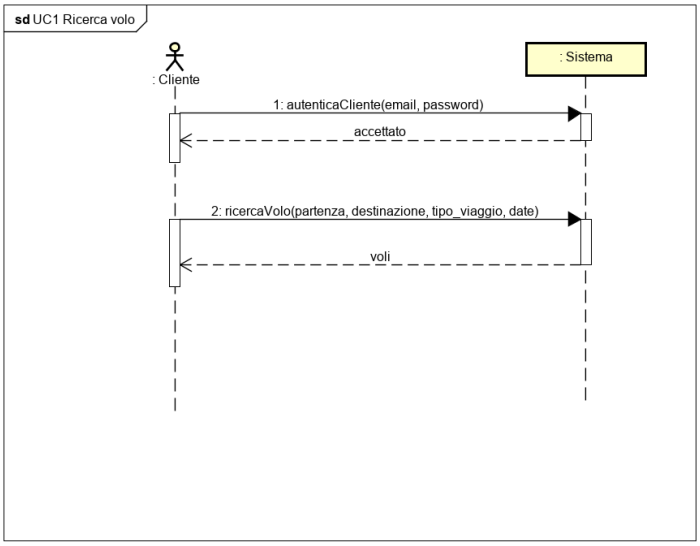
Ulteriori due elaborati prodotti in corrispondenza della fase di analisi di ciascuna iterazione sono i Diagrammi di sequenza di Sistema (SSD) e i Contratti delle operazioni.

Con i diagrammi di sequenza di sistema vengono descritte le operazioni svolte in ciascun caso d'uso, individuato con l'analisi dei requisiti, in termini di interazione tra un dato attore (Cliente o Amministratore) e il Sistema stesso. La descrizione realizzata mediante SSD viene quindi completata con i Contratti delle Operazioni al fine di individuare, laddove presenti, in corrispondenza di ciascuna operazione degli SSD cosa accade in termini di creazione/rimozione associazioni, oggetti istanziati e attributi modificati.

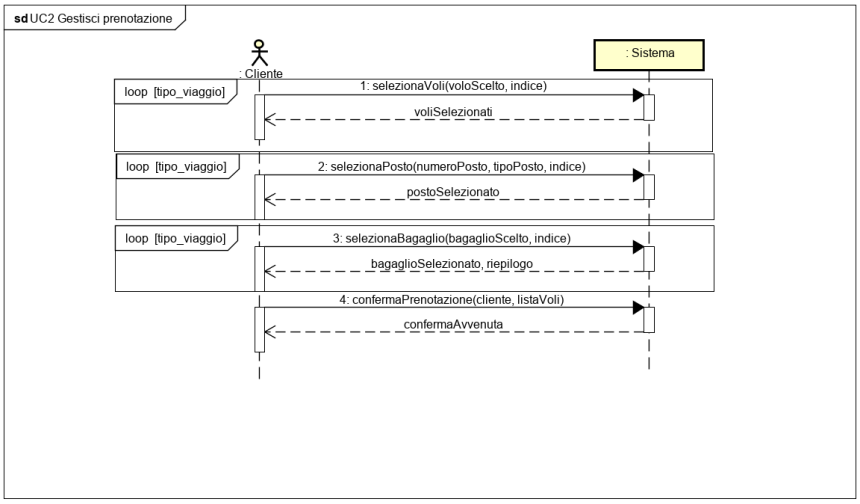
Per le singole iterazioni si ha:

- **Iterazione 1**

  Con il primo SSD si mostra l'interazione tra il Cliente e il Sistema al fine di eseguire l'operazione di ricerca di un nuovo volo, che prevede il fatto che vengano forniti dati quali aeroporto di partenza, aeroporto di destinazione e date per ogni tratta. In questa prima iterazione si suppone che l'operazione avvenga solo dopo che l'utente si è autenticato.



Con il secondo SSD si rappresenta l'interazione in termini di gestione della prenotazione, nel caso di una nuova prenotazione . L'interazione prevede l'esecuzione della scelta di Voli (precedentemente ricercati), posto e bagaglio per la singola tratta.



Contratto CO1: selezionaVoli

Operazione: *selezionaVoli(voloScelto:* Volo*, indice*: int*)*

Riferimenti: caso d'uso: Gestisci prenotazione

Pre-condizioni: È stata effettuata una ricerca di voli

Post-condizioni:

- È stata creata un'istanza di *VoloPrenotato*
- L'istanza di *Volo* è stata associata a *VoloPrenotato*

- *VoloPrenotato* è stato associato a *listaVoli[indice]*

Contratto CO2: selezionaPosto

Operazione: *selezionaPosto(numeroPosto:* String, *tipoPosto:* String, *indice*: int*)*

Riferimenti: caso d'uso: Gestisci prenotazione

Pre-condizioni: Sono stati selezionati dei voli

Post-condizioni:

- È stata creata un'istanza *posto* di *Posto*
- *posto* è stato associato a *listaVoli[indice]*
- Gli attributi *numeroPosto* e *tipoPosto* sono stati aggiornati con *numeroPosto* e *tipoPosto*

Contratto CO3: selezionaBagaglio

Operazione: *selezionaBagaglio (bagaglioScelto:* String, *indice*: int*)*

Riferimenti: caso d'uso: Gestisci prenotazione

Pre-condizioni: Sono stati selezionati i voli e posti

Post-condizioni:

- È stata creata un'istanza *bagaglio* di *Bagaglio*
- *bagaglio* è stato associato a *listaVoli[indice]*
- L'attributo *bagaglioScelto* è stato aggiornato con *bagaglioScelto*

Contratto CO4: confermaPrenotazione

Operazione: *confermaPrenotazione (cliente:* Cliente, *listaVoli*: VoloPrenotato[]*)*

Riferimenti: caso d'uso: Gestisci prenotazione

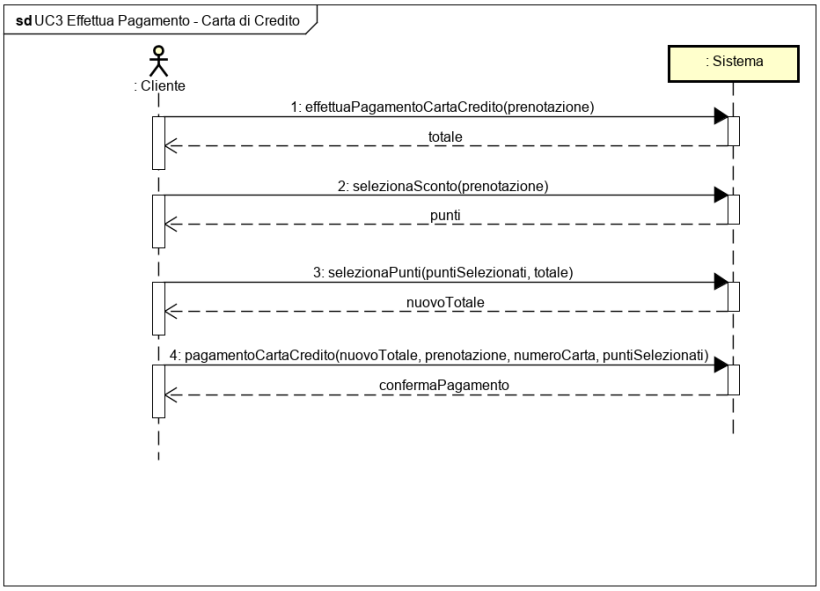
Pre-condizioni: Sono stati selezionati i voli**,** posti e bagagli

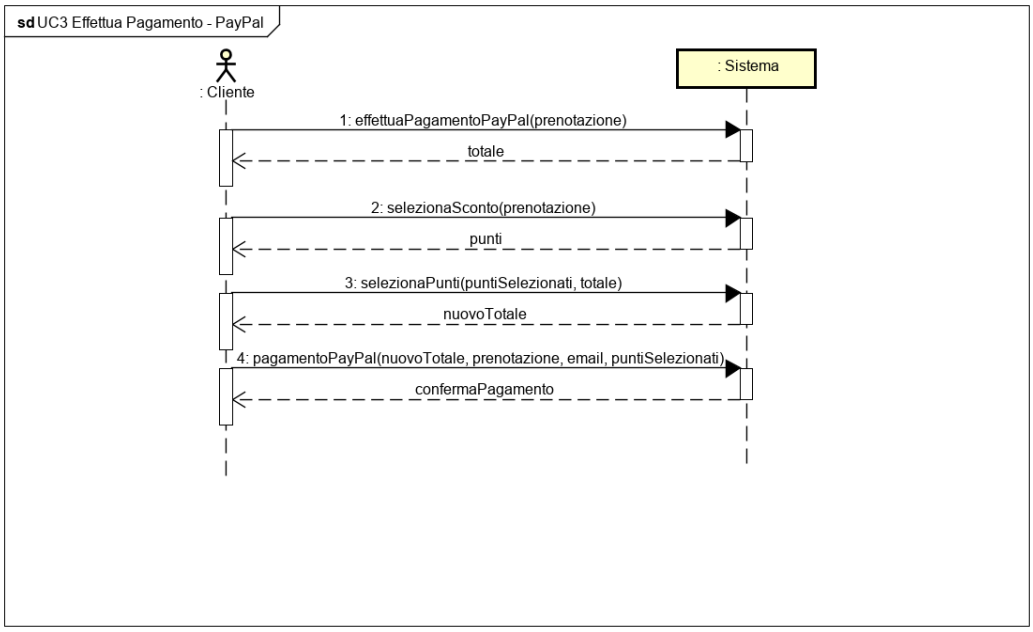
Post-condizioni:

- È stata creata un'istanza *prenotazione* di *Prenotazione*
- *prenotazione* è stata associata a *finchAirlines*
- *cliente* è stato associato a *prenotazione*
- *listaVoli[]* è stato associato a *prenotazione*
- L'attributo *numeroPrenotazione* di *prenotazione* è stato aggiornato
- L'attributo *totale* di *prenotazione* è stato aggiornato

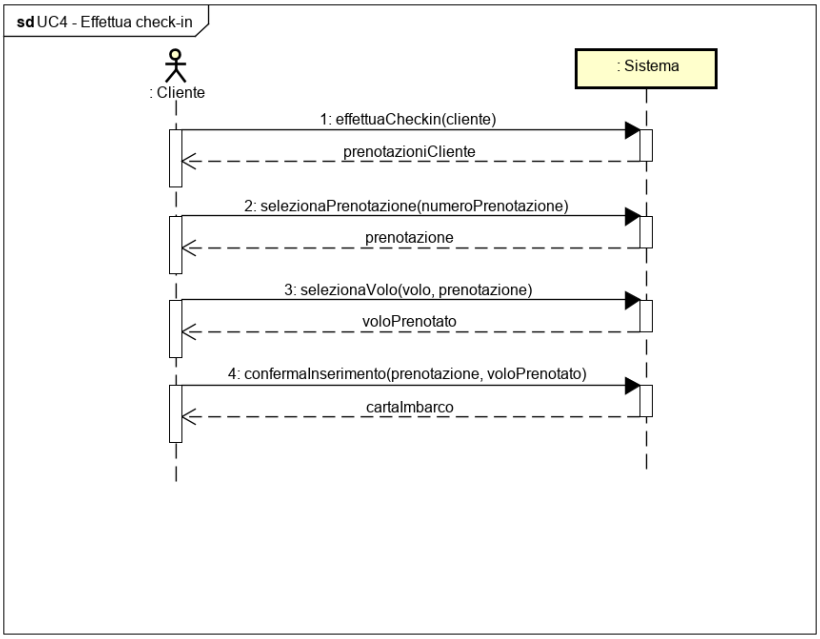
- **Iterazione 2**

Per il caso d'uso UC3 si prendono in considerazione gli scenari alternativi riguardanti le due possibilità per il pagamento di una prenotazione: Carta di Credito e PayPal.





Nel diagramma di sequenza di sistema del caso d'uso UC4 si mette in evidenza l'interazione in termini di operazioni eseguite dal cliente per effettuare il check-in e ottenere la relativa carta di imbarco.



Contratto CO1: effettuaPagamentoCartaDiCredito

Operazione: *effettuaPagamentoCartaDiCredito(prenotazione: Prenotazione)*

Riferimenti: caso d'uso: Effettua pagamento

Pre-condizioni: È stata selezionata una prenotazione

Post-condizioni:

- È stata creata un'istanza di *PagamentoCartaDiCredito*
- È stata associata l'istanza di *PagamentoCartaDiCredito* a *Prenotazione*
- È stata creata un'istanza di *CartaCreditoAdapter*
- È stata associata l'istanza di *CartaCreditoAdapter* a *PagamentoCartaDiCredito*

Contratto CO2: effettuaPagamentoPaypal

Operazione: *effettuaPagamentoPaypal(prenotazione: Prenotazione)*

Riferimenti: caso d'uso: Effettua pagamento

Pre-condizioni: È stata selezionata una prenotazione

Post-condizioni:

- È stata creata un'istanza di *PagamentoPaypal*
- È stata associata l'istanza di *PagamentoPayPal* a *Prenotazione*
- È stata creata un'istanza di *PayPalAdapter*
- È stata associata l'istanza di *PayPalAdapter* a *PagamentoPayPal*

Contratto CO3: pagamentoCartaCredito

Operazione: *pagamentoCartaCredito(nuovoTotale: double, prenotazione: Prenotazione, numeroCarta: String, puntiSelezionati: int)*

Riferimenti: caso d'uso: Effettua pagamento

Pre-condizioni: È stato selezionato il metodo di pagamento

Post-condizioni:

- L'attributo *Pagato* di *Prenotazione è stato aggiornato*
- L'attributo *punti* di *Cliente* è stato aggiornato
- È stata creata un'istanza di *CartaDiCredito*
- È stata associata l'istanza di *CartaDiCredito* a *PagamentoCartaDiCredito*

Contratto CO4: pagamentoPaypal

Operazione: *pagamentoPaypal(nuovoTotale: double, prenotazione: Prenotazione, email: String, puntiSelezionati: int)*

Riferimenti: caso d'uso: Effettua pagamento

Pre-condizioni: È stato selezionato il metodo di pagamento

Post-condizioni:

- L'attributo *pagato* di *Prenotazione è stato aggiornato*
- L'attributo *punti* di *Cliente* è stato aggiornato
- È stata creata un'istanza di *ContoPayPal*
- È stata associata l'istanza di *ContoPayPal* a *PagamentoPayPal*

Contratto CO5: confermaInserimento

Operazione: *confermaInserimento*(*prenotazione*: Prenotazione, *voloPrenotato*: VoloPrenotato)

Riferimenti: caso d'uso: Effettua check-in

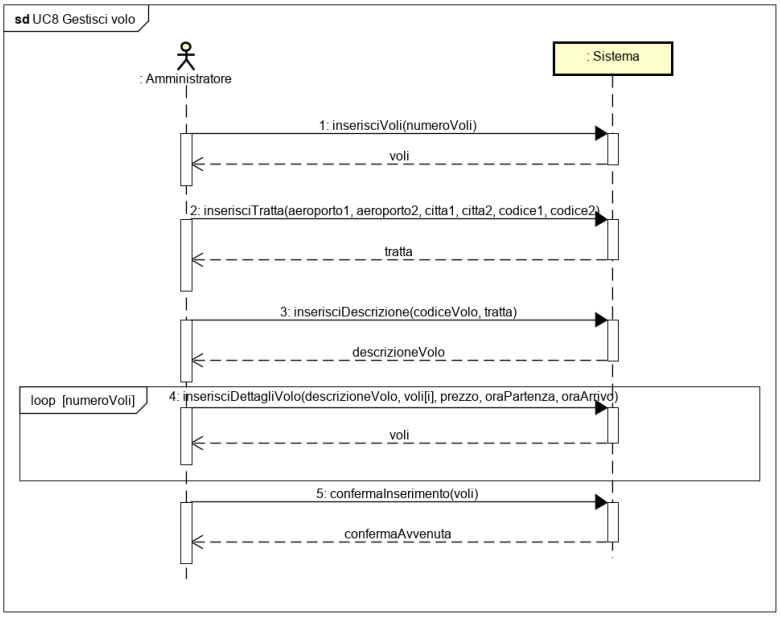
Pre-condizioni: È stato selezionato un volo prenotato

Post-condizioni:

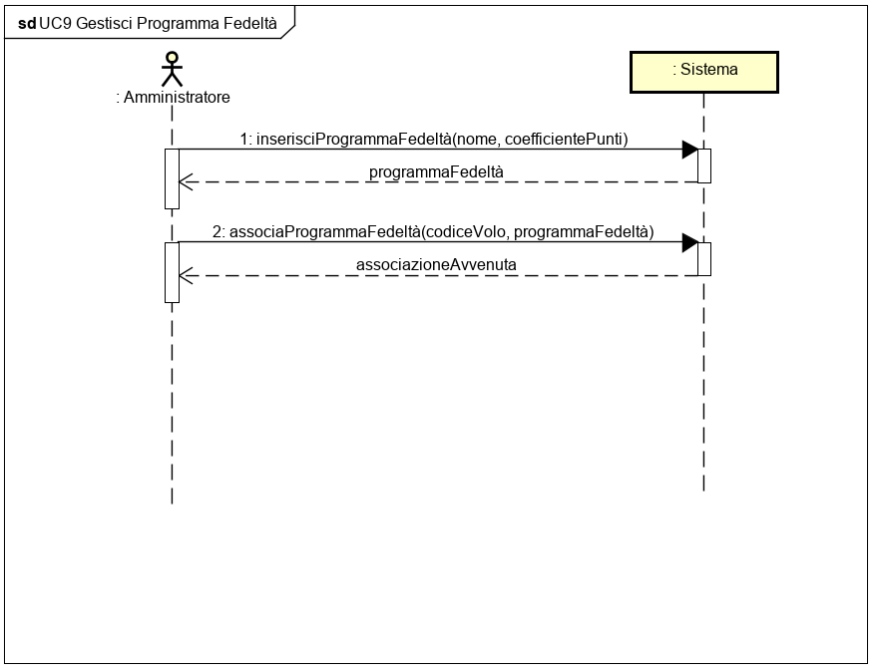
- È stata creata un'istanza *CartaDiImbarco*
- *prenotazione* e *voloPrenotato* sono state associate a *CartaDiImbarco*
- L'attributo *checkin* di *VoloPrenotato* è stato aggiornato

- **Iterazione 3**

Il diagramma di sequenza di sistema del caso d'uso UC8 mette in risalto l'interazione tra un altro attore, l'amministratore, e il sistema nelle operazioni di gestione di un volo. In particolare, per la creazione di un nuovo volo acquistabile dai clienti e inseribile nel sistema.



Si rappresenta anche l'interazione tra Amministratore e Sistema per la creazione di un nuovo programma fedeltà da inserire nel sistema e associare ad un volo identificato da un particolare numero di volo (Descrizione Volo).



Contratto CO1: inserisciVoli

Operazione: *inserisciVoli(numeroVoli: int)*

Riferimenti: caso d'uso: Gestisci Volo

Pre-condizioni: Nessuna

Post-condizioni:

- È stata creata un'istanza o più istanze di *Volo*

Contratto CO2: inserisciTratta

Operazione: *inserisciTratta(aeroporto1: String, aeroporto2: String, citta1: String, citta2: String, codice1: String, codice2: String)*

Riferimenti: caso d'uso: Gestisci Volo

Pre-condizioni: È stata creata una o più istanze di volo

Post-condizioni:

- Sono state create due istanze di *Aeroporto*
- Gli attributi *nome, città e codice* di *Aeroporto* sono stati aggiornati

Contratto CO3: inserisciDescrizione

Operazione: *inserisciDescrizione(codiceVolo: String, tratta[]: Aeroporto)*

Riferimenti: caso d'uso: Gestisci Volo

Pre-condizioni: È stata inserita una tratta

Post-condizioni:

- È stata creata un'istanza di *DescrizioneVolo*
- L'attributo *codice* di *DescrizioneVolo* è stato aggiornato
- Sono state associate le istanze di *Aeroporto* a *DescrizioneVolo*

Contratto CO4: inserisciDettagliVolo

Operazione: *inserisciDettagliVolo(descrizioneVolo: DescrizioneVolo, voli[i]: Volo, prezzo: double, oraPartenza: Date, oraArrivo: Date)*

Riferimenti: caso d'uso: Gestisci Volo

Pre-condizioni: È stata inserita una descrizione di un volo

Post-condizioni:

- L'istanza di *DescrizioneVolo* è stata associata all'istanza *voli[i]* di *Volo*
- Gli attributi *oraPartenza, oraArrivo, prezzo* di *Volo* sono stati aggiornati

Contratto CO5: confermaInserimento

Operazione: *confermaInserimento(voli: ArrayList<Volo>)*

Riferimenti: caso d'uso: Gestisci Volo

Pre-condizioni: Sono stati inseriti i dettagli del volo creato

Post-condizioni:

- Ogni istanza di *Volo* è stata associata a *FinchAirlines*

Contratto CO6: inserisciProgrammaFedeltà

Operazione: *inserisciProgrammaFedeltà(nome: String, coefficientePunti: double)*

Riferimenti: caso d'uso: Gestisci Programma Fedeltà

Pre-condizioni: Deve esistere almeno un volo nel sistema

Post-condizioni:

- È stata creata un'istanza di *ProgrammaFedeltà*
- È stata creata un'istanza di *DescrizioneProgrammaFedeltà*
- L'attributo *nome* di *ProgrammaFedeltà* è stato aggiornato
- L'attributo *coefficientePunti* di *DescrizioneProgrammaFedeltà* è stato aggiornato
- L'istanza di *DescrizioneProgrammaFedeltà* è stata associata a *ProgrammaFedeltà*

Contratto CO7: associaProgrammaFedeltà

Operazione: a*ssociaProgrammaFedeltà(codiceVolo: String, programmaFedeltà: ProgrammaFedeltà)*

Riferimenti: caso d'uso: Gestisci Programma Fedeltà

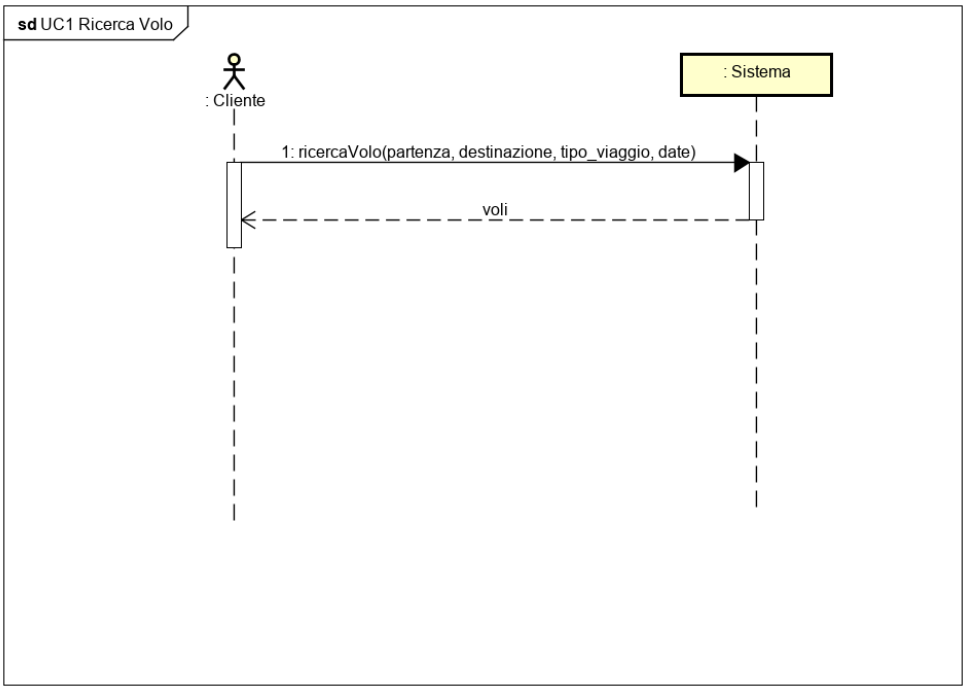
Pre-condizioni: È stato creato un programma fedeltà

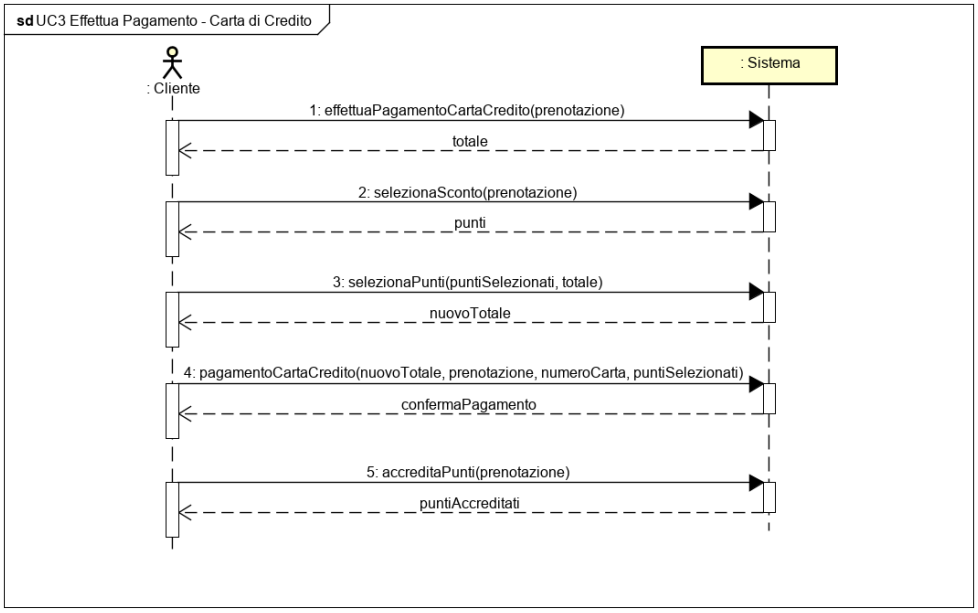
Post-condizioni:

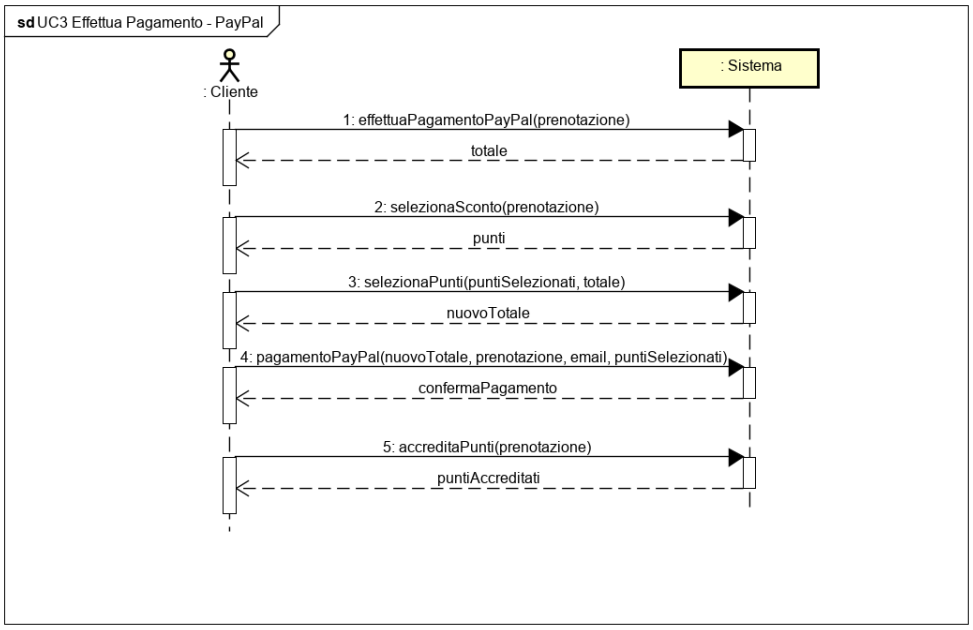
- L' istanza di *ProgrammaFedeltà* è stata associata a *DescrizioneVolo*

- **Iterazione 4**

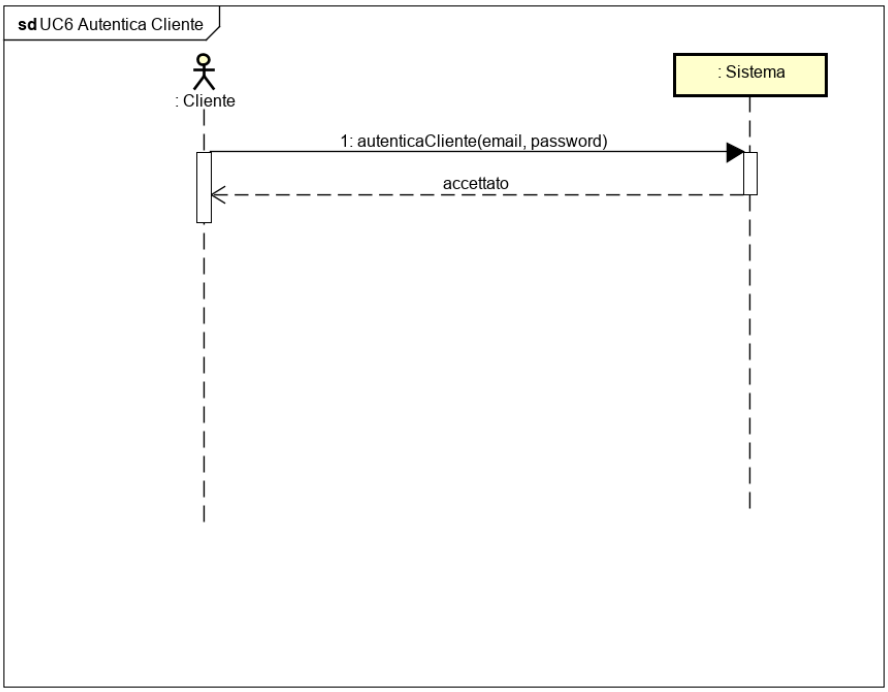
I primi due SSD mostrati di seguito sono relativi a due casi d'uso già analizzati e implementati in precedenza: UC1 e UC3. In particolare, li si prende nuovamente in considerazione al fine di aggiungere delle modifiche che possano rendere coerente il tutto allo stato di evoluzione del sistema alla presente iterazione. Nel caso d'uso UC1 viene rimossa l'operazione di autenticazione, in quanto già eseguita nel caso d'uso UC6 modellato nella presente iterazione. Nel caso d'uso UC3 viene aggiunta una nuova operazione, *accreditaPunti(prenotazione)*, che permette di accreditare dei punti ad un Cliente nel momento in cui effettua una prenotazione (e il relativo pagamento) che prevede un programma fedeltà.

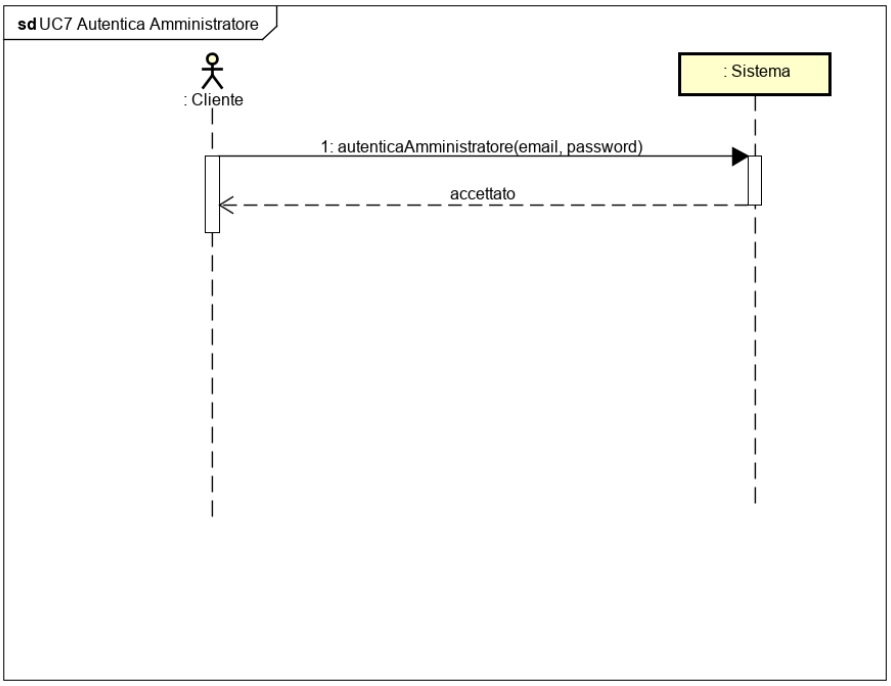






Gli SSD dei casi d'uso UC6 e UC7 mostrano la stessa tipologia di interazione tra il Sistema e due attori differenti: il Cliente e l'Amministratore per l'esecuzione delle operazioni di autenticazione.





Contratto CO1: accreditaPunti

Operazione: *accreditaPunti(prenotazione: Prenotazione)*

Riferimenti: caso d'uso: Effettua Pagamento

Pre-condizioni: È stato effettuato il pagamento di una prenotazione

Post-condizioni:

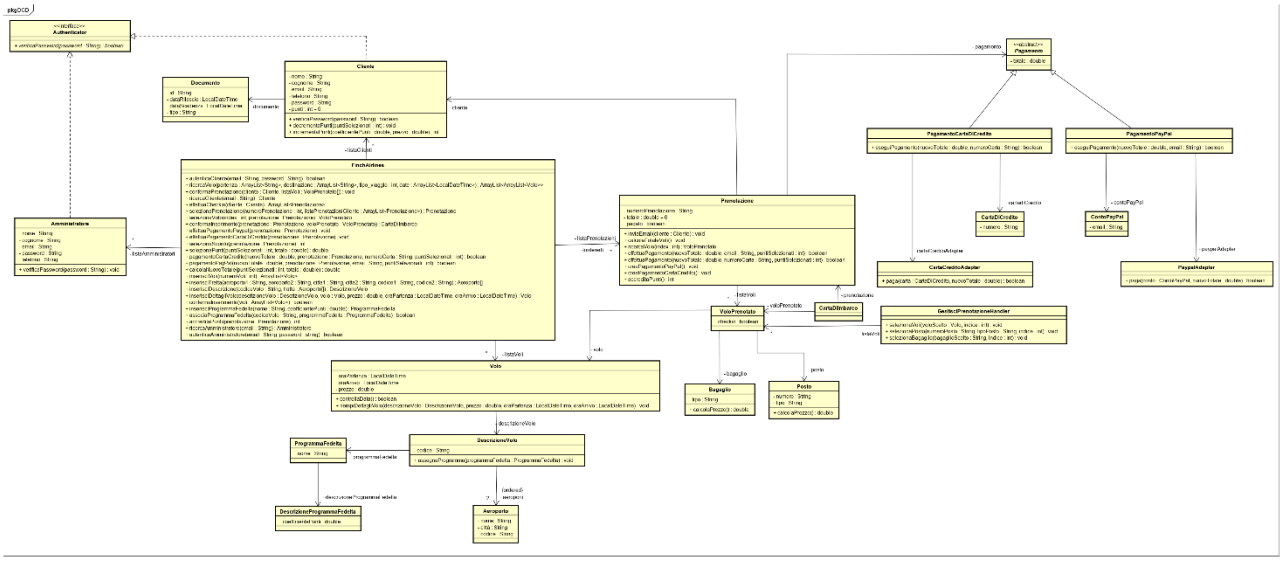
- L'attributo *punti* di *Prenotazione* è stato aggiornato

3. Progettazione

Diagramma delle Classi

Una ulteriore fase eseguita nello sviluppo del presente progetto è quella della Progettazione Orientata agli oggetti. Questa viene fatta al fine di mostrare quelli che sono gli oggetti software definibili a partire dagli elaborati della precedente fase di Analisi. Gli oggetti software sono gli elementi costituenti il sistema e tale fase di sviluppo ha come obiettivo la loro definizione e quella delle loro proprietà e responsabilità, attraverso il Diagramma delle Classi di Progetto (DCD). Un ulteriore elaborato, realizzato parallelamente al DCD, è quello dei Diagrammi di Sequenza, con l'obiettivo di mostrare le interazioni tra le varie classi software in corrispondenza delle singole operazioni di sistema.

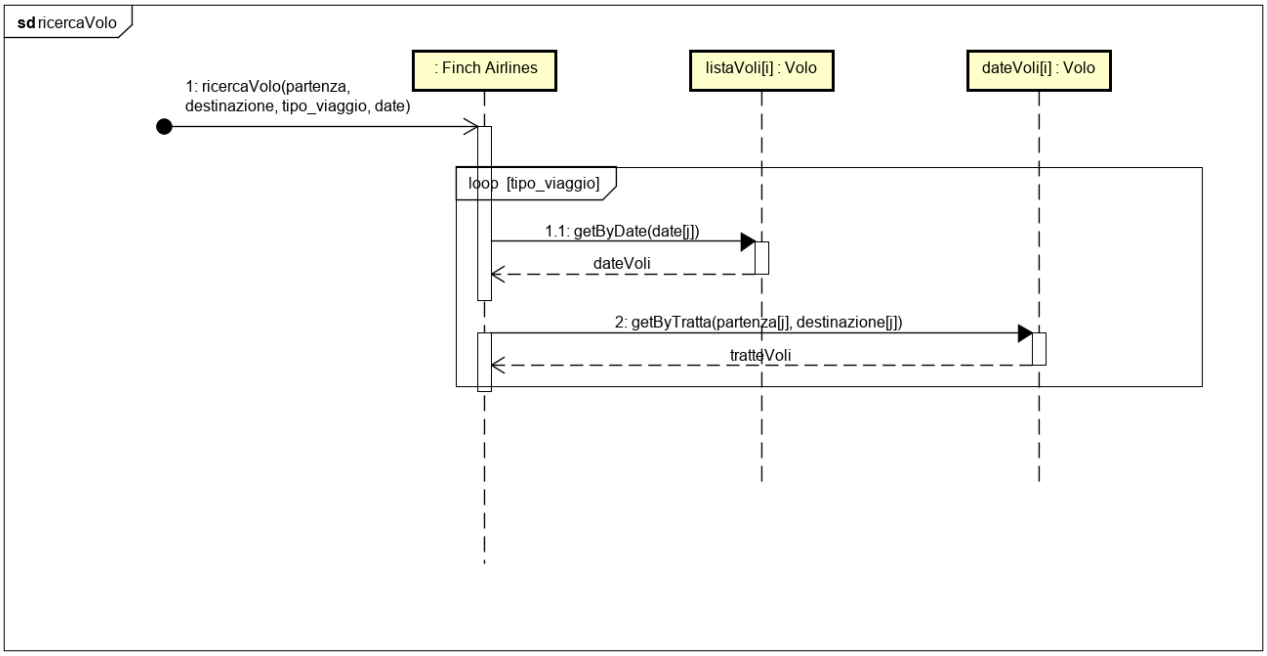
Il DCD finale è quindi il seguente:



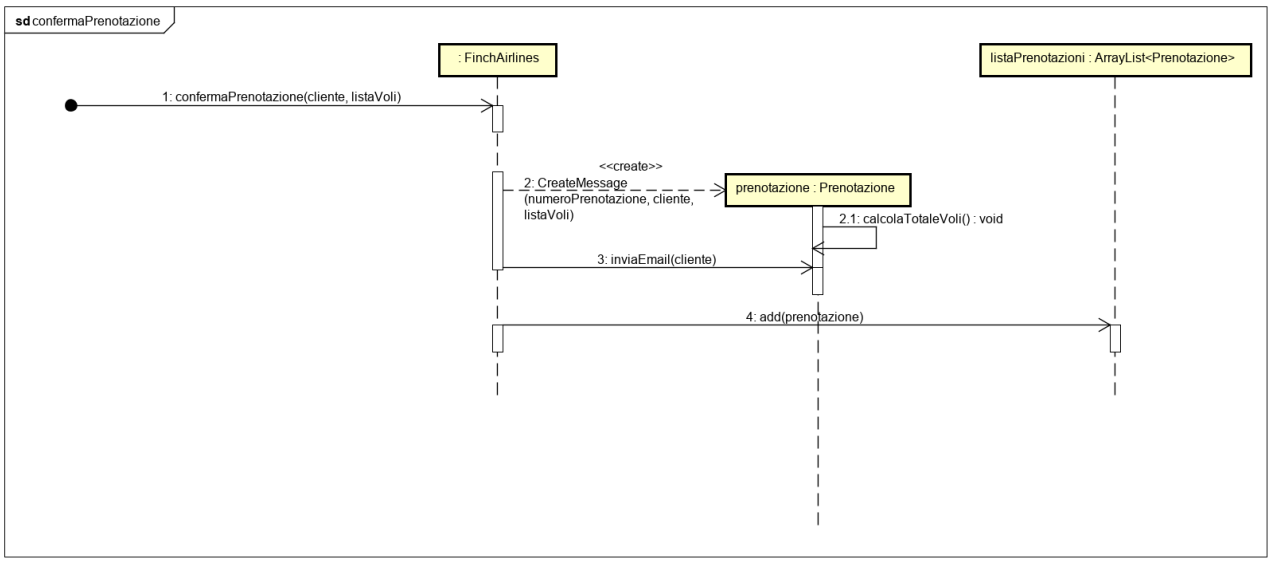
Diagrammi di Sequenza

Qui di seguito vengono mostrati i diagrammi di sequenza delle operazioni di sistema più rilevanti. Per visionarli tutti si veda i file .astah presenti all'interno delle cartelle delle singole iterazioni.

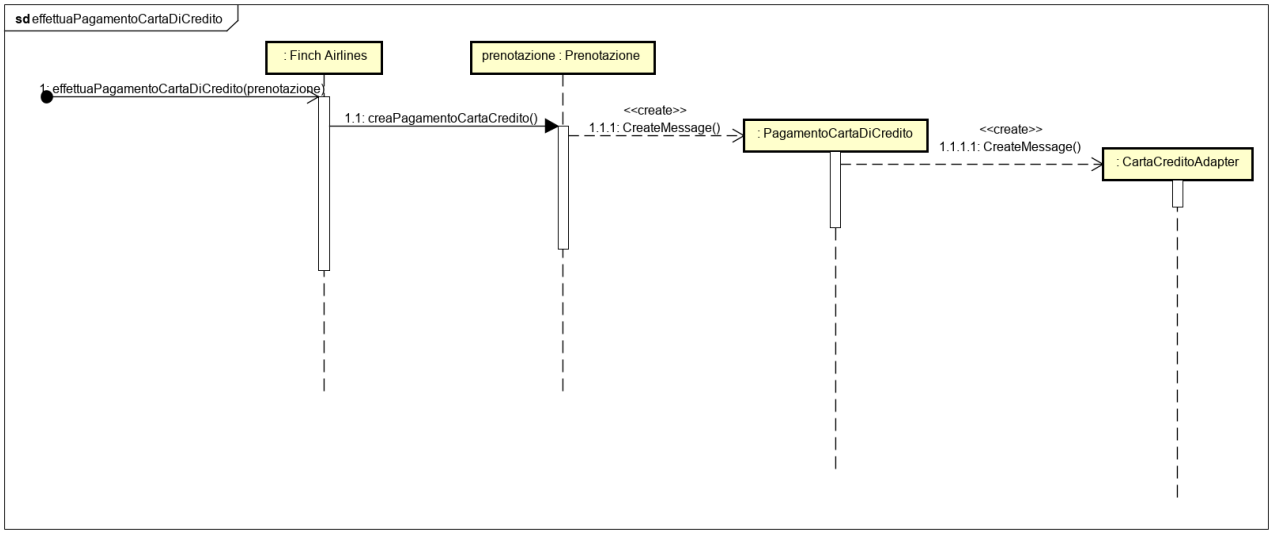
- Ricerca di voli date le informazioni di interesse per il cliente



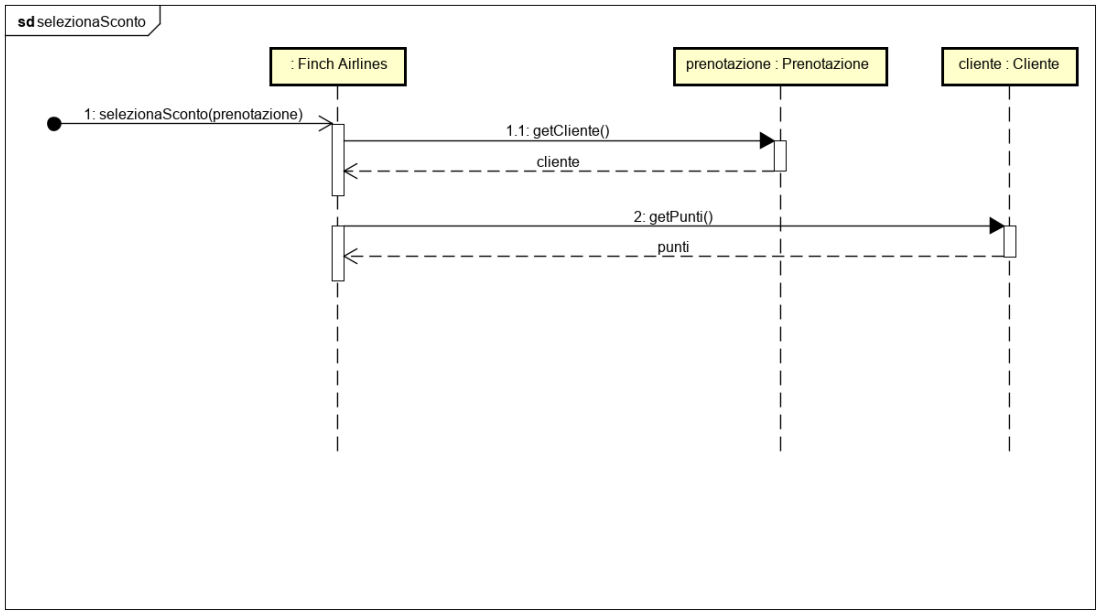
- Conferma e creazione della prenotazione effettuata dal cliente



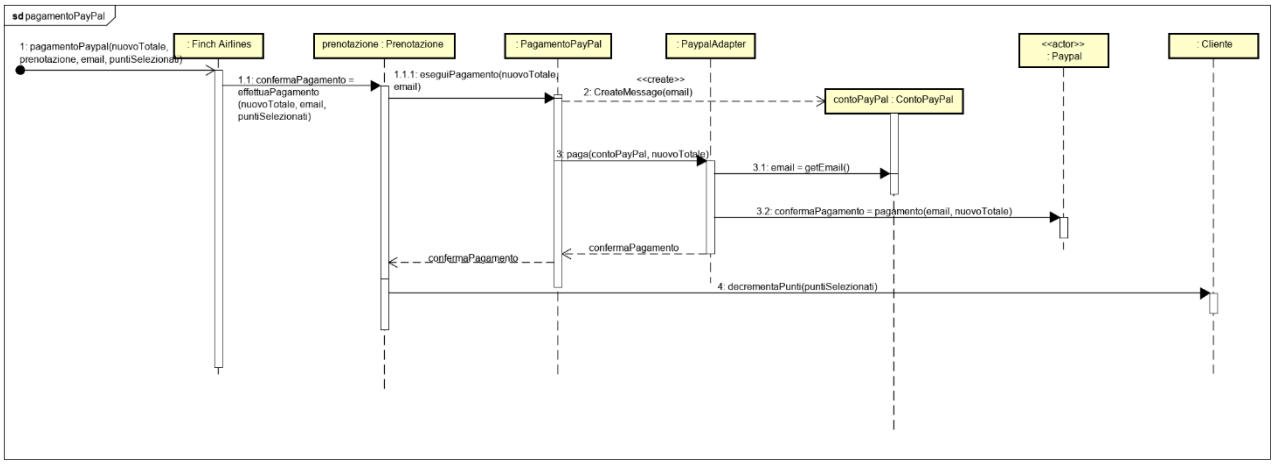
- Creazione di un pagamento con Carta di Credito



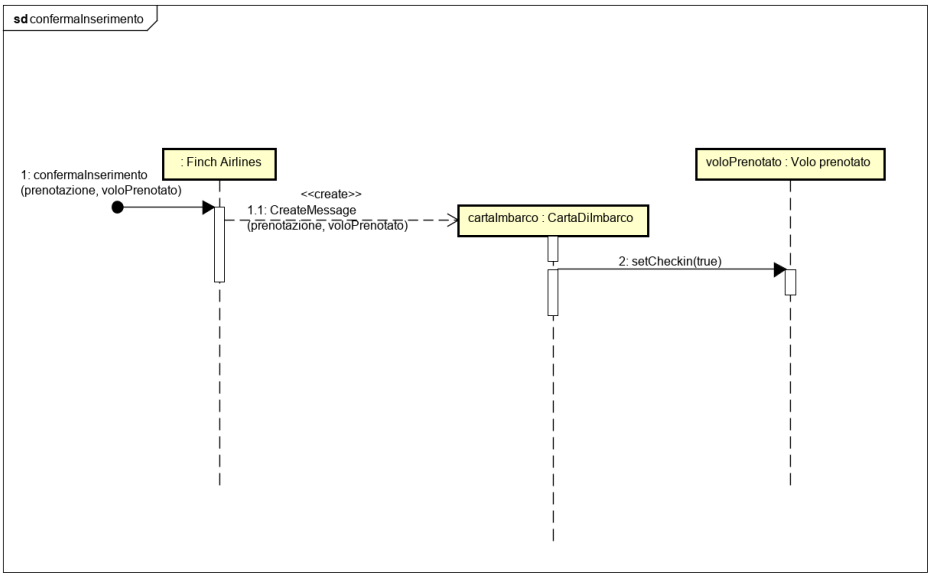
- Lettura dei punti posseduti da un Cliente



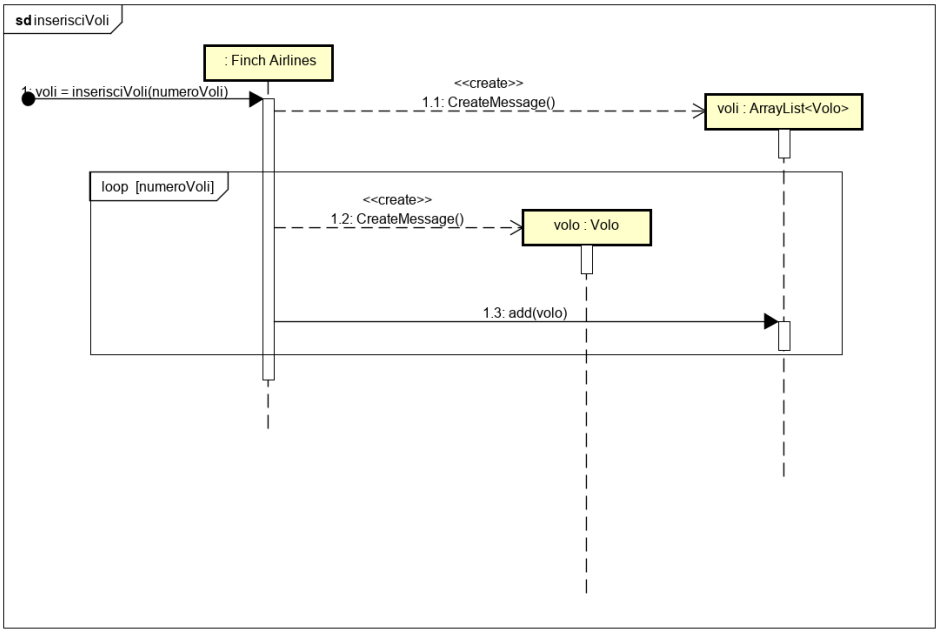
- Completamento del pagamento della prenotazione presso il servizio esterno con PayPal



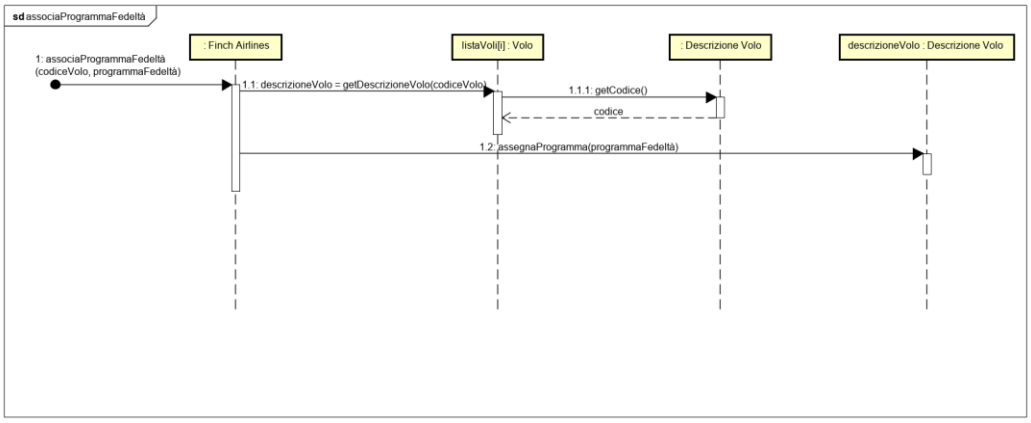
- Creazione della carta di imbarco e conferma della operazione di check-in effettuato



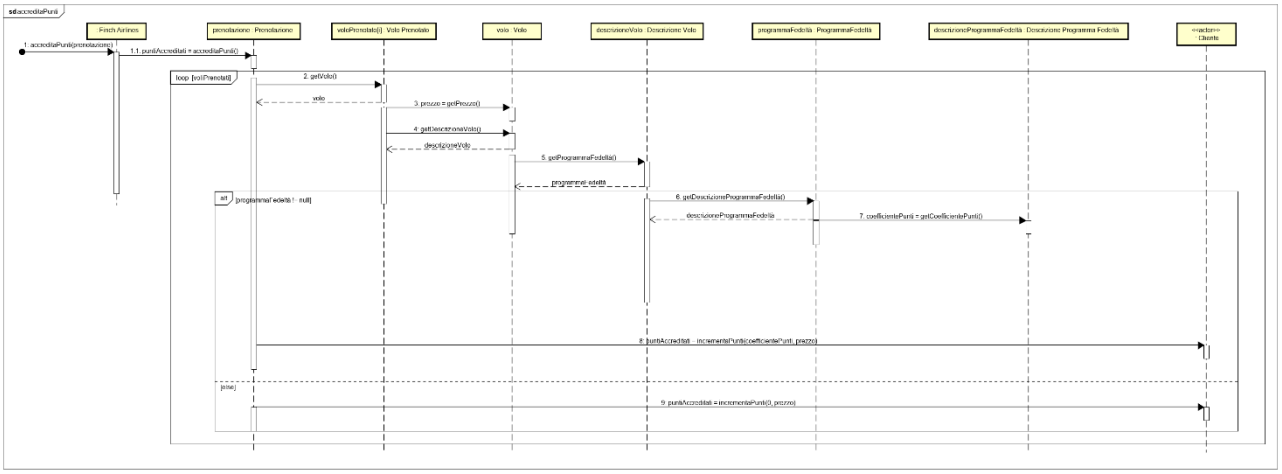
- Creazione di un nuovo volo o di un set di nuovi voli (per voli periodici)



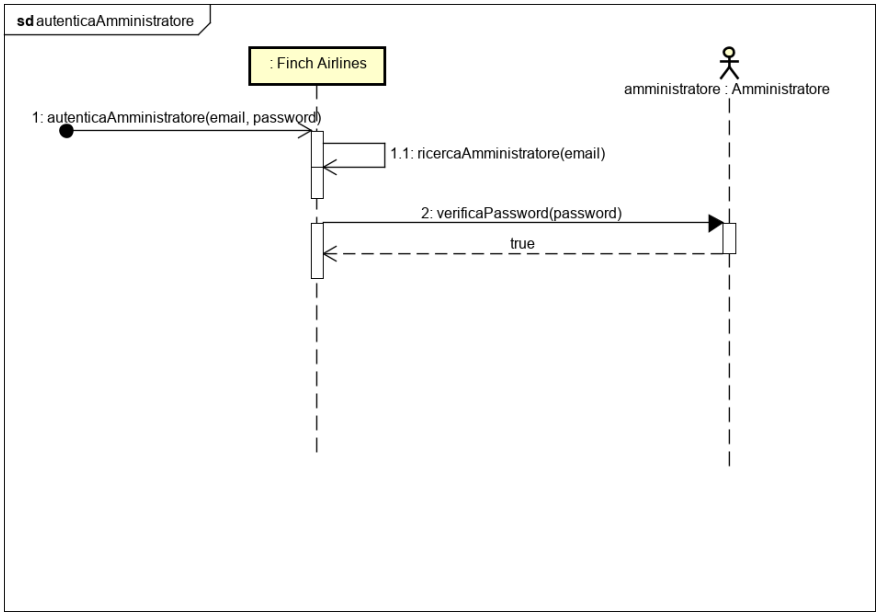
- Associazione del Programma Fedeltà creato ad un volo identificato mediante Codice Volo



- Accredito dei punti al Cliente per ciascun volo della prenotazione che presenta Programma Fedeltà



- Ricerca dell'Amministratore mediante e-mail (identificazione) e verifica della password (autenticazione)



4. Testing

Introduzione

Il testing è una delle fasi fondamentali eseguite in coda alla singola iterazione di un processo di sviluppo software. Il suo obiettivo è quello di verificare che il software funzioni correttamente, ovvero in accordo con le specifiche dettate dal committente.

Un buon processo di sviluppo software richiede l'esecuzione di un elevato numero di test al fine di rilevare tutti i possibili malfunzionamenti presenti al suo interno, tuttavia l'esecuzione di tale elevato numero di operazioni richiederebbe troppo tempo. A tal fine, quindi, si opta per una buona progettazione di testing, ovvero una scelta accurata e oculata delle componenti che, se testate, contribuiscono ad un funzionamento del software complessivamente ottimale.

In modo specifico i test che è possibile effettuare sono:

- **Test Unitari**: quelli eseguiti all'interno del presente progetto e che hanno l'obiettivo di andare a testare singole unità di codice al fine di rilevare malfunzionamenti. Quello che tipicamente viene fatto è quindi l'esecuzione di alcuni metodi presenti nel codice fornendo in ingresso dati tali da testare tutti i loro possibili comportamenti. Questi vengono poi comparati con i valori attesi per quelle date combinazioni di dati in input. In modo specifico, all'interno del progetto si fa uso del framework *JUnit* che, in un processo di sviluppo software in Java, permette l'esecuzione di test in modo automatizzato.
- **Test di integrazione**: test eseguiti per verificare il funzionamento complessivo del sistema. In questo tipo di test si valuta una caratteristica completa del sistema anziché la singola unità. Il testing verte quindi su sottoinsiemi via via crescenti in modo tale da verificare la loro corretta interazione.
- **Test di sistema:** un ulteriore soprainsieme di test è quello di Sistema, che comprende tutti i casi di test esposti precedentemente e che mira a convalidare il rispetto dei requisiti funzionali e non funzionali. In particolare, si verifica che il sistema funzioni correttamente lanciando manualmente l'applicazione in un normale processo di esecuzione.
- **Test di accettazione:** altra tipologia di test eseguita al fine di simulare il comportamento dell'applicazione finale quando utilizzata dagli attori a cui è destinata (Cliente e Amministratore).

Testing Unitario

Per l'esecuzione di questo tipo di test è necessario individuare un insieme di classi, quindi di metodi da testare. In particolare, si è scelto di testare i metodi appartenenti a classi legate alle operazioni di calcolo. Per questo motivo, allora, si individua un insieme di valori da impiegare per il testing e raggruppabili all'interno di **classi di equivalenza**. Si ha allora:

- Bagaglio
    - CalcolaPrezzo
        - È stata scelta l'opzione con bagaglio "Mano" (a mano)
        - È stata scelta l'opzione con bagaglio "10kg"
        - È stata scelta l'opzione con bagaglio "20kg"
        - È stata scelta l'opzione con bagaglio identificato da una stringa errata
- Posto
    - CalcolaPrezzo

        - È stata scelta l’opzione con posto “base” (a mano)
        - È stata scelta l’opzione con posto “premium”
        - È stata scelta l’opzione con posto “optimum”
        - È stata scelta l’opzione con posto identificato da una stringa errata
- Prenotazione
    - CalcolaTotaleVoli
        - Viene inserito un insieme di voli con informazioni corrette (prezzo volo positivo, bagaglio e posto corretti)
        - Viene inserito un insieme di voli con informazioni non corrette (prezzo volo positivo, bagaglio e posto non corretti)
    - AccreditaPunti
        - È presente una programma fedeltà per il volo dato
        - Non è presente un programma fedeltà per il volo dato
- FinchAirlins
    - CalcolaNuovoTotale
        - Viene calcolato il nuovo totale con punti positivi
        - Viene calcolato il nuovo totale con punti negativi
        - Viene calcolato il nuovo totale in presenza di un totale di partenza negativo o nullo
- Cliente
    - IncrementaPunti
        - Il coefficiente di calcolo dei punti e il prezzo sono positivi
        - Il coefficiente di calcolo dei punti e/o il prezzo sono negativi
    - DecrementaPunti
        - I punti da decrementare sono positivi e maggiori dei punti attualmente posseduti dal cliente
        - I punti da decrementare sono positivi e minori o uguali dei punti attualmente posseduti dal cliente
        - I punti da decrementare sono negativi

Test di Sistema

Da questo particolare tipo di test, eseguito per ogni iterazione sull'applicativo finito, non sono emerse criticità rilevanti. Tuttavia, laddove presenti, sono stati utili ad individuare le singole unità da verificare ed eventualmente correggere. Una caratteristica emersa da tali test è il fatto che alcune incongruenze sono dovute al fatto che per una buona parte dei casi d'uso non sono stati fattivamente implementati eventuali scenari alternativi.

5. Refactoring e Conclusioni

Database e Refactoring

In ultima istanza si è prevista la possibilità di integrare il software sviluppato con un database che potesse gestire la permanenza dei dati del sistema. In particolare, si è fatto uso di un database locale MySQL con interfacciamento mediante la libreria *java.sql*. Si è inoltre effettuata una operazione di miglioramento di presentazione delle informazioni su un'interfaccia a riga di comando. Infine, sono state corrette tutte le anomalie messe in risalto dai Test di Sistema.

Test di accettazione

Il Test di accettazione, descritto nella sezione precedente, è stato eseguito come fase ultima e conclusiva, simulando il comportamento del Cliente o dell'Amministratore del Sistema nell'utilizzo dell'applicazione per l'esplicazione delle varie funzioni (es. acquisto di un volo, inserimento di un programma fedeltà, ecc.). Tale test ha permesso la verifica della capacità del sistema a portare a termine le operazioni descritte nei casi d'uso.