

5. Elaborazione – Iterazione 4

Introduzione

In questa iterazione si affronta la modellazione dei seguenti casi 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.
- 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.

Inoltre, si aggiornano i seguenti casi 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.
- UC3: Effettua Pagamento, viene modificato per includere la funzionalità di assegnazione punti del programma fedeltà all'atto dell'acquisto.

Aggiornamento casi d'uso UC1: Ricerca Volo, UC3: Effettua Pagamento e UC7: Autentica Amministratore

UC1: Ricerca volo

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

Scenari alternativi:

*a. In qualsiasi momento il sistema fallisce

1. Il sistema avvisa il cliente dell'errore

- 2. Il sistema ritorna alla schermata di ricerca iniziale
- 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

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

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

Analisi Orientata agli Oggetti

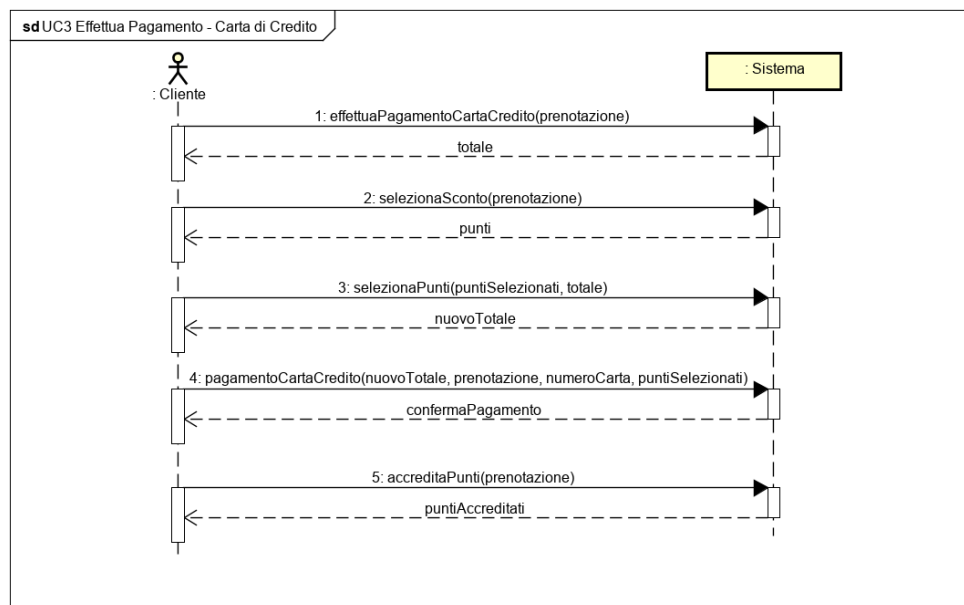
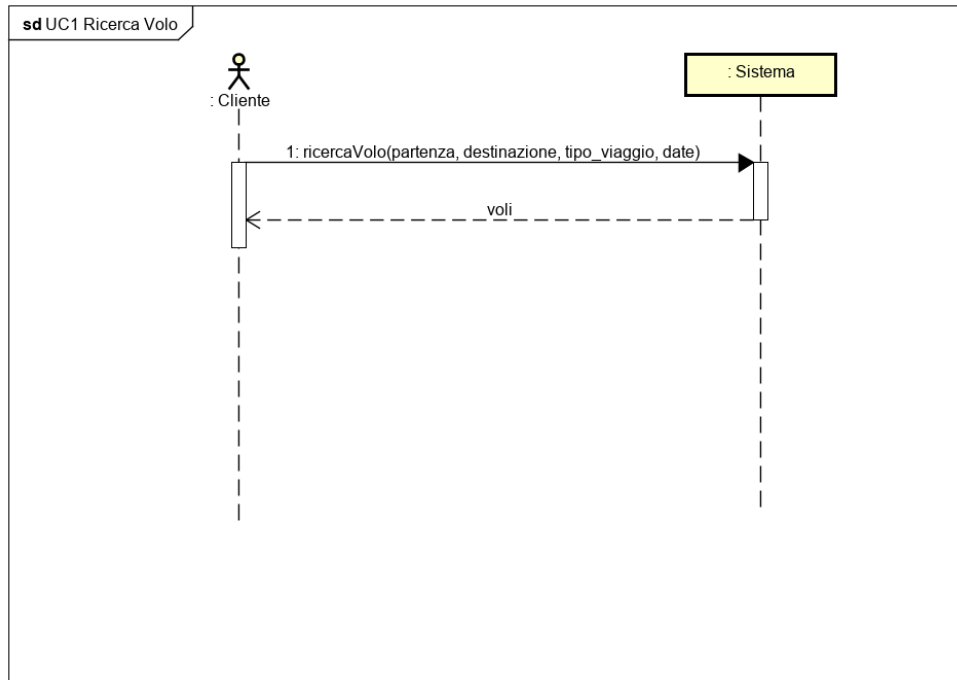
Modello di Dominio

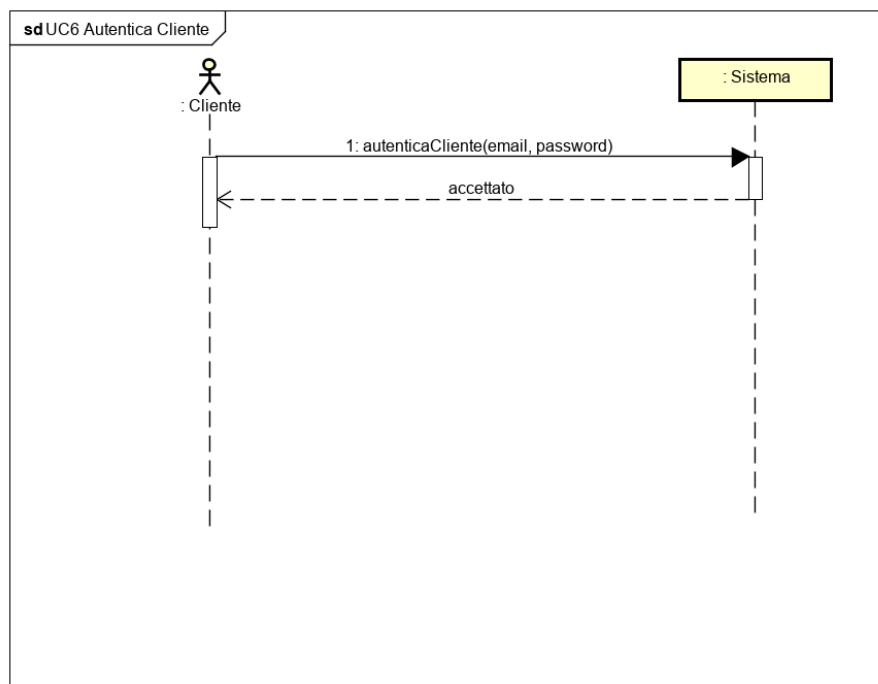
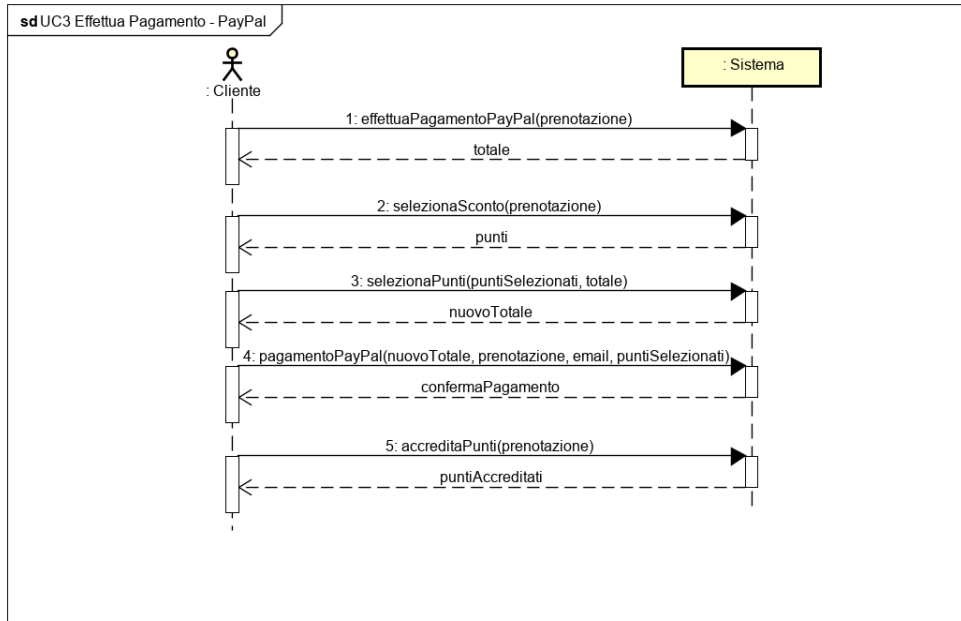
In questa iterazione non è prevista alcuna aggiunta al modello di dominio.

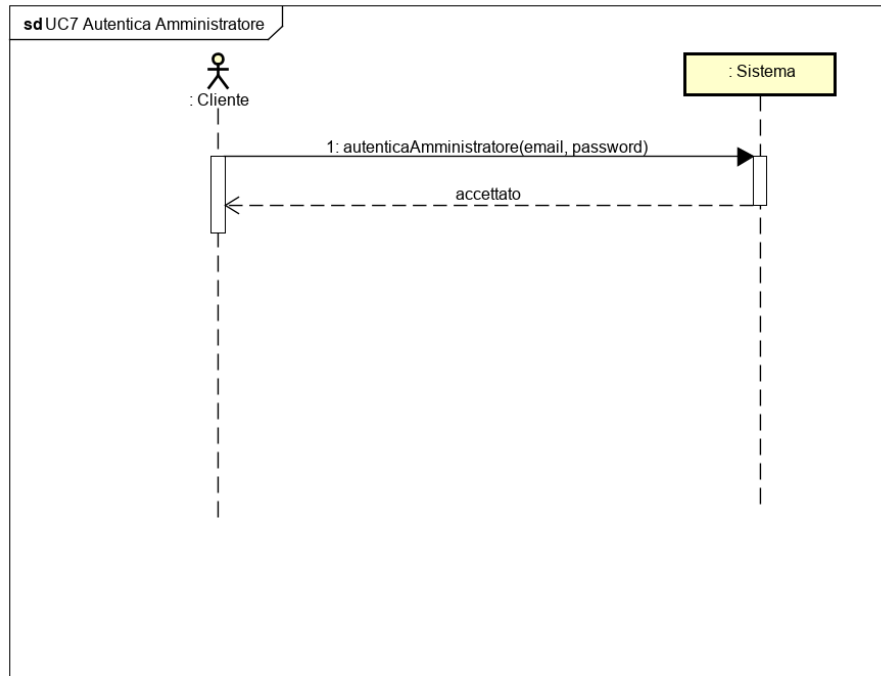
Diagrammi di Sequenza di Sistema

In questa sezione vengono inseriti i Diagrammi di Sequenza di Sistema (SSD) dei casi d'uso *UC1: Ricerca volo* e *UC3: Effettua pagamento*, *UC6: Autentica Cliente* e *UC7: Autentica Amministratore*.

Si ricorda che gli SSD dei casi d'uso UC1 e UC3 sono stati già realizzati nelle precedenti iterazioni, quindi i diagrammi di seguito rappresentati ne presentano solo le modifiche effettuate.







Contratti delle operazioni

I contratti delle operazioni vengono descritti per il solo caso d'uso *UC3: Effettua Pagamento* relativamente alle operazioni di sistema aggiunte.

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:

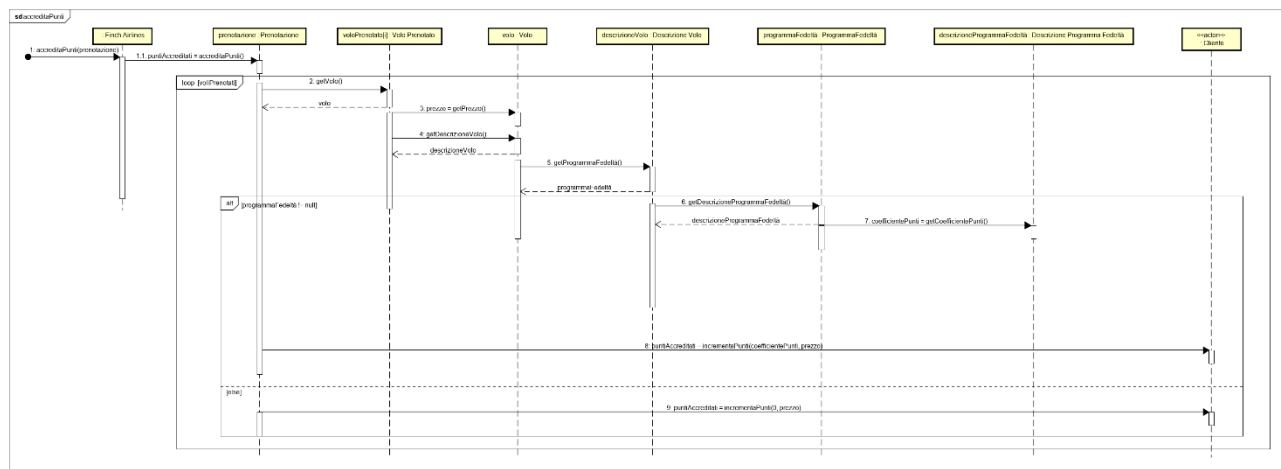
- L'attributo *punti* di *Prenotazione* è stato aggiornato

Progettazione Orientata agli Oggetti

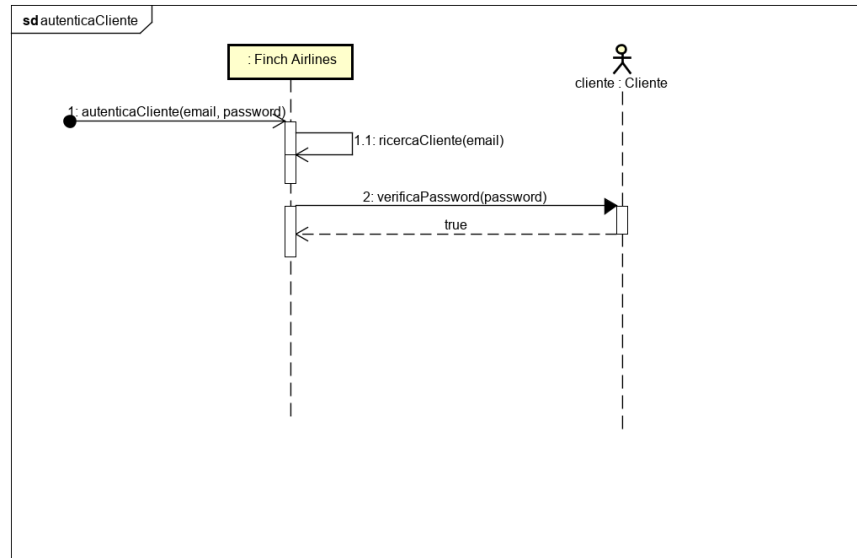
Di seguito vengono rappresentati i diagrammi di interazione e delle classi di progetto (DCD) relativi ai casi d'uso *UC3: Effettua Pagamento*, *UC6: Autentica Cliente* e *UC7: Autentica Amministratore*.

Diagrammi di sequenza

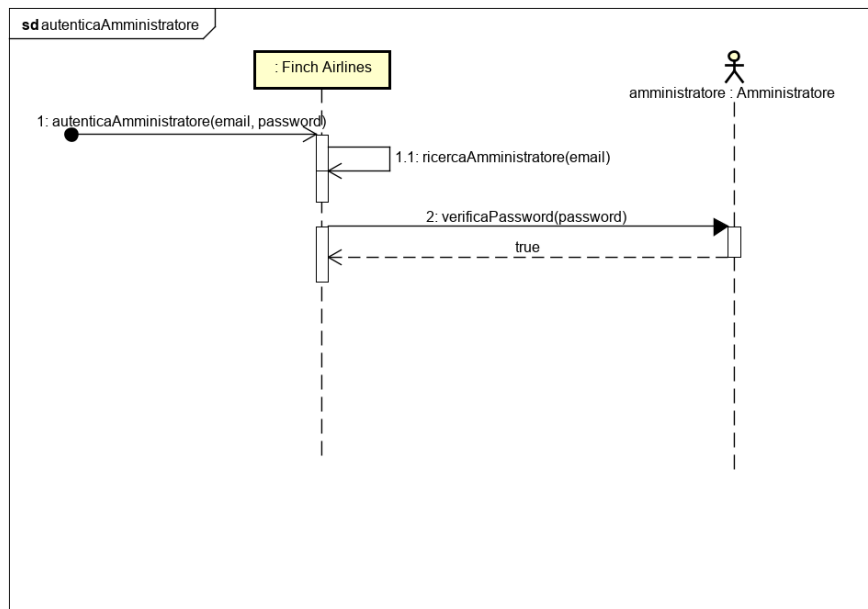
- Accredito dei punti al Cliente per ciascun volo della prenotazione che presenta Programma Fedeltà



- Ricerca del Cliente mediante e-mail (identificazione) e verifica della password (autenticazione)



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




```

classDiagram
    class Client {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class Documents {
        +id: String
        +idForm: LocalizedText
        +description: LocalizedText
        +type: String
    }
    class Anamnesis {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class Prescriptions {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class Payments {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class CardCredit {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class CardDebit {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class CashCredit {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class CashDebit {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class Voucher {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class Receipt {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }
    class ReceiptDebit {
        -name: String
        -age: String
        -sex: String
        -password: String
        +login: String
        +logout: String
        +last: String
    }

    Client "1" -- "2" Documents : documents
    Client "1" -- "2" Anamnesis : anamnesis
    Client "1" -- "2" Prescriptions : prescriptions
    Client "1" -- "2" Payments : payments
    Client "1" -- "2" CardCredit : cardCredit
    Client "1" -- "2" CardDebit : cardDebit
    Client "1" -- "2" CashCredit : cashCredit
    Client "1" -- "2" CashDebit : cashDebit
    Client "1" -- "2" Voucher : voucher
    Client "1" -- "2" Receipt : receipt
    Client "1" -- "2" ReceiptDebit : receiptDebit

    Documents --> Anamnesis : documents
    Documents --> Prescriptions : prescriptions
    Documents --> Payments : payments
    Documents --> CardCredit : cardCredit
    Documents --> CardDebit : cardDebit
    Documents --> CashCredit : cashCredit
    Documents --> CashDebit : cashDebit
    Documents --> Voucher : voucher
    Documents --> Receipt : receipt
    Documents --> ReceiptDebit : receiptDebit

    Anamnesis --> Prescriptions : anamnesis
    Anamnesis --> Payments : payments
    Anamnesis --> CardCredit : cardCredit
    Anamnesis --> CardDebit : cardDebit
    Anamnesis --> CashCredit : cashCredit
    Anamnesis --> CashDebit : cashDebit
    Anamnesis --> Voucher : voucher
    Anamnesis --> Receipt : receipt
    Anamnesis --> ReceiptDebit : receiptDebit

    Prescriptions --> Payments : prescriptions
    Prescriptions --> CardCredit : cardCredit
    Prescriptions --> CardDebit : cardDebit
    Prescriptions --> CashCredit : cashCredit
    Prescriptions --> CashDebit : cashDebit
    Prescriptions --> Voucher : voucher
    Prescriptions --> Receipt : receipt
    Prescriptions --> ReceiptDebit : receiptDebit

    Payments --> CardCredit : cardCredit
    Payments --> CardDebit : cardDebit
    Payments --> CashCredit : cashCredit
    Payments --> CashDebit : cashDebit
    Payments --> Voucher : voucher
    Payments --> Receipt : receipt
    Payments --> ReceiptDebit : receiptDebit

    CardCredit --> CashCredit : cardCredit
    CardDebit --> CashDebit : cardDebit
    CashCredit --> Voucher : cashCredit
    CashDebit --> Voucher : cashDebit
    Voucher --> Receipt : voucher
    Voucher --> ReceiptDebit : voucher
    Receipt --> ReceiptDebit : receipt
  
```

The diagram illustrates a system architecture with the following components and relationships:

- Client** (yellow box) is the central entity, associated with **Documents**, **Anamnesis**, **Prescriptions**, **Payments**, **CardCredit**, **CardDebit**, **CashCredit**, **CashDebit**, **Voucher**, **Receipt**, and **ReceiptDebit** via associations labeled with their respective class names.
- Documents** (yellow box) is associated with **Anamnesis**, **Prescriptions**, **Payments**, **CardCredit**, **CardDebit**, **CashCredit**, **CashDebit**, **Voucher**, **Receipt**, and **ReceiptDebit** via associations labeled with their respective class names.
- Anamnesis** (yellow box) is associated with **Prescriptions**, **Payments**, **CardCredit**, **CardDebit**, **CashCredit**, **CashDebit**, **Voucher**, **Receipt**, and **ReceiptDebit** via associations labeled with their respective class names.
- Prescriptions** (yellow box) is associated with **Payments**, **CardCredit**, **CardDebit**, **CashCredit**, **CashDebit**, **Voucher**, **Receipt**, and **ReceiptDebit** via associations labeled with their respective class names.
- Payments** (yellow box) is associated with **CardCredit**, **CardDebit**, **CashCredit**, **CashDebit**, **Voucher**, **Receipt**, and **ReceiptDebit** via associations labeled with their respective class names.
- CardCredit** (yellow box) is associated with **CashCredit** and **CashDebit** via associations labeled with their respective class names.
- CardDebit** (yellow box) is associated with **CashDebit** and **CashCredit** via associations labeled with their respective class names.
- CashCredit** (yellow box) is associated with **CashDebit** and **Voucher** via associations labeled with their respective class names.
- CashDebit** (yellow box) is associated with **CashCredit** and **Voucher** via associations labeled with their respective class names.
- Voucher** (yellow box) is associated with **Receipt** and **ReceiptDebit** via associations labeled with their respective class names.
- Receipt** (yellow box) is associated with **ReceiptDebit** via an association labeled with its class name.
- ReceiptDebit** (yellow box) is associated with **Receipt** via an association labeled with its class name.

Each class has a set of attributes and methods, with some methods being abstract. The diagram also shows a hierarchy of classes, with **CardCredit** and **CardDebit** being subclasses of **Payments**, and **CashCredit** and **CashDebit** being subclasses of **Payments**.