

Università degli Studi di Napoli Federico II

**Corso di Laurea in Ingegneria Informatica**

**Corso di Ingegneria del Software**

**Prof. S. Russo - A.A. 2020 - 21**

***Progetto***

**APPLICAZIONE PER LA GESTIONE DEI CONTI CORRENTI**

Studente:

Riccardo Francesco Ruocco N46006008 ricca.ruocco@studenti.unina.it

Versione 3 del *3/10/2021*

**INDICE**

1. Specifiche informali 1

2. Analisi e specifica dei requisiti 2

2.1 Analisi nomi-verbi 2

2.2 Revisione dei requisiti 2

2.3 Glossario dei termini 3

2.4 Classificazione dei requisiti 3

2.4.1 Requisiti funzionali 4

2.4.2 Requisiti sui dati 4

2.4.3 Vincoli / Altri requisiti 4

2.5 Modellazione dei casi d’uso 5

2.5.1 Attori e casi d’uso 5

2.5.2 Diagramma dei casi d’uso 5

2.5.3 Scenari 5

2.6 Diagramma delle classi 2

2.7 Diagrammi di sequenza 2

3. Stima dei costi 3

4. Piano di test funzionale 4

5. Progettazione 5

5.1 Diagramma delle classi 5

5.2 Diagrammi di sequenza 5

6. Implementazione 6

7. Testing 7

7.1 Test strutturale 7

7.1.1 Complessità ciclomatica 7

7.2 Test funzionale 8

# Specifiche informali

Si vuole realizzare un'applicazione per la gestione dei conti correnti di una piccola banca.   
Il direttore può visualizzare tutti i conti correnti e registrare nel sistema gli impiegati che operano agli   
sportelli della banca, indicando nome, cognome, email e numero di telefono aziendali.   
Un conto corrente ha un codice IBAN e deve avere un cliente titolare; un conto può anche avere al più   
un cliente cointestatario (contitolare). Dei clienti occorre memorizzare nome e cognome, numero di   
telefono, email, codice fiscale.

Gli impiegati della banca possono aprire e chiudere conti correnti per i clienti che si presentano allo   
sportello. All’atto dell’apertura di un conto il sistema genera le credenziali (username e password) che   
il cliente potrà utilizzare per accedere online al proprio conto corrente, e il codice della carta di   
pagamento associata al conto per pagamenti online. La chiusura di un conto corrente può essere   
effettuata dall’impiegato (sempre su richiesta di un cliente che si presenta allo sportello) solo in caso di   
conto corrente non negativo (saldo disponibile maggiore o uguale a zero). I clienti della banca possono   
richiedere operazioni di prelievo e deposito di contanti presentandosi allo sportello (l’operazione è   
svolta dall’impiegato, e il sistema deve tener traccia dell’impiegato che la ha effettuata).   
Dalla propria area riservata del sito della banca, dopo aver effettuato l’accesso con le credenziali, i   
clienti possono visualizzare il saldo, la lista dei movimenti del proprio conto negli ultimi 30 giorni, ed   
effettuare bonifici online specificando il codice IBAN del conto del destinatario e l’importo da   
bonificare.

I clienti possono inoltre utilizzare la carta associata al proprio conto per effettuare pagamenti online su   
siti commerciali. Per il completamento di un pagamento online, il sistema (esterno) del sito   
commerciale richiede l’autorizzazione alla banca, che la concederà solo se l’importo del pagamento   
non supera il saldo del conto; il sistema della banca deve registrare ogni richiesta di pagamento   
(autorizzata o meno), e per quelle autorizzate provvederà a registrare l’operazione associandola al   
conto del cliente.   
Il direttore della banca può richiedere al sistema di generare un report di tutte le operazioni effettuate   
da un certo cliente in un certo intervallo di tempo (prelievo, deposito, pagamenti online).   
Ogni ultimo giorno del mese il sistema genera un report del totale di entrate ed uscite registrate dalla   
banca nell'ultimo mese per ogni cliente.

# Analisi e specifica dei requisiti

## Analisi nomi-verbi

Si vuole realizzare un' applicazione per la gestione dei conti correnti di una piccola banca.   
Il direttore può visualizzare tutti i conti correnti e registrare nel sistema gli impiegati che operano agli   
sportelli della banca, indicando nome, cognome, email e numero di telefono aziendali.   
Un conto corrente ha un codice IBAN e deve avere un cliente titolare; un conto può anche avere al più   
un cliente cointestatario (contitolare).

Dei clienti occorre memorizzare nome e cognome, numero di   
telefono, email, codice fiscale.

Gli impiegati della banca possono aprire conti e chiudere conti correnti per i clienti che si presentano allo   
sportello.

All’atto dell’apertura di un conto il sistema genera le credenziali (username e password) che   
il cliente potrà utilizzare per accedere online al proprio conto corrente, e il codice della carta di   
pagamento associata al conto per pagamenti online.

La chiusura di un conto corrente può essere effettuata dall’ impiegato (sempre su richiesta di un cliente che si presenta allo sportello) solo in caso di conto corrente non negativo (saldo disponibile maggiore o uguale a zero).

I clienti della banca possono richiedere operazioni di prelievo e deposito di contanti presentandosi allo sportello (l’operazione è svolta dall’ impiegato, e il sistema deve tener traccia dell’ impiegato che la ha effettuata).   
Dalla propria area riservata del sito della banca, dopo aver effettuato l’accesso con le credenziali, i   
clienti possono visualizzare il saldo,visualizzare la lista dei movimenti del proprio conto negli ultimi 30 giorni, ed effettuare bonifici online specificando il codice IBAN del conto del destinatario e l’ importo da   
bonificare.   
I clienti possono inoltre utilizzare la carta associata al proprio conto per effettuare pagamenti online su   
siti commerciali.

Per il completamento di un pagamento online, il sistema (esterno) del sito commerciale richiede l’autorizzazione alla banca , che la concederà solo se l’importo del pagamento   
non supera il saldo del conto; il sistema della banca deve registrare ogni richiesta di pagamento   
(autorizzata o meno), e per quelle autorizzate provvederà a registrare l’operazione associandola al   
conto del cliente.   
Il direttore della banca può richiedere al sistema di generare un report di tutte le operazioni effettuate   
da un certo cliente in un certo intervallo di tempo (prelievo, deposito, pagamenti online).   
Ogni ultimo giorno del mese il sistema genera un report del totale di entrate ed uscite registrate dalla   
banca nell'ultimo mese per ogni cliente.

*LEGENDA:  
Classe  
Attributo  
Funzionalità  
Attore*

*Classe-Attore*

## Revisione dei requisiti

Con gli incontri avvenuti con il committente si è cercato di disambiguare alcuni aspetti del sistema.

Sono stati chiariti seguenti punti:

1. Il sistema deve permettere al direttore di visualizzare tutti i conti correnti
2. Un conto corrente è identificato da un codice IBAN e da un saldo.
3. Un conto corrente deve avere almeno un cliente intestatario
4. Un conto corrente può essere al massimo due clienti intestatari
5. Il sistema deve permettere al direttore di registrare gli impiegati che lavorano agli sportelli della banca
6. Un impiegato è identificato da nome,cognome,email e numero di telefono aziendale
7. I clienti sono identificati da nome,cognome,numero di telefono,email,codice fiscale,username
8. Il sistema deve offrire agli impiegati presenti allo sportello una funzionalità per aprire conti correnti
9. Il sistema deve offrire agli impiegati presenti allo sportello una funzionalità per chiudere conti correnti
10. I conti correnti possono essere chiusi solo se hanno saldo disponibile maggiore o uguale di zero
11. Il sistema deve generare, all’atto dell’apertura di un nuovo conto, le credenziali affinché i clienti intestatari possano accedere alla propria area riservata sul sito della banca
12. Le credenziali che i clienti possono utilizzare per accedere alla propria area riservata sul sito della banca sono una username ed una password
13. Il sistema deve generare, all’atto dell’apertura di un nuovo conto, il codice della carta di pagamento associata al conto
14. Il sistema deve offrire all’impiegato allo sportello una funzionalità per registrare un operazione di prelievo contanti
15. Il sistema deve offrire all’impiegato allo sportello una funzionalità per registrare un operazione di deposito contanti
16. Per ciascuna operazione di prelievo contanti il sistema deve tener traccia dell’impiegato che l’ha effettuata e della data in cui è stata eseguita
17. Per ciascuna operazione di deposito contanti il sistema deve tener traccia dell’impiegato che l’ha effettuata della data in cui è stata eseguita
18. Il sistema deve offrire al cliente autenticato sul sito della banca una funzionalità per visualizzare il saldo del proprio conto corrente
19. Il sistema deve offrire al cliente autenticato sul sito della banca una funzionalità per visualizzare la lista dei movimenti del proprio conto negli ultimi 30 giorni
20. Il sistema deve offrire al cliente autenticato sul sito della banca una funzionalità per

effettuare bonifici online

1. Un bonifico è identificato dall’IBAN del destinatario, dall’importo da bonificare e dalla data in cui viene eseguito
2. Il sistema deve offrire una funzionalità ai siti commerciali per richiedere il pagamento
3. Il sistema deve registrare ogni richiesta di pagamento che arriva dai siti commerciali
4. Una richiesta di pagamento è caratterizzata dal nome del sito commerciale, dall’importo richiesto, dalla data.
5. Il sistema deve associare le richieste di pagamento autorizzate al conto del cliente
6. Il sistema deve offrire una funzionalità al direttore per generare un report di tutte le operazioni effettuate(prelievo,deposito,pagamenti online)da un certo cliente in un certo intervallo di tempo
7. Il sistema deve generare ogni ultimo giorno del mese un report per ogni cliente del totale di entrate ed uscite registrate dalla banca nell’ultimo mese

## Glossario dei termini

|  |  |  |
| --- | --- | --- |
| Termine | Descrizione | Sinonimi |
| Cliente Intestatario | possessore del conto | Cliente,Correntista |
| Saldo | saldo del conto corrente che si ottiene dal calcolo della somma algebrica di tutte le operazioni a credito e a debito |  |
| Lista dei movimenti | Elenco delle operazioni a credito e a debito con relativa data |  |

## Classificazione dei requisiti

## Requisiti funzionali

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RF01 | Il sistema deve permettere al direttore di visualizzare tutti i conti correnti | 1 |
| RF02 | Il sistema deve permettere al direttore di registrare gli impiegati che lavorano agli sportelli della banca | 5 |
| RF03 | Il sistema deve offrire agli impiegati presenti allo sportello una funzionalità per aprire conti correnti | 8 |
| RF04 | Il sistema deve offrire agli impiegati presenti allo sportello una funzionalità per chiudere conti correnti se questi hanno saldo non negativo | 9 |
| RF05 | Il sistema deve generare, all’atto dell’apertura di un nuovo conto, le credenziali affinché i clienti intestatari possano accedere alla propria area riservata sul sito della banca | 11 |
| RF06 | Il sistema deve generare, all’atto dell’apertura di un nuovo conto, il codice della carta di pagamento associata al conto | 13 |
| RF07 | Il sistema deve offrire all’impiegato allo sportello una funzionalità per registrare un operazione di prelievo contanti | 14 |
| RF08 | Il sistema deve offrire all’impiegato allo sportello una funzionalità per registrare un operazione di deposito contanti | 15 |
| RF09 | Il sistema deve offrire al cliente autenticato sul sito della banca una funzionalità per visualizzare il saldo del proprio conto corrente | 18 |
| RF10 | Il sistema deve offrire al cliente autenticato sul sito della banca una funzionalità per visualizzare la lista dei movimenti del proprio conto negli ultimi 30 giorni | 19 |
| RF11 | Il sistema deve offrire al cliente autenticato sul sito della banca una funzionalità per  effettuare bonifici online | 20 |
| RF12 | Il sistema deve offrire una funzionalità ai siti commerciali per richiedere il pagamento | 22 |
| RF13 | Il sistema deve offrire una funzionalità al direttore per generare un report di tutte le operazioni effettuate(prelievo,deposito,pagamenti online)da un certo cliente in un certo intervallo di tempo | 26 |
| RF14 | Il sistema deve generare ogni ultimo giorno del mese un report per ogni cliente del totale di entrate ed uscite registrate dalla banca nell’ultimo mese | 27 |
|  |  |  |

## Requisiti sui dati

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RD01 | Un conto corrente è identificato da un codice IBAN e da un saldo. | 2 |
| RD02 | Un conto corrente deve avere almeno un cliente intestatario | 3 |
| RD03 | Un conto corrente può essere al massimo due clienti intestatari | 4 |
| RD04 | Un impiegato è identificato da nome,cognome,email e numero di telefono aziendale | 6 |
| RD05 | I clienti sono identificati da nome,cognome,numero di telefono,email,codice fiscale,username e password | 7,12 |
| RD06 | Per ciascuna operazione di prelievo contanti il sistema deve tener traccia dell’impiegato che l’ha effettuata e della data in cui è stata eseguita | 16 |
| RD07 | Per ciascuna operazione di deposito contanti il sistema deve tener traccia dell’impiegato che l’ha effettuata della data in cui è stata eseguita | 17 |
| RD08 | Un bonifico è identificato dall’IBAN del destinatario, dall’importo da bonificare e dalla data in cui viene eseguito | 21 |
| RD09 | Una richiesta di pagamento è caratterizzata dal nome del sito commerciale, dall’importo richiesto e dalla data. | 24 |
| RD10 | Il sistema deve registrare ogni richiesta di pagamento che arriva dai siti commerciali | 23 |

## Vincoli / Altri requisiti

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine(n.frase  dei requisiti  revisionati) |
| RNF1 |  | 10 |

## Modellazione dei casi d’uso

## Attori e casi d’uso

***Attori Primari:***

* Direttore
* Impiegato
* ClienteAutenticatoOnline
* SitoCommerciale
* Tempo

***Attori Secondari:***

* …
* …
* …
* …

***Casi d’uso:***

* UC1:VisualizzaContiCorrenti
* UC2:RegistraImpiegati
* UC3:GeneraReportOperazioniCliente
* UC4:ApriContoCorrente
* UC5:ChiudiContoCorrente
* UC6:PrelevaContanti
* UC7:DepositaContanti
* UC8:GeneraReportOperazioniClientiUltimoMese
* UC9:VisualizzaSaldo
* UC10:VisualizzaListaMovimentiUltimoMese
* UC11:EffettuaBonifico
* UC12:RichiediPagamento

***Casi d’ uso di inclusione:***

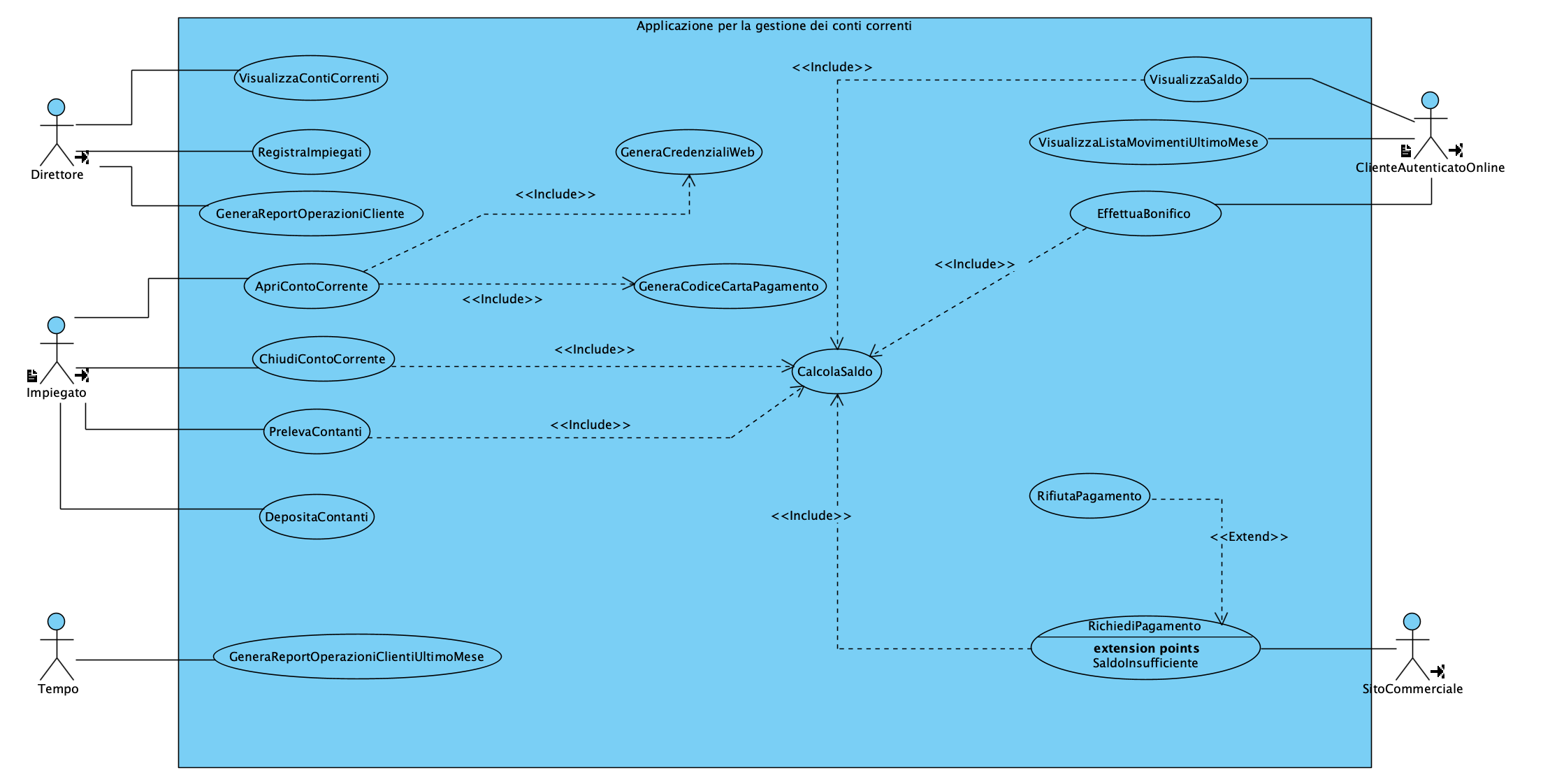
* UC13:GeneraCredenzialiWeb
* UC14:GeneraCodiceCartaPagamento
* UC15:CalcolaSaldo

***Casi d’uso di estensione:***

* UC16:RifiutaPagamento

|  |  |  |  |
| --- | --- | --- | --- |
| Caso d’uso | Attori Primari | Attori Secondari | Incl. / Ext. |
| UC1:VisualizzaContiCorrenti | Direttore |  |  |
| UC2:RegistraImpiegati | Direttore |  |  |
| UC3:GeneraReportOperazioniCliente | Direttore |  |  |
| UC4:ApriContoCorrente | Impiegato |  | Include  GeneraCredenzialiWeb  GeneraCodiceCartaPagamento |
| UC5:ChiudiContoCorrente | Impiegato |  | Include  CalcolaSaldo |
| UC6:PrelevaContanti | Impiegato |  | Include  CalcolaSaldo |
| UC7:DepositaContanti | Impiegato |  |  |
| UC8:GeneraReportOperazioniClientiUltimoMese | Tempo |  |  |
| UC9:VisualizzaSaldo | ClienteAutenticatoOnline |  | Include  CalcolaSaldo |
| UC10:VisualizzaListaMovimentiUltimoMese | ClienteAutenticatoOnline |  |  |
| UC11:EffettuaBonifico | ClienteAutenticatoOnline |  | Include  CalcolaSaldo |
| UC12:RichiediPagamento | SitoCommerciale |  | Esteso da  RifiutaPagamento |
| UC13:GeneraCredenzialiWeb | Impiegato |  | Incluso da  ApriContoCorrente |
| UC14:GeneraCodiceCartaPagamento | Impiegato |  | Incluso da  ApriContoCorrente |
| UC15:CalcolaSaldo | Impiegato  ClienteAutenticatoOnline  SitoCommerciale |  | Incluso da:  ChiudiContoCorrente  PrelevaContanti  VisualizzaSaldo  EffettuaBonifico  RichiediPagamento |
| UC16:RifiutaPagamento | SitoCommerciale |  | Estensione di  RichiediPagamento |
|  |  |  |  |

## Diagramma dei casi d’uso



## Scenari

Riportare lo scenario principale per ogni caso d’uso sviluppato fino alla codifica in Java, come nel seguente esempio

|  |  |
| --- | --- |
| Caso d’uso: | VisualizzaListaMovimentiUltimoMese |
| Attore primario | ClienteAutenticatoOnline |
| Attore secondario | … |
| Descrizione | Il cliente chiede al sistema di mostrargli la lista dei movimenti del proprio conto negli ultimi 30 giorni |
| Pre-Condizioni | Il cliente deve essersi autenticato |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia alle quando il cliente richiede la lista dei movimenti dell’ultimo mese al sistema 2. Il sistema cerca le operazioni compiute allo sportello negli ultimi 30 giorni 3. Il sistema cerca i bonifici inviati negli ultimi 30 giorni 4. Il sistema cerca i pagamenti a siti commerciali effettuati negli ultimi 30 giorni 5. Il sistema mostra la lista dei movimenti del conto corrente al cliente |
| Post-Condizioni | Viene visualizzata la lista dei movimenti |
| Casi d’uso correlati | *nessuno* |
| Sequenza di eventi  alternativi | nessuna |

## Diagramma delle classi

Diagramma delle classi di analisi

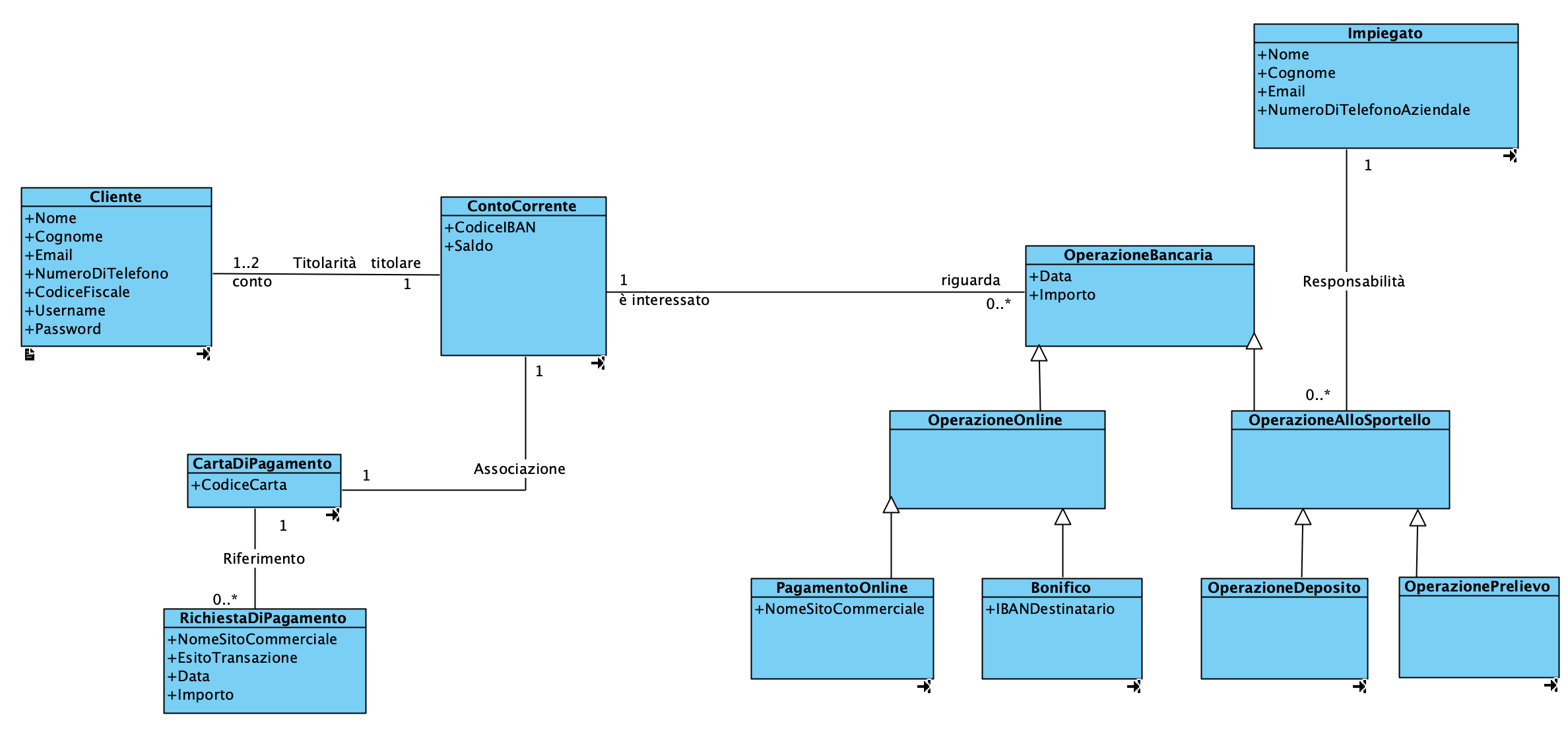
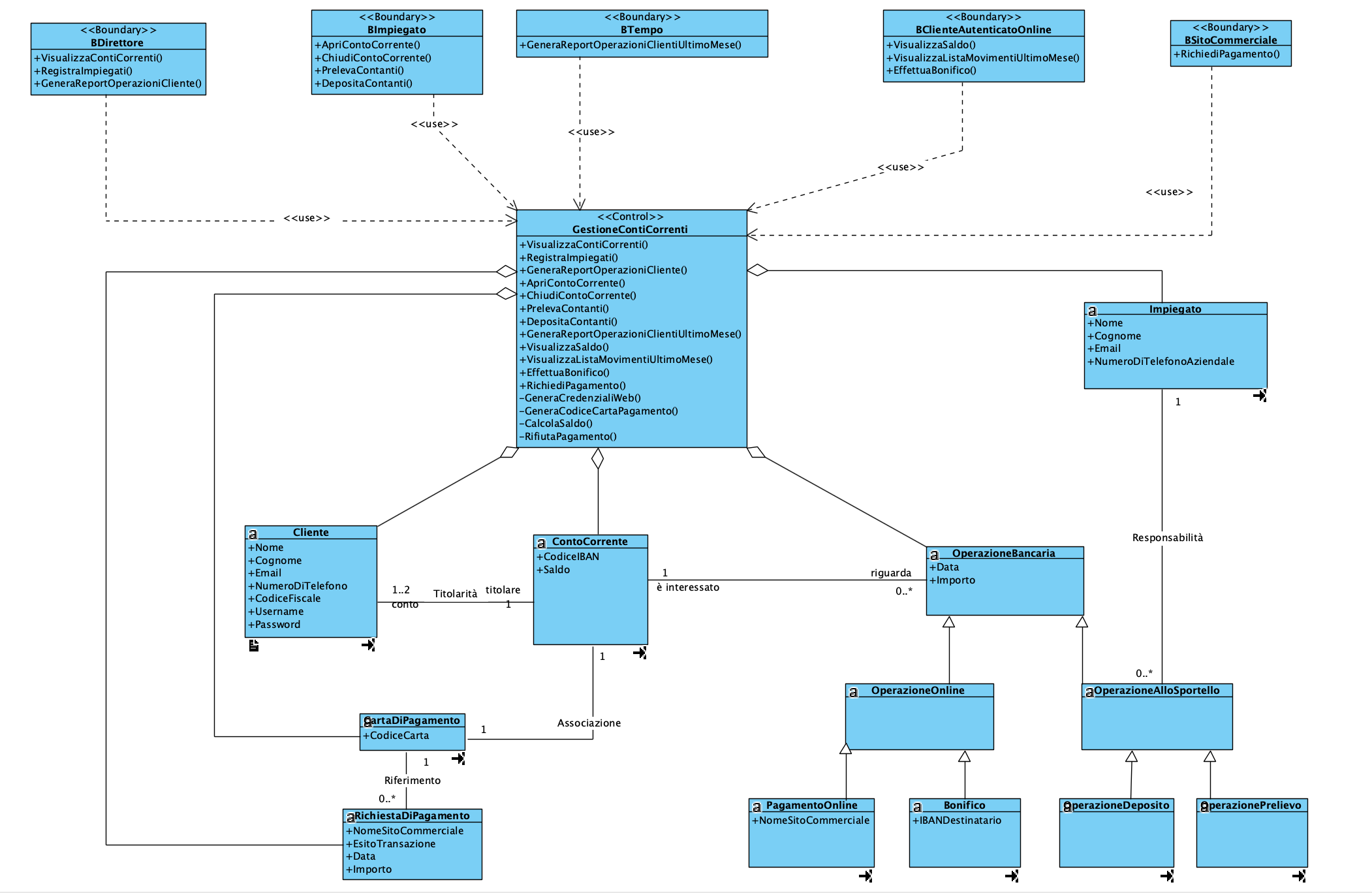
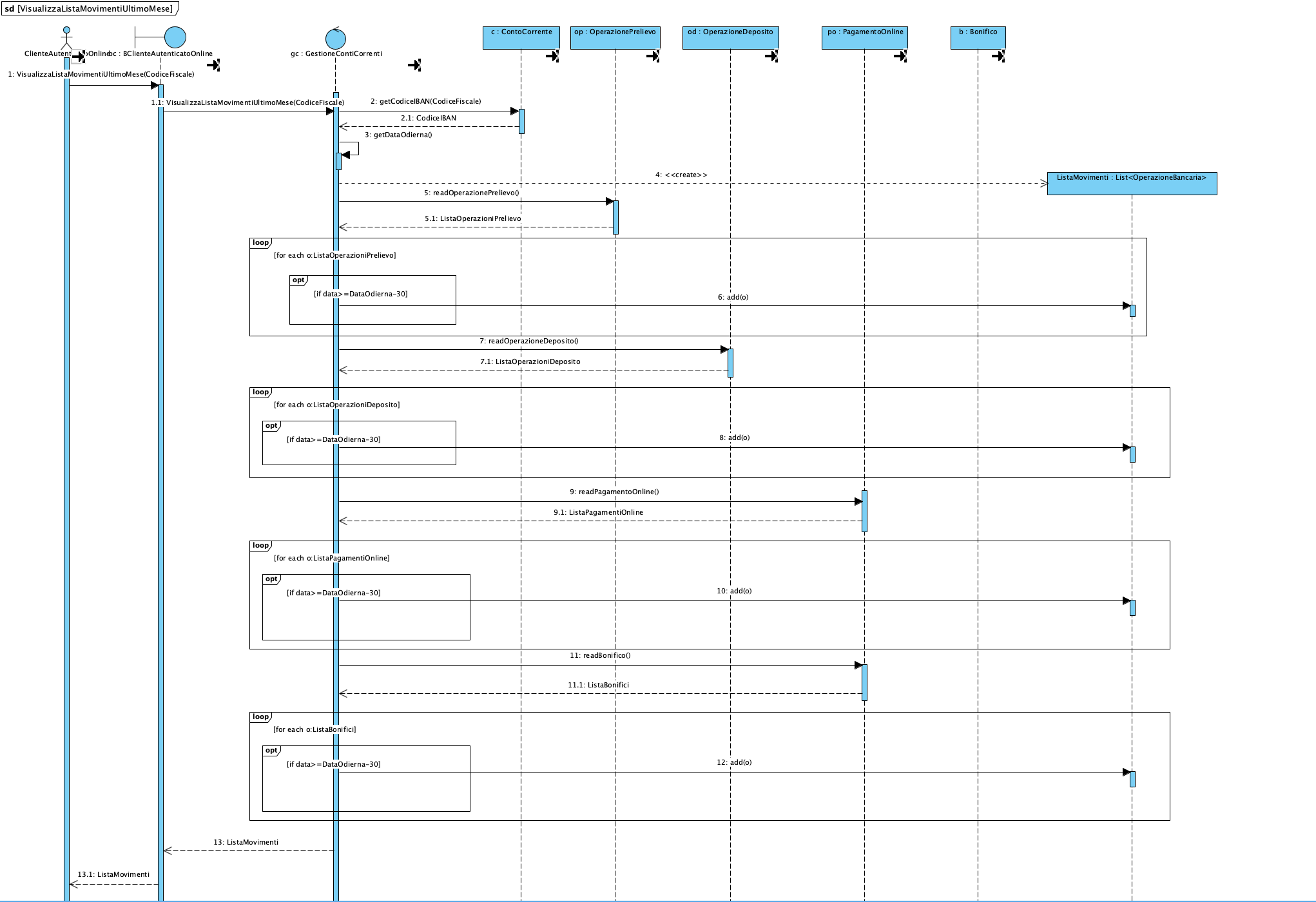


Diagramma delle classi **raffinato**



## Diagrammi di sequenza



## Verifica della completezza dei requisiti

Legenda UCD=Use Case Diagram, CD=Class Diagram, SD=Sequence Diagram

* RF1 è modellato nell’UCD con l’attore “Direttore” e con il caso d’uso UC1
* RF2 è modellato nell’UCD con l’attore “Direttore” e con il caso d’uso UC2
* RF3 è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC4
* RF4 è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC5
* RF5 è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC13
* RF6 è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC14
* RF7 è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC6
* RF8 è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC7
* RF9 è modellato nell’UCD con l’attore “ClienteAutenticatoOnline” e con il caso d’uso UC9
* RF10 è modellato nell’UCD con l’attore “ClienteAutenticatoOnline” e con il caso d’uso UC10
* RF11 è modellato nell’UCD con l’attore “ClienteAutenticatoOnline” e con il caso d’uso UC11
* RF12 è modellato nell’UCD con l’attore “SitoCommerciale” e con il caso d’uso UC12
* RF13 è modellato nell’UCD con l’attore “Direttore” e con il caso d’uso UC3
* RF14 è modellato nell’UCD con l’attore “Tempo” e con il caso d’uso UC8
* RD1 è modellato nel CD con la classe ContoCorrente
* RD2 è modellato nel con la molteplicità dell’associazione Titolarità
* RD3 è modellato nel con la molteplicità dell’associazione Titolarità
* RD4 è modellato con la classe Impiegato
* RD5 è modellato nel CD con la classe Impiegato
* RD6 è modellato nel CD con l’associazione Responsabilità
* RD7 è modellato nel CD con la classe ContoCorrente
* RD8 è modellato nel CD con la classe Bonifico
* RD9 è modellato nel CD con la classe RichiestaDiPagamento
* RD10 è modellato nel CD con la classe RichiestaDiPagamento

# Stima dei costi

**VisualizzaListaMovimentiUltimoMese**

|  |  |  |  |
| --- | --- | --- | --- |
|  | SEMPLICE | MEDIO | COMPLESSO |
| NILF | **7** | **10** | **15** |
| NEIF | **5** | **7** | **10** |
| NEI | **3** | **4** | **6** |
| NEO | **4** | **5** | **7** |
| NEQ | **4** | **4** | **6** |

Conteggio:

**Internal Logical File(NILF)-**Aggregati logici di dati generati, usati e gestiti internamente dal sistema.

In questo caso specifico **NILF=3**

**External Interface File(EIF)**-Aggregati logici di dati scambiati o condivisi con altre applicazioni. In questo caso specifico **NEIF=0**

**External Input(EI)**-Informazioni distinte fornite dall’utente o da altre parti del sistema usate come dati di ingresso.Nel nostro caso, come parametro di input vado a considerare il codice fiscale.

Quindi **NEI=1**

**External Output(NEO)-**output distinti che il sistema restituisce all’utente come risultato delle proprie elaborazioni.

In questo caso **NEO=0**

**External Inquiry(NEQ)-**interrogazioni in linea che producono una risposta immediate del sistema, senza necessità di elaborazione.

In questo caso **NEQ=3**

Scelta dei pesi

|  |  |  |  |
| --- | --- | --- | --- |
|  | SEMPLICE | MEDIO | COMPLESSO |
| NILF |  | **10** |  |
| NEIF | **0** |  |  |
| NEI | **3** |  |  |
| NEO | **0** |  |  |
| NEQ | **4** |  |  |

Calcolo UFP

UFP=3\*10+1\*3+3\*4=45

|  |  |  |  |
| --- | --- | --- | --- |
| FATTORI CORRETTIVI | | | |
| COMUNICAZIONE DATI | | | 1 |
| DISTRIBUZIONE ELABORAZIONE | | | 0 |
| PRESTAZIONI | | | 3 |
| UTILIZZO INTENSIVO CONFIGURAZIONE | | | 2 |
| FREQUENZA DELLE TRANSAZIONI | | | 3 |
| INSERIMENTO DATI INTERATTIVO | | | 3 |
| EFFICIENZA PER L’UTENTE FINALE | | | 4 |
| AGGIORNAMENTO INTERATTIVO | | | 2 |
| COMPLESSITA’ ELABORATIVA | | | 0 |
| RIUSABILITA’ | | | 3 |
| FACILITA’ INSTALLAZIONE | | | 2 |
| FACILITA’ GESTIONE OPERATIVA |  |  | 2 |
| MOLTEPLICITA’ DI SITI | | | 0 |
| FACILITA’ DI MODIFICA | | | 3 |

28

FP=UFP x (0.65+0.01 x )=41,85

Il numero di line di codice stimate se l’applicazione è sviluppata in Java è: 251

LLOC/FP=2218

# Piano di test funzionale

Progettare i casi di test funzionale con la tecnica del *Category Partition Testing*. Descrivere il procedimento di calcolo.

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “VisualizzaListaMovimentiUltimoMese”.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoria...** | **Categoria...** | **Categoria...** | **Categoria...** | **Categoria...** |
| * ... * ... * ... | * ... * ... * ... | * ... * ... * ... | * ... * ... * ... | * ... * ... * ... |

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Progettazione

## Diagramma delle classi

Riportare il diagramma delle classi di progettazione. Reificare eventuali classi associative del diagramma delle classi di analisi. Specificare argomenti e tipo di ritorno delle operazioni (per quelle più significative, coinvolte nei casi d’uso sviluppati fino alla implementazione). Includere classi del dominio della soluzione, come strutture dati e classi DAO. Raggruppare le classi in package.

## Diagrammi di sequenza

Riportare i diagrammi di sequenza di progetto per il/i casi d’uso sviluppati fino alla codifica in Java.

# Implementazione

Non includere il codice sorgente, ma descrivere l’implementazione in Java, descrivendo gli artefatti di codifica:

* Elencare:
  + package, classi, tipi di eccezione definiti
* Elencare gli artefatti necessari per l’installazione ed esecuzione del programma, senza ovviamente l’ambiente di sviluppo come Eclipse (DB h2, eventuali librerie e versioni di Java che l’utilizzatore deve avere installati, file .class, .jar, …)
* Produrre un eventuale diagramma di deployment
* Eventualmente inserire la documentazione del codice prodotta con Javadoc (relativamente alle funzionalità implementate)
* Riportare il numero di LOC e di LLOC scritte in Java
* Confrontare con la stima dei costi effettuata e commentare eventuali scostamenti

# Testing

## Test strutturale

## Complessità ciclomatica

Costruire il Control Flow Graph per uno o due dei metodi delle classi implementate (si scelgano metodi non proprio banali), e:

- si mostri il calcolo del numero ciclomatico;

- si indichino i percorsi linearmente indipendenti;

Prima o a fianco del CFG riportare il codice Java del metodo.

Es.:

public static boolean CalcolaStatistica(String stringaData, int CAP, boolean scelta) {

if(scelta)

stringaData = "01/"+stringaData;

LocalDate data = null;

DateTimeFormatter dateTimeFormatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");

try {

if(!(MonitoraggioAmbiente.getInstance().VerificaCAP(CAP))) { //Verifica CAP

System.err.println("CAP non valido");

return false;

}

}

catch(IOException e) {

System.err.println("Errore di I/O durante il controllo del CAP" + e.getMessage());

return false;

}

try {

data = LocalDate.parse(stringaData, dateTimeFormatter); //Dalla sequenza data.parse() != null

if(data==null) { //Verifica DATA

System.err.println("Data non valida");

return false;

}

}catch(DateTimeParseException e) {

System.err.println("Data non valida");

return false;

}

java.util.Map<String, Float> Dati = null;

try {

if(scelta) {

Dati = MonitoraggioAmbiente.getInstance().CalcolaStatisticaMese(CAP, data);

System.out.println("Statistica del mese inserito:\n");

}

else {

Dati = MonitoraggioAmbiente.getInstance().CalcolaStatisticaGiorno(CAP, data);

System.out.println("Statistica del giorno inserito:\n");

}

for(String key:Dati.keySet()) {

System.out.println(key +":"+(key.length()<14?"\t\t":"\t") + Dati.get(key)); //Output

}

} catch (StatisticaException e) {

System.err.println(e.getMessage()); return false;

} catch (MisuraException e) {

System.err.println(e.getMessage()); return false;

}

return true;

}

}

Control Flow Graph



NUMERO CICLOMATICO:

numero di regioni chiuse del grafo = 6

numero di nodi predicati (0,2,4,6,10) +1 = 6

# archi – # nodi + 2 = (15 - 11) + 2 = 6

CAMMINI:

1. 0-1-2-3
2. 0-1-2-4-5
3. 0-1-2-4-6-7-9
4. 0-1-2-4-6-8-9
5. 0-1-2-4-6-7-9-10-9
6. 0-2-4-6-7-9

## Test funzionale

Descrivere i risultati dell’esecuzione dei test funzionali precedentemente pianificati adoperando lo schema di tabella seguente.

Descrivere le eventuali azioni di *debugging* a seguito di casi di test con esito *FAIL*.

Commentare se gli eventuali difetti rilevati dal test funzionale potevano essere rilevati anche da un test strutturale.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Output Ottenuti** | **Post-condizioni Ottenute** | **Esito**  **(*FAIL*, *PASS*)** |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |