****

**CORSO DI LAUREA:** Ingegneria Informatica e delle Telecomunicazione

**CORSO:** Programmazione Web e Mobile

**ANNO ACCADEMICO:** 2018/2019

**DOCENTE:** Prof. Roberto Pirrone

**TESINA:** Applicazione di Pagamento Elettronico (DSPpay)

**GRUPPO:** DSP

**A cura di:**

Lavalle Davide

Peri Giuseppe

Pisciotta Salvatore

**INDICE**

1. **Obiettivi Generali**
2. **Sistema Corrente**
3. **Sistema Proposto**
   1. Panoramica del Sistema
   2. Requisiti funzionali
   3. Requisiti non funzionali
4. **Modelli del Sistema**
   1. Modello dei casi d’uso
      1. Attori coinvolti
      2. Casi d’uso – Vista d’insieme
      3. Casi d’uso comuni
      4. Caso d’uso – Visualizza Movimenti
      5. Caso d’uso – Creazione estratti conto
      6. Casi d’uso – Gestione Profilo
      7. Casi d’uso – Gestione Metodi Pagamento
      8. Casi d’uso – Invio/Richiesta Denaro
      9. Casi d’uso – Pagamento
   2. Modello dinamico – Diagrammi di sequenza
5. **Architettura software proposta**
   1. Panoramica
   2. Gestione dei dati persistenti – Progetto del Database
      1. Modello ER
      2. Progetto Logico
      3. Descrizione delle Tabelle
   3. Decomposizione in sottoinsiemi
   4. Mappatura Hardware/Software
6. **Implementazione**
   1. Guida all’installazione
   2. Applicazione Android
7. **Divisione dei compiti**
8. **Obiettivi Generali**

Lo scopo del sistema proposto è quello di fornire un servizio di gestione di un proprio conto online tramite un browser web o tramite l’apposita applicazione Android. Il sistema deve, inoltre, permettere agli utenti di associare una carta di credito/debito o un iban di un c/c bancario.

1. **Sistema Corrente**

Si suppone che attualmente non esista alcun software con le stesse funzionalità.

1. **Sistema Proposto**
   1. **Panoramica del Sistema**
   2. **Requisiti funzionali**

* **Gestione del profilo**

Creazione del profilo, e possibilità di modifica dei dati anagrafici e della password di accesso all’area riservata dell’utente, aggiunta dei metodi di pagamento e settaggio di un metodo di default.

* **Movimenti**

Gestione e visualizzazione dei movimenti sul conto dell’utente con funzioni di cronologia delle operazioni e operazioni di filtraggio.

* **Estratti conto**

Creazione di estratti conto generati ogni mese che posso essere scaricati navigando nell’apposita area.

* **Invio di denaro**

Possibilità di inviare denaro a utenti registrati e non, con modifica dei saldi nel metodo di pagamento scelto dal donatore e nel metodo di pagamento di default del ricevente.

Nel caso di utente non registrato viene mandata una email con il link di registrazione.

* **Richiesta di denaro**

Possibilità di richiedere denaro ad utenti registrati e non, con modifica dei saldi nel metodo di pagamento scelto dal ricevente e nel metodo di pagamento di default del donatore.

Nel caso di utente non registrato viene mandata una email con il link di registrazione.

* **Pagamento esercizi commerciali**

Funzionalità di pagamento di un esercizio commerciale dove l’utente specifica il nome dell’esercizio, l’ammontare del pagamento e il metodo di pagamento.

* **Rateizzazioni**

Funzionalità di avvio di pagamenti che avvengono periodicamente, dove l’ammontare del pagamento totale viene diviso in un numero di rate scelto dall’utente, distanziate l’una dall’altra da un numero di giorni, anch’esso scelto dall’utente.

Inoltre è possibile bloccare la rateizzazione e allora l’importo ancora non pagato verrà sottratto dal metodo di pagamento scelto per intero.

**3.3. Requisiti non funzionali**

* **Visualizzazione responsive**

L’app deve adattare la propria visualizzazione allo schermo sul quale è visualizzato. Il layout grafico deve quindi essere piacevole e comodamente fruibile su qualunque schermo (mobile, tablet, desktop).

* **Controllo validità dei dati**

L’app deve impedire che l’utente possa inserire dei dati sbagliati o inconsistenti, come per esempio un numero di cellulare in cui compaiono lettere o una password che non rispetta dei determinati canoni.

* **Usabilità**

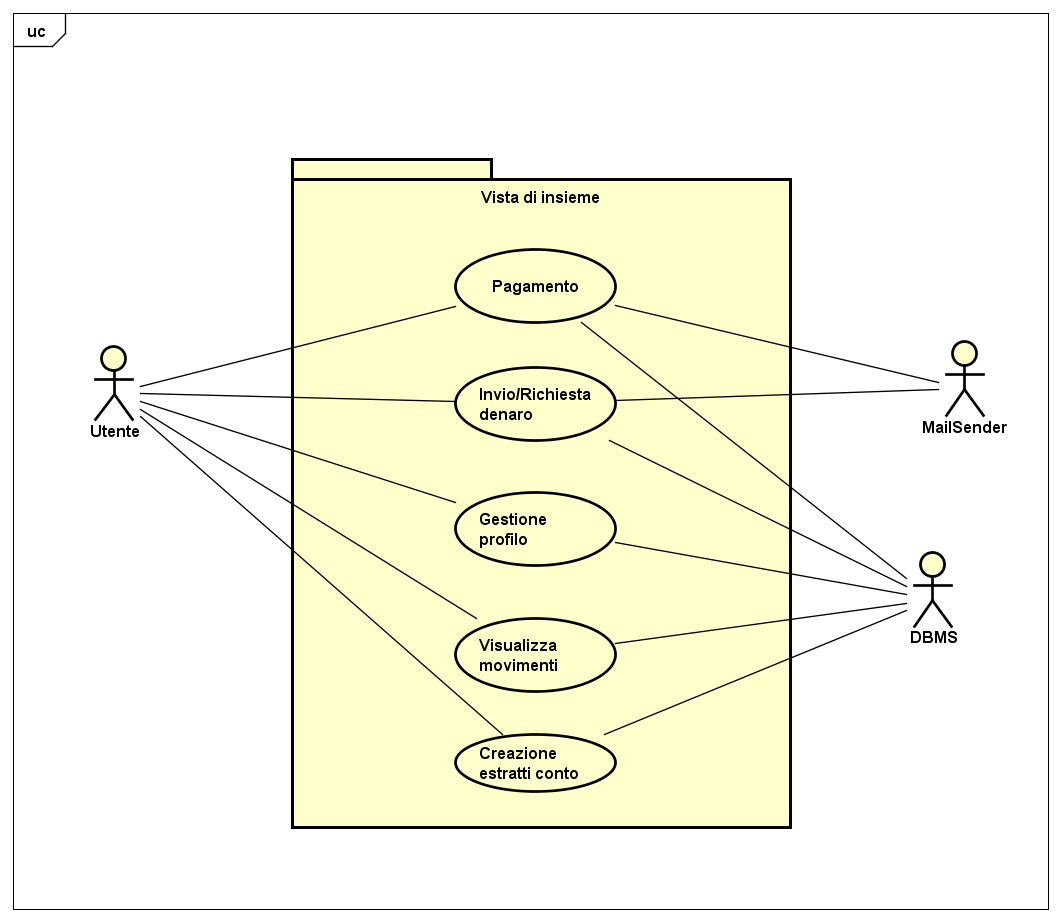
L’app deve fornire una GUI semplice ed intuitiva.

**4. Modelli del Sistema**

**4.1. Modello dei casi d’uso**

**4.1.1. Attori coinvolti**

* **DBMS:** Software che si occupa di conservare e distribuire i dati contenuti sul database agli altri attori, quando richiesto.
* **UTENTE:** Individuo registrato alla pagina web o all’applicazione che può usufruire dei servizi che esso offre
* **MAIL SERVER:** Servizio che si occupa di inviare email a più utenti, utilizzato per il ripristino della password, per mandare il link di registrazione a utenti non registrati quando si effettua un’invio/richiesta di denaro verso tale utente,
* per indicare all’utente la mancanza di soldi nel metodo di pagamento o del superamento dei suoi limiti spesa all’atto di effettuare un pagamento.

**4.1.2. Casi d’uso – Vista d’insieme**

I casi d’uso sono divisi in 3 macro casi (Pagamento, Invio/Richiesta di denaro e Gestione Profilo) e altri due casi d’uso semplici (Visualizza Movimenti e Creazione Estratti Conto).

**4.1.3 Casi d’uso comuni**

In questa sezione si descrivono i casi d’uso che compaiono in diverse parti del programma.

|  |
| --- |
| **CASO D’USO: INVIO E-MAIL** |
| **ATTORI:**  Utente, DBMS, MailSender |
| **PRECONDIZIONE:**   1. L’utente ha effettuato un’operazione che richiede una richiesta/aggiornamento/eliminazione dal DBMS. |
| **SEQUENZA DEGLI EVENTI:**   1. Il sistema riceve una risposta ad una query dal DBMS. 2. Il sistema invia una mail utilizzando come dati di invio, quelli presenti in specifici campi compilati dall’utente nella pagina dell’operazione eseguita. 3. Il MailSender spedisce l’email e il sistema manda a video un messaggio di conferma dell’invio della e-mail |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma di invio e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente. |

|  |
| --- |
| **CASO D’USO: INVIO NOTIFICA DI ERRORE** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**   1. L’Utente ha effettuato un’operazione che richiede una richiesta/aggiornamento/eliminazione dal DBMS |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme un pulsante che fa mandare al sistema una query al DBMS. 2. Il sistema manda al DBMS la query e i dati presenti. 3. Il DBMS risponde con un messaggio di errore. 4. Il sistema mette a video dell’utente una notifica di errore.   4.1 Se l’operazione lo prevede si rimanda al caso d’uso “invia e-mail”. |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di errore e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente. |

|  |
| --- |
| **CASO D’USO: CHECK IMPORTO E LIMITE SPESA** |
| **ATTORI:** Utente, DBMS |
| **PRECONDIZIONE:**  L’utente deve effettuare un pagamento |
| **SEQUENZA DEGLI EVENTI:**   1. Il sistema, dall’operazione chiamante, verifica che il saldo del metodo di pagamento scelto o di default sia superiore all’importo da pagare    1. Se il risultato è positivo, il sistema effettua il controllo sui limiti spesa giornalieri e mensili che sono propri del metodo di pagamento scelto       1. Se il risultato è positivo il caso d’uso ritorna al precedente    2. Altrimenti viene mandato un messaggio di errore |
| **PUNTO DI ESTENSIONE:**  Al punto 1.2 viene chiamato il caso d’uso “Invio Email/Messaggio di errore” |

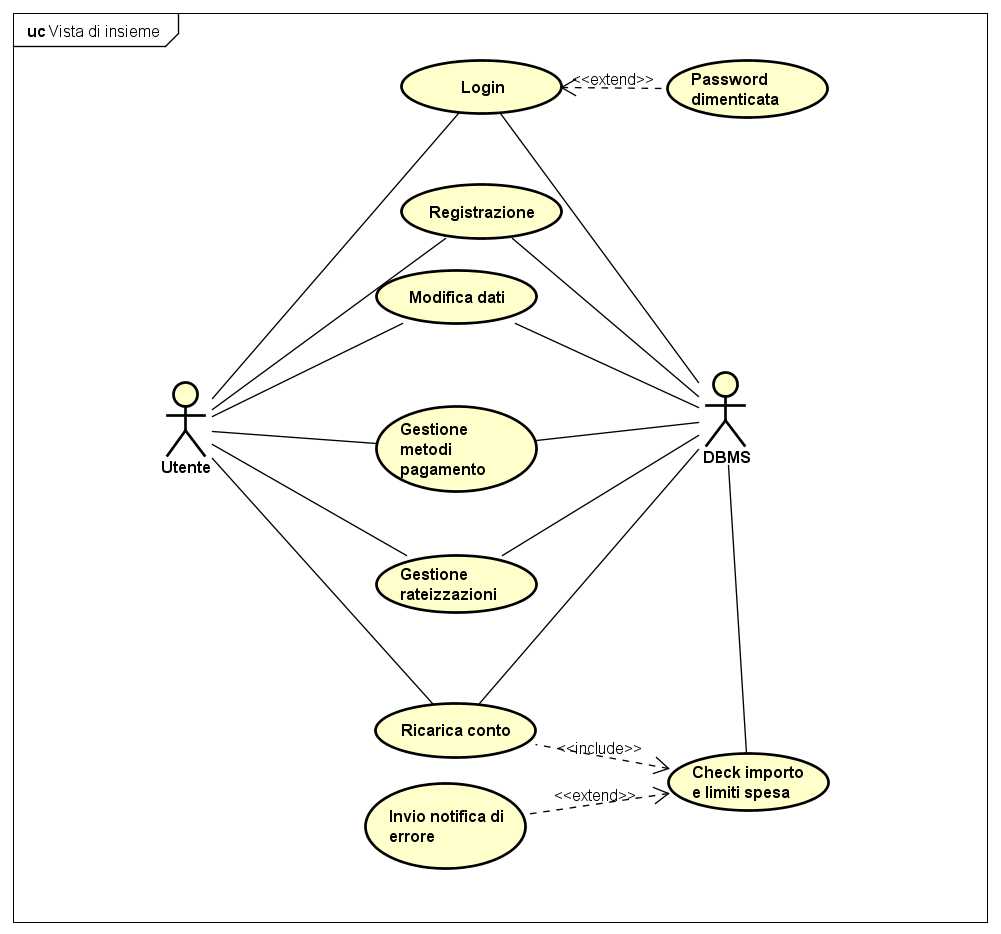
**4.1.4 Caso d’uso – Visualizza movimenti**

|  |
| --- |
| **CASO D’USO: VISUALIZZA MOVIMENTI** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**   1. L’utente deve essere autenticato |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme sul link dell’opzione “Visualizza Movimenti” presente sulla pagina principale 2. Il sistema invia al DBMS i dati dell’utente 3. Il sistema compila la tabella della pagina dei movimenti con i dati forniti dal DBMS, e la mostra all’utente    1. L’utente seleziona un metodo di filtraggio    2. Il sistema filtra le righe della tabella in funzione del filtraggio selezionato |

**4.1.5. Caso d’uso – Creazione estratti conto**

|  |
| --- |
| **CASO D’USO: CREAZIONE ESTRATTI CONTO** |
| **ATTORI:** Utente, DBMS |
| **PRECONDIZIONE:**  L’utente è autenticato |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme, dalla pagina principale, su link “Effettua pagamenti periodici” 2. Il sistema reindirizza l’utente alla pagina, dove ogni mese vengono creati gli estratti conti del mese conclusosi 3. L’utente premendo su Download scarica l’estratto conto corrispondente |

**4.1.6. Casi d’uso – Gestione profilo**

****

|  |
| --- |
| **CASO D’USO: LOGIN** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**  L’utente vuole accedere al sistema DSPay |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme sul pulsante di LOGIN. 2. Il sistema fa apparire in sovrimpressione una schermata con i campi per il login. 3. L’utente compila i campi con le credenziali di accesso e preme il pulsante “ACCEDI” una volta compilati    1. Se l’utente preme su “PASSWORD DIMENTICATA?”, si rimanda al caso d’uso “PASSWORD DIMENTICATA” 4. Il sistema manda al DBMS i dati inseriti    1. In caso di dati sbagliati si rimanda al caso d’uso “INVIO NOTIFICA ERRORE” 5. Il sistema mostra una finestra di avvenuto accesso. |
| **PUNTO DI ESTENSIONE:**  Nel punto 6.1 viene chiamato il caso d’uso “INVIO NOTIFICA DI ERRORE” |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma di invio e premuto OK sulla rispettiva finestra il sistema fa scomparire la finestra di login e rimanda sulla pagina principale. |

|  |
| --- |
| **CASO D’USO: PASSWORD DIMENTICATA** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**  L’utente preme su “PASSWORD DIMENTICATA?” sulla pagina di login |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente compila il campo EMAIL presente nella pagina per recuperare la password 2. Il sistema genera una password temporanea casuale 3. Il sistema manda una email a quella specificata dall’utente, si rimanda al caso d’uso INVIO EMAIL 4. Il sistema legge la email e la manda al DBMS per aggiornare la password dell’account corrispondente |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma di invio dell’email e premuto OK sulla rispettiva finestra il sistema fa scomparire la finestra di recupero password e rimanda sulla pagina principale. |

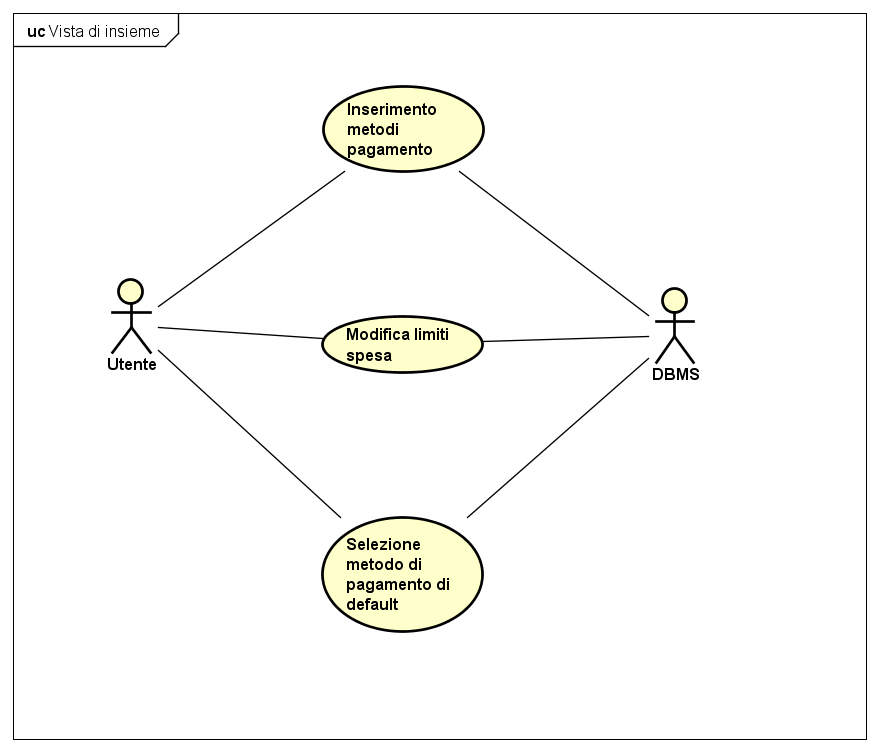
|  |
| --- |
| **CASO D’USO: REGISTRAZIONE** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**   1. L’utente dalla pagina principale preme sul tasto “Registrati” |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme il pulsante “Registrati” dalla pagina principale. 2. Il sistema mostra a video una finestra con i campi di compilazione vuoti associati ai dati necessari per la registrazione. 3. L’utente compila i dati.    1. Il sistema verifica, ogni qual volta l’utente inserisce un dato in un campo, la correttezza del formato del dato inserito.    2. Se il formato risulta essere errato il sistema lo segnala con un messaggio all’interno della pagina. 4. L’utente una volta compilati i campi, preme sul pulsante “Registrati”.    1. Se il sistema identifica dei formati errati di dati, disabilita il pulsante e lascia i messaggi di errore all’interno della pagina 5. Il sistema manda al DBMS i dati presenti nella pagina    1. Se sono presenti dati di un utente già registrato si rimanda al caso “INVIO NOTIFICA DI ERRORE” 6. Il sistema mette a video un messaggio di conferma dell’operazione |
| **PUNTO DI ESTENSIONE:**  Nel punto 6.1 viene chiamato il caso d’uso “INVIO NOTIFICA DI ERRORE” |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema fa scomparire la finestra di registrazione e rimanda a video la schermata iniziale |

|  |
| --- |
| **CASO D’USO: MODIFICA DATI** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**   1. L’utente deve essere autenticato |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme sul tasto Gestisci Account del proprio menù personale. 2. Il sistema manda al DBMS i dati dell’utente. 3. Il sistema compila i campi con i dati ricevuti dal DBMS. 4. L’utente modifica il campo/i corrispondente/i al dato/i da modificare.    1. Il sistema verifica la correttezza dei formati dei dati presenti nei campi    2. Se sono presenti formati errati, il sistema li segnala 5. Una volta terminate le modifiche, l’utente preme il tasto di conferma.    1. Se sono presenti formati di dati errati il sistema disabilita il pulsante 6. Il sistema manda al DBMS i dati presenti nella pagina 7. Il sistema mostra un messaggio di conferma di avvenuta modifica |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente. |

|  |
| --- |
| **CASO D’USO: GESTIONE RATEIZZAZIONI** |
| **ATTORI:** Utente, DBMS |
| **PRECONDIZIONE:**  L’utente deve essere autenticato  L’utente si trova sulla pagina di “Gestione Rateizzazione” |
| **SEQUENZA DEGLI EVENTI:**   1. L’utente visualizza le sue rateizzazioni e preme sul pulsante per stornare il pagamento 2. Il sistema prova ad effettuare il pagamento della somma rimanente verso quell’esercizio commerciale, viene invocato il caso d’uso “Check importo e limiti spesa”    1. Se non ci sono messaggi di errore viene registrata la transazione e si aggiorna quindi il saldo dell’utente |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente, aggiornando eventualmente la tabella. |

|  |
| --- |
| **CASO D’USO: RICARICA CONTO** |
| **ATTORI:** Utente, DBMS |
| **PRECONDIZIONE:**  L’utente deve essere autenticato  L’utente si trova sulla pagina di “Ricarica conto” |
| **SEQUENZA DEGLI EVENTI:**   1. L’utente inserisce il tipo di metodo di pagamento col quale vuole ricaricare il suo conto online 2. Il sistema manda al DBMS il tipo di pagamento scelto dall’utente e riempie il menu a tendina con i metodi di pagamento disponibili dell’utente 3. L’utente inserisce l’importo da ricaricare e preme su conferma 4. Viene invocato il caso d’uso “Check importo e limiti spesa”    1. Se non ci sono messaggi di errore viene registrata la transazione e si aggiorna quindi il saldo dell’utente |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente, svuotando i campi. |

**4.1.7 Casi d’uso - Gestione Metodi Pagamento**

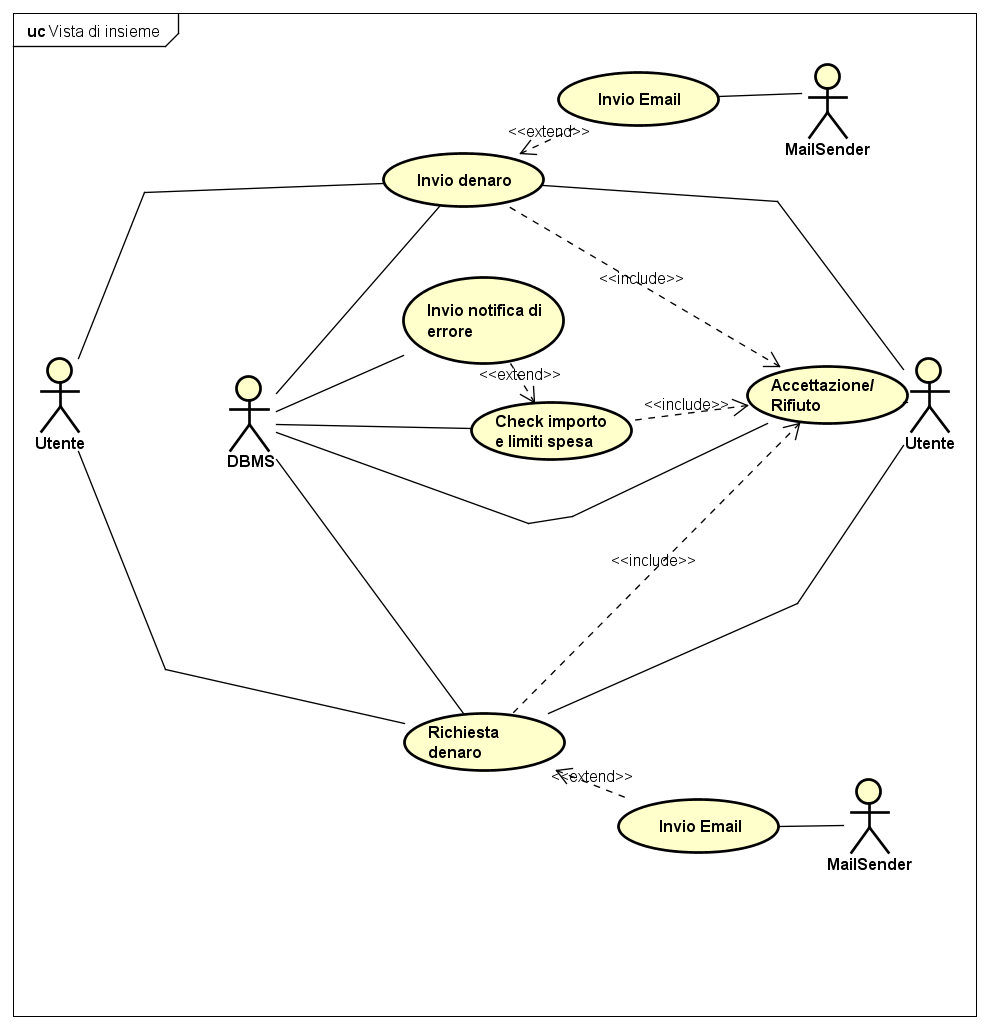
**** Questo caso d’uso è in realtà un macro caso che comprende altri casi d’uso.

|  |
| --- |
| **CASO D’USO: INSERIMENTO METODO PAGAMENTO** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**  1.L’utente deve essere autenticato  2.L’utente si deve trovare nella pagina “Metodi Pagamento” |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quanto l’utente inserisce quale tipologia di metodo di pagamento vuole inserire. 2. In funzione della scelta il sistema abilita i campi necessari all’inserimento. 3. L’utente compila i campi abilitati. 4. Il sistema verifica la correttezza dei valori inseriti ogni qual volta l’utente cambia campo di compilazione.    1. Se vi sono errori nei formati dei dati inseriti il sistema stamperà un avviso per correggerli 5. L’utente preme sul pulsante “+”.    1. Se non vi sono errori il sistema invia i dati inseriti dall’utente al DBMS    2. Se vi sono errori il sistema non invia i dati ma lascia stampati gli avvisi per la correzione. 6. Il sistema legge i dati e li manda al DBMS    1. In caso di carta già inserita si rimanda al caso d’uso “INVIO NOTIFICA DI ERRORE” 7. Il sistema mostra a sistema un messaggio di conferma. |
| **PUNTO DI ESTENSIONE:**  Nel punto 6.1 viene chiamato il caso d’uso “INVIO NOTIFICA DI ERRORE” |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente, svuotando i campi. |

|  |
| --- |
| **CASO D’USO: SELEZIONE METODO PAGAMENTO DI DEFAULT** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**   1. L’utente deve essere autenticato 2. L’utente deve trovarsi nella pagina “Metodi Pagamento” 3. Il sistema manda al DBMS i dati dell’utente 4. Il sistema compila la tabella con i metodi di pagamento inseriti dall’utente |
| **SEQUENZA DEGLI EVENTI:**   1. IL caso d’uso inizia quando l’utente apre il menu a tendina della carta che vuole settare come carta di default    1. Se l’utente sceglie il metodo di default già impostato il sistema disabilita il pulsante default 2. L’utente preme sul pulsante default 3. Il sistema invia le informazioni della carta da settare al DBMS 4. Il sistema mostra un messaggio di avvenuta modifica |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente. |

|  |
| --- |
| **CASO D’USO: MODIFICA LIMITI DI SPESA** |
| **ATTORI:**  Utente, DBMS |
| **PRECONDIZIONE:**   1. L’utente deve essere autenticato 2. L’utente deve trovarsi nella pagina “Metodi Pagamento” 3. Il sistema manda al DBMS i dati dell’utente 4. Il sistema compila la tabella con i metodi di pagamento inseriti dall’utente |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente apre il menu a tendina della carta a cui vuole modificare i limiti 2. L’utente preme sul pulsante limiti di spesa 3. Il sistema apre la pagina per cambiare i limiti di spesa alla carta selezionata 4. L’utente seleziona i valori dei limiti di spesa per i limiti giornalieri e mensili 5. Il sistema manda i dati al DBMS 6. Il sistema manda il dato selezionato al DBMS 7. Il sistema mostra un messaggio di avvenuta modifica |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente. |

**4.1.8. Casi d’uso – Invio/Richiesta Denaro**

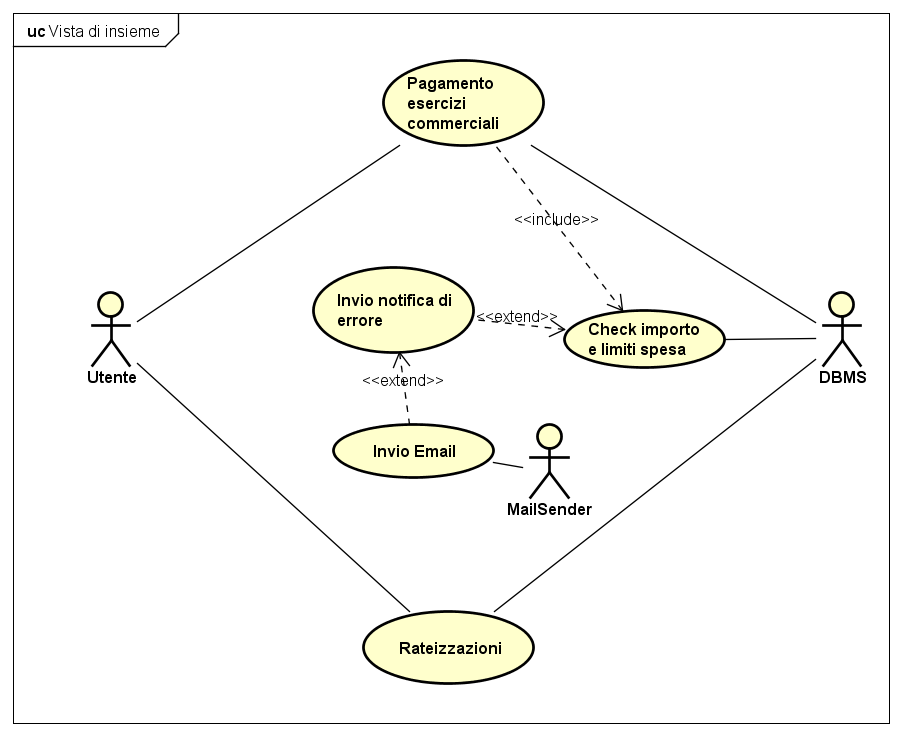
****

|  |
| --- |
| **CASO D’USO: INVIO DENARO** |
| **ATTORI:** Utente1, Utente2, DBMS |
| **PRECONDIZIONE:** Utente1 deve essere autenticato |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando Utente1 preme, dalla pagina principale, su link “Invia denaro ad un amico” 2. Il sistema reindirizza l’Utente1 alla pagina 3. L’Utente1 inserisce nel form email e/o cellulare del destinatario, l’importo e seleziona il tipo di pagamento 4. Il sistema manda al DBMS il tipo di pagamento scelto dall’utente e riempie il menu a tendina con i metodi di pagamento disponibili dell’utente 5. L’Utente1 seleziona il metodo e preme su conferma 6. Il sistema verifica la validità dei campi e se tutto è conforme il DBMS effettua la richiesta.    1. Se l’Utente2 non è registrato viene mandata un’email all’indirizzo specificato da Utente1, dove viene inserito il link di registrazione    2. Se l’Utente2 è registrato, viene invocato il caso d’uso “Accettazione/Rifiuto Richiesta”       * 1. Se viene effettuata la transazione, il sistema mostra un messaggio di conferma dell’operazione |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma di invio e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente, svuotando i campi. |

|  |
| --- |
| **CASO D’USO: ACCETTAZIONE/RIFIUTO RICHIESTA** |
| **ATTORI:** Utente1, Utente2, DBMS |
| **PRECONDIZIONE:** L’Utente2 ha ricevuto una richiesta da parte di Utente1 |
| **SEQUENZA DEGLI EVENTI:**   1. Utente2 è nella pagina di “Gestione delle richieste” 2. Il sistema carica la pagina con le richieste fatte a nome di Utente2, recuperandole tramite il DBMS 3. L’Utente2 può accettare o rifiutare le richieste    1. Se rifiuta, il DBMS cancella la richiesta e non avviene nessuna variazione di saldo    2. Se accetta, viene invocato il caso d’uso “check Importo e limite spesa”       * 1. Se termina correttamente il DBMS effettua la transazione e modifica i saldi di Utente1 e Utente2 |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma di invio e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente aggiornando la lista di richieste da accettare/rifiutare |

|  |
| --- |
| **CASO D’USO: RICHIESTA DENARO** |
| **ATTORI:** Utente1, Utente2, DBMS |
| **PRECONDIZIONE:** Utente1 deve essere autenticato |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando Utente1 preme, dalla pagina principale, su link “Richiedi denaro da un amico” 2. Il sistema reindirizza l’Utente1 alla pagina 3. L’Utente1 inserisce nel form email e/o cellulare del destinatario, l’importo e seleziona il tipo di pagamento sul quale vuole ricevere il denaro 4. Il sistema manda al DBMS il tipo di pagamento scelto dall’utente e riempie il menu a tendina con i metodi di pagamento disponibili dell’utente 5. L’Utente1 seleziona il metodo e preme su conferma 6. Il sistema verifica la validità dei campi e se tutto è conforme il DBMS effettua la richiesta.    1. Se l’Utente2 non è registrato viene mandata un’email all’indirizzo specificato da Utente1, dove viene inserito il link di registrazione    2. Se l’Utente2 è registrato, viene invocato il caso d’uso “Accettazione/Rifiuto Richiesta”       * 1. Se viene effettuata la transazione, il sistema mostra un messaggio di conferma dell’operazione. |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma di invio e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente, svuotando i campi. |

**4.1.9. Casi d’uso – Pagamento**

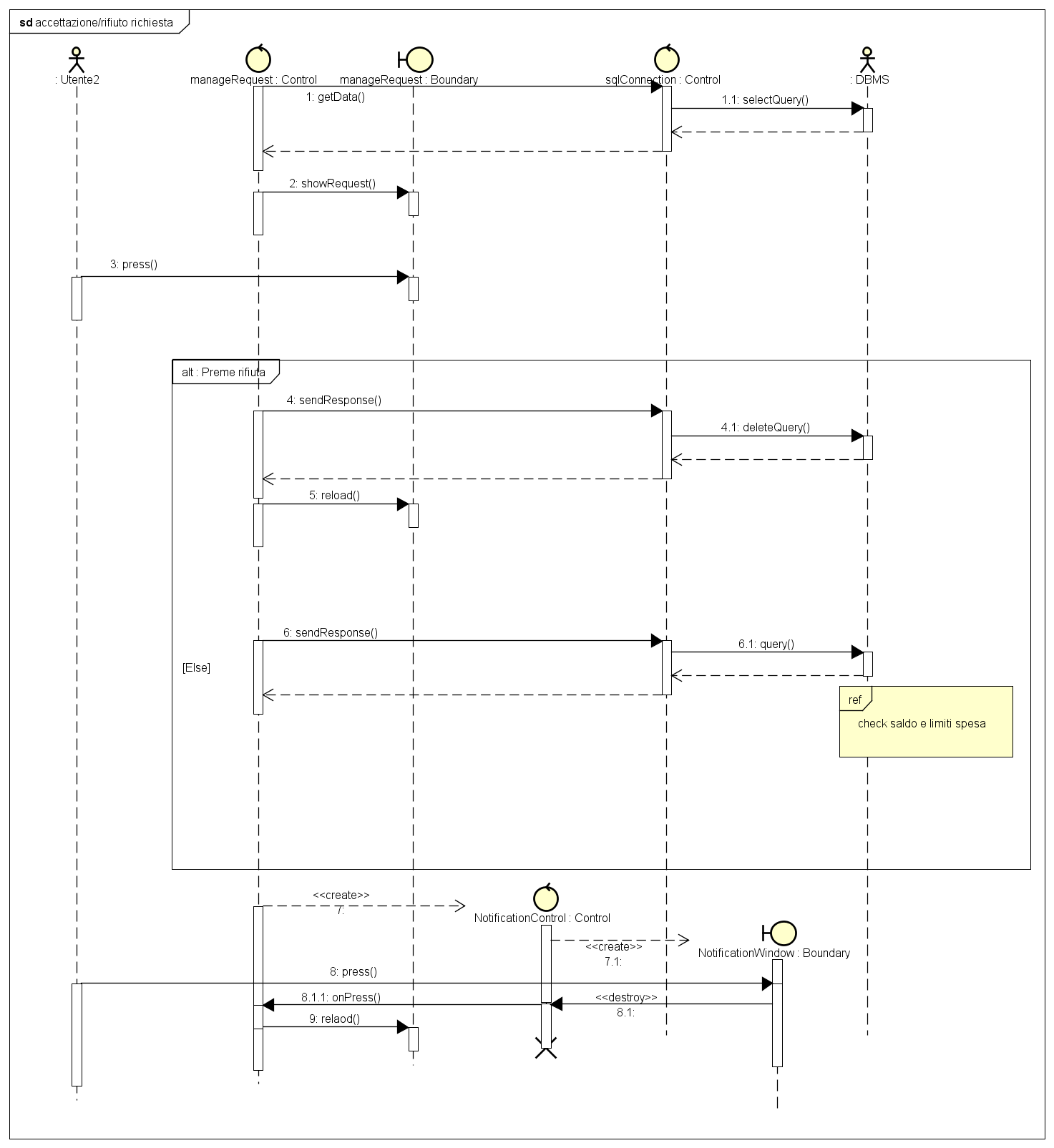
****

|  |
| --- |
| **CASO D’USO: PAGAMENTO ESERCIZI COMMERCIALI** |
| **ATTORI:** Utente, DBMS |
| **PRECONDIZIONE:** L’utente deve essere autenticato |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme, dalla pagina principale, su link “Effettua i tuoi pagamenti” 2. Il sistema reindirizza l’utente alla pagina 3. L’utente inserisce nel form il nome dell’esercizio commerciale, l’importo e seleziona il tipo di pagamento 4. Il sistema manda al DBMS il tipo di pagamento scelto dall’utente e riempie il menu a tendina con i metodi di pagamento disponibili dell’utente 5. L’utente seleziona il metodo e preme su conferma 6. Viene invocato il caso d’uso “Check importo e limiti spesa”    1. Se non ci sono messaggi di errore viene registrata la transazione e si aggiorna quindi il saldo dell’utente |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente, svuotando i campi. |

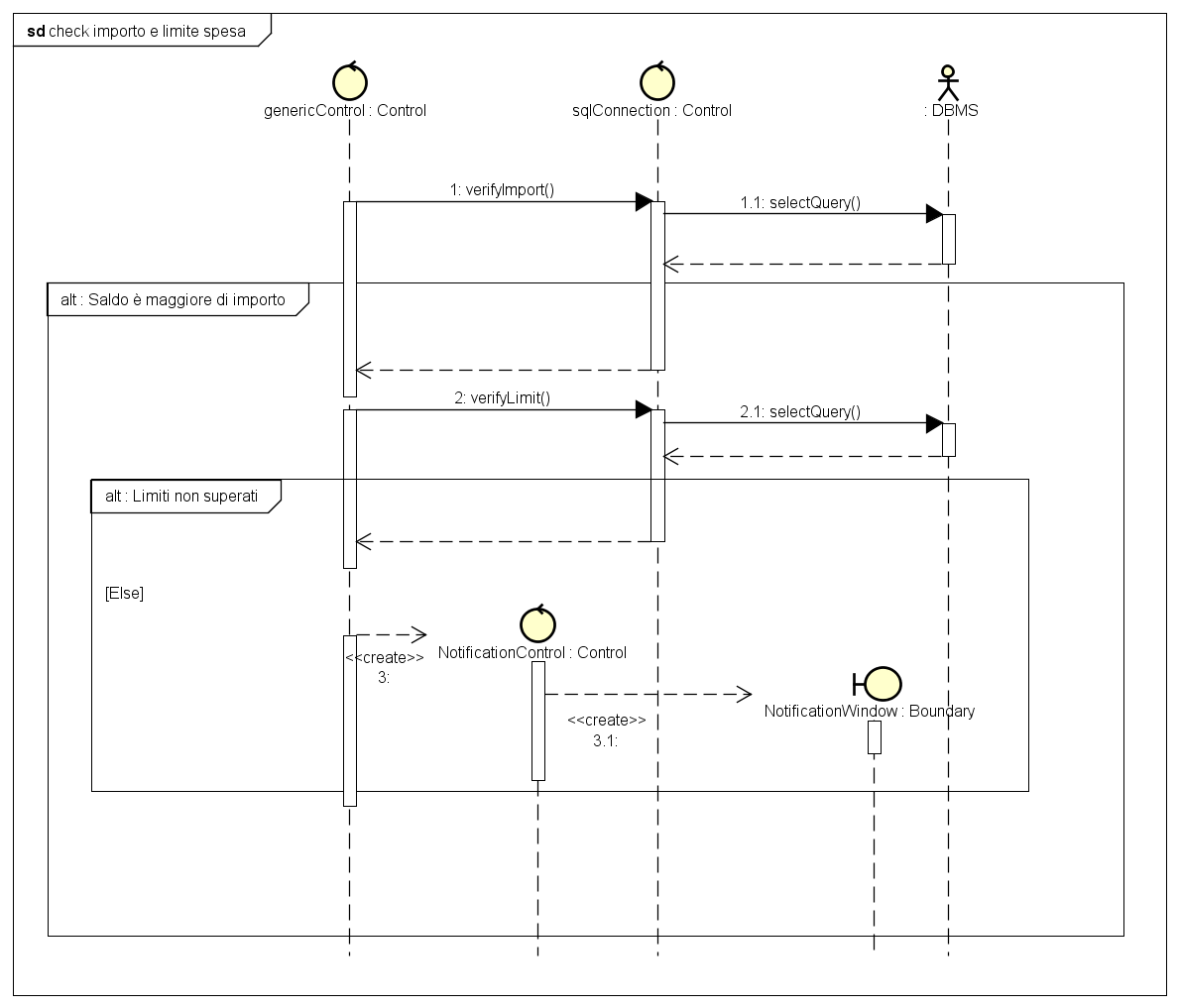
|  |
| --- |
| **CASO D’USO: RATEIZZAZIONI** |
| **ATTORI:** Utente, DBMS |
| **PRECONDIZIONE:** L’utente deve essere autenticato |
| **SEQUENZA DEGLI EVENTI:**   1. Il caso d’uso inizia quando l’utente preme, dalla pagina principale, su link “Effettua pagamenti periodici” 2. Il sistema reindirizza l’utente alla pagina 3. L’utente inserisce nel form il nome dell’esercizio commerciale, l’importo, la data di avvio, ogni quanti giorni avviene il pagamento, il numero di rate e seleziona il tipo di pagamento 4. Il sistema manda al DBMS il tipo di pagamento scelto dall’utente e riempie il menu a tendina con i metodi di pagamento disponibili dell’utente 5. L’utente seleziona il metodo e preme su conferma 6. Il sistema ogni 24h controlla se è venuto il giorno del pagamento, effettuando la transazione invocando il caso d’uso “Check importo e limiti spesa”    1. Se non ci sono messaggi di errore viene registrata la transazione e si aggiorna quindi il saldo dell’utente |
| **POSTCONDIZIONE:**  Una volta visualizzato il messaggio di conferma e premuto OK sulla rispettiva finestra il sistema rimane sulla pagina corrente, svuotando i campi. |

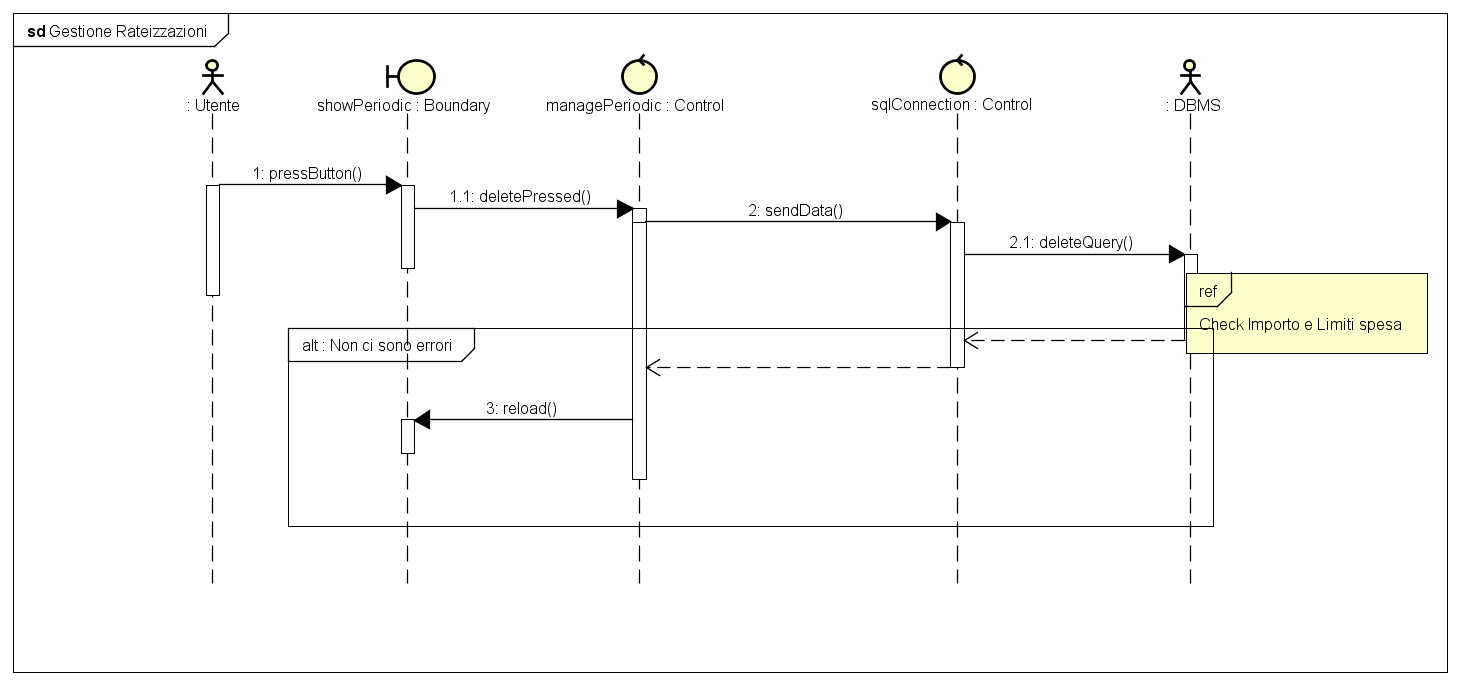
**4.2. Modello dinamico – Diagrammi di sequenza**

**Accettazione/Rifiuto Richiesta**

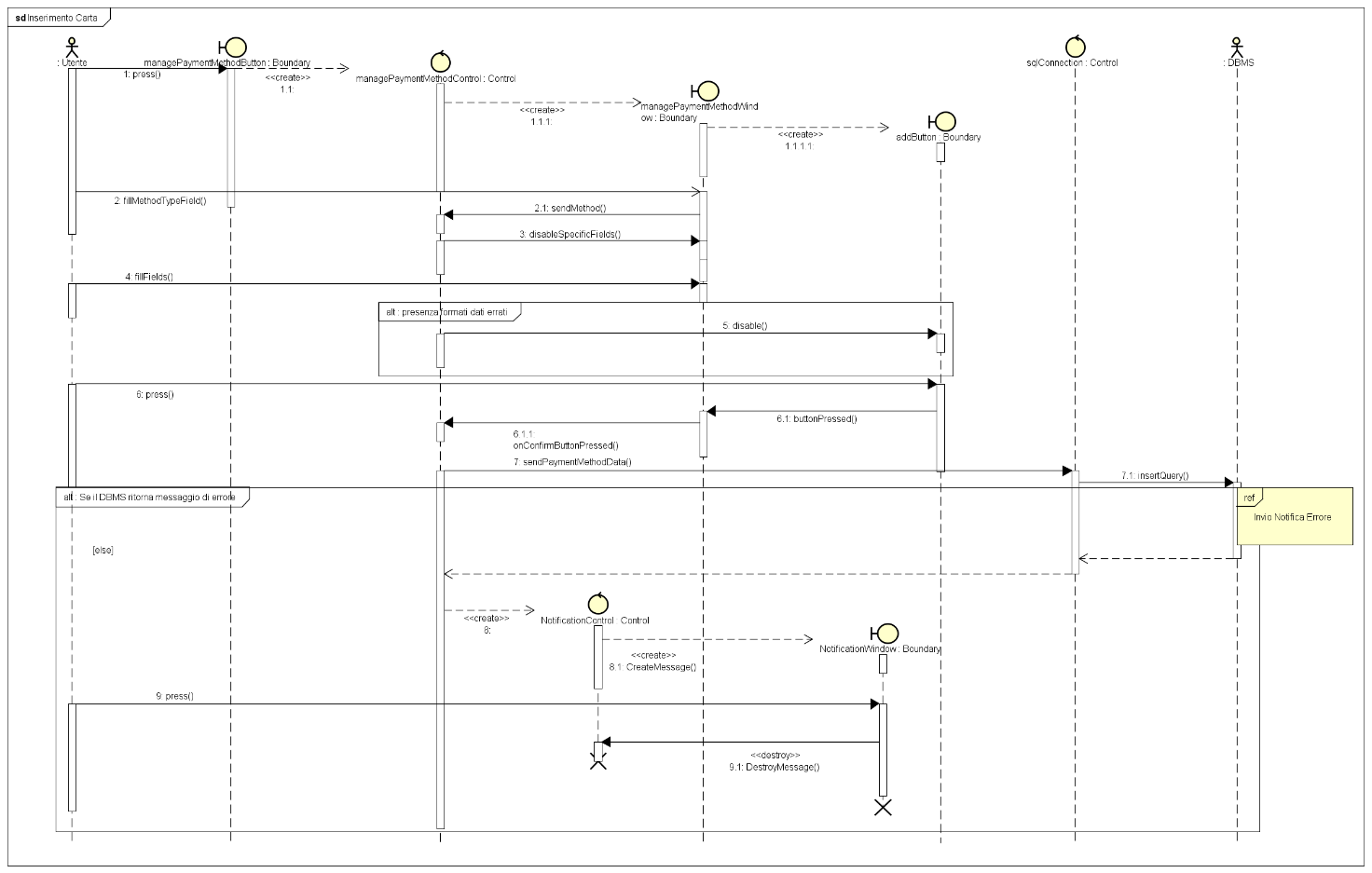
****

**Check Importo e limiti spesa**

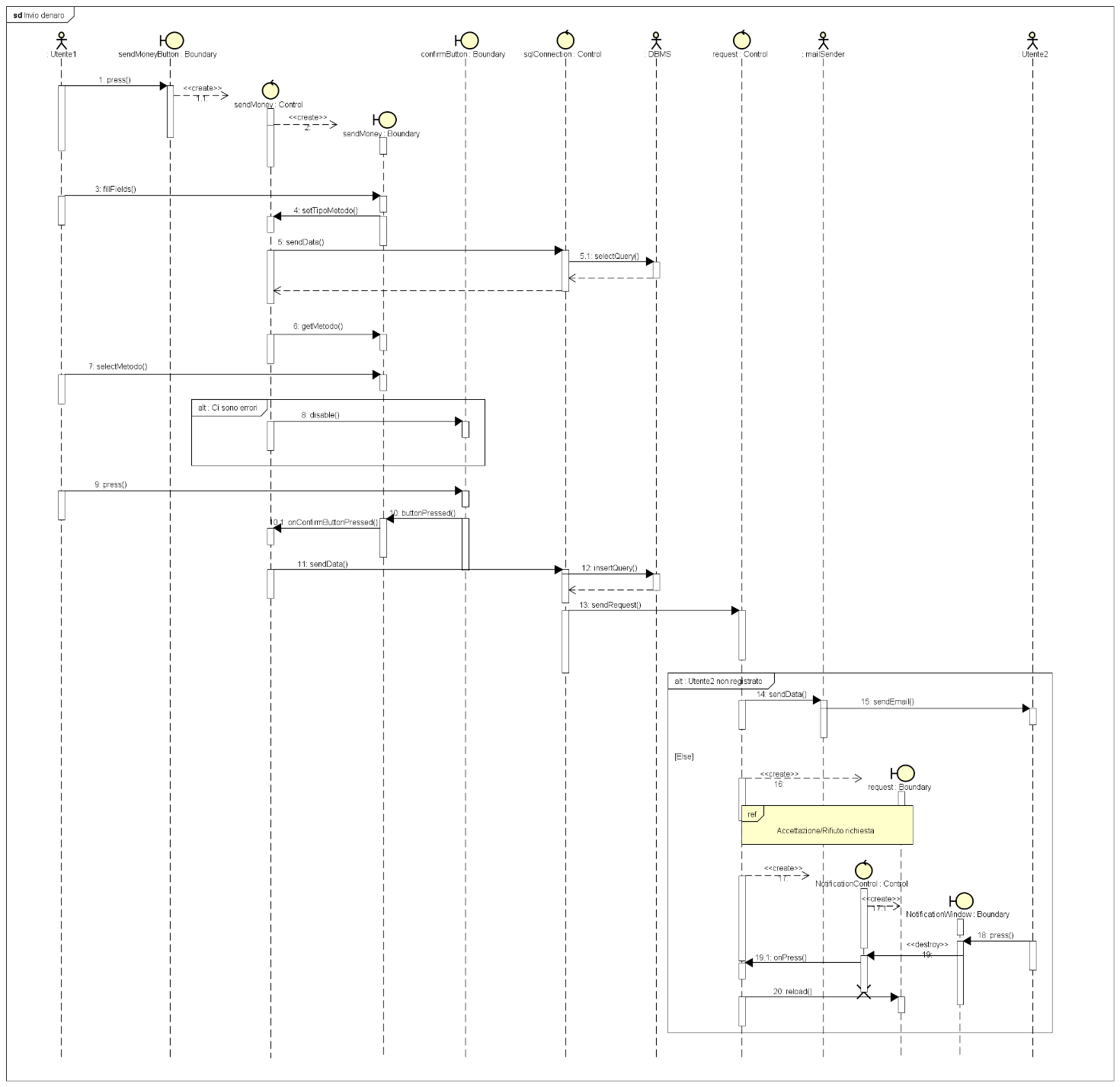
****

**Gestione Rateizzazioni**

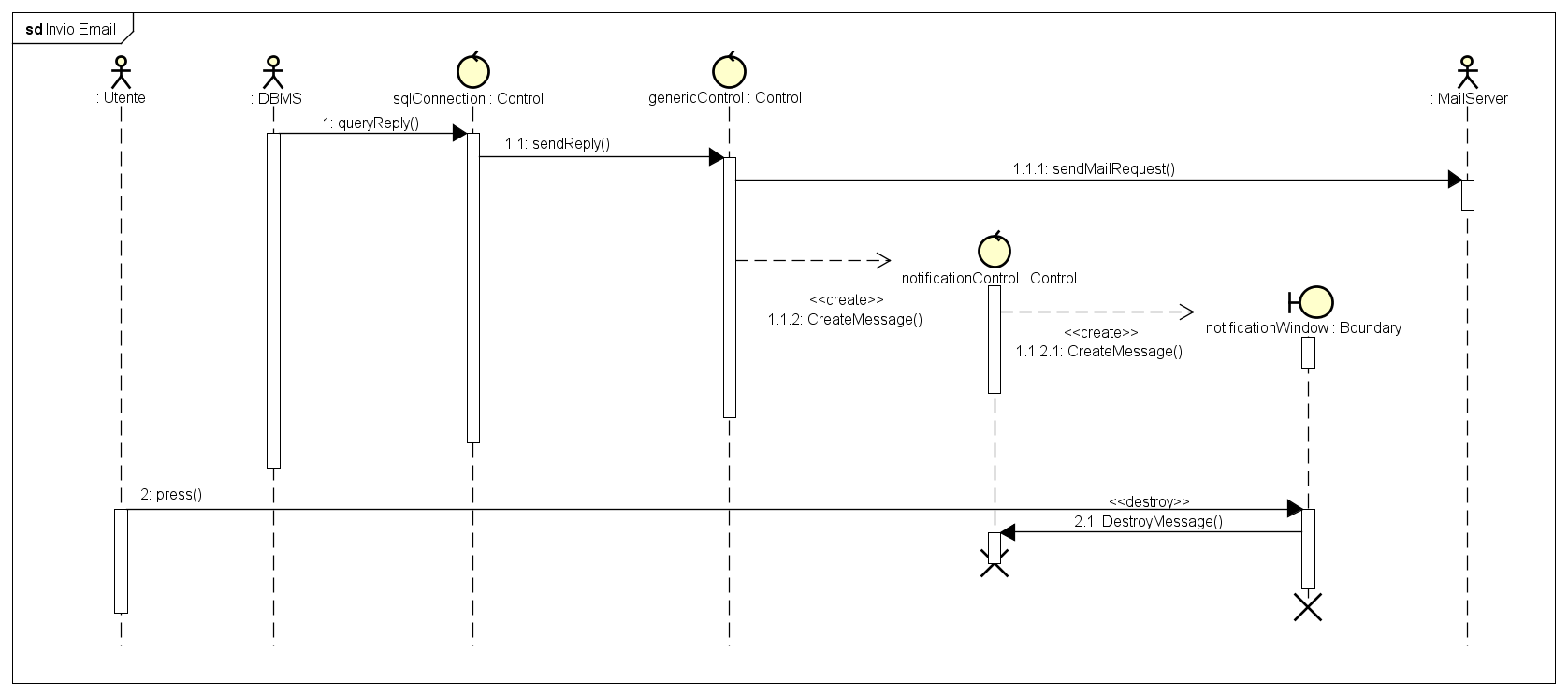
**Inserimento metodo di Pagamento**



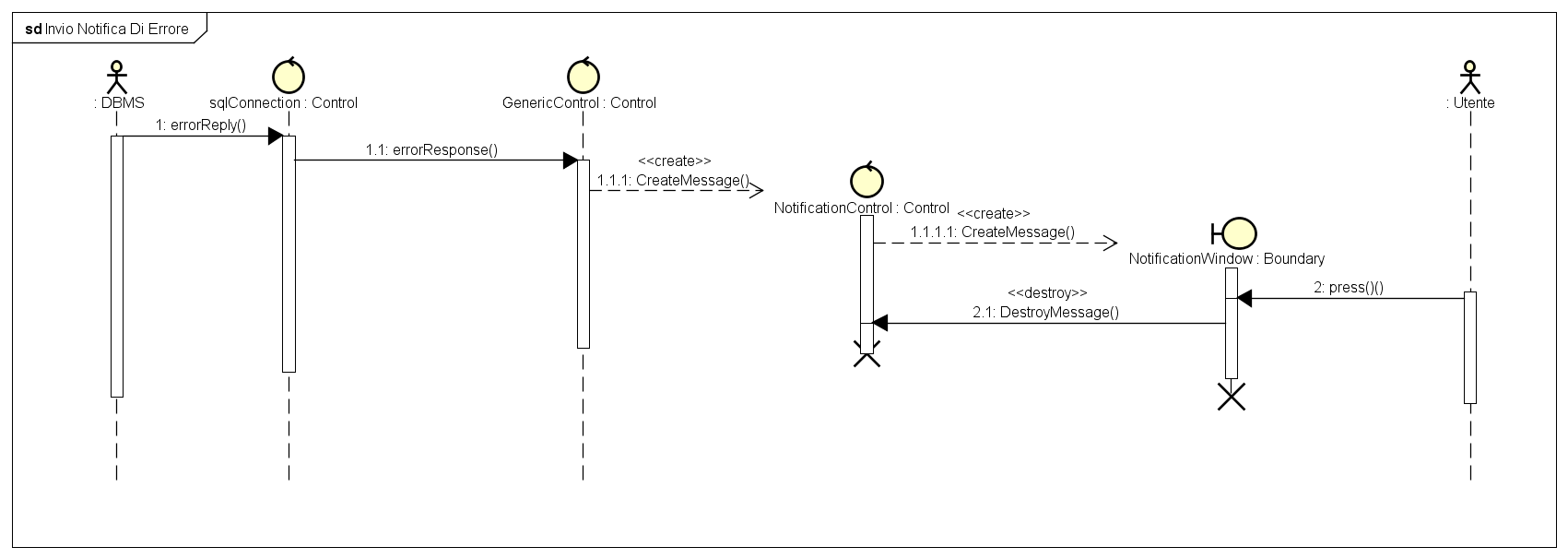
**Invio Denaro**



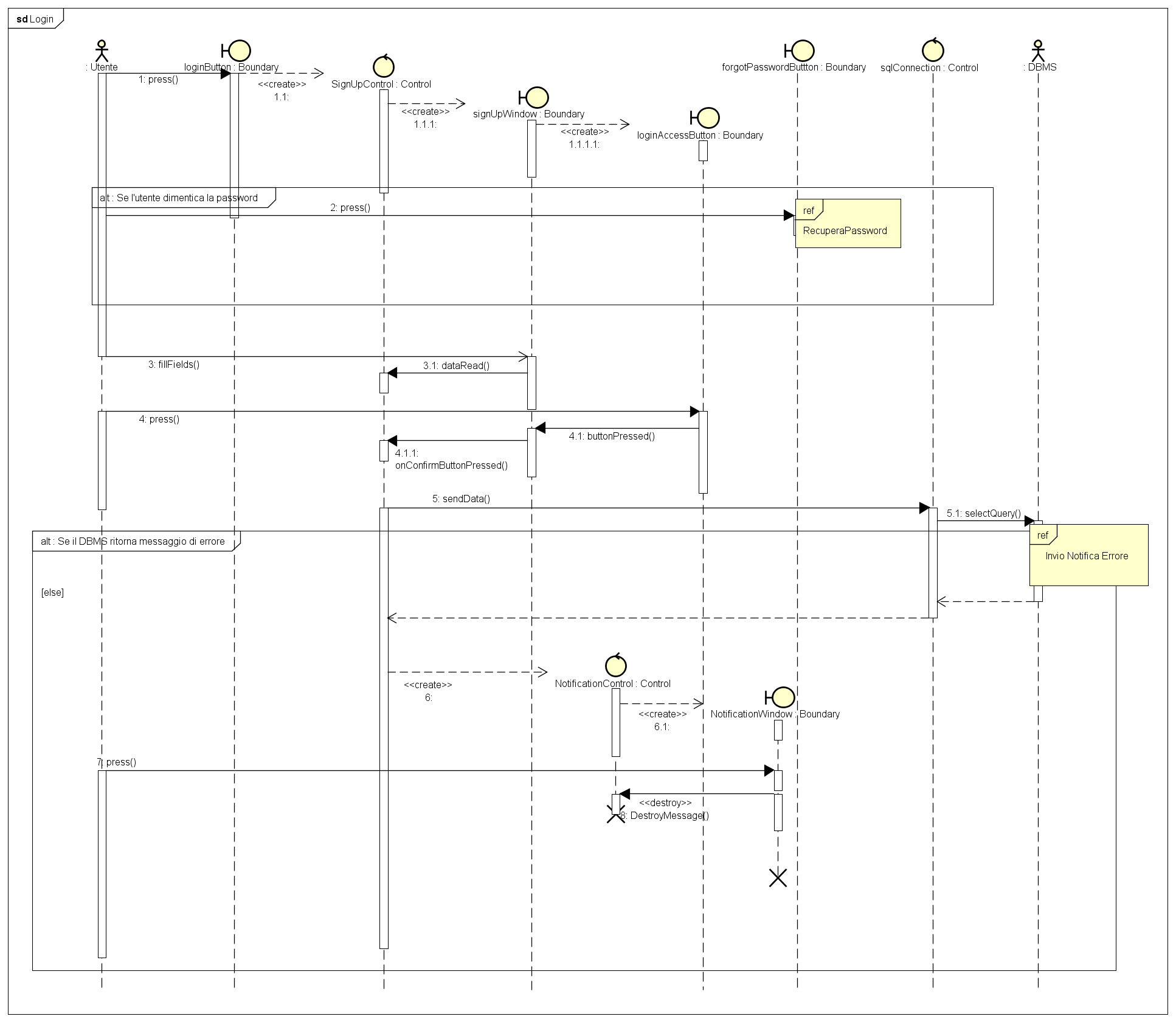
**Invio Email**

****

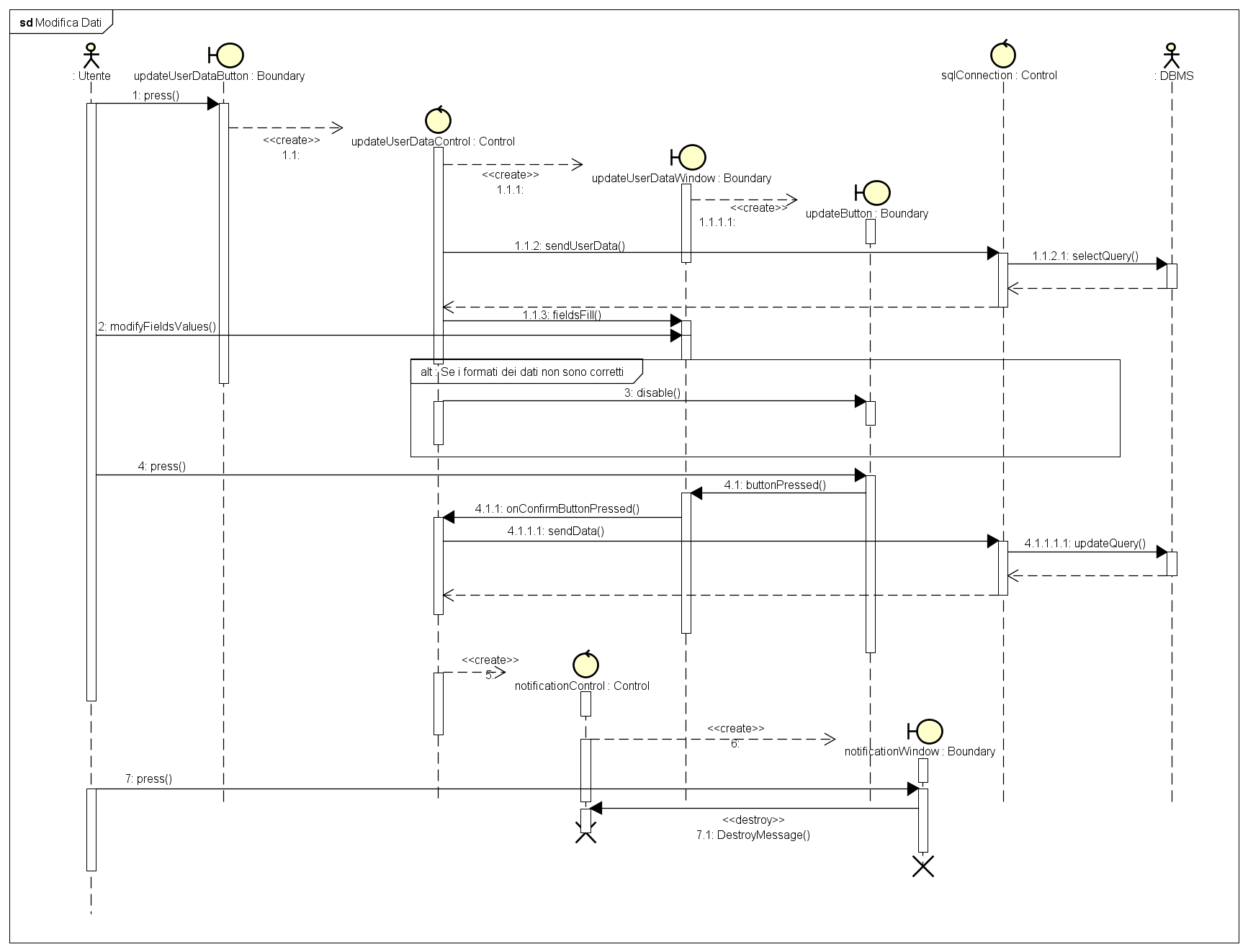
**Invio Notifica di Errore**

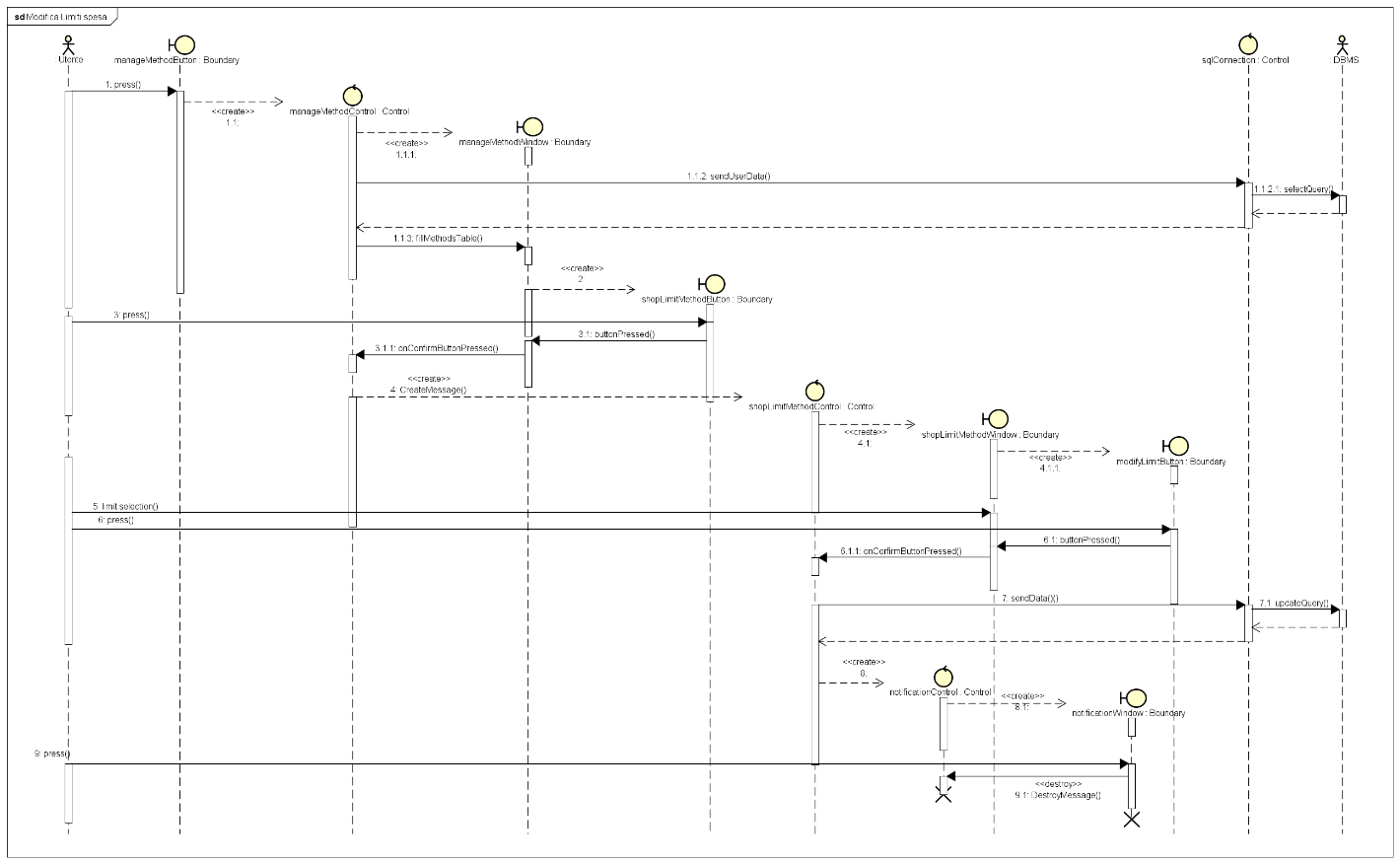
****

**Login**

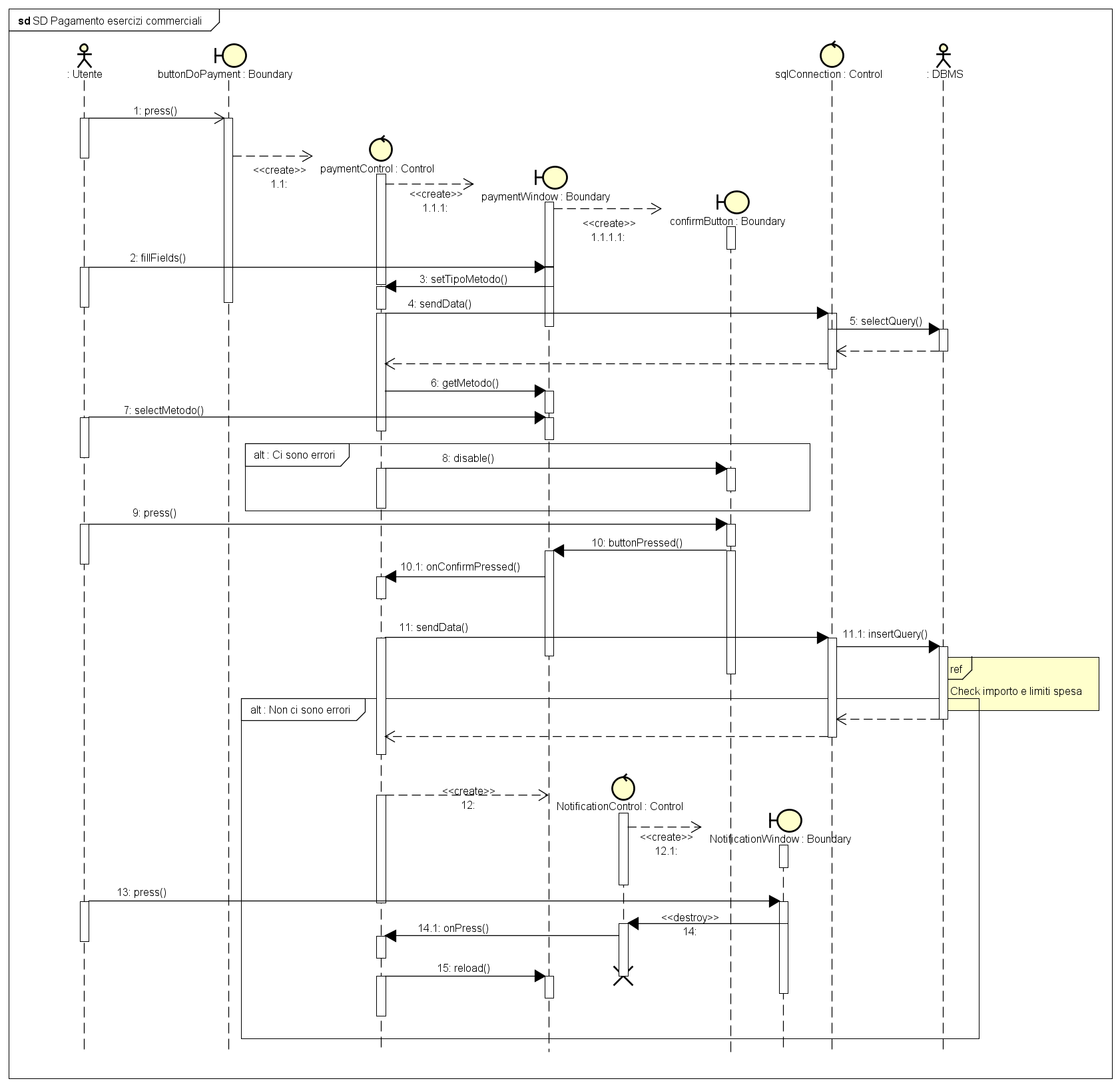
****

**Modifica Dati**

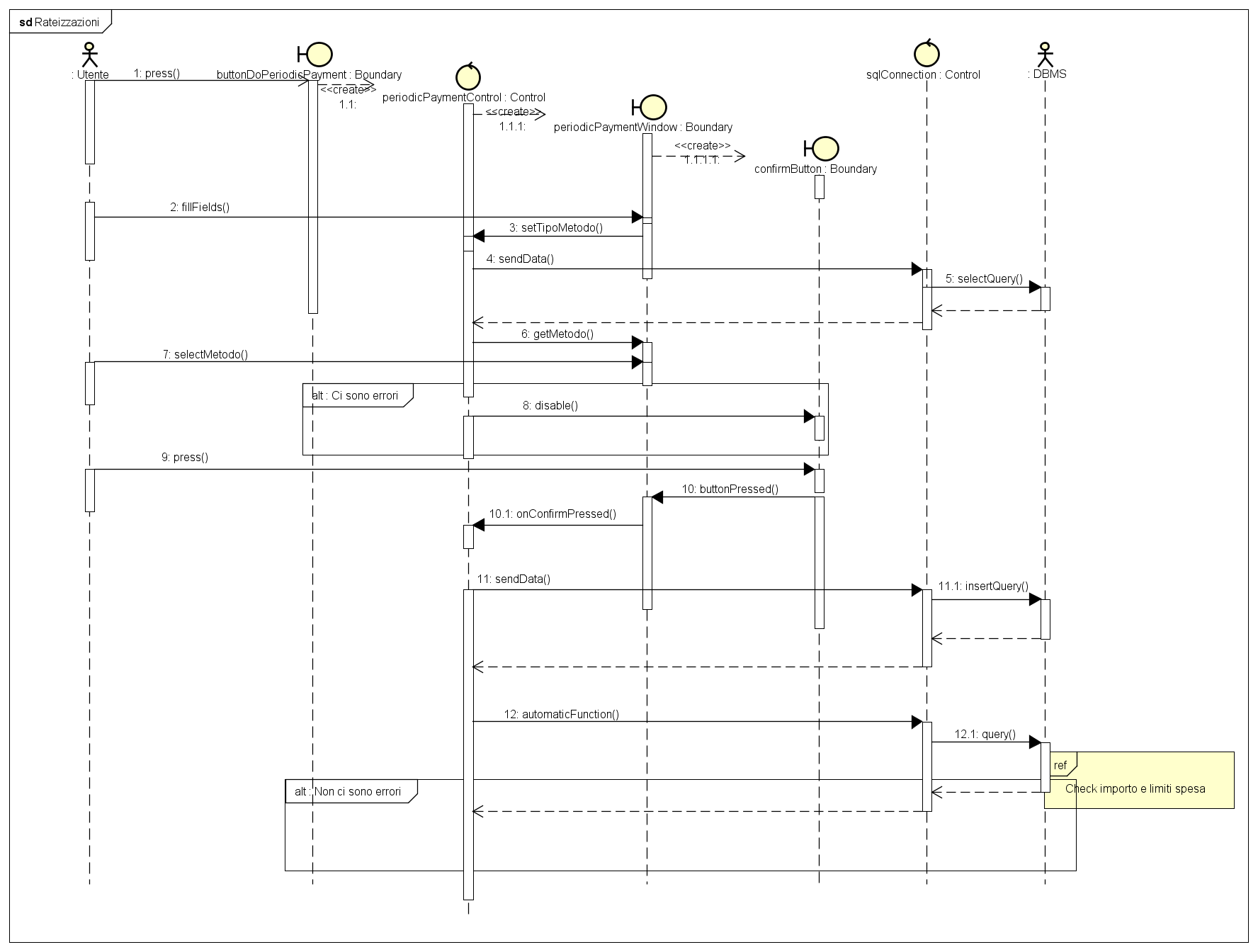
****

**Modifica Limiti Spesa**

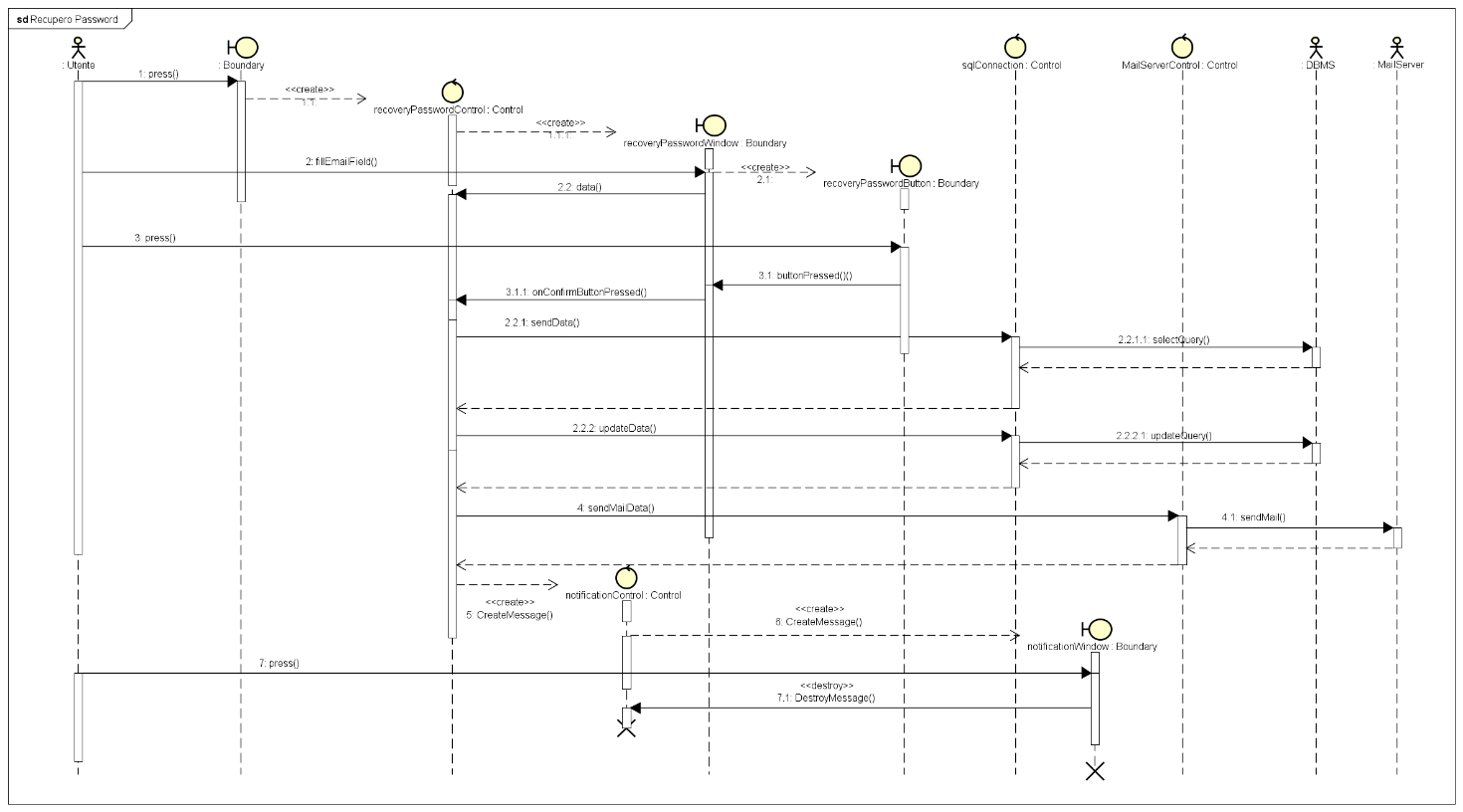
**Pagamento Esercizi Commerciali**

****

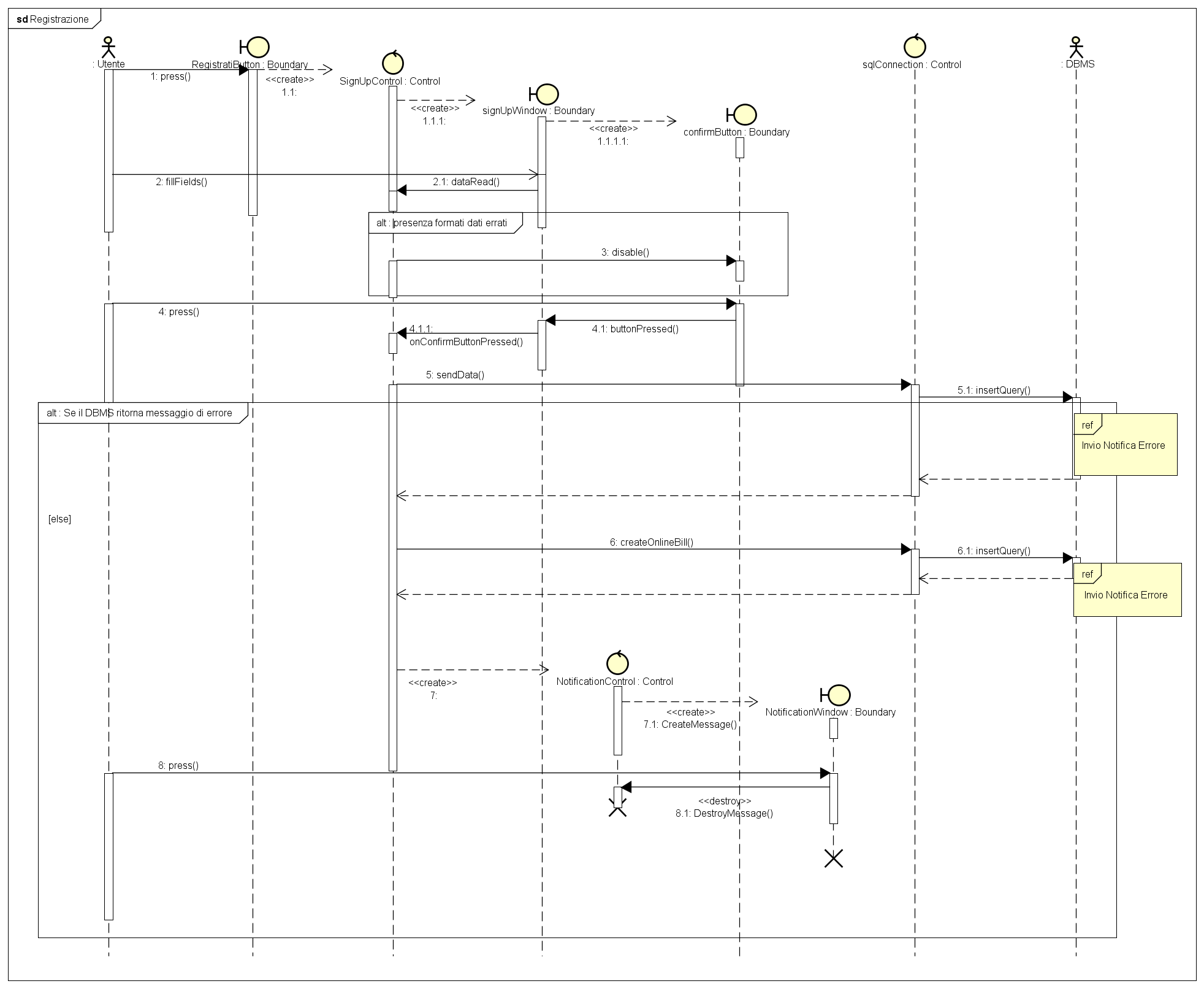
**Rateizzazioni**

****

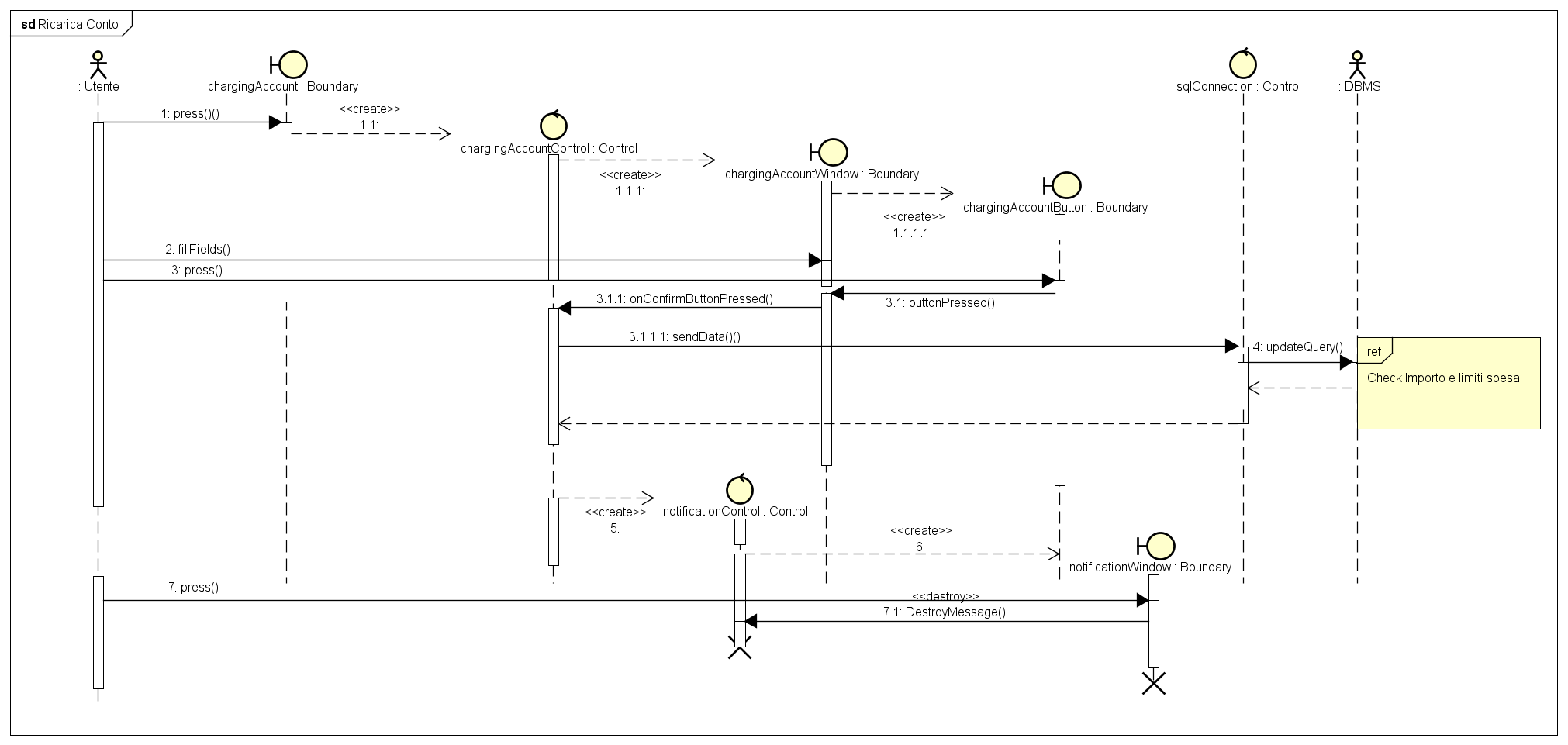
**Recupero Password**



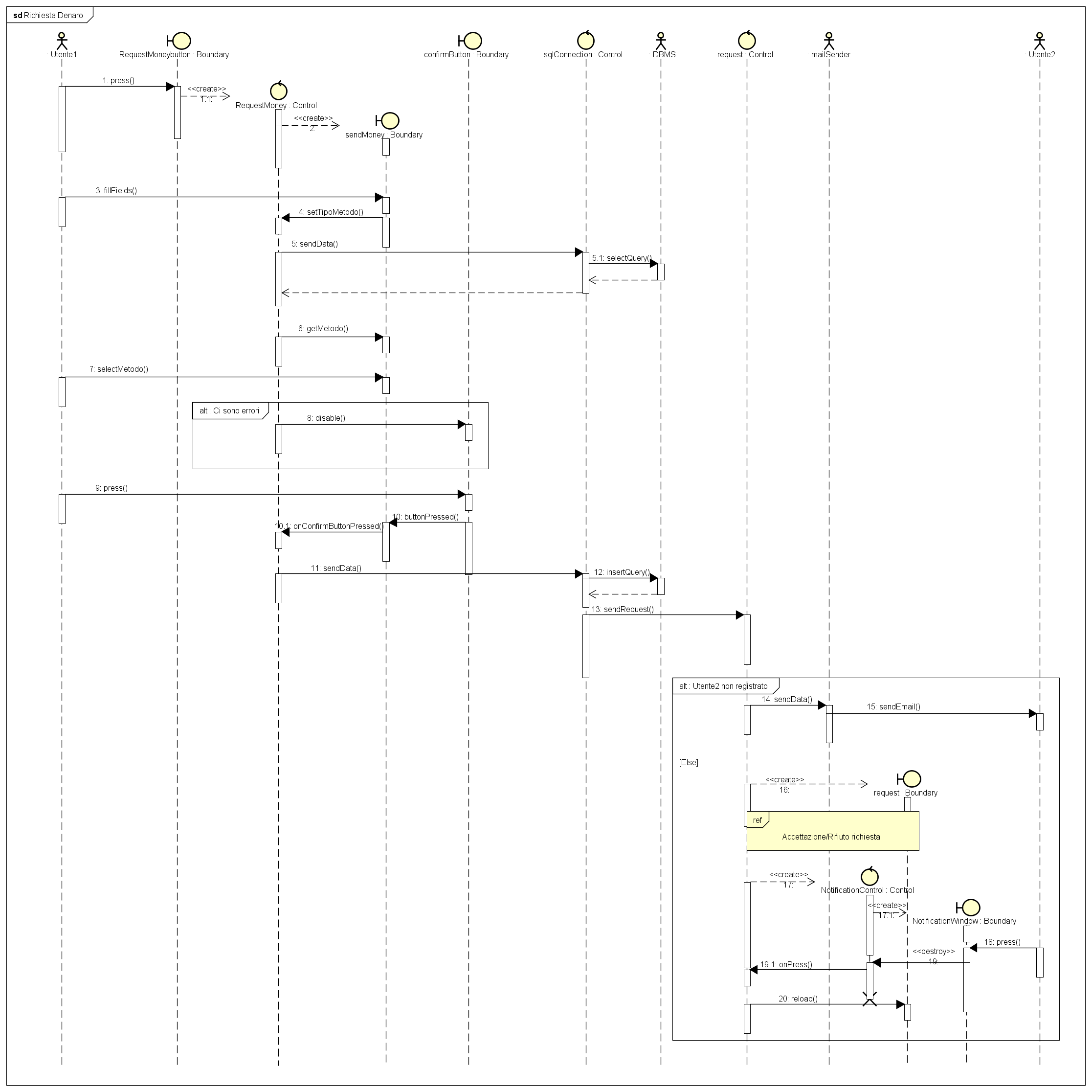
**Registrazione**

****

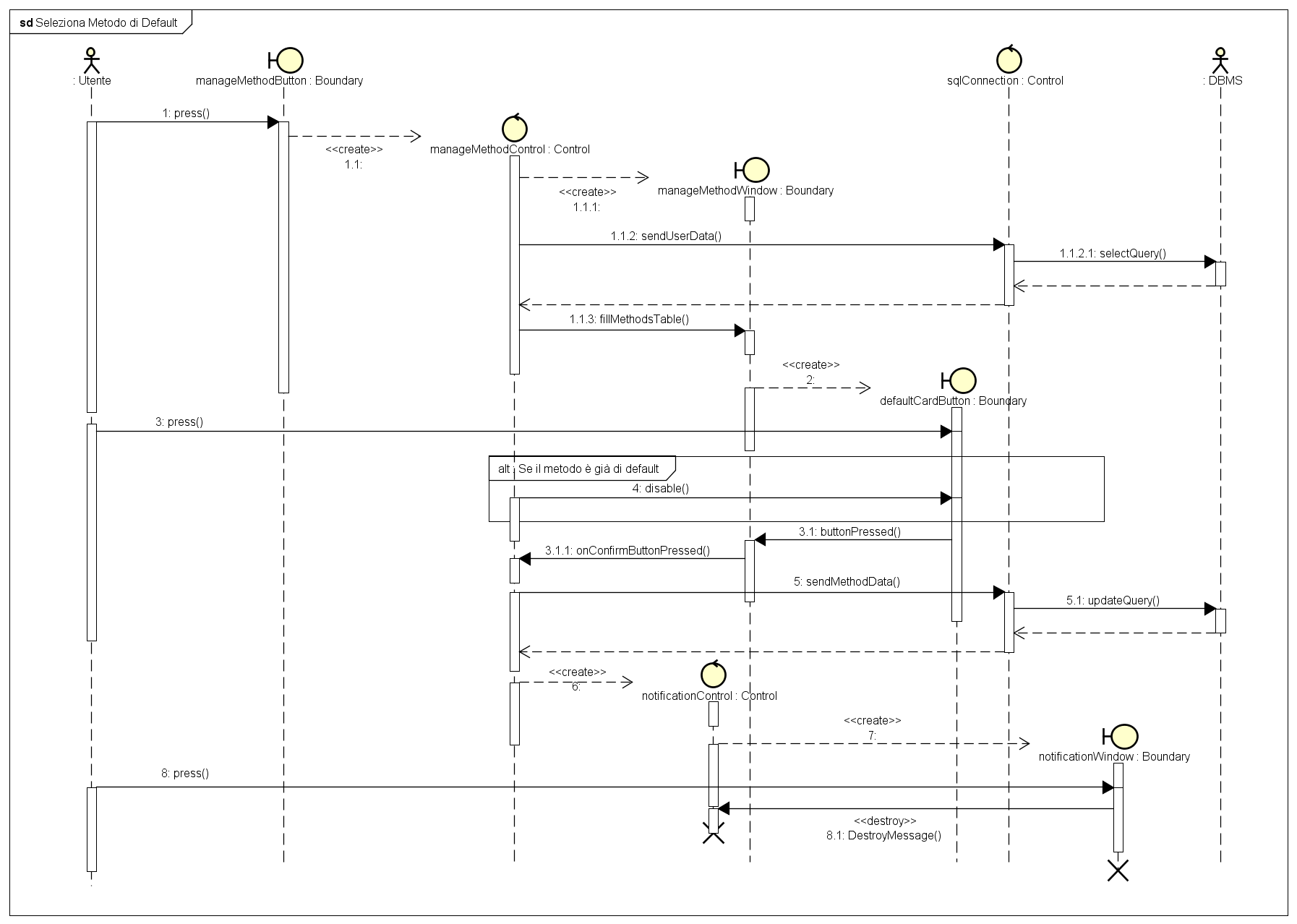
**Ricarica Conto**

****

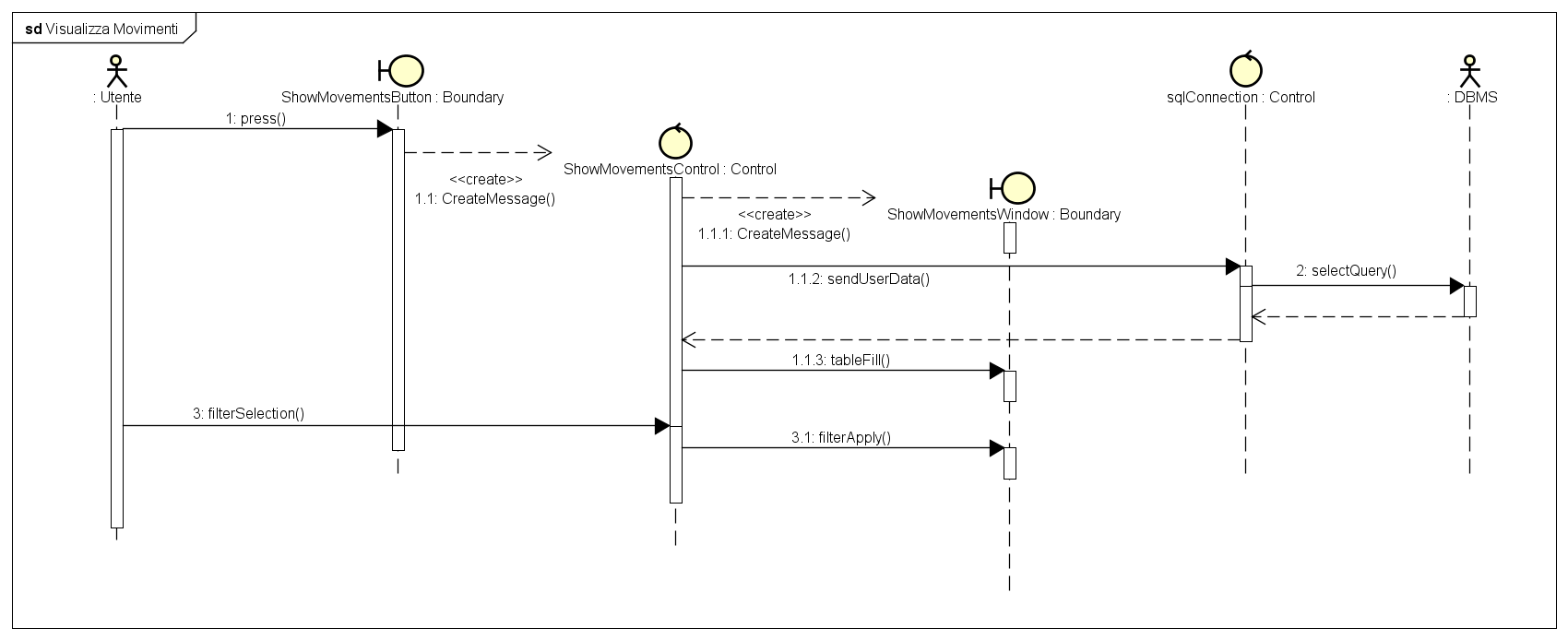
**Richiesta Denaro**

****

**Seleziona Metodo Default**

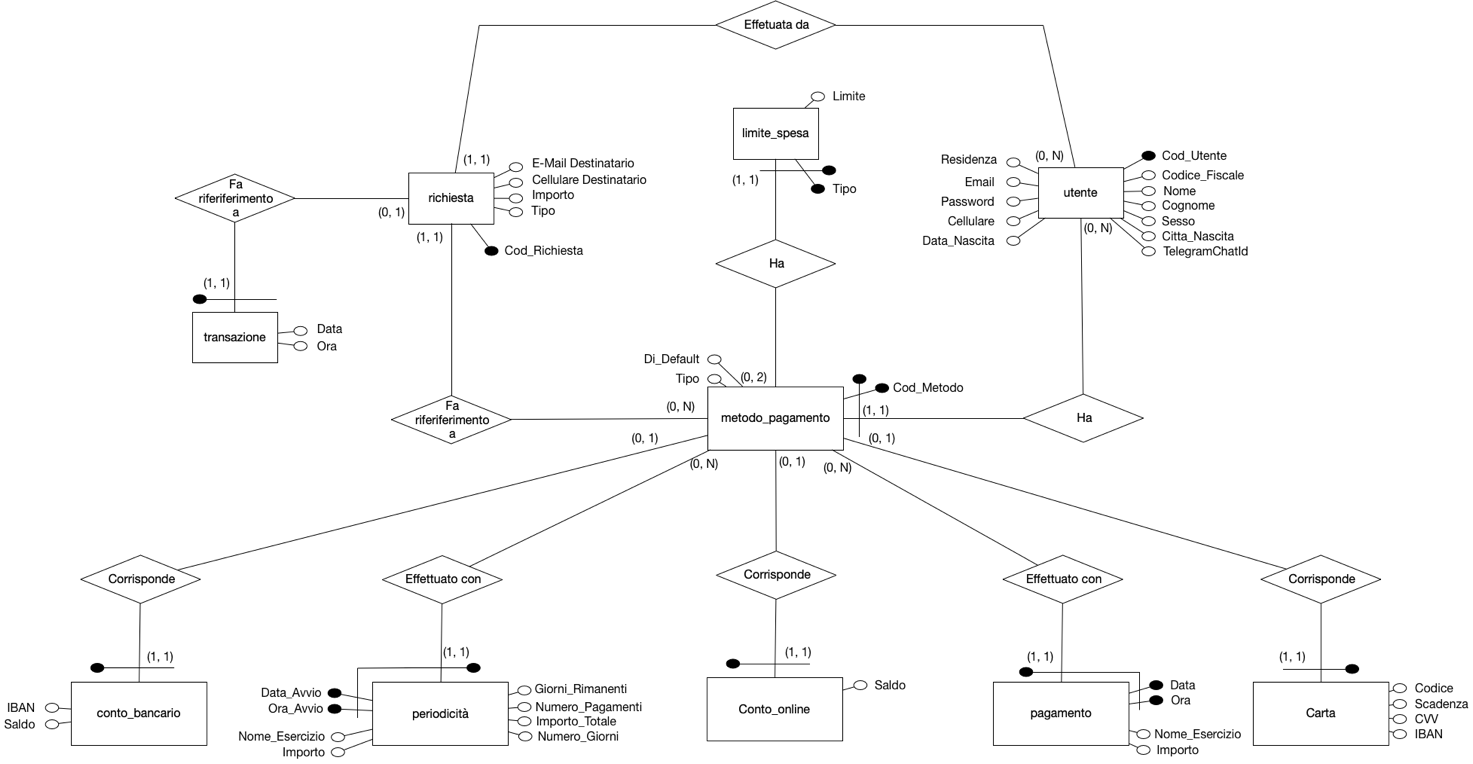
****

**Visualizza Movimenti**

****

1. **Architettura software proposta**
   1. **Panoramica**

Per il sistema è stata scelta un’architettura client-server. I nodi client sono i dispositivi degli utenti che accedono al sito web dell’applicazione, o gli smartphone/tablet su cui è installata l’app. Il nodo server è rappresentato dal server del committente.

* 1. **Gestione dei dati persistenti – Progetto del Database**
     1. **Modello** **ER**
     2. **Progetto logico**

Si utilizza un DB relazionale, basato sulle relazioni esistenti fra le varie tabelle. Il DB è stato realizzato con l’ausilio di MySQL.

Date le cardinalità uno a uno, sono state eliminate le seguenti relazioni:

1. La relazione “**Effettuata da**” tra le entità “richiesta” e “utente” viene rimossa mediante la chiave esterna da “Ref\_Mittente” di “richiesta” a “Cod\_Utente” di “utente”.
2. La relazione “**Ha**” tra le entità “metodo\_pagamento” e “utente” viene rimossa mediante la chiave esterna da “Ref\_Utente” di “metodo\_pagamento” a “Cod\_Utente” di “utente”.
3. La relazione “**Fa riferimento a**” tra le entità “transazione” e “richiesta” viene rimossa mediante la chiave esterna da “Ref\_Richiesta” di “transazione” a “Cod\_Richiesta” di “richiesta”.
4. La relazione “**Effettuata da**” tra le entità “richiesta” e “utente” viene rimossa mediante la chiave esterna da “Ref\_Mittente” di “richiesta” a “Cod\_Utente” di “utente”.
5. La relazione “**Corrisponde**” tra le entità “conto\_bancario” e “metodo\_pagamento” viene rimossa mediante la chiave esterna da “Ref\_Utente” di “conto\_bancario” a “Cod\_Utente” di “utente” e “Ref\_Metodo” di “conto\_bancario” e “Cod\_Metodo” di “metodo\_pagamento”.
6. La relazione “**Corrisponde**” tra le entità “conto\_online” e “metodo\_pagamento” viene rimossa mediante la chiave esterna da “Ref\_Utente” di “conto\_online” a “Cod\_Utente” di “utente” e “Ref\_Metodo” di “conto\_online” e “Cod\_Metodo” di “metodo\_pagamento”.
7. La relazione “**Corrisponde**” tra le entità “carta” e “metodo\_pagamento” viene rimossa mediante la chiave esterna da “Ref\_Utente” di “conto\_bancario” a “Cod\_Utente” di “utente” e “Ref\_Metodo” di “conto\_bancario” e “Cod\_Metodo” di “metodo\_pagamento”.
8. La relazione “**Effettuato con**” tra le entità “periodicità” e “metodo\_pagamento” viene rimossa mediante la chiave esterna da “Ref\_Mittente” di “periodicità” a “Cod\_Utente” di “utente” e “Ref\_Metodo” di “periodicità” e “Cod\_Metodo” di “metodo\_pagamento”.
9. La relazione “**Effettuato con**” tra le entità “pagamento” e “metodo\_pagamento” viene rimossa mediante la chiave esterna da “Ref\_Mittente” di “pagamento” a “Cod\_Utente” di “utente” e “Ref\_Metodo” di “pagamento” e “Cod\_Metodo” di “metodo\_pagamento”.

In conclusione, il database progetto è costituito dalle seguenti entità:

**Utente** (Cod\_Utente, Codice\_Fiscale, Nome, Cognome, Sesso, Data\_Nascita, Città\_Nascita, Residenza, Email, Password, Cellulare)

**Metodo\_Pagamento** (Ref\_Utente, Cod\_Metodo, Di\_Default, Tipo)

**Limite\_Spesa** (Ref\_Utente, Ref\_Metodo, Limite, Tipo)

**Richiesta** (Cod\_Richiesta, Ref\_Mittente, Ref\_Metodo, Email\_Destinatario, Cellulare\_Destinatario, Importo, Tipo)

**Transazione** (Ref\_Richiesta, Data, Ora)

**Conto\_Online** (Ref\_Utente, Ref\_Metodo, Saldo)

**Conto\_Bancario** (Ref\_Utente, Ref\_Metodo, IBAN, Saldo)

**Pagamento** (Ref\_Mittente, Ref\_Metodo, Nome\_Esercizio, Importo, Data, Ora)

**Periodicità** (Ref\_Mittente, Ref\_Metodo, Data, Ora, Nome\_Esercizio, Importo, Numero\_Giorni, Giorni\_Rimanenti, Numero\_Pagamenti, Importo\_Totale)

**Carta** (Ref\_Utente, Ref\_Metodo, Codice, Scadenza, CVV, Saldo, IBAN)

* + 1. **Descrizione delle Tabelle**

**Utente**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Cod\_Utente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Codice\_Fiscale** | Char(16) (richiesto: si, chiave primaria: no) |
| **Nome** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Cognome** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Sesso** | Char(1) (richiesto: si, chiave primaria: no) |
| **Data\_Nascita** | Date (richiesto: si, chiave primaria: no) |
| **Città\_Nascita** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Residenza** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Email** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Password** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Cellulare** | Char(10) (richiesto: si, chiave primaria: no) |

**Metodo\_Pagamento**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Utente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Cod\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **Di\_Default** | TINYINT(1) (richiesto: no, chiave primaria: no) |
| **Tipo** | Varchar(45) (richiesto: si, chiave primaria: no) |

**Limite\_Spesa**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Utente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **Limite** | Float (richiesto: no, chiave primaria: no) |
| **Tipo** | Char(1) (richiesto: si, chiave primaria: si) |

**Richiesta**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Cod\_Richiesta** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Mittente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **Email\_Destinatario** | Varchar(45) (richiesto: no, chiave primaria: no) |
| **Cellulare\_Destinatario** | Char(10) (richiesto: no, chiave primaria: no) |
| **Importo** | Float (richiesto: si, chiave primaria: si) |
| **Tipo** | Varchar(45) (richiesto: si, chiave primaria: si) |

**Transazione**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Richiesta** | INT(11) (richiesto: si, chiave primaria: si) |
| **Data** | Date (richiesto: si, chiave primaria: no) |
| **Ora** | Time (richiesto: si, chiave primaria: no) |

**Conto\_Online**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Utente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **Saldo** | Float (richiesto: si, chiave primaria: no) |

**Conto\_Bancario**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Utente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **IBAN** | Char(27) (richiesto: si, chiave primaria: no) |
| **Saldo** | Float (richiesto: si, chiave primaria: no) |

**Pagamento**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Mittente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **Nome\_Esercizio** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Importo** | Float (richiesto: si, chiave primaria: no) |
| **Data** | Date (richiesto: si, chiave primaria: si) |
| **Ora** | Time (richiesto: si, chiave primaria: si) |

**Periodicità**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Mittente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **Data** | Date (richiesto: si, chiave primaria: si) |
| **Ora** | Time (richiesto: si, chiave primaria: si) |
| **Nome\_Esercizio** | Varchar(45) (richiesto: si, chiave primaria: no) |
| **Importo** | Float (richiesto: si, chiave primaria: no) |
| **Numero\_Giorni** | INT(11) (richiesto: no, chiave primaria: no) |
| **Giorni\_Rimanenti** | INT(11) (richiesto: no, chiave primaria: no) |
| **Numero\_Pagamenti** | INT(11) (richiesto: no, chiave primaria: no) |
| **Importo\_Totale** | Float (richiesto: no, chiave primaria: no) |

**Carta**

|  |  |
| --- | --- |
| **Campo** | **Tipo** |
| **Ref\_Utente** | INT(11) (richiesto: si, chiave primaria: si) |
| **Ref\_Metodo** | INT(11) (richiesto: si, chiave primaria: si) |
| **Codice** | BIGINT(20) (richiesto: si, chiave primaria: no) |
| **Scadenza** | DATE (richiesto: si, chiave primaria: no) |
| **CVV** | INT(11) (richiesto: si, chiave primaria: no) |
| **Saldo** | Float (richiesto: si, chiave primaria: no) |
| **IBAN** | Char(27) (richiesto: no, chiave primaria: no) |

* 1. **Decomposizione in sottoinsiemi**

**//BHOOOO**

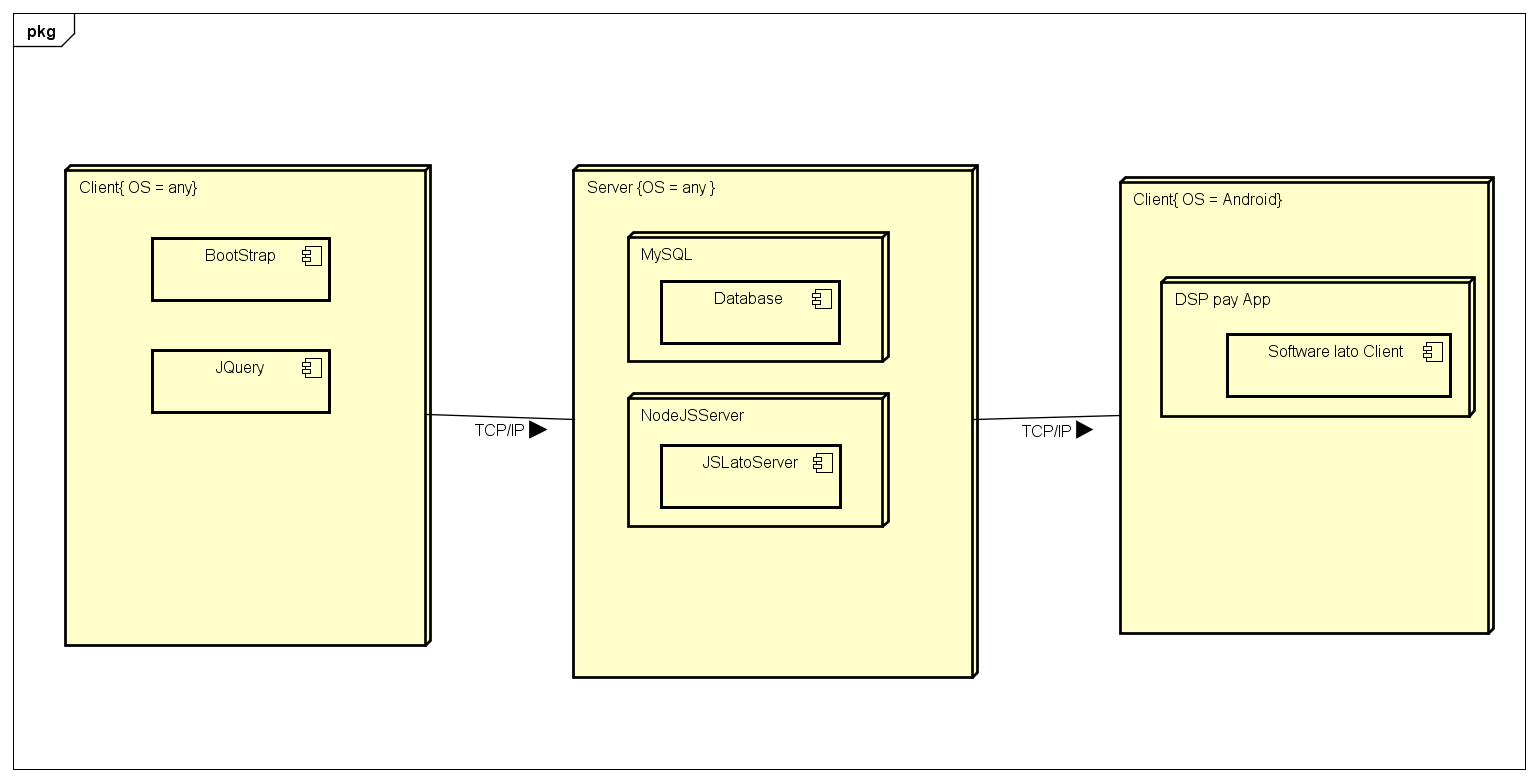
* 1. **Mappatura Hardware/Software**

La mappatura è stata effettuata sulla base del modello architetturale client/server.

Il nodo server contiene:

* un’istanza di un server NodeJS, che si occuperà di servire le richieste dei client,
* un’istanza di MySQL, che si occuperà della gestione dei contenuti del Database, utilizzando il modulo mysql di NodeJS
* il software lato client

I nodi client rappresentano i dispositivi degli utenti che si collegano ad internet e accedono al sito web dell’applicazione.



1. **Implementazione**
   1. **Guida all’installazione**
2. Eseguire lo script “create.sql” sul DB.
3. Installare node
4. Installare cordova da terminale con:
5. npm install -g cordova
6. Aprire il file “server.js” e andare in fondo al file e cambiare i dati di accesso al server MySQL impostando username e password corretti.
7. Aprire un terminale e navigare nella cartella “Server” del progetto e digitare:

node server.js

1. Aprire un altro terminale e navigare nella cartella “Sito” del progetto e digitare:

cordova run browser

**Account del DB**

**E-mail:** davide.lavalle.45@gmail.com

**Password:** 123Davide

**E-mail:** kiro90@gmail.com

**Password:** 123Angelo

**E-mail:** salvatorepisciotta1997@gmail.com

**Password:** ciaoGatto3

**E-mail:** salvopisciotta97@gmail.com

**Password:** ciao123A

* 1. **Applicazione Android**

Per poter usufruire dell’applicazione DSPpay è necessario inserire nella schermata iniziale l’indirizzo IP del server.

Per ottenere tale indirizzo IP digitare ipconfig da terminale Windows oppure ifconfig da terminale UNIX.

1. **Divisione dei compiti**

Il progetto è stato sviluppato concordando inizialmente che non ci sarebbero stati ruoli fissi ma ognuno avrebbe contribuito in ogni parte dello sviluppo.

Con il progredirsi della complessità del progetto si è reso necessario che un elemento del gruppo (Davide Lavalle) si occupasse principalmente della parte grafica, mentre, gli altri due componenti (Salvatore Pisciotta e Giuseppe Peri), si sono occupati dello sviluppo funzionale del software, infine, durante la fase di testing e di sviluppo dell’apk, tutto il team è stato coinvolto per appunto testare e risolvere i vari bug riscontrati indipendentemente dalla natura grafica o funzionale del bug.