

Progetto per il corso di Programmazione, servizio di home banking

# BANKING

MIRKO DI LUCIA

**Università di Pisa**

Ing. Informatica

## DOCUMENTO DI ANALISI

L'applicazione permette ad un utente di visionare i movimenti bancari del proprio conto corrente e delle proprie carte di credito nonché di inviare denaro ad altri utenti iscritti al servizio.

## MOCKUP DELL'APPLICAZIONE

Banking

Data inizio:

1/06/2017

Data fine:

10/06/2017

Carta selezionata

Tutte

Data esecuzione	Nome	Beneficiario	Descrizione
2017-06-30 14:41	Pagamento POS		Pagamento POS presso farmacia S.Maria
2017-06-14 22:01	Bonifico bancario	aaa@aaa.com	Bonifico bancario da parte di Gincarlo Fumagalli
2017-05-27 10:34	Ricarica prepagata	aaa@aaa.com	Ricarica prepagata presso sportello bancario
2017-05-13 17:49	Pagamento e-commerce		Pagamento E-Commerce a favore di Amazon.com

Saldo cliente: 1025.89€

Tutte

Cliente

Iban

Quantita

Invia Denaro

Revoca

Tabella movimenti

Categorizzazione spese

Periodo	Barra Blu	Barra Grigia
2017-04	~920	~1050
2017-05	~720	~800
2017-06	~1150	~980

## CASI D'USO

- All'avvio dell'applicazione, vengono mostrati nella tabella i movimenti degli ultimi 30 giorni viene mostrato anche il saldo contabile, nel grafico a barre i movimenti di un intervallo di date presente nella configurazione (ultima settimana, ultimo mese, ultimi 6 mesi), nel grafico a torta verranno

mostrate il totale sulle categorie di spesa negli ultimi 30 giorni.

- IF l'utente seleziona una delle proprie carte di credito
  - L'applicazione mostra gli ultimi movimenti bancari associati a quella carta, in particolare vengono aggiornati la tabella dei movimenti, il saldo contabile, il saldo disponibile, il grafico a barre e il grafico a torta.
- IF L'utente seleziona un intervallo di date
  - L'applicazione mostra i movimenti inerenti al solo intervallo di date sulla tabella, sul grafico a barre e sul saldo contabile.
- IF L'utente seleziona una carta di credito e inserisce un utente, un iban e una quantità di denaro e preme il pulsante "Invia Denaro" l'applicazione aggiornerà la tabella dell'utente inserendo il nuovo movimento generato e aggiornerà il saldo contabile all'ultimo movimento.
- IF L'utente seleziona un movimento sulla tabella e preme il pulsante "Revoca" (se il movimento corrisponde a un trasferimento di denaro con data di esecuzione minore di 6 ore) l'applicazione aggiornerà la tabella rimuovendo il movimento bancario selezionato.

•

- 

## **FILE DI CONFIGURAZIONE IN XML**

All'avvio l'applicativo legge dal file di configurazione le seguenti proprietà

- Indirizzo IP del client.
- Indirizzo IP e porta del server di log.
- Indirizzo IP e porta del dbms.
- Username e password del dbms.
- Username e password per l'accesso alle proprie carte di credito e ai propri movimenti
- Modalità del grafico a barre (quotidiano, settimanale o mensile) e l'intervallo di mesi da prendere in considerazione.

## **SERVER DI LOG REMOTO**

L'applicativo invia una attività come stringa di testo in formato XML, la stringa contiene informazioni come il tipo di evento il nome dell'applicativo, l'indirizzo IP del client e timestamp, che viene inviato al server remoto per ogni evento di seguito:

- Apertura / Chiusura dell'applicativo.
- Login.
- Cambiamento della carta di credito.
- Cambiamento intervallo di date.
- Selezione di un elemento della tabella.
- Invio denaro.
- Revoca transazione.
- Sollevazione dell'eccezione `ClienteNonTrovatoException`.
- Sollevazione dell'eccezione `SaldoNonSufficienteException`.

## **DATABASE**

L'applicativo effettua alcune operazioni con il DATABASE:

- Accesso: viene verificato se email e password corrispondo all'utente.
- Ottieni movimenti per intervallo di date e per carta di credito selezionata.
- Ottieni movimenti (quotidiani, settimanali o mensili) cliente per il grafico a barre.
- Ottieni somma delle spese raggruppate per categoria e filtrati per intervallo di dati e carta di credito selezionata.
- Ottieni Carte di credito del cliente.

- Invia denaro effettuando dei controlli sul possesso della carta di credito sull'esistenza del destinatario e sul saldo sufficiente per la carta di credito.
- Revoca l'invio di denaro se il movimento selezionato corrisponde a un trasferimento interno e il movimento è stato effettuato nelle ultime 6 ore
- Aggiornamento del saldo sulla carta di credito e sul totale dell'utente.
- Ottenimento del saldo della carta di credito e sul totale dell'utente.

## **CACHE LOCALE DEGLI INPUT**

- Alla chiusura il sistema salva su file binario i filtri precedentemente applicati (intervallo di date e carta selezionata).
- All'avvio il sistema carica dal file binario i dati precedentemente salvati e prova ad effettuare il login con le credenziali precedentemente inseriti e imposta i filtri di ricerca.

## DOCUMENTO DI PROGETTO

### Classi e package

- **Backend**
  - **GestoreDatabase**: permette all'applicativo di eseguire query sul database.
  - **LogServer**: programma con esecuzione autonoma, permette di tenere traccia degli eventi che si verificano sull'applicativo a seguito di azioni intraprese dell'utente. Appende ad un file xml le comunicazioni provenienti dal LogClient tramite socket.
- **Middleware**
  - **Exception**
    - **ClienteNonTrovatoException** classe rappresentante l'eccezione per il mancato ritrovamento dell'utente durante l'autenticazione o l'invio di denaro.
    - **IntervalloSuperatoException** classe rappresentante l'eccezione per il superamento delle 6 ore durante la revoca di un movimento bancario tra utenti.
    - **SaldoNonSufficienteException** classe rappresentante l'eccezione per il credito non sufficiente nella carta di credito durante l'invio di denaro ad un altro utente.
  - **LogClient** : gestisce l'invio di eventi alla classe LogServer tramite socket.
  - **Cliente** : contenitore per l'elemento cliente, contiene email, cognome, nome e saldo del cliente.
  - **Carta**: contenitore per l'elemento carta di credito, contiene email del cliente, nome della carta, iban, saldo.
  - **DatoGrafico** : contenitore per i dati da applicare al grafico, contiene l'etichetta e il valore numerico da mostrare.
  - **Movimento** : contenitore per i movimenti bancari delle carte di credito, contiene il nome dell'operazione la descrizione, il beneficiario, la cifra
  - **Configurazione** : viene istanziata da GestoreConfigurazione tramite file XML dopo aver validato lo schema associato, contiene i parametri per impostare la comunicazione con il database e la configurazione per la comunicazione con il LogServer, usa Serializable.
  - **Evento** : elemento serializzato in xml e inviato al server di log tramite il ClientLog, usa Serializable.

- **GestoreXML** : permette il caricamento di file xml a LogServer e GestoreConfigurazione, nonché la validazione delle Stringhe xml sia lette da file sia ricevute dal LogServer e passate per argomento.
- **GestoreCache** : legge e scrive i filtri applicati alla visualizzazione dei dati sui movimenti bancari come l'intervallo di date e la carta di credito selezionata, usa Serializable.
- **GestoreConfigurazione** : carica dal xml e valida tramite GestoreXML la configurazione per poter stabilire una connessione con il database e il server di log.
- **Frontend**
  - **SelettoreCarte** : rappresenta la ChoiceBoxContenente le carte di credito dell'utente, è possibile impostare come prima opzione "Tutte" per poter selezionare tutte le carte insieme.
  - **SelettoreIntervalloDate**: rappresenta l'oggetto per poter selezionare un intervallo di date, contiene 2 datepicker che rappresentano la data di inizio e la data di fine per l'intervallo stesso la loro selezione comporta la chiamata delle funzioni cambioDataIniziale(), cambioDataFinale() e cambioIntervallo() contenuti nell'interfaccia AscoltatoreCambioData.
  - **GraficoTorta**: implementa un grafico a torta contenente le spese sostenute dall'utente raggruppate per categoria di spesa, estende PieChart.
  - **GraficoBarre**: implementa un grafico a barre che mostra a seconda della configurazione il totale dei movimenti quotidiani, mensili oppure settimanali per un intervallo di date espresso in mesi, estende BarChart.
  - **Main**: entry point dell'applicazione, crea tutti gli elementi grafici mostrati all'utente come il selettore delle carte di credito, i selettori di data per selezionare l'intervallo di date il grafico a barre e la tabella dei movimenti, la sezione per l'invio del denaro e il grafico a torta contenente le categorizzazioni delle spese.
  - **TabellaMovimentiCliente**: permette la visualizzazione dei movimenti del cliente e della carta di credito selezionata, estende TableView.

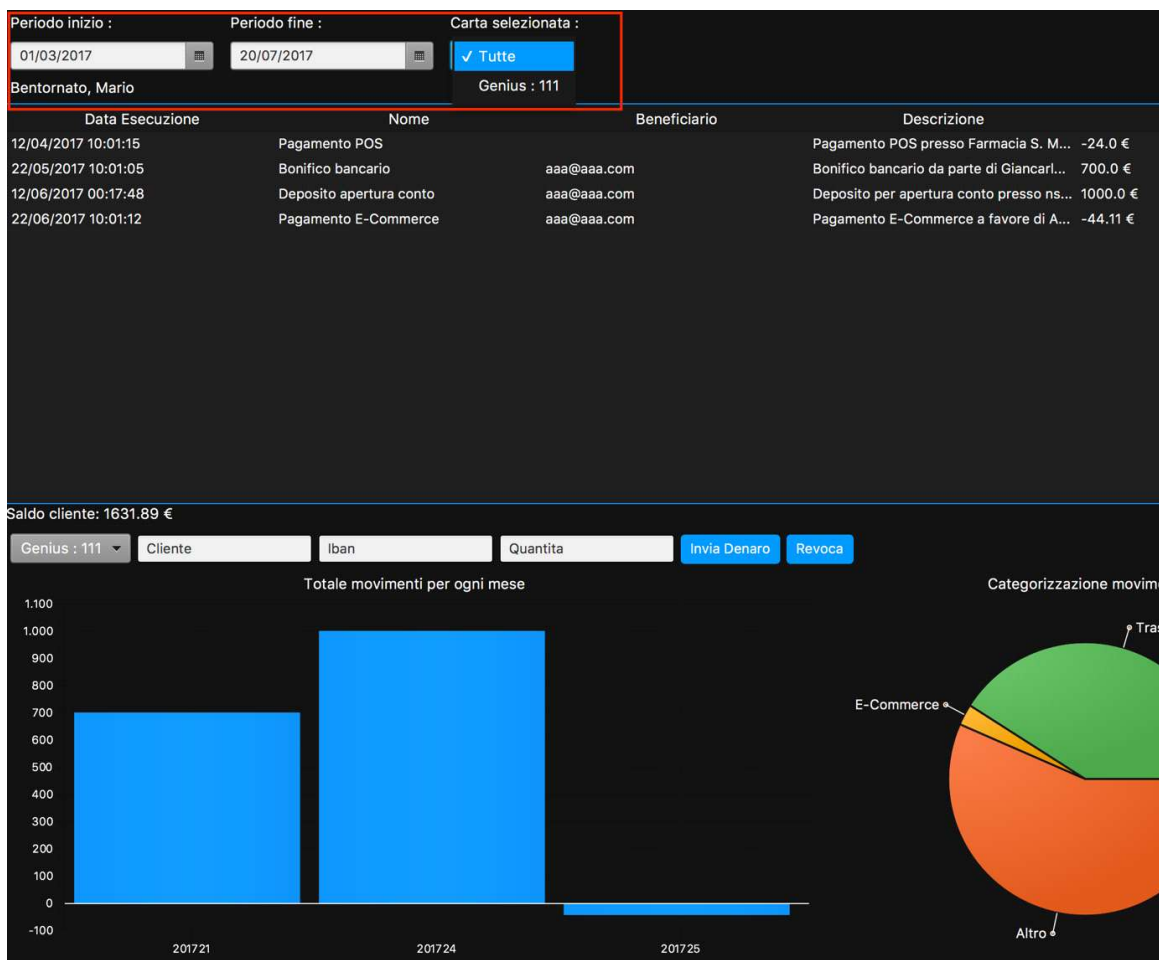
**[DIAGRAMMA UML ALLA FINE DEL DOCUMENTO]**



## MANUALE UTENTE

- **Consultazione selettiva dei movimenti**

Selezionare l'intervallo di date e la carta di credito per visualizzare i movimenti nella tabella inerenti all'intervallo di date e alla carta selezionata.



```

<?xml version="1.0" encoding="UTF-8"?>
<Evento nomeApp="BANKING">
  <tag>Cambio intervallo date 2017-06-23 -- 2017-07-25</tag>
  <clientIP>0.0.0.0</clientIP>
  <timestamp>1500820410547</timestamp>
</Evento>

<?xml version="1.0" encoding="UTF-8"?>
<Evento nomeApp="BANKING">
  <tag>Cambio intervallo date 2017-05-01 -- 2017-07-25</tag>
  <clientIP>0.0.0.0</clientIP>
  <timestamp>1500820415321</timestamp>
</Evento>

<?xml version="1.0" encoding="UTF-8"?>
<Evento nomeApp="BANKING">
  <tag>Cambio intervallo date 2017-05-01 -- 2017-07-24</tag>
  <clientIP>0.0.0.0</clientIP>
  <timestamp>1500820422281</timestamp>
</Evento>

```

Le query eseguite sul database cambiano in base alla configurazione impostata e alla selezione della carta di credito.

- Caso in cui la carta di credito non è impostata e si seleziona un intervallo di date, questa query viene eseguita per la popolazione della tabella dei movimenti.

100% 121:1

SELECT \* FROM movimento WHERE Email = "aaa@aaa.com" AND DataEsecuzione BETWEEN DATE("2017-06-01") AND

Result Grid Filter Rows: Search Edit: Export/Import:

Email	Iban	DataEsecuzione	Beneficiario	Nome	Categoria	Descrizione	Cifra
aaa@aaa.com	111	2017-06-12 00:17:48	aaa@aaa.com	Deposito apertura conto	Altro	Deposito per apertura conto presso ns. filiale 51550	1000
aaa@aaa.com	111	2017-06-22 10:01:12	aaa@aaa.com	Pagamento E-Commerce	E-Commerce	Pagamento E-Commerce a favore di Amazon	-44.11
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### Modalità quotidiana:

```

<?xml version="1.0" encoding="UTF-8"?>
<Configurazione>
  <portaServerLog>8080</portaServerLog>
  <indirizzoServerLog>localhost</indirizzoServerLog>
  <indirizzoClient>111.111.111.111</indirizzoClient>
  <hostnameDatabase>localhost</hostnameDatabase>
  <portaDatabase>3306</portaDatabase>
  <utenteDatabase>root</utenteDatabase>
  <passwordDatabase></passwordDatabase>
  <utenteAccesso>aaa@aaa.com</utenteAccesso>
  <passwordAccesso>aaa</passwordAccesso>
  <modalitaGrafico>0</modalitaGrafico>
  <intervalloGrafico>3</intervalloGrafico>
</Configurazione>

```

- Caso in cui la carta di credito non è impostata ed è impostata la visualizzazione quotidiana dei movimenti, l'intervallo di date impostato è tra oggi e 3 mesi fa, questa query viene eseguita per popolare il grafico a barre.

1	•	SELECT DATE_FORMAT(DataEsecuzione, '%d-%m-%Y') AS Mese, SUM(Cifra) AS Cifra
2		FROM Movimento
3		WHERE Email = "aaa@aaa.com"
4		AND DataEsecuzione BETWEEN NOW() - INTERVAL 3 MONTH AND NOW()
5		GROUP BY (DATE_FORMAT(DataEsecuzione, '%d-%m-%Y'))

100%	27:3
<b>Result Grid</b>	Filter Rows: <input type="text" value="Search"/> Export:
Mese	Cifra
▶ 12-06-2017	1000
22-05-2017	700
22-06-2017	-44.11000061035156
23-07-2017	-30

- Caso in cui la carta di credito è impostata ed è impostata la visualizzazione quotidiana dei movimenti, l'intervallo di date impostato è tra oggi e 3 mesi fa, questa query viene eseguita per popolare il grafico a barre

1	•	SELECT DATE_FORMAT(DataEsecuzione, '%d-%m-%Y') AS Mese, SUM(Cifra) AS Cifra
2		FROM Movimento
3		WHERE Email = "aaa@aaa.com" AND Iban = "111"
4		AND DataEsecuzione BETWEEN NOW() - INTERVAL 3 MONTH AND NOW()
5		GROUP BY (DATE_FORMAT(DataEsecuzione, '%d-%m-%Y'))

100%	44:3
<b>Result Grid</b>	Filter Rows: <input type="text" value="Search"/> Export:
Mese	Cifra
▶ 12-06-2017	1000
22-05-2017	700
22-06-2017	-44.11000061035156
23-07-2017	-30

## Modalità settimanale:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configurazione>
  <portaServerLog>8080</portaServerLog>
  <indirizzoServerLog>localhost</indirizzoServerLog>
  <indirizzoClient>111.111.111.111</indirizzoClient>
  <hostnameDatabase>localhost</hostnameDatabase>
  <portaDatabase>3306</portaDatabase>
  <utenteDatabase>root</utenteDatabase>
  <passwordDatabase></passwordDatabase>
  <utenteAccesso>aaa@aaa.com</utenteAccesso>
  <passwordAccesso>aaa</passwordAccesso>
  <modalitaGrafico>1</modalitaGrafico>
  <intervalloGrafico>3</intervalloGrafico>
</Configurazione>
```

- Caso in cui la carta di credito non è impostata ed è impostata la visualizzazione settimanale dei movimenti, l'intervallo di date impostato è tra oggi e 3 mesi fa, questa query viene eseguita per popolare il grafico a barre.

```
1 • SELECT YEARWEEK(DataEsecuzione) AS Mese, SUM(Cifra) AS Cifra
2 FROM Movimento
3 WHERE Email = "aaa@aaa.com"
4 AND DataEsecuzione BETWEEN NOW() - INTERVAL 3 MONTH AND NOW()
5 GROUP BY (YEARWEEK(DataEsecuzione))
```

100% 47:4

Result Grid Filter Rows: Search Export:

Mese	Cifra
201721	700
201724	1000
201725	-44.11000061035156
201730	-30

- Caso in cui la carta di credito è impostata ed è impostata la visualizzazione settimanale dei movimenti, l'intervallo di date impostato è tra oggi e 3 mesi fa, questa query viene eseguita per popolare il grafico a barre.

1	•	SELECT YEARWEEK(DataEsecuzione) AS Mese, SUM(Cifra) AS Cifra
2		FROM Movimento
3		WHERE Email = "aaa@aaa.com" AND Iban = "111"
4		AND DataEsecuzione BETWEEN NOW() - INTERVAL 3 MONTH AND NOW()
5		GROUP BY(YEARWEEK(DataEsecuzione))

100%
1:5

**Result Grid**

Mese	Cifra
▶ 201721	700
201724	1000
201725	-44.11000061035156
201730	-30

## Modalità mensile:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configurazione>
  <portaServerLog>8080</portaServerLog>
  <indirizzoServerLog>localhost</indirizzoServerLog>
  <indirizzoClient>111.111.111.111</indirizzoClient>
  <hostnameDatabase>localhost</hostnameDatabase>
  <portaDatabase>3306</portaDatabase>
  <utenteDatabase>root</utenteDatabase>
  <passwordDatabase></passwordDatabase>
  <utenteAccesso>aaa@aaa.com</utenteAccesso>
  <passwordAccesso>aaa</passwordAccesso>
  <modalitaGrafico>2</modalitaGrafico>
  <intervalloGrafico>3</intervalloGrafico>
</Configurazione>
```

- Caso in cui la carta di credito non è impostata ed è impostata la visualizzazione mensile dei movimenti, l'intervallo di date impostato è tra oggi e 3 mesi fa, questa query viene eseguita per popolare il grafico a barre

```
1 SELECT DATE_FORMAT(DataEsecuzione, '%m-%Y') AS Mese, SUM(Cifra) AS Ci
2 FROM Movimento
3 WHERE Email = "aaa@aaa.com"
4 AND DataEsecuzione BETWEEN NOW() - INTERVAL 3 MONTH AND NOW()
5 GROUP BY (DATE_FORMAT(DataEsecuzione, '%m-%Y'))
```

100% 27:3

Result Grid Filter Rows: Search Export:

Mese	Cifra
05-2017	700
06-2017	955.8899993896484
07-2017	-30

- Caso in cui la carta di credito è impostata ed è impostata la visualizzazione mensile dei movimenti, l'intervallo di date impostato è tra oggi e 3 mesi fa, questa query viene eseguita per popolare il grafico a barre

```
1 SELECT DATE_FORMAT(DataEsecuzione, '%m-%Y') AS Mese, SUM(Cifra) AS Cifra
2 FROM Movimento WHERE Email = "aaa@aaa.com" AND Iban = "111"
3 AND DataEsecuzione BETWEEN NOW() - INTERVAL 3 MONTH AND NOW()
4 GROUP BY (DATE_FORMAT(DataEsecuzione, '%m-%Y'))
```

100% 47:3

Result Grid Filter Rows: Search Export:

Mese	Cifra
05-2017	700
06-2017	955.8899993896484
07-2017	-30

- Query eseguita per la popolazione del grafico a torta secondo l'intervallo di date.

```

1 SELECT Categoria, SUM(ABS(Cifra)) AS Cifra
2 FROM Movimento
3 WHERE Email = "aaa@aaa.com"
4 AND DataEsecuzione BETWEEN DATE("2017-06-01") AND DATE("2017-0
5 GROUP BY Categoria

```

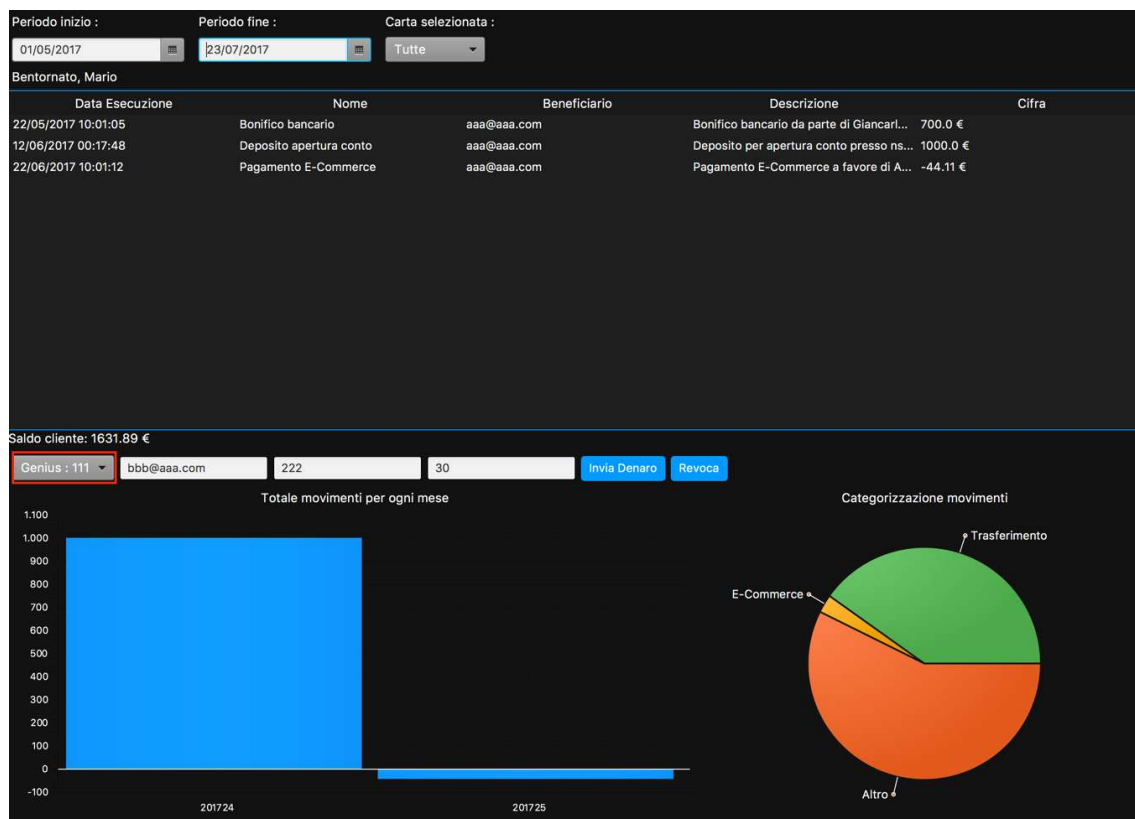
100% 1:5

Result Grid Filter Rows: Search Export:

Categoria	Cifra
Altro	1000
E-Commerce	44.11000061035156

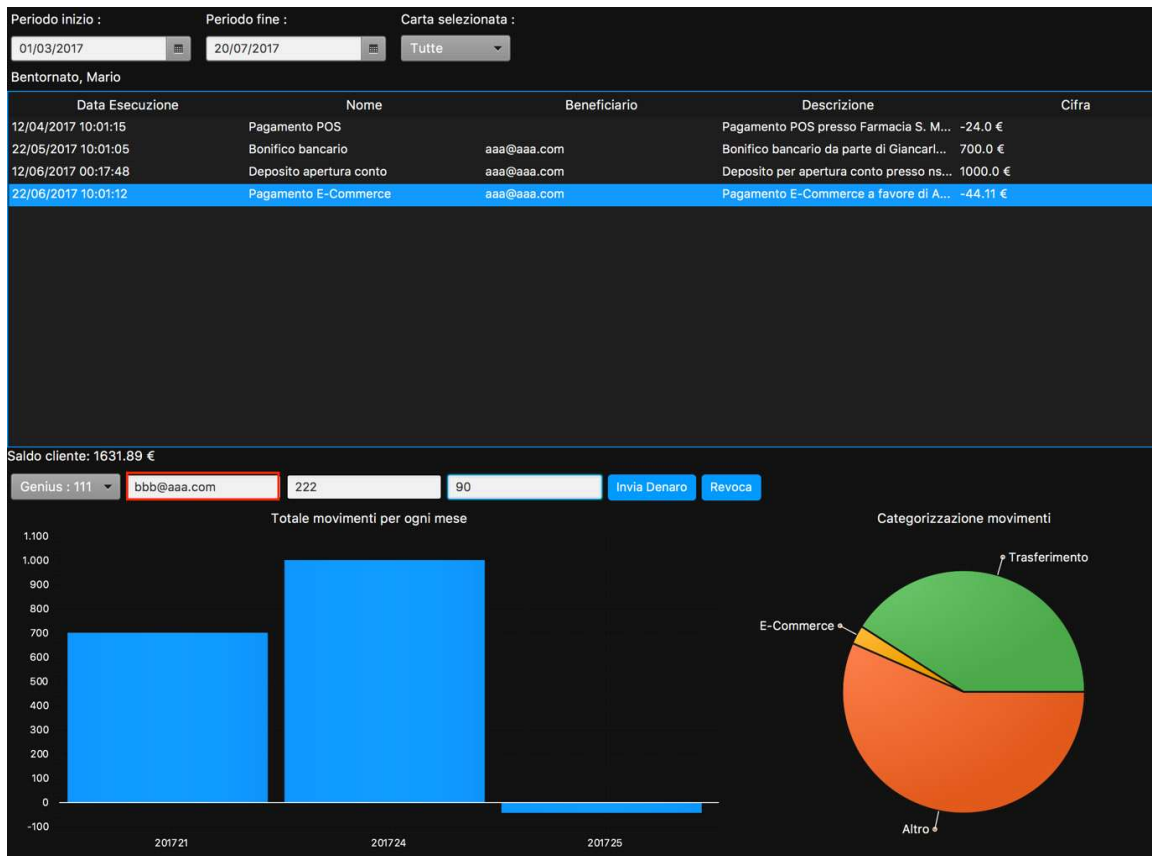
- Invio denaro ad un utente**

Selezionare la carta di credito dalla quale inviare il denaro.



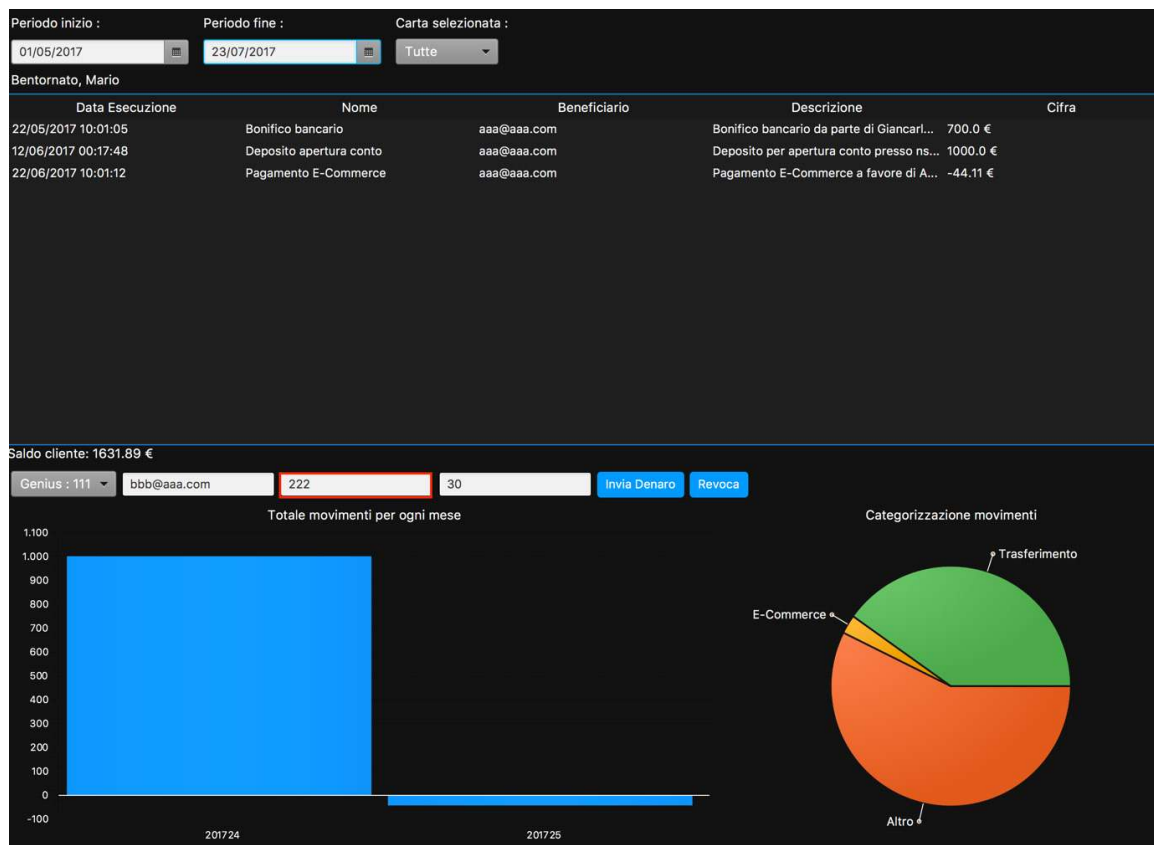
Inserire l'email del cliente che riceverà il denaro.





Inserire l'iban della carta di credito del cliente che riceverà il denaro.

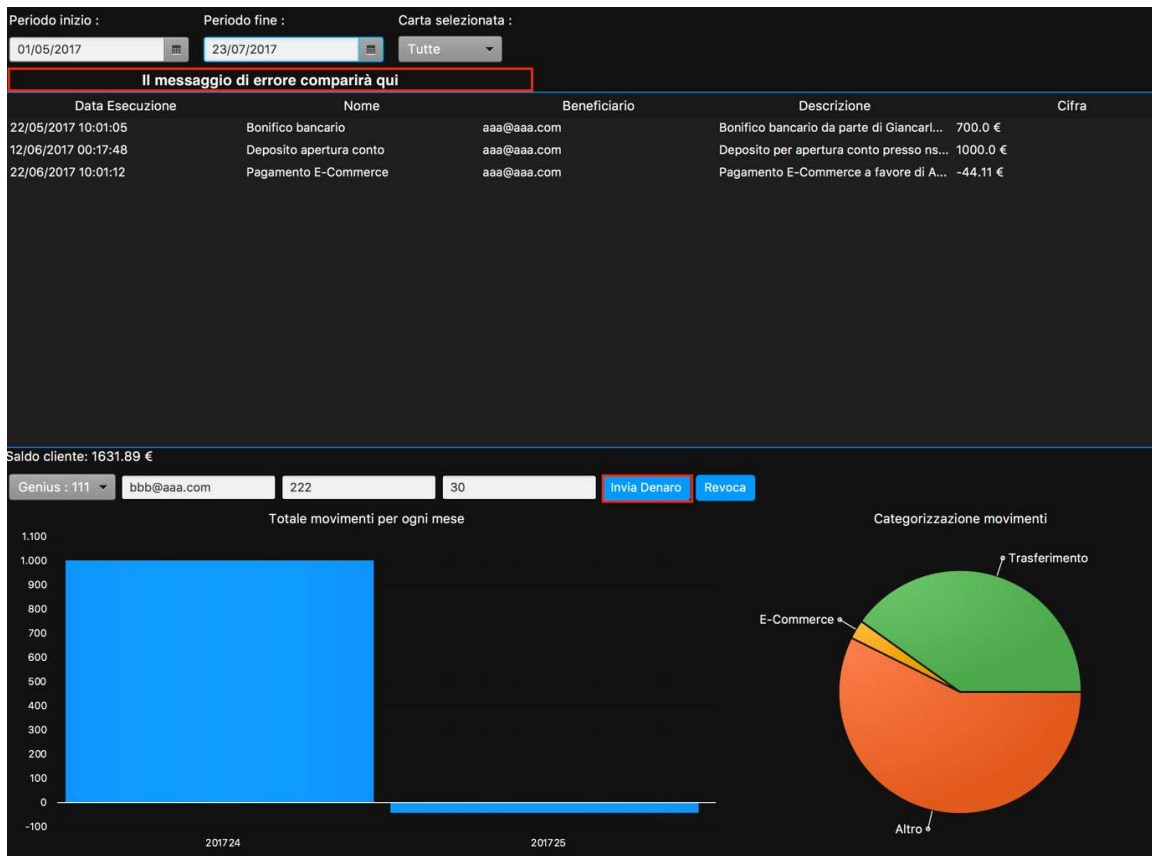




Inserire la quantità di denaro da inviare.



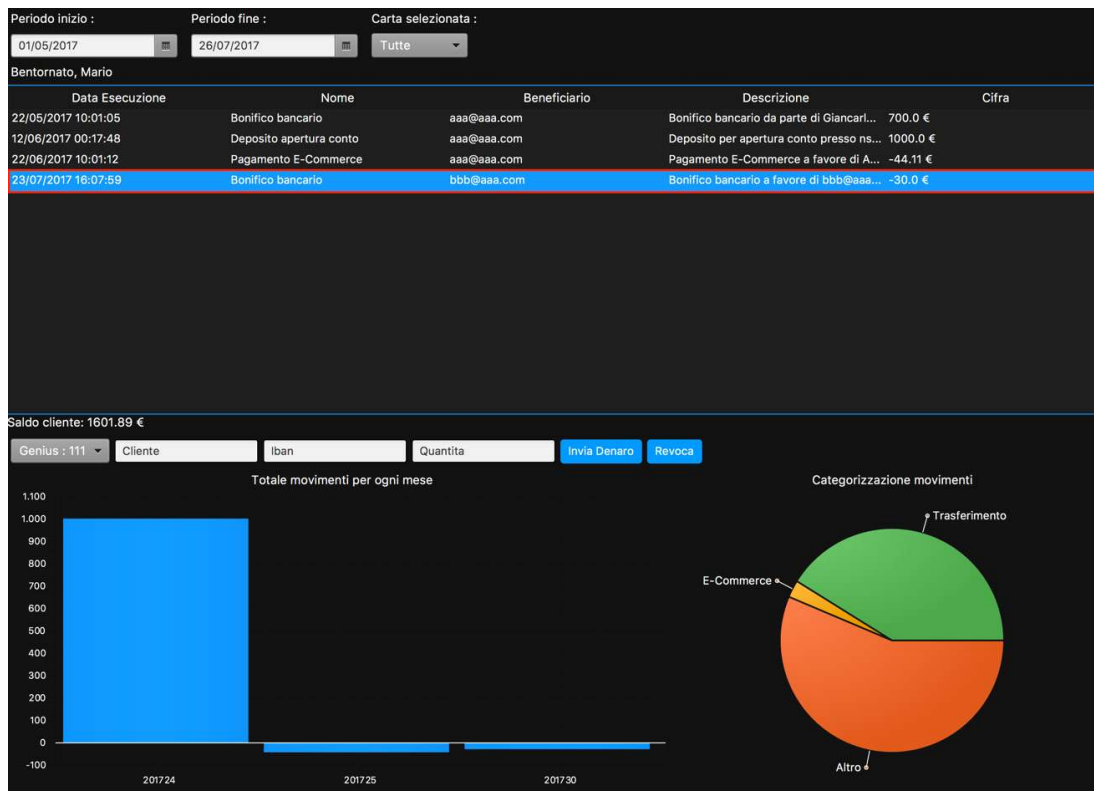
Fare click su [Invia denaro](#) per completare l'operazione, in caso di denaro insufficiente o di cliente non trovato verrà mostrato un messaggio di errore.



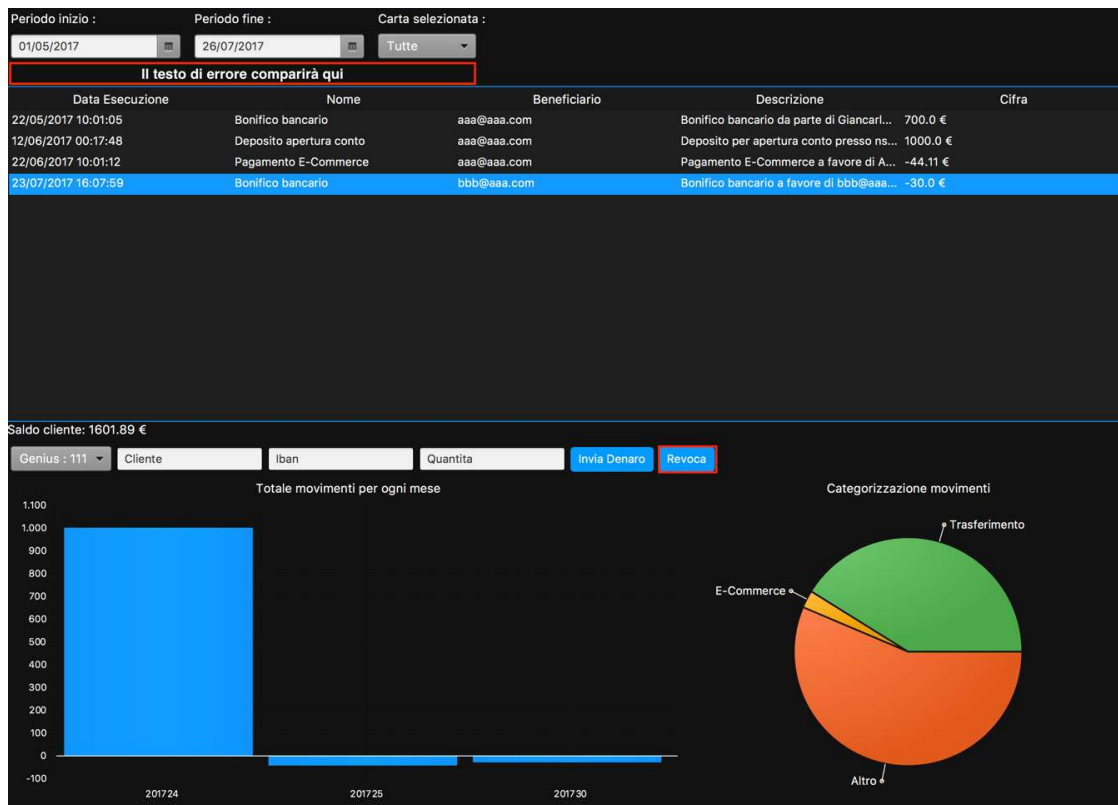
```
<?xml version="1.0" encoding="UTF-8"?>
<Evento nomeApp="BANKING">
  <tag>Invio denaro</tag>
  <clientIP>0.0.0.0</clientIP>
  <timestamp>1500820487546</timestamp>
</Evento>
```

- **Revoca di una transazione**

Selezionare una riga nella tabella delle transazioni.



Fare click su Revoca per revocare la transazione selezionata, se la transazione è stata fatta da più di 6 ore oppure non corrisponde ad una transazione tra utenti del servizio verrà mostrato un messaggio di errore.



```
<?xml version="1.0" encoding="UTF-8"?>
<Evento nomeApp="BANKING">
  <tag>Revoca</tag>
  <clientIP>0.0.0.0</clientIP>
  <timestamp>1500820491523</timestamp>
</Evento>
```