

2^a Prova in Itinere

Pay Steam

Lorenzo Benvenuti / 0046403

Sommario

Fase 1: Specifica dei requisiti	3
Requisiti utente	3
Profili e funzioni	3
Considerazioni preliminari	3
Fase 2: Progettazione concettuale	4
Analisi dei requisiti	4
Glossario dei termini	5
Considerazioni	6
Revisione della traccia	6
Operazioni	6
Schema E-R	7
Fase 3: Progettazione logica	8
Tavola dei volumi	8
Tavola delle operazioni	8
Tavola degli accessi	9
Ristrutturazione schema E-R	10
Schema E-R ristrutturato	11
Schema relazionale	12
Fase 4: Progettazione fisica	13
Implementazione della base di dati	13
Script SQL	14

Nota: Lo sviluppo di questo progetto seguirà gli stessi passaggi della prima prova, quindi, onde evitare ripetizioni inutili, non viene considerato necessario l'utilizzo di paragrafi introduttivi.

Fase 1: Specifica dei requisiti

Requisiti utente

“Si chiede di realizzare una web application per il pagamento on line di servizi e beni.

Il servizio sarà utilizzabile da utenti registrati. Ogni utente avrà un suo conto corrente con un deposito in euro/una disponibilità di carta di credito.

Il servizio potrà essere chiamato da una qualunque applicazione che passerà i parametri dell'esercizio commerciale e la somma da riconoscere.

Il servizio verificherà le credenziali del consumatore e mostrerà il riepilogo del bene/servizio da acquistare corredato dal valore economico.

Il consumatore dovrà accettare la transazione esplicitamente ed il servizio comunicherà all'applicazione chiamante l'avvenuto pagamento.

L'esercente dovrà avere un suo account sull'applicazione con un conto su cui ricevere le somme dalle transazioni.”

Profili e funzioni

- **Profilo visitatore**
 - Può leggere le condizioni e registrarsi.
- **Profilo utente registrato**
 - Può visualizzare il suo conto con i movimenti in ingresso ed uscita.
 - Può autorizzare transazioni di uscita.
 - Può memorizzare carte di credito.
 - Diamo per scontato la capacità di spesa e la copertura della carta di credito senza ulteriori controlli.
- **Profilo esercente**
 - Può visualizzare il suo conto con i movimenti in ingresso ed uscita.
 - Può autorizzare transazioni di uscita.
- **Profilo M2M (web-api)**
 - Espone un servizio che riceve chiamate così formattate:
 - url inviante, url di risposta, id esercente, id transazione, descrizione del bene, prezzo transazione.
 - Risponde a chiamate così formattate:
 - url inviante, id transazione, esito transazione (possibili risposte OK, KO).

Considerazioni preliminari

- Per semplicità non sono previsti vincoli e verifiche di alcuna sorta sulle carte di credito (plafond, correttezza degli estremi, ecc.).
- La registrazione dell'utente è di livello elementare. Non sono previste autenticazioni a due fattori (per messaggio, e-mail, ecc.) o modalità più sofisticate.
- L'inserimento dei dati in fase di registrazione avviene tramite protezione crittografica opzionale.
- Le web API descritte comunicano con autenticazione.

Fase 2: Progettazione concettuale

Analisi dei requisiti

Legenda: **Entità**, **Relazioni**, Attributi

Si chiede di realizzare una **web application** per il **pagamento** on line di **servizi** e **beni**.

Il **servizio** sarà utilizzabile da **utenti registrati**. Ogni **utente** avrà un suo **conto corrente** con un deposito in euro/una disponibilità di carta di credito.

Il **servizio** potrà essere **chiamato** da una qualunque applicazione che passerà i parametri dell'**esercizio commerciale** e la somma da riconoscere.

Il **servizio** **verificherà** le credenziali del **consumatore** e **mostreerà** il riepilogo del **bene/servizio** da acquistare corredato dal valore economico.

Il **consumatore** dovrà **accettare** la **transazione** esplicitamente ed il **servizio** **comunicherà** all'applicazione chiamante l'avvenuto **pagamento**.

L'**esercente** dovrà **avere** un suo **account** sull'applicazione con un **conto** su cui ricevere le somme dalle **transazioni**.

- **Profilo visitatore**
 - o Può **leggere** le condizioni e **registrarsi**.
- **Profilo utente registrato**
 - o Può **visualizzare** il suo **conto** con i **movimenti** in ingresso ed uscita.
 - o Può **autorizzare** **transazioni** di uscita.
 - o Può **memorizzare** carte di credito.
 - o ~~Diamo per scontato la capacità di spesa e la copertura della carta di credito senza ulteriori controlli.~~
- **Profilo esercente**
 - o Può **visualizzare** il suo **conto** con i **movimenti** in ingresso ed uscita.
 - o Può **autorizzare** **transazioni** di uscita.
- **Profilo M2M (web-api)**
 - o Espone un **servizio** che **riceve** chiamate così formattate:
 - url inviante, url di risposta, id esercente, id transazione, descrizione del bene, prezzo transazione.
 - o **Risponde** a chiamate così formattate:
 - url inviante, id transazione, esito transazione (possibili risposte OK, KO).

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Account	Profilo utilizzato per accedere a servizi digitali		Carta di credito, Conto corrente, Consumatore, Esercente, Movimento, Utente
Bene	Prodotto fisico scambiato in una operazione commerciale		Consumatore, Esercente, Transazione
Carta di credito	Dispositivo elettronico usato per i pagamenti		Account, Transazione
Conto corrente	Strumento per il deposito e prelievo del denaro	Conto	Account, Movimento, Transazione
Consumatore	Persona che usufruisce di un bene/servizio	Utente	Bene, Servizio, Transazione
Esercente	Persona che fornisce un bene/servizio	Esercizio commerciale, Utente	Bene, Servizio, Transazione
Movimento	Flusso di denaro tra conti correnti		Conto corrente, Transazione
Servizio	Prestazione scambiata in una operazione commerciale		Consumatore, Esercente, Transazione
Transazione	Esecuzione di una operazione commerciale		Bene, Carta di credito, Conto corrente, Consumatore, Esercente, Movimento, Servizio
Utente	Persona che utilizza l'applicazione e/o i suoi servizi	Consumatore, Esercente, Visitatore	Account
Visitatore	Persona che consulta l'applicazione	Utente	

Considerazioni

- Consumatore, Esercente, Utente registrato e Visitatore sono tutte tipologie diverse di Utente e ciascuna espleta una funzione specifica collegata ad esso.
- Esercizio commerciale ed Esercente sono ritenuti assimilabili.
- Il termine “servizio” viene utilizzato sia per riferirsi alle prestazioni scambiate che all’applicazione stessa.
- I visitatori (consumatori o esercenti) potranno creare un unico account ciascuno.
- Le transazioni conterranno url inviante, url di risposta, id esercente, id transazione, descrizione del bene e prezzo, poiché anche se sono requisiti del profilo M2M, essi sono effettivamente collegati alla specifica transazione, mentre tale profilo si occupa esclusivamente di veicolarla all’esterno.
- Per semplicità, si considera arbitrariamente che tutte le transazioni vadano a buon fine e ciascuna di esse generi almeno un movimento.
- Inoltre si presuppone che ogni utente memorizzi almeno una carta di credito.
- Un’unica API si occupa della comunicazione con le applicazioni chiamanti (compresa quella della 1^a prova).

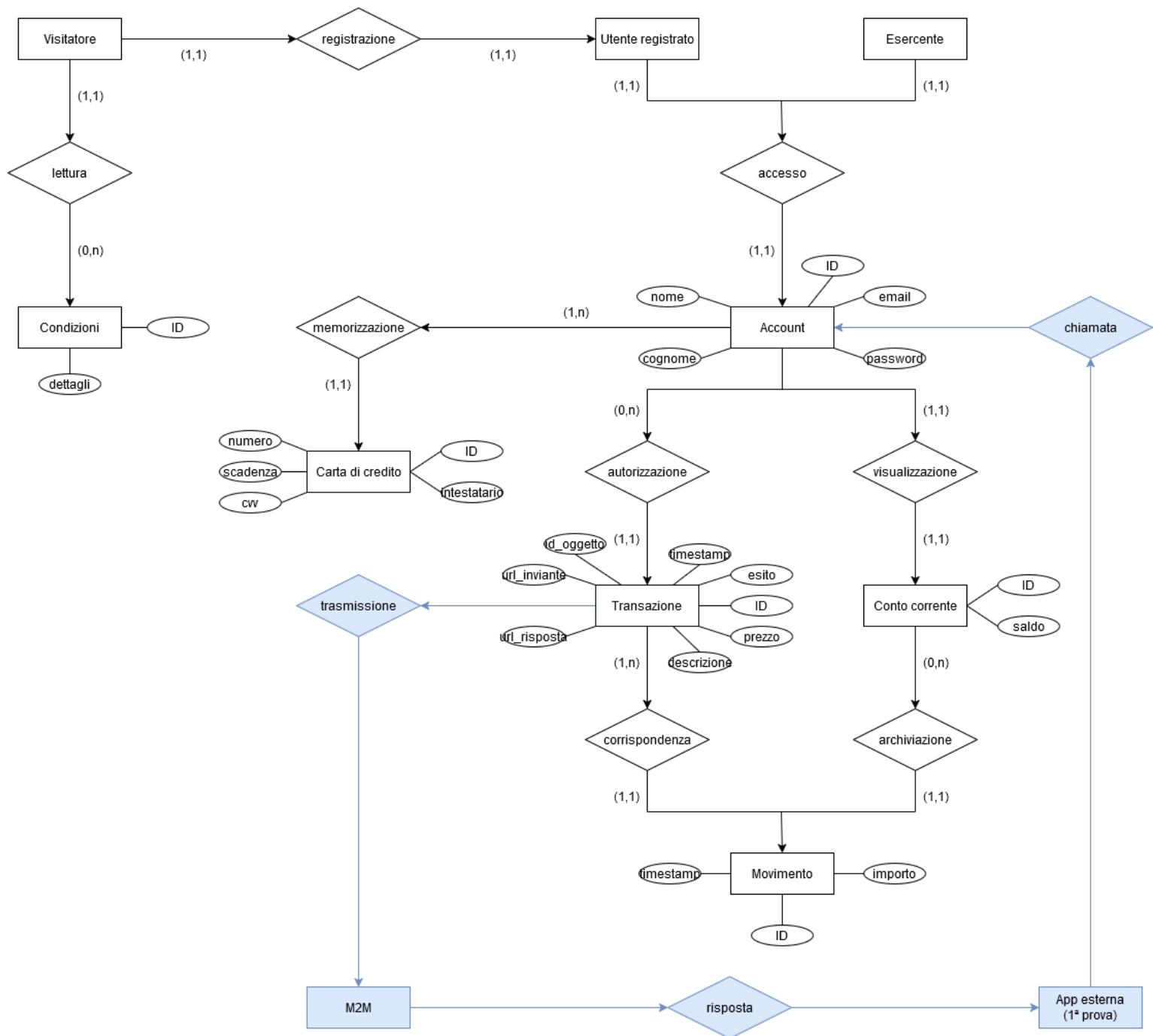
Revisione della traccia

- Il visitatore legge le condizioni e/o si registra sull’applicazione (aprendo un account e diventando così un utente registrato).
- L’utente registrato visualizza il conto ed i movimenti, autorizza le transazioni e memorizza le carte di credito.
- L’esercente visualizza il conto ed i movimenti e autorizza le transazioni.
- Il profilo M2M (API) riceve le transazioni da pagare ed emette le ricevute dei pagamenti.

Operazioni

- 1) Lettura condizioni
- 2) Registrazione utente
- 3) Memorizzazione carta di credito
- 4) Visualizzazione movimenti
- 5) Autorizzazione transazione

Schema E-R



Nota: La parte evidenziata mostra il collegamento tramite API con l'altra applicazione (SFT, 1ª prova).

Fase 3: Progettazione logica

- Si anticipa in parte la ristrutturazione dello schema E-R, considerando nelle tavole solo le entità e le relazioni principali. La spiegazione sulle modifiche verrà data dettagliatamente prima della ristrutturazione effettiva.

Tavola dei volumi

Concetto	Tipo	Volume/gg
Visitatore	E	1000
Condizioni	E	1
Account	E	500
Carta di credito	E	600
Transazione	E	3000
lettura	R	800
registrazione	R	300
memorizzazione	R	600
visualizzazione	R	2000
autorizzazione	R	3000

Tavola delle operazioni

Operazione	Descrizione	Tipo	Frequenza/gg
Op. 1	Lettura condizioni	I	800
Op. 2	Registrazione utente	I	300
Op. 3	Memorizzazione carta di credito	I	600
Op. 4	Visualizzazione movimenti	I	5000
Op. 5	Autorizzazione transazione	I	3000

Tavola degli accessi

Lettura condizioni

Consiste nella sola lettura delle condizioni, senza effettuare l'accesso all'account:

Concetto	Costrutto	Accesso	Tipo
Condizioni	E	1	L

Registrazione utente

Prevede una lettura per controllare che non ci siano credenziali corrispondenti a quelle immesse ed una scrittura per registrarle in caso di assenza di duplicati:

Concetto	Costrutto	Accesso	Tipo
Account	E	1	L
Account	E	1	S

Memorizzazione carta di credito

Dopo una lettura preventiva, onde evitare duplicati, si inseriscono i dati della nuova carta:

Concetto	Costrutto	Accesso	Tipo
Carta di credito	E	1	L
Carta di credito	E	1	S

Visualizzazione movimenti

È necessario prima accedere all'account (poiché contiene il conto corrente), e poi leggere le transazioni passate (le quali corrispondono ad una serie di movimenti):

Concetto	Costrutto	Accesso	Tipo
Account	E	1	L
Transazione	E	1	L

Autorizzazione transazione

Per approvare una transazione, e quindi effettuare un pagamento, bisogna prima accedere al proprio account e controllare che la transazione contenga tutti i campi necessari:

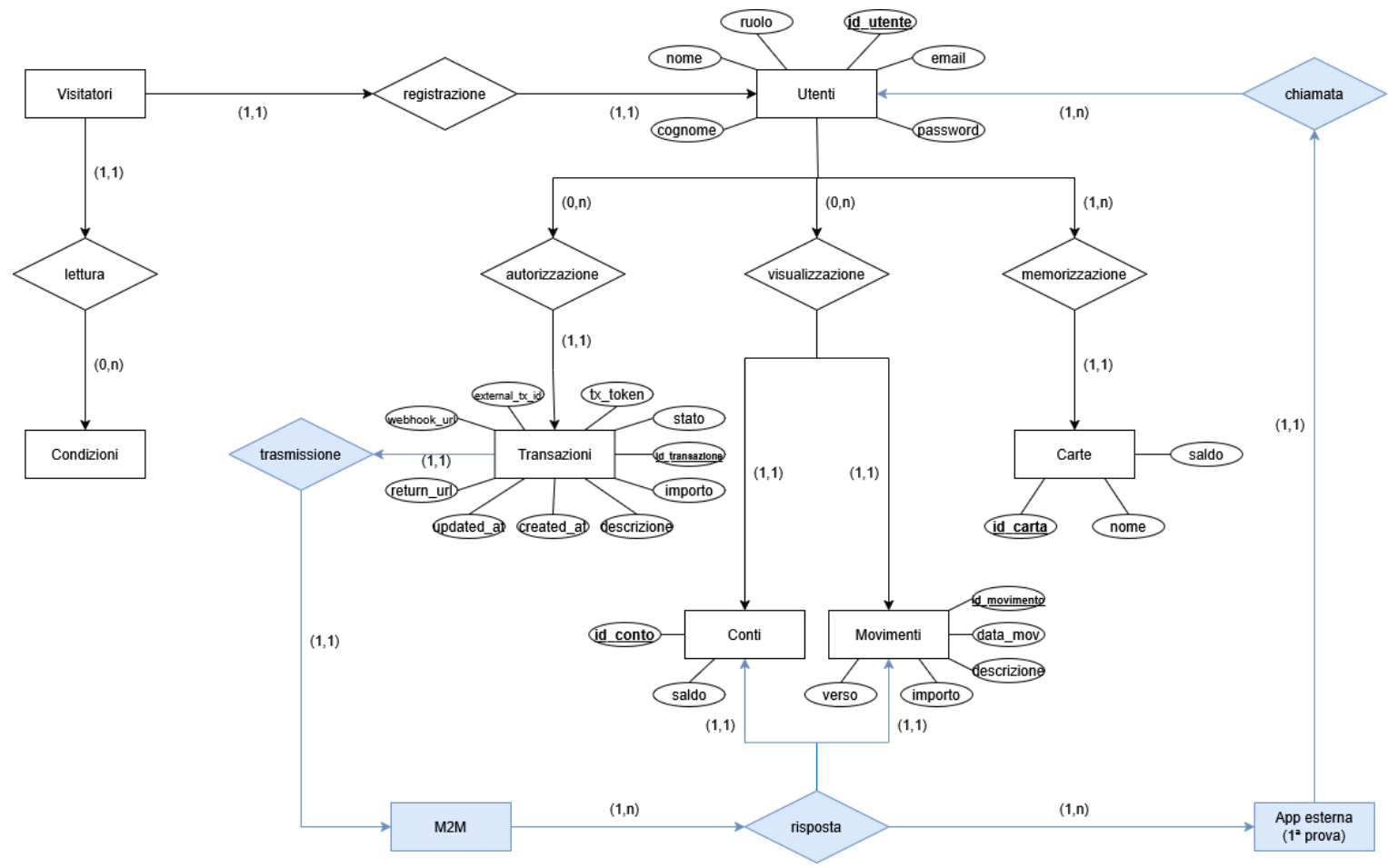
Concetto	Costrutto	Accesso	Tipo
Account	E	1	L
Transazione	E	1	S

Ristrutturazione schema E-R

Come già anticipato, per semplificare la struttura:

- Le entità “Conto corrente” e “Movimento” diventano rispettivamente “Conti” e “Movimenti” e vengono poste sullo stesso livello, poiché è più preciso considerarle come sezioni indipendenti consultabili dall’utente, in modo da poter inserire in esse maggiori informazioni dedicate (ovvero un maggior numero di attributi ciascuna) e poterle collegare direttamente al profilo M2M per essere aggiornate ad ogni transazione effettuata. Inoltre il “timestamp” di “Movimento” si trasforma in “data_mov”.
- Le entità “Utente registrato”, “Esercente” ed “Account” vengono accorpate nell’entità “Utenti”, a cui verrà aggiunto l’attributo “ruolo” per permetterne la distinzione, mentre l’attributo “saldo” viene rimosso per essere inserito in “Conti” e “Carte” ed avere così un maggiore dettaglio.
- Il ramo del diagramma dedicato all’entità “Visitatore” non verrà considerato in fase di popolamento del database, poiché le informazioni ad esso legate non sono state specificate in fase di raccolta dei requisiti. Si adotta quindi un approccio arbitrario, inserendo direttamente una breve descrizione per l’utilizzo dell’applicazione nella sua landing page. Viene comunque mantenuto nel diagramma ristrutturato per completezza e per mostrarne l’esistenza.
- L’entità “Transazione” diventa “Transazioni” e riceve una modifica agli attributi per ottenere una maggiore coerenza e fedeltà ai requisiti. In particolare “url_inviante”, “url_risposta” e “id_oggetto” vengono tradotti rispettivamente in “webhook_url”, “return_url” e “external_id_tx” mentre “timestamp” viene dettagliato in “created_at” e “updated_at”, in modo da fornire una descrizione più dettagliata e professionale delle transazioni.
- L’entità “Carta di credito” ed il suo attributo “intestatario” diventano rispettivamente “Carte” e “nome”, mentre i restanti attributi “numero”, “scadenza” e “cvv” vengono sostituiti da “saldo” per semplicità.
- L’entità “Transazioni” presenterà sia gli attributi legati al consumatore che quelli associati all’esercente (sotto forma di foreign keys), in modo tale da rappresentare entrambe le operazioni di entrata e uscita.
- La “risposta” del profilo M2M (che corrisponderà all’effettiva web API al termine dello sviluppo fisico dell’applicazione) alle transazioni viene collegata a “Conti” e “Movimenti”, perché operativamente è più sensato e corretto avere le due sezioni aggiornate automaticamente ad ogni pagamento validato.
- Sarà assegnato a tutte le entità uno specifico attributo “id”, il quale rappresenterà il loro identificatore primario, grazie alla sua capacità di distinguerle tramite una semplice codifica.

Schema E-R ristrutturato



Schema relazionale

Schema logico definitivo, corrispondente alle tabelle che popoleranno il database.

Vengono applicate ulteriori ottimizzazioni per renderlo più performante e riportate le foreign keys:

Utenti(id_utente, nome, cognome, email, password, ruolo)

Conti(id_conto, id_utente, saldo)

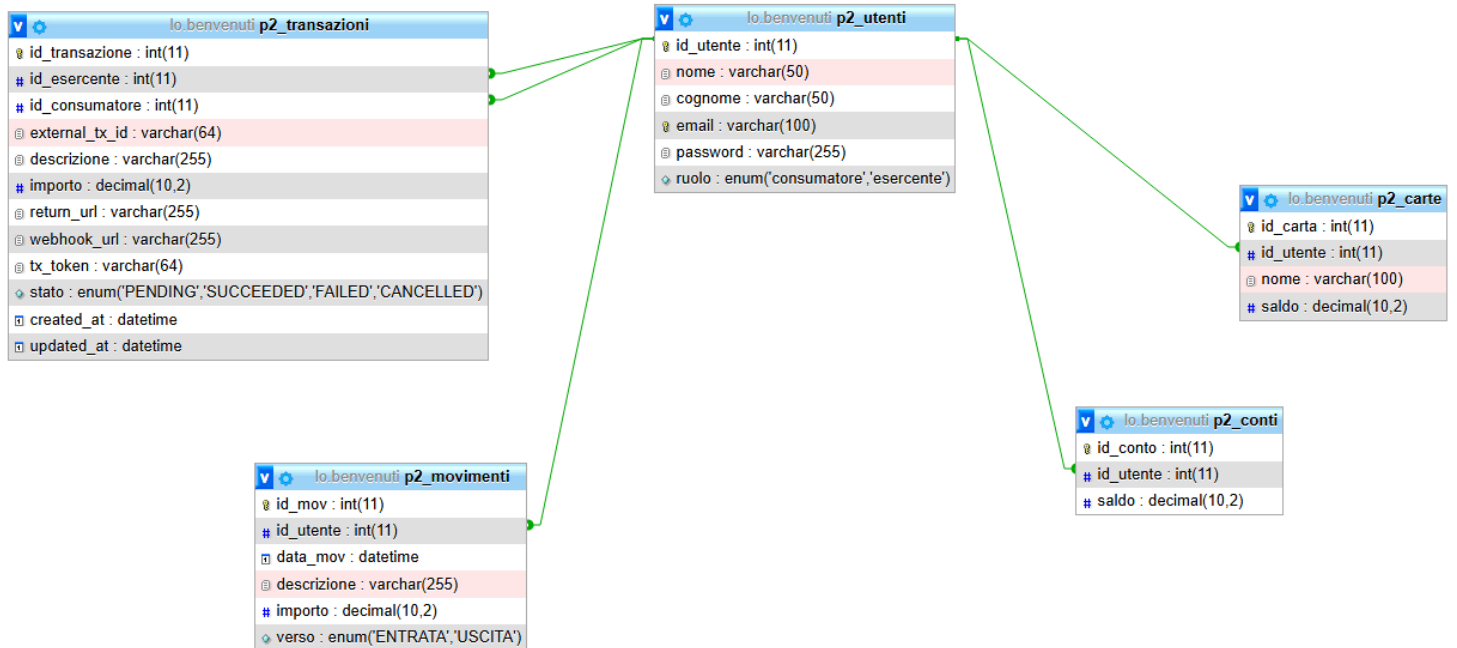
Carte(id_carta, id_utente, nome, saldo)

Transazioni(id_transazione, id_consumatore, id_esercente, external_tx_id, descrizione, importo, return_url, webhook_url, tx_token, stato, created_at, updated_at)

Movimenti(id_mov, id_utente, data_mov, descrizione, importo, verso)

Fase 4: Progettazione fisica

Implementazione della base di dati



Script SQL

```
-- SCHEMA PAYSTEAM

-- N.B. Allinea l'ID consumatore PaySteam all'ID utente SFT (id_utente = 3)
SET @OLD_FK_CHECKS = @@FOREIGN_KEY_CHECKS;
SET FOREIGN_KEY_CHECKS = 0;

CREATE TABLE IF NOT EXISTS p2_utenti (
  id_utente INT AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(50) NOT NULL,
  cognome VARCHAR(50) NOT NULL,
  email VARCHAR(100) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL,
  ruolo ENUM('consumatore','esercente') NOT NULL DEFAULT 'consumatore'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p2_conti (
  id_conto INT AUTO_INCREMENT PRIMARY KEY,
  id_utente INT NOT NULL,
  saldo DECIMAL(10,2) NOT NULL DEFAULT 0.00,
  CONSTRAINT fk_p2_conti_utente FOREIGN KEY (id_utente) REFERENCES p2_utenti(id_utente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p2_movimenti (
  id_mov INT AUTO_INCREMENT PRIMARY KEY,
  id_utente INT NOT NULL,
  data_mov DATETIME NOT NULL,
  descrizione VARCHAR(255) NOT NULL,
  importo DECIMAL(10,2) NOT NULL,
  verso ENUM('ENTRATA','USCITA') NOT NULL,
  CONSTRAINT fk_p2_mov_utente FOREIGN KEY (id_utente) REFERENCES p2_utenti(id_utente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS p2_carte (
  id_carta INT AUTO_INCREMENT PRIMARY KEY,
  id_utente INT NOT NULL,
  nome VARCHAR(100) NOT NULL,
  saldo DECIMAL(10,2) NOT NULL DEFAULT 0.00,
  CONSTRAINT fk_p2_carte_utente FOREIGN KEY (id_utente) REFERENCES p2_utenti(id_utente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS p2_transazioni (  
  id_transazione INT AUTO_INCREMENT PRIMARY KEY,  
  id_esercente INT NOT NULL,  
  id_consumatore INT NOT NULL,  
  external_tx_id VARCHAR(64) NOT NULL,  
  descrizione VARCHAR(255) NOT NULL,  
  importo DECIMAL(10,2) NOT NULL,  
  return_url VARCHAR(255) NOT NULL,  
  webhook_url VARCHAR(255) NOT NULL,  
  tx_token VARCHAR(64) NOT NULL,  
  stato ENUM('PENDING','SUCCEEDED','FAILED','CANCELLED') NOT NULL DEFAULT 'PENDING',  
  created_at DATETIME NOT NULL,  
  updated_at DATETIME NOT NULL,  
  CONSTRAINT fk_p2_tx_esercente FOREIGN KEY (id_esercente) REFERENCES p2_utenti(id_utente),  
  CONSTRAINT fk_p2_tx_consumatore FOREIGN KEY (id_consumatore) REFERENCES p2_utenti(id_utente)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```