

Progetto DALI : Servizi bancari/postali

Giuseppe Costanzo – Mat. 284785

Sistema operativo: Windows

Link github: <https://github.com/GiuseppeCostanzo/DaliProjectCostanzo>

Descrizione

L'idea del progetto è quella di simulare un ambiente, in particolare un servizio bancario (o postale) dove poter effettuare le classiche operazioni.

In tale dominio vi sono 3 tipologie di agenti : **cliente**, **sportellista** e **ticket** (il **bigliettaio**).

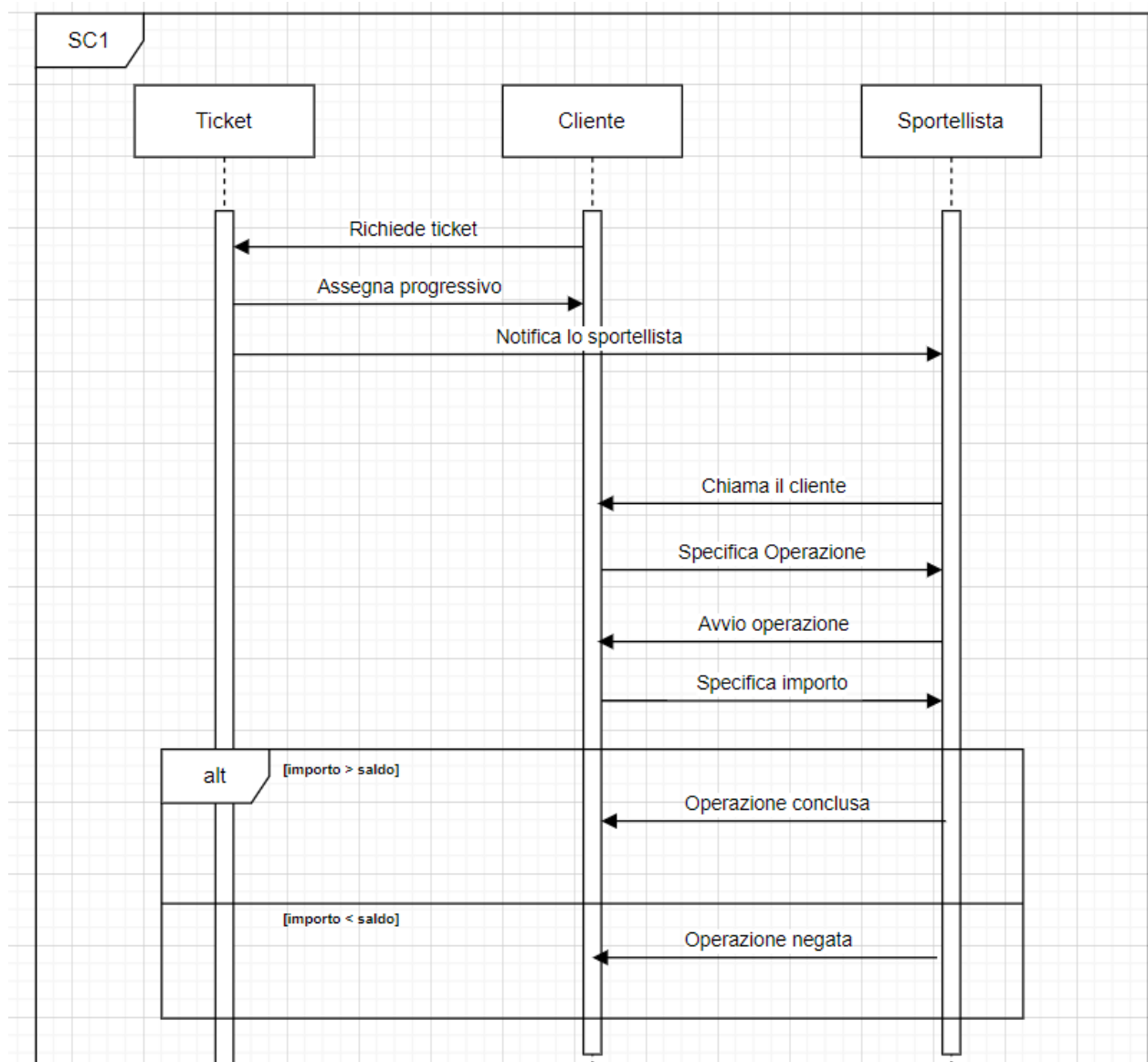
Quando un **cliente** arriva in banca per effettuare una operazione, si reca prima dal **ticket** per richiedere un biglietto (un numero intero progressivo), che servirà allo sportellista per gestire la clientela. L'agente ticket rilascia dunque tale progressivo al cliente (che parte da 1 e aumenta man mano che arrivano clienti), e il cliente attenderà dunque il suo turno.

Nel mentre, l'agente ticket notifica l'agente sportellista che vi sono nuovi clienti in coda da servire. Lo sportellista dunque, se libero, chiamerà il primo cliente utile (ossia quello avente progressivo più piccolo). A questo punto il cliente che viene chiamato si reca dallo sportellista, e può effettuare tre tipologie di operazione dal proprio conto corrente: **pagamento**, **deposito** (versamento), e **prelievo**. Tali operazioni non sono deterministiche, ma sono scelte dall'agente in modo casuale

Il cliente, una volta scelta la tipologia di operazione, specifica l'importo, e nel caso di **pagamento** e **prelievo**, se sceglie un valore maggiore di ciò che ha nel proprio conto corrente l'operazione verrà **negata**.

Nella specifica implementazione, vi sono un agente sportellista, un agente ticket, e 3 agenti clienti. Si noti che questo sistema di gestione della clientela può potenzialmente gestire n clienti

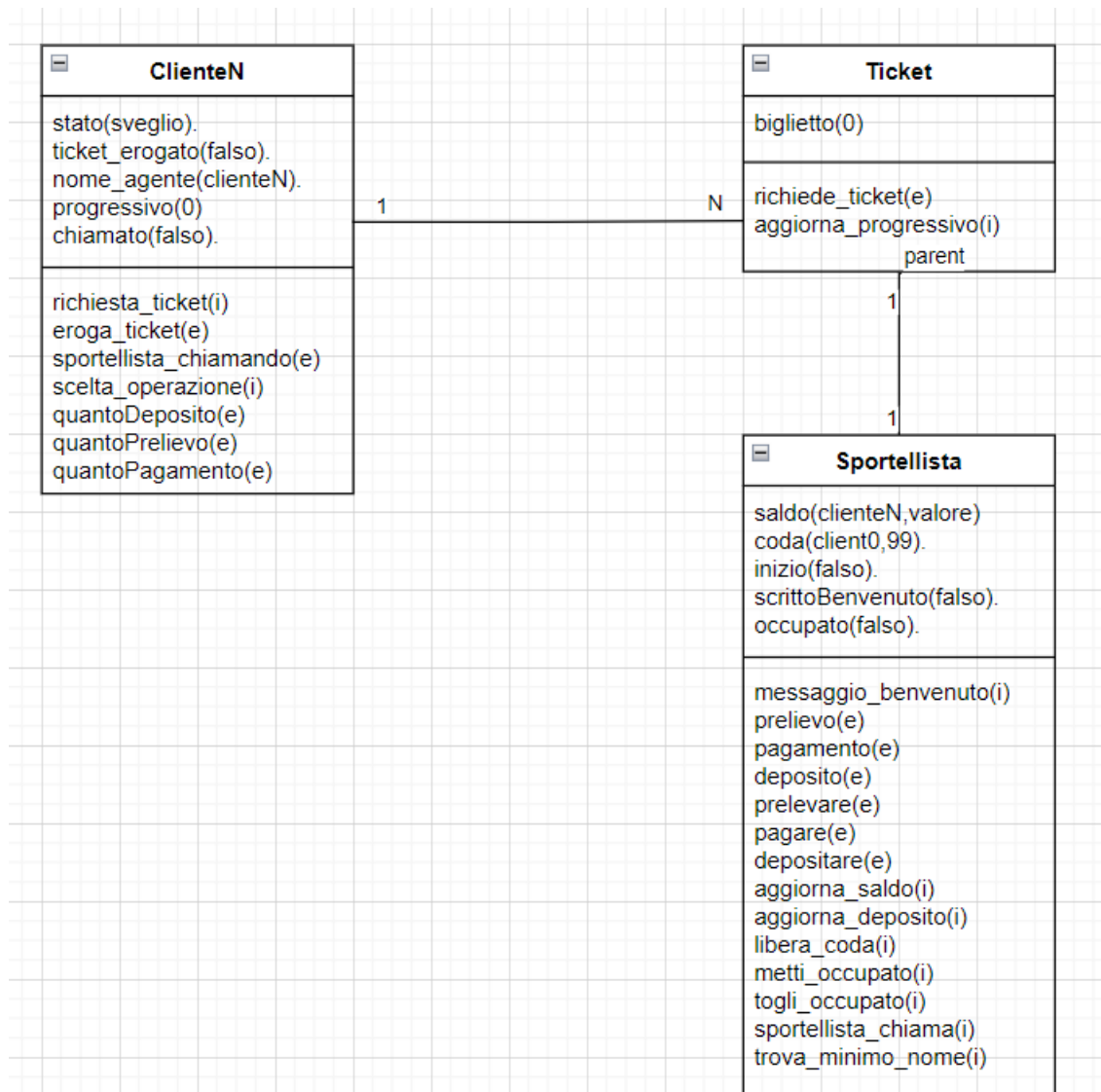
Sequence Diagram



In questo scenario SC1 viene descritto il sequence diagram degli agenti nel caso di una operazione di pagamento o prelievo che include dunque anche un sotto-scenario a seconda dell'esito dell'operazione.

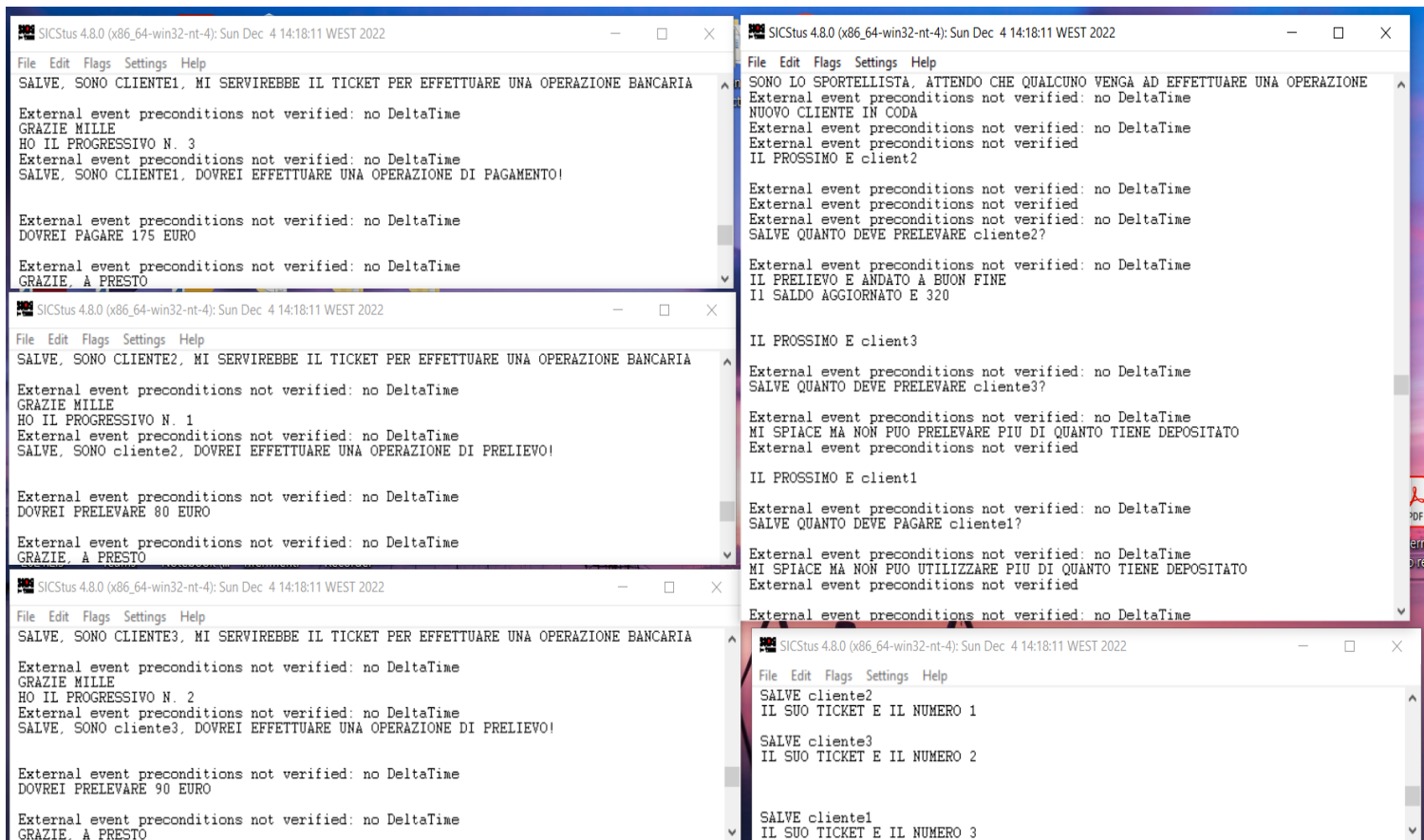
Nel caso dello scenario di deposito, l'operazione verrà direttamente conclusa

Class Diagram



Nel seguente class diagram sono riportate le entità principali del dominio. In una specifica classe sono riportati i **fatti** nella parte superiore, e gli **eventi** (i sta a indicare evento interno, e sta a indicare evento esterno)

Screenshot di esecuzione

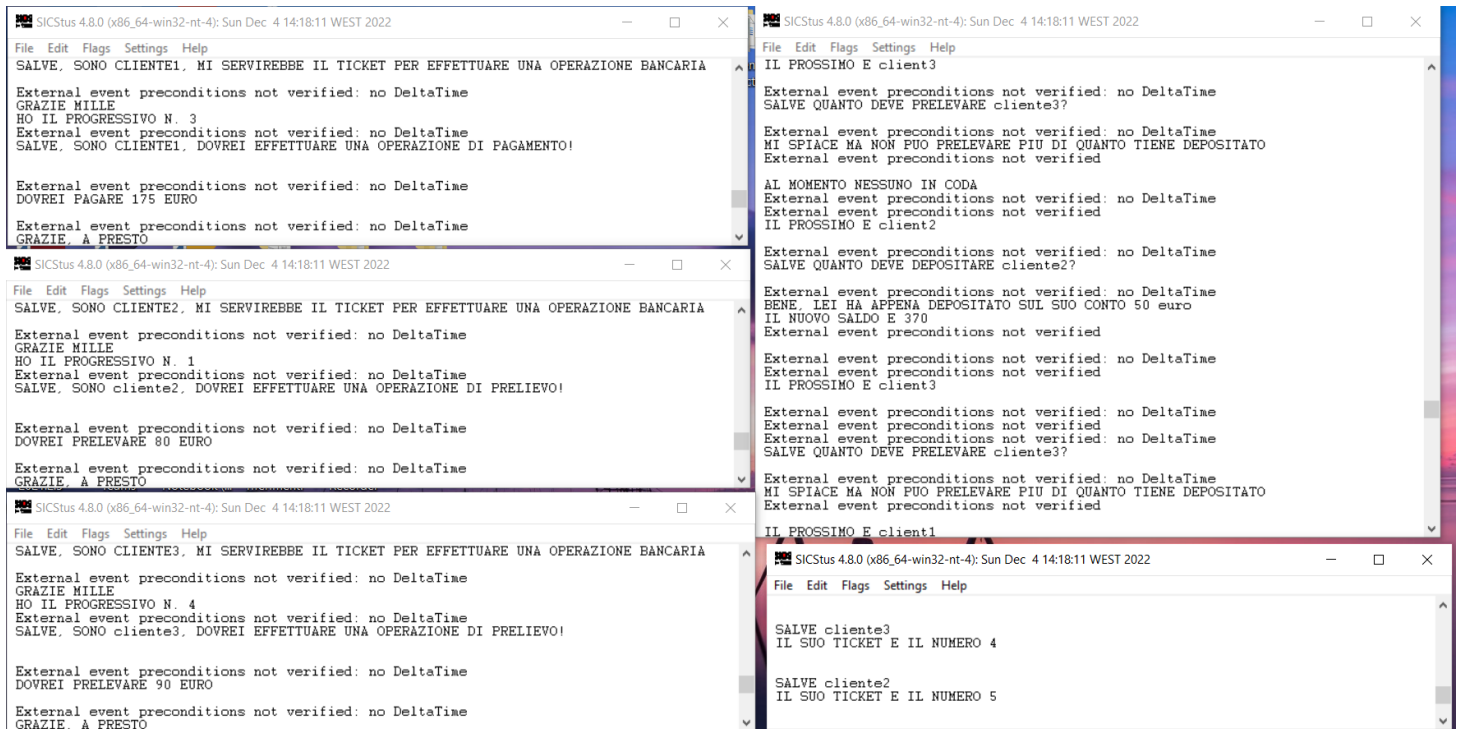


In questo esempio di esempio di esecuzione ci sono 3 clienti che vogliono effettuare alcune operazioni.

Lo sportellista si rende dunque disponibile per i clienti, mentre il ticket opera con i numeri progressivi per creare la coda. Seguendo il flusso di esecuzione, l'agente ticket riceve delle richieste di ticket, e assegna per ordine di arrivo il progressivo. Il primo della coda è cliente2, che una volta preso il ticket viene chiamato dallo sportellista per avviare l'operazione scelta (in questo caso di prelievo) con informazione di saldo aggiornato dopo il prelievo (320 euro).

Verrà poi dunque chiamato il cliente3 (poiché ha il progressivo successivo n.2) e viene avviata una nuova trattativa per poi concludere che cliente3 richiedeva di prelevare più di quanto possedeva, per cui l'operazione di prelievo viene negata.

Arriva poi il cliente (con progressivo n.3) che vuole effettuare invece una operazione di pagamento, che anche in questo caso non va a buon fine poiché non può usare più di quanto tiene in deposito.



L'esecuzione continua con il cliente 3 che richiede nuovamente una operazione, dunque si reca dal cliente ticket, gli viene assegnato il progressivo n.4 e effettua di nuovo una operazione di prelievo che non andrà a buon fine, e così via.