Progetto di Ingegneria del Software

ANNO ACCADEMICO 2017/2018

Destinatario: Sergio di Martino

DOCUMENTO DI DESIGN DEL SISTEMA

gruppo 17

Stefano Falangone	N86002437
Alessandro Caputo	N86001934
Crescenzo Cipolletta	N86001866

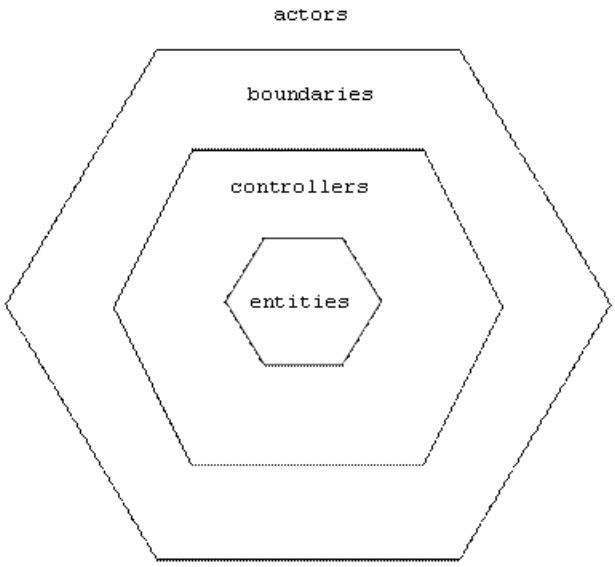
Documento di Design del Sistema

Sezioni:

1. Analisi dell'architettura del sistema
2. Diagramma delle classi
3. Diagrammi di stato
4. Diagrammi di sequenza

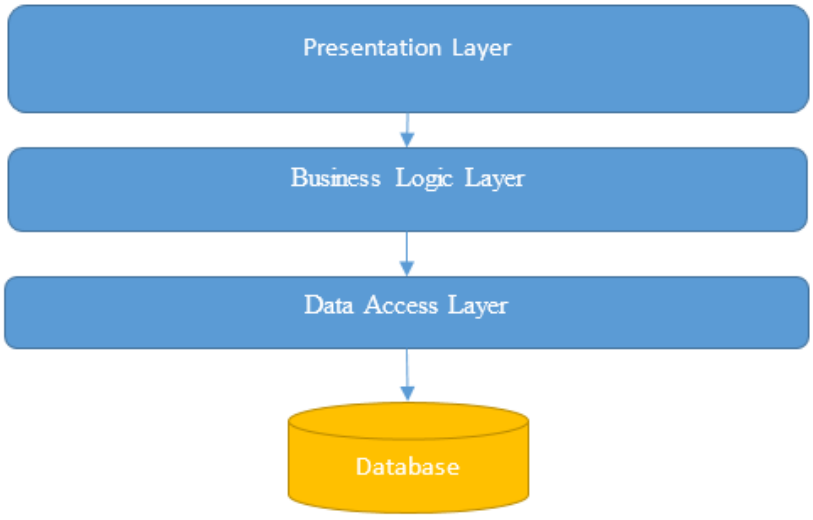
1. Analisi dell'architettura del sistema

L'architettura scelta è del tipo "client/server", vi è un server (AWS) che fornisce servizi ad una serie di Client che all'avvio si collegano agli Amazon Web Services.

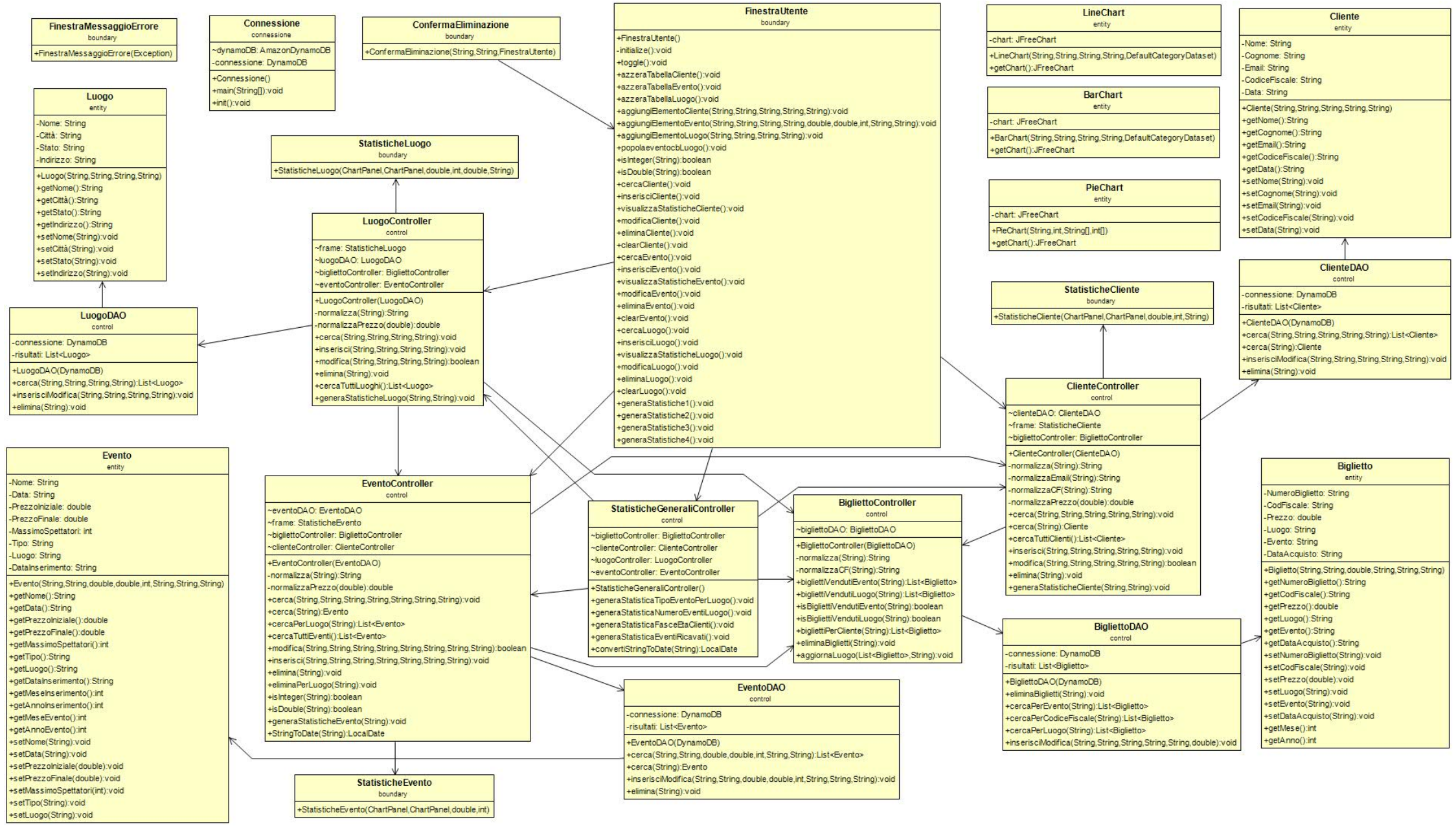


Gli utenti interagiscono con i diversi Client che possono essere utilizzati in concorrenza. Il carico di operazioni in locale è basso, mentre vi è un intenso scambio di informazioni tra client e server in quanto quest'ultimo dovrà rispondere a tutte le richieste di dati ospitando un database (DynamoDB).
L'architettura interna del software segue lo schema a 3 livelli nel quale vi è un primo strato di interfaccia grafica (boundary), un secondo di logica (controller) e un terzo di dati (entity).L'interfaccia grafica interagisce con la logica per poter effettuare le diverse operazioni senza aver alcun accesso ai dati. Per disaccoppiare il software dal database esterno è stata utilizzato il design pattern DAO.
Ogni controller non interagisce direttamente sui dati del database, ma vi è un Layer aggiuntivo che contiene - per ogni classe controller di una entity - una DAO che fornisce tutte le operazioni di CRUD sul database utilizzando le API fornite da AWS per DynamoDB.

Ogni Layer inoltre comunica soltanto fino ad un livello più in basso al fine di preservare la chiusura dell'architettura.

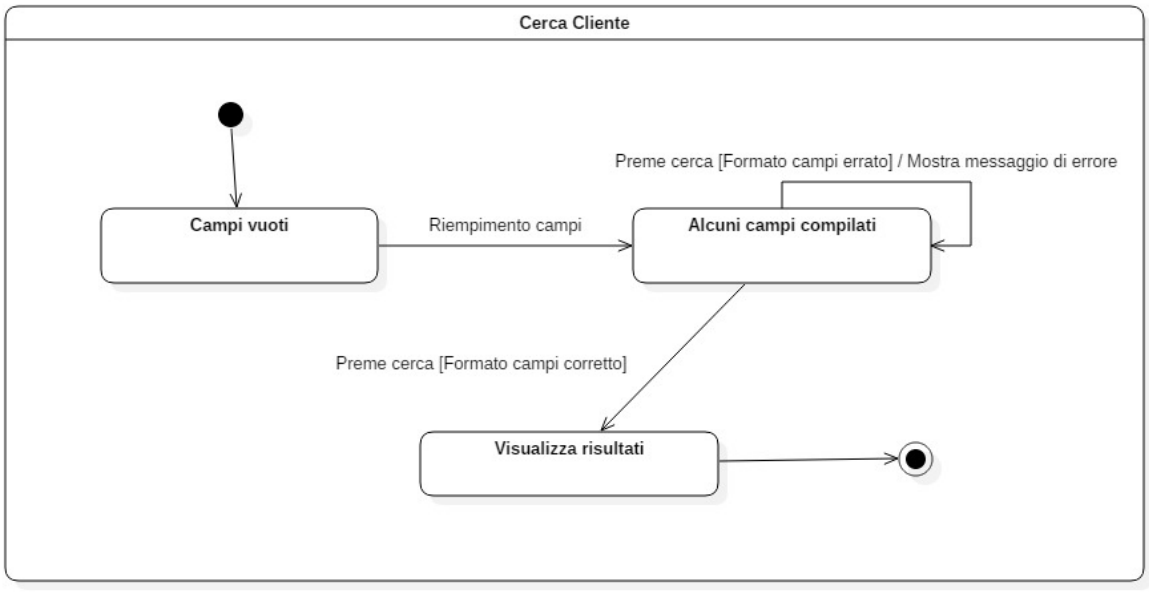


2. Diagramma delle classi

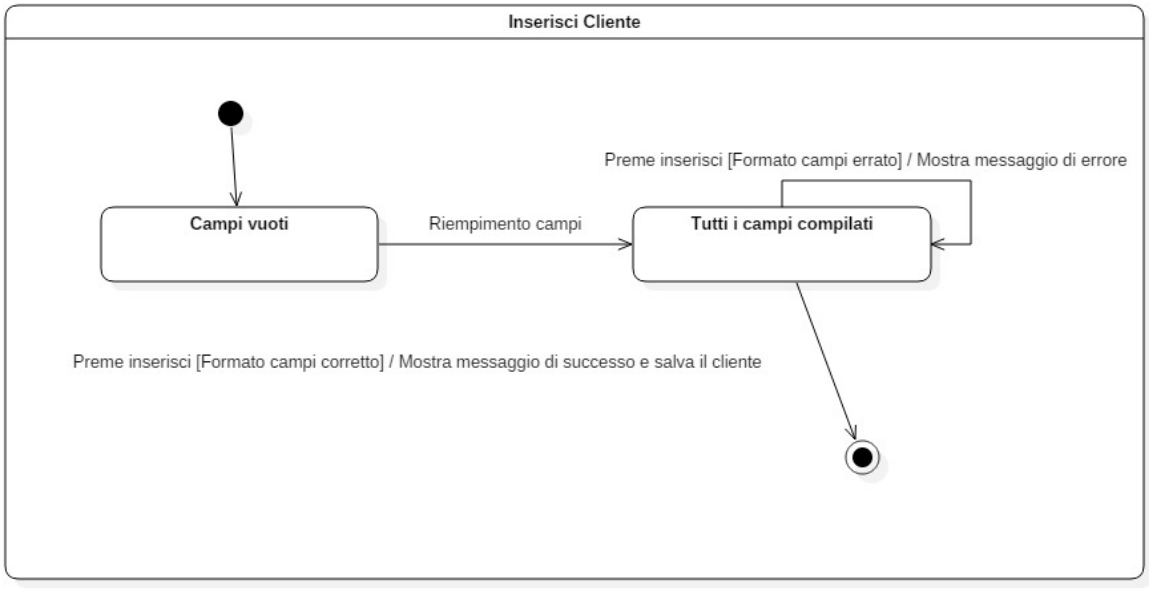

FinestraMessaggioErrore
boundary
+FinestraMessaggioErrore(Exception)
Connessione
connessione
~dynamoDB: AmazonDynamoDB
-connessione: DynamoDB
+Connessione()
+main(String[]):void
+init():void
ConfermaEliminazione
boundary
+ConfermaEliminazione(String,String,FinestraUtente)
FinestraUtente
boundary
+FinestraUtente()
-initialize():void
+toggle():void
+azzeraTabellaCliente():void
+azzeraTabellaEvento():void
+azzeraTabellaLuogo():void
+aggiungiElementoCliente(String,String,String,String,String):void
+aggiungiElementoEvento(String,String,String,String,double,double,int,String,String):void
+aggiungiElementoLuogo(String,String,String,String):void
+popolaeventocbLuogo():void
+isInteger(String):boolean
+isDouble(String):boolean
+cercaCliente():void
+inserisciCliente():void
+visualizzaStatisticheCliente():void
+modificaCliente():void
+eliminaCliente():void
+clearCliente():void
+cercaEvento():void
+inserisciEvento():void
+visualizzaStatisticheEvento():void
+modificaEvento():void
+eliminaEvento():void
+clearEvento():void
+cercaLuogo():void
+inserisciLuogo():void
+visualizzaStatisticheLuogo():void
+modificaLuogo():void
+eliminaLuogo():void
+clearLuogo():void
+generaStatistiche1():void
+generaStatistiche2():void
+generaStatistiche3():void
+generaStatistiche4():void
LineChart
entity
-chart: JFreeChart
+LineChart(String,String,String,String,DefaultCategoryDataset)
+getChart():JFreeChart
BarChart
entity
-chart: JFreeChart
+BarChart(String,String,String,String,DefaultCategoryDataset)
+getChart():JFreeChart
PieChart
entity
-chart: JFreeChart
+PieChart(String,int,String[],int[])
+getChart():JFreeChart
Cliente
entity
-Nome: String
-Cognome: String
-Email: String
-CodiceFiscale: String
-Data: String
+Cliente(String,String,String,String,String)
+getNome():String
+getCognome():String
+getEmail():String
+getCodiceFiscale():String
+getData():String
+setNome(String):void
+setCognome(String):void
+setEmail(String):void
+setCodiceFiscale(String):void
+setData(String):void
Luogo
entity
-Nome: String
-Città: String
-Stato: String
-Indirizzo: String
+Luogo(String,String,String,String)
+getNome():String
+getCittà():String
+getStato():String
+getIndirizzo():String
+setNome(String):void
+setCittà(String):void
+setStato(String):void
+setIndirizzo(String):void
StatisticheLuogo
boundary
+StatisticheLuogo(ChartPanel,ChartPanel,double,int,double,String)
LuogoController
control
~frame: StatisticheLuogo
~luogoDAO: LuogoDAO
~bigliettoController: BigliettoController
~eventoController: EventoController
+LuogoController(LuogoDAO)
-normalizza(String):String
-normalizzaPrezzo(double):double
+cerca(String,String,String,String):void
+inserisci(String,String,String,String):void
+modifica(String,String,String,String):boolean
+elimina(String):void
+cercaTuttiLuoghi():List<Luogo>
+generaStatisticheLuogo(String,String):void
LuogoDAO
control
-connessione: DynamoDB
-risultati: List<Luogo>
+LuogoDAO(DynamoDB)
+cerca(String,String,String,String):List<Luogo>
+inserisciModifica(String,String,String,String):void
+elimina(String):void
StatisticheCliente
boundary
+StatisticheCliente(ChartPanel,ChartPanel,double,int,String)
ClienteDAO
control
-connessione: DynamoDB
-risultati: List<Cliente>
+ClienteDAO(DynamoDB)
+cerca(String,String,String,String,String):List<Cliente>
+cerca(String):Cliente
+inserisciModifica(String,String,String,String,String):void
+elimina(String):void
ClienteController
control
~clienteDAO: ClienteDAO
~frame: StatisticheCliente
~bigliettoController: BigliettoController
+ClienteController(ClienteDAO)
-normalizza(String):String
-normalizzaEmail(String):String
-normalizzaCF(String):String
-normalizzaPrezzo(double):double
+cerca(String,String,String,String,String):void
+cerca(String):Cliente
+cercaTuttiClienti():List<Cliente>
+inserisci(String,String,String,String,String):void
+modifica(String,String,String,String,String):boolean
+elimina(String):void
+generaStatisticheCliente(String,String):void
Evento
entity
-Nome: String
-Data: String
-PrezzoIniziale: double
-PrezzoFinale: double
-MassimoSpettatori: int
-Tipo: String
-Luogo: String
-DataInserimento: String
+Evento(String,String,double,double,int,String,String,String)
+getNome():String
+getData():String
+getPrezzoIniziale():double
+getPrezzoFinale():double
+getMassimoSpettatori():int
+getTipo():String
+getLuogo():String
+getDataInserimento():String
+getMeseInserimento():int
+getAnnoInserimento():int
+getMeseEvento():int
+getAnnoEvento():int
+setNome(String):void
+setData(String):void
+setPrezzoIniziale(double):void
+setPrezzoFinale(double):void
+setMassimoSpettatori(int):void
+setTipo(String):void
+setLuogo(String):void
EventoController
control
~eventoDAO: EventoDAO
~frame: StatisticheEvento
~bigliettoController: BigliettoController
~clienteController: ClienteController
+EventoController(EventoDAO)
-normalizza(String):String
-normalizzaPrezzo(double):double
+cerca(String,String,String,String,String,String,String):void
+cerca(String):Evento
+cercaPerLuogo(String):List<Evento>
+cercaTuttiEventi():List<Evento>
+modifica(String,String,String,String,String,String,String,String):boolean
+inserisci(String,String,String,String,String,String,String):void
+elimina(String):void
+eliminaPerLuogo(String):void
+isInteger(String):boolean
+isDouble(String):boolean
+generaStatisticheEvento(String):void
+StringToDate(String):LocalDate
StatisticheGeneraliController
control
~bigliettoController: BigliettoController
~clienteController: ClienteController
~luogoController: LuogoController
~eventoController: EventoController
+StatisticheGeneraliController()
+generaStatisticaTipoEventoPerLuogo():void
+generaStatisticaNumeroEventiLuogo():void
+generaStatisticaFasceEtaClienti():void
+generaStatisticaEventiRicavati():void
+convertiStringToDate(String):LocalDate
BigliettoController
control
~bigliettoDAO: BigliettoDAO
+BigliettoController(BigliettoDAO)
-normalizza(String):String
-normalizzaCF(String):String
+bigliettiVendutiEvento(String):List<Biglietto>
+bigliettiVendutiLuogo(String):List<Biglietto>
+isBigliettiVendutiEvento(String):boolean
+isBigliettiVendutiLuogo(String):boolean
+bigliettiPerCliente(String):List<Biglietto>
+eliminaBiglietti(String):void
+aggiornaLuogo(List<Biglietto>,String):void
Biglietto
entity
-NumeroBiglietto: String
-CodFiscale: String
-Prezzo: double
-Luogo: String
-Evento: String
-DataAcquisto: String
+Biglietto(String,String,double,String,String,String)
+getNumeroBiglietto():String
+getCodFiscale():String
+getPrezzo():double
+getLuogo():String
+getEvento():String
+getDataAcquisto():String
+setNumeroBiglietto(String):void
+setCodFiscale(String):void
+setPrezzo(double):void
+setLuogo(String):void
+setEvento(String):void
+setDataAcquisto(String):void
+getMese():int
+getAnno():int
BigliettoDAO
control
-connessione: DynamoDB
-risultati: List<Biglietto>
+BigliettoDAO(DynamoDB)
+eliminaBiglietti(String):void
+cercaPerEvento(String):List<Biglietto>
+cercaPerCodiceFiscale(String):List<Biglietto>
+cercaPerLuogo(String):List<Biglietto>
+inserisciModifica(String,String,String,String,String,double):void
EventoDAO
control
-connessione: DynamoDB
-risultati: List<Evento>
+EventoDAO(DynamoDB)
+cerca(String,String,double,double,int,String,String):List<Evento>
+cerca(String):Evento
+inserisciModifica(String,String,double,double,int,String,String,String):void
+elimina(String):void
StatisticheEvento
boundary
+StatisticheEvento(ChartPanel,ChartPanel,double,int)

3. Diagrammi di stato

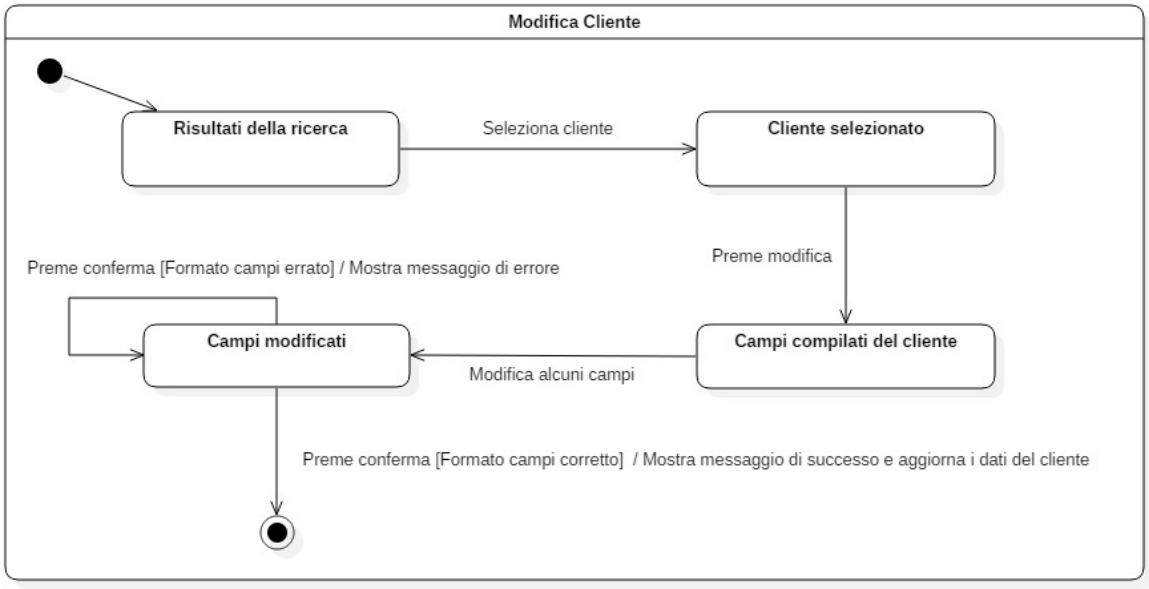
Cerca cliente



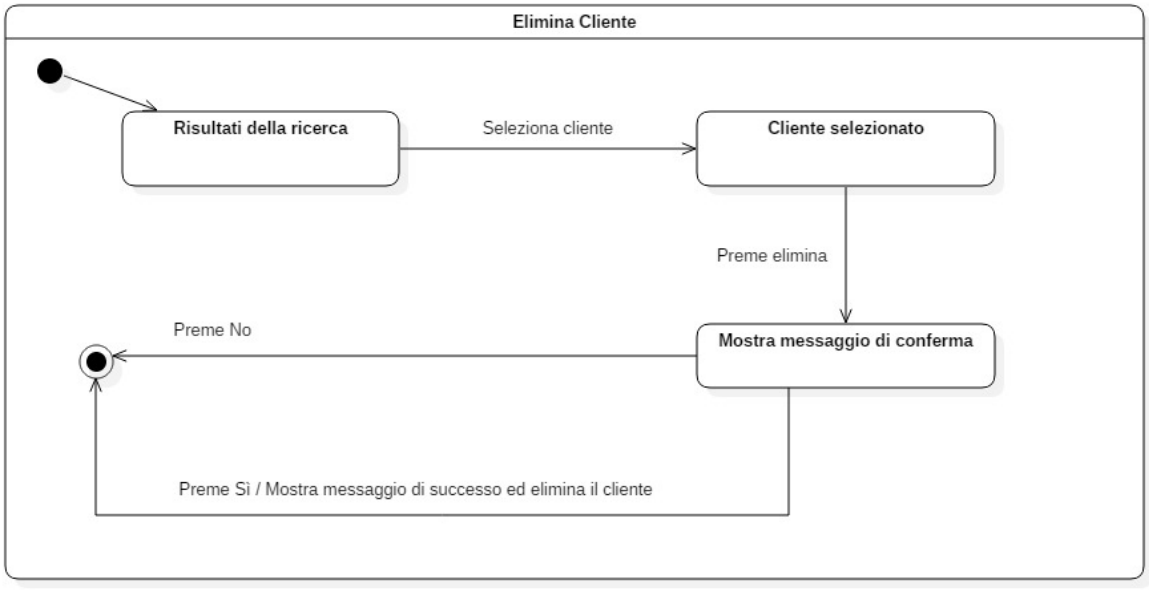
Inserisci cliente



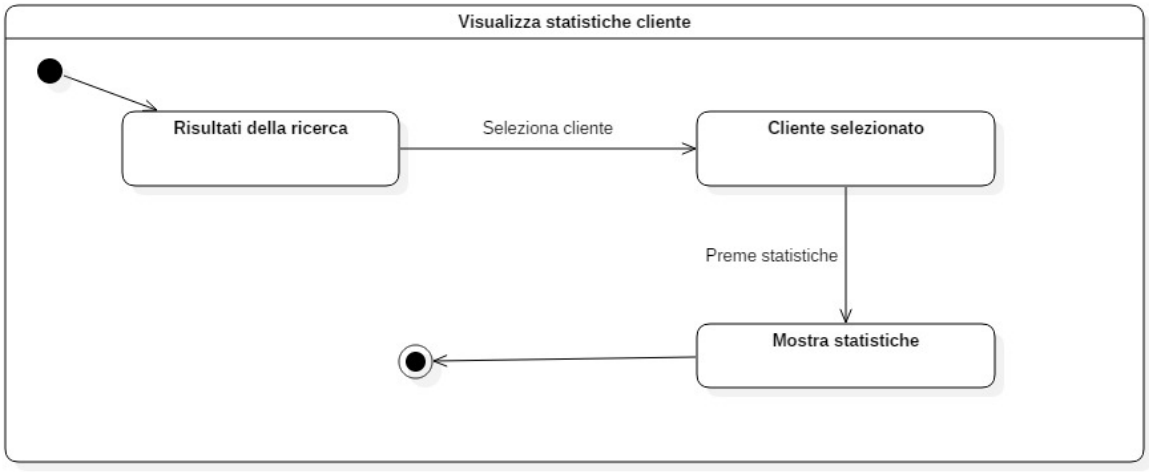
Modifica cliente



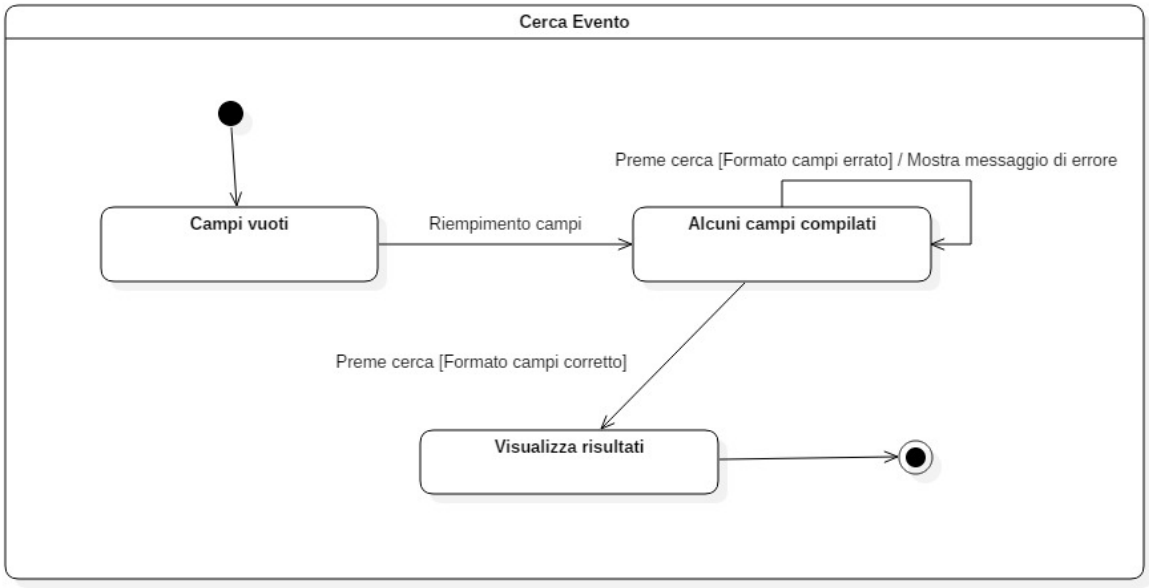
Elimina cliente



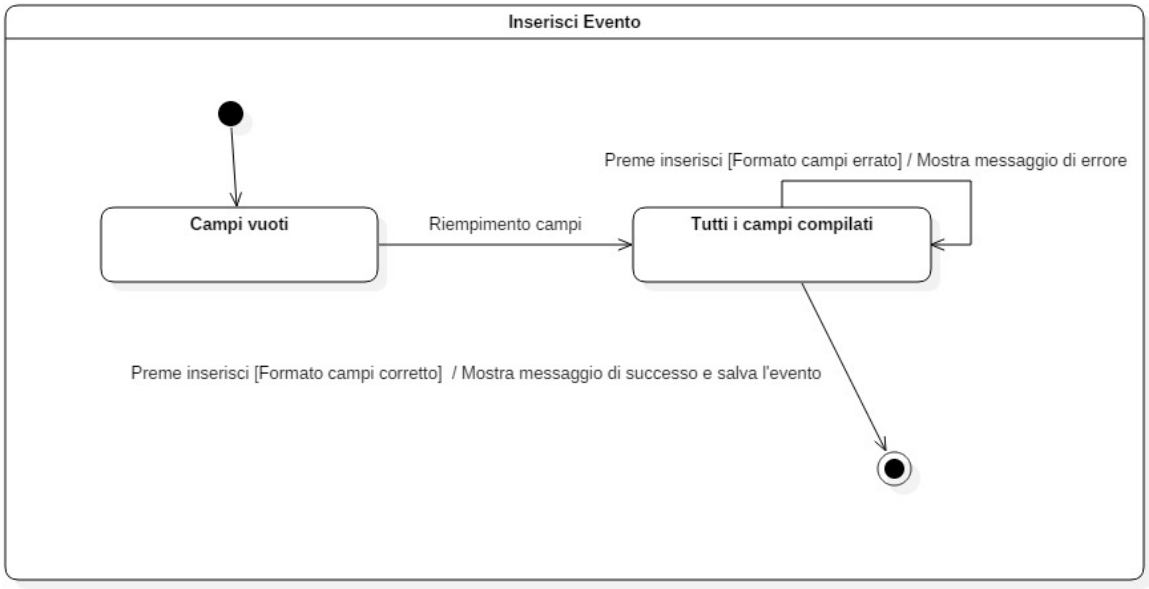
Visualizza statistiche cliente



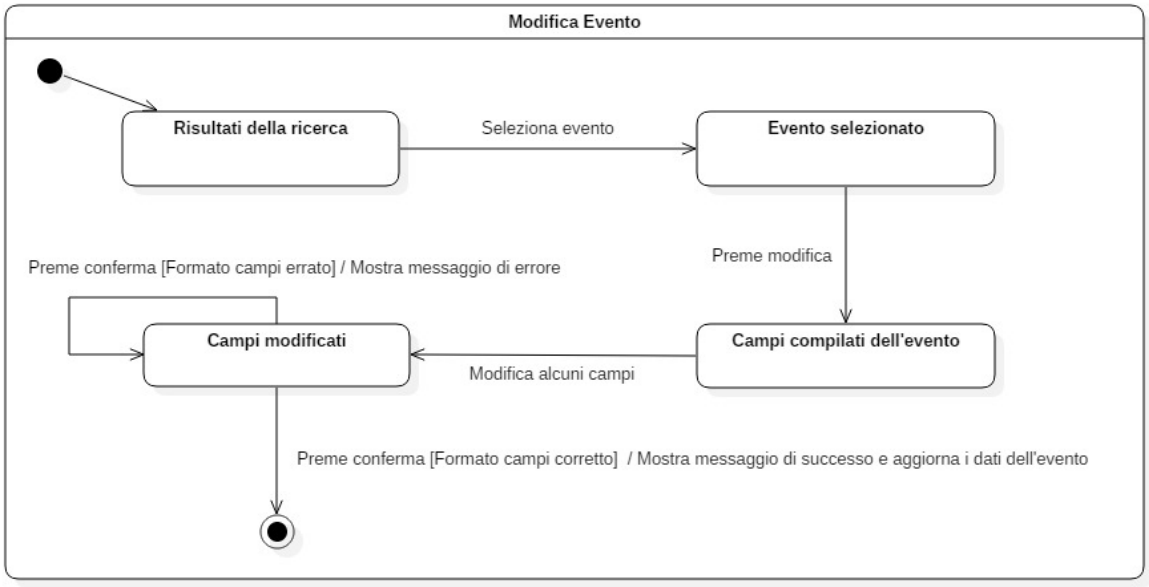
Cerca evento



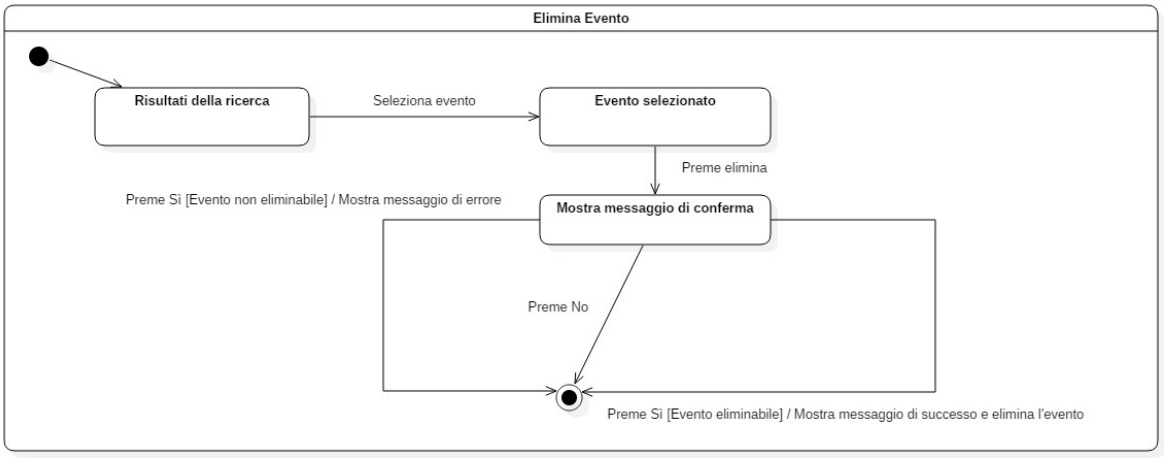
Inserisci evento



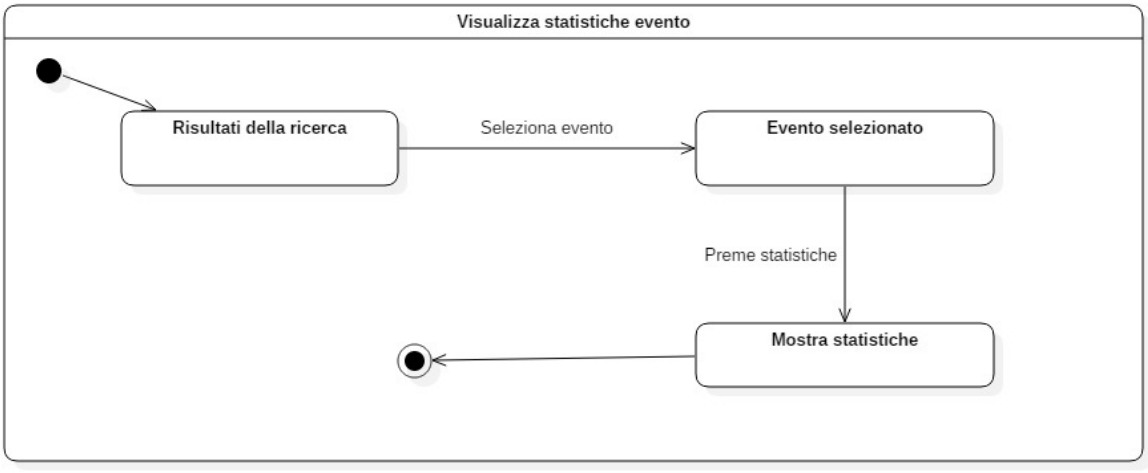
Modifica evento



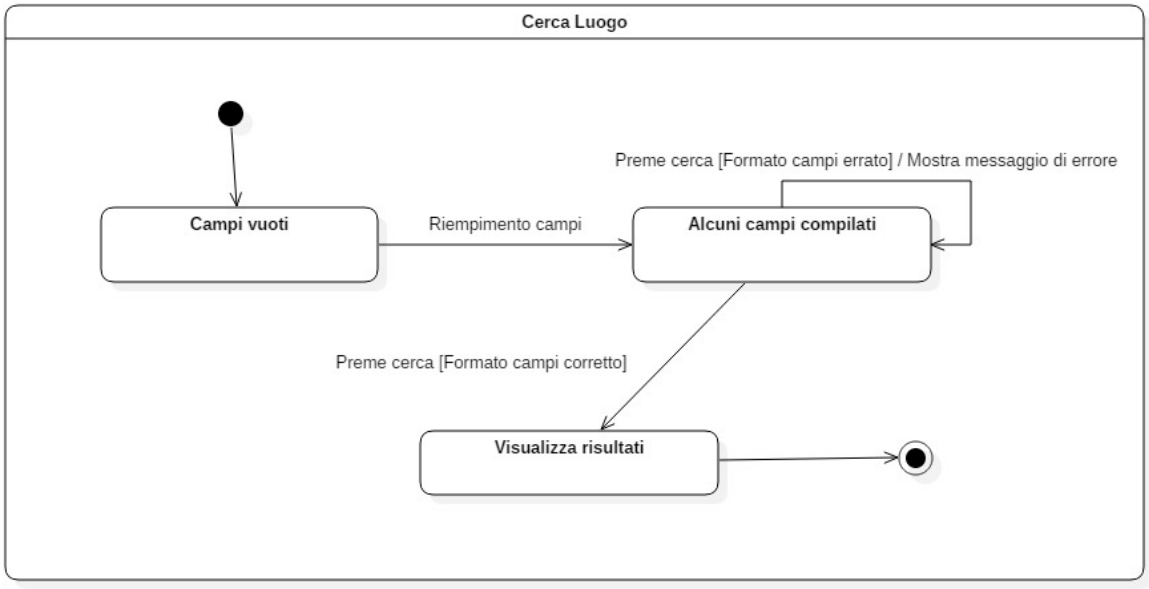
Elimina evento



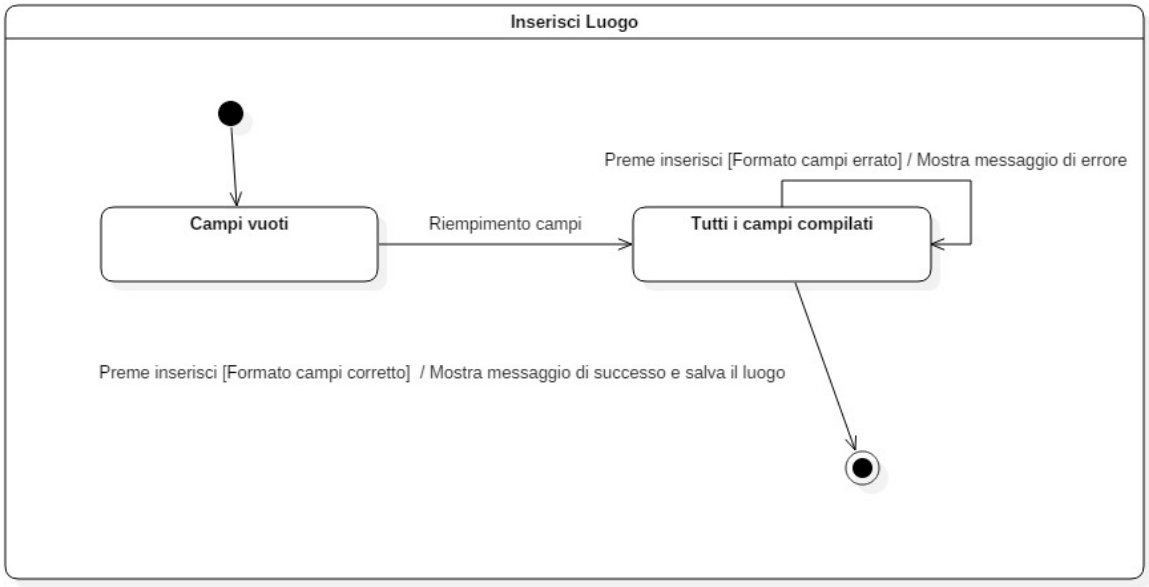
Visualizza statistiche evento



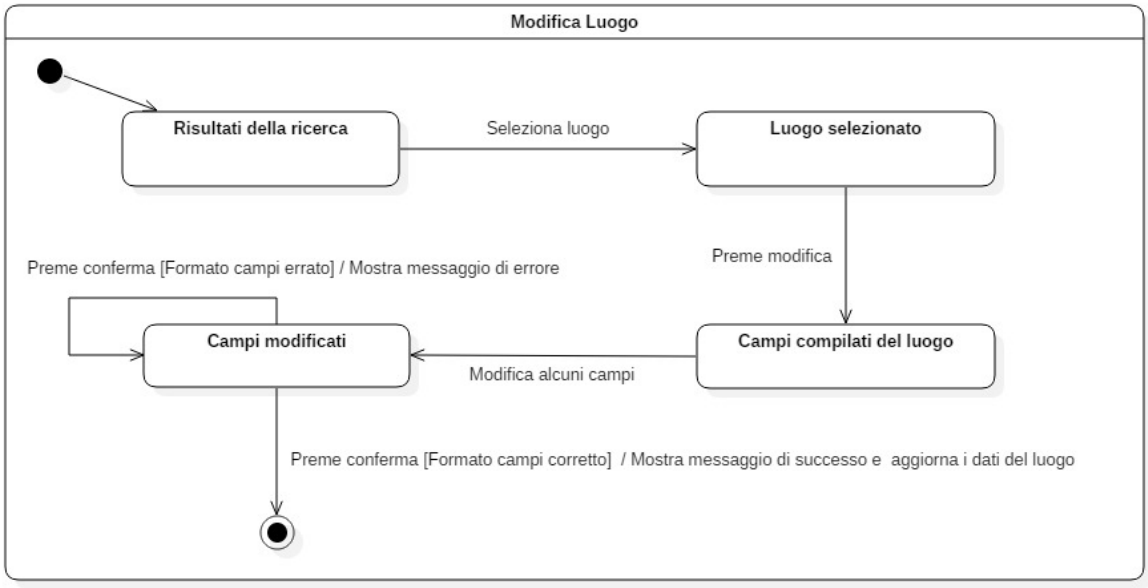
Cerca luogo



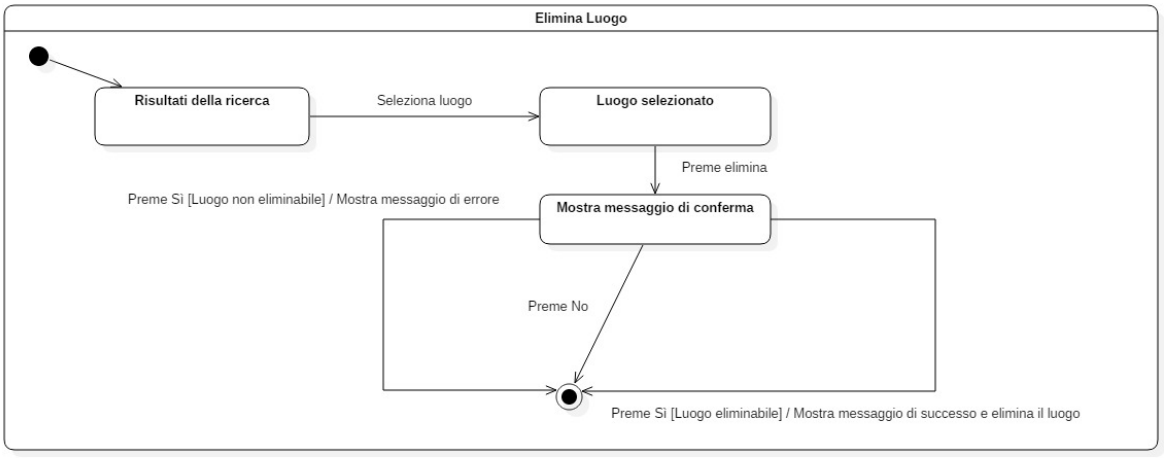
Inserisci luogo



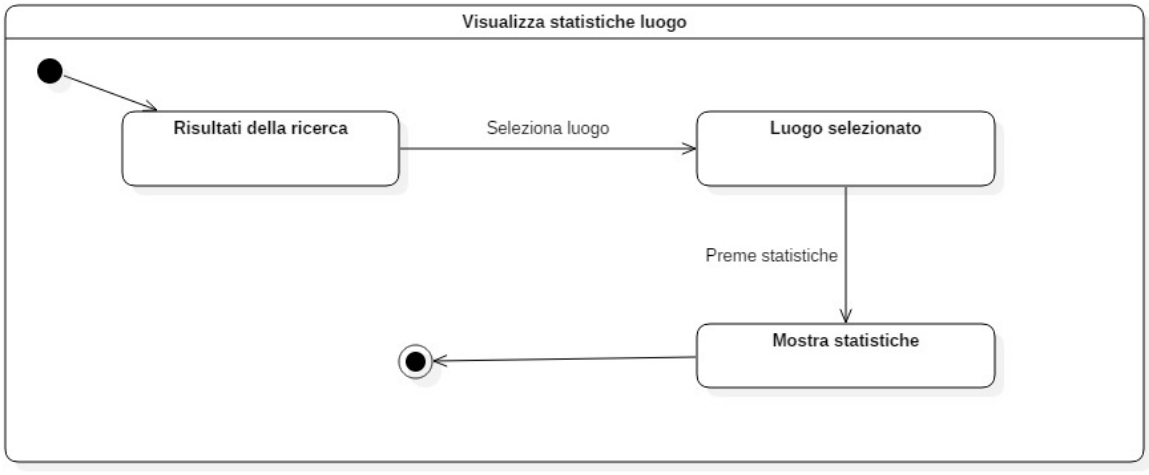
Modifica luogo



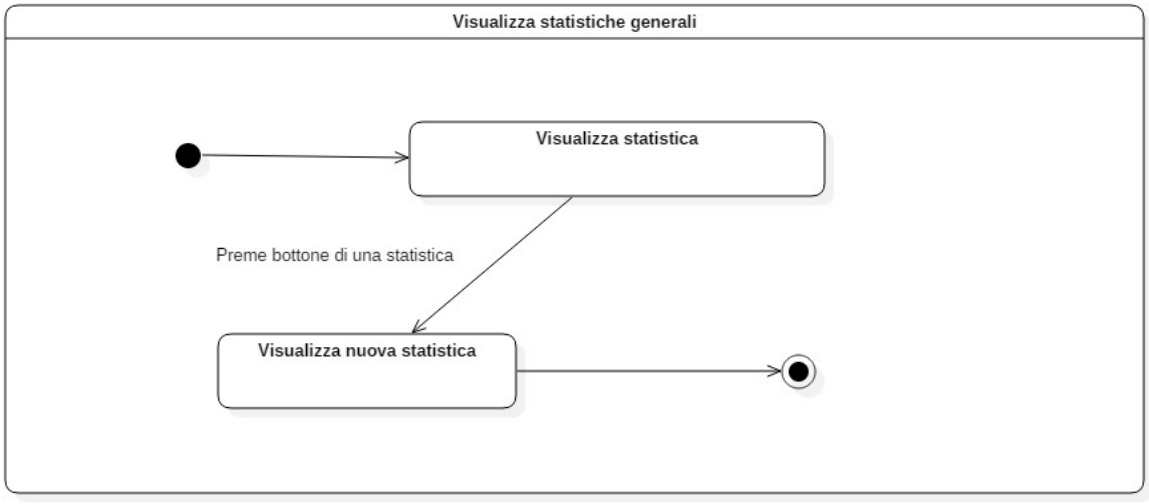
Elimina luogo



Visualizza statistiche luogo

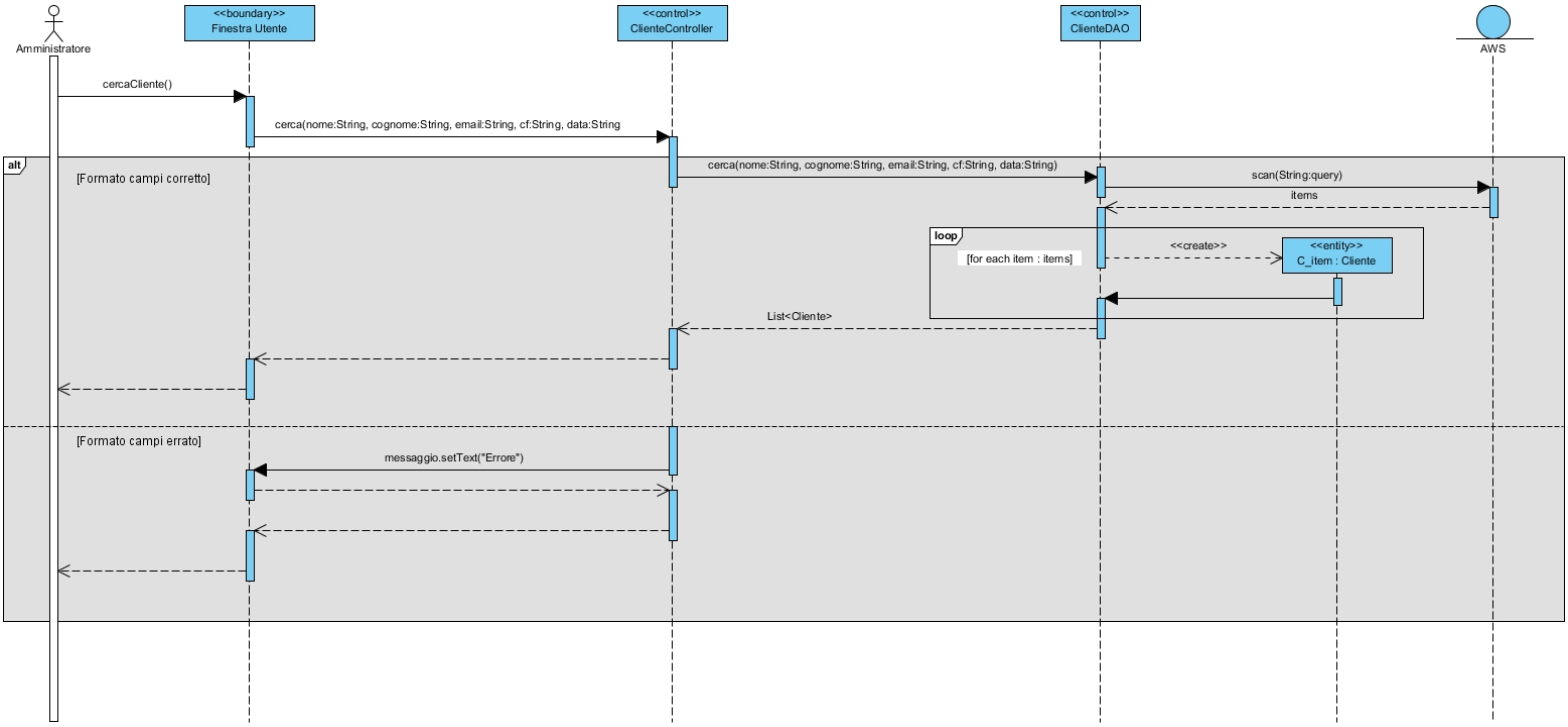


Visualizza statistiche generali

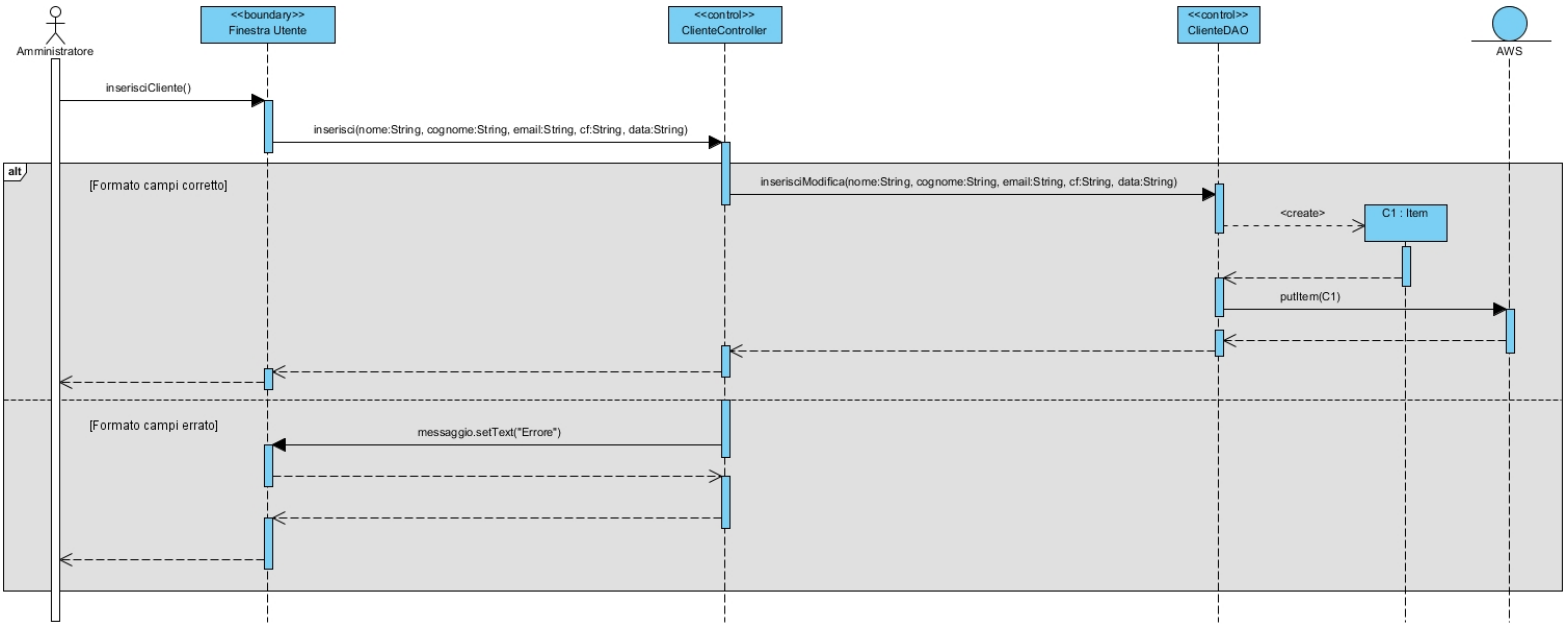


4. Diagrammi di sequenza

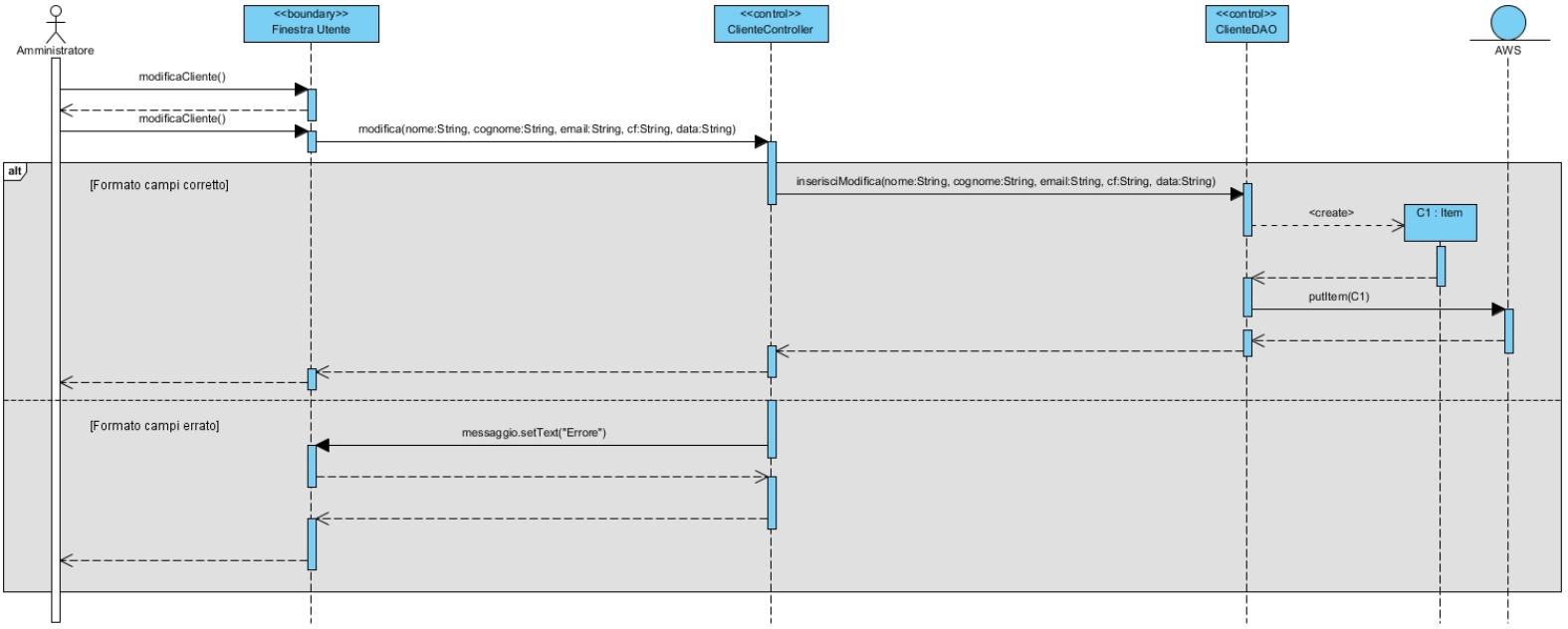
Cerca cliente



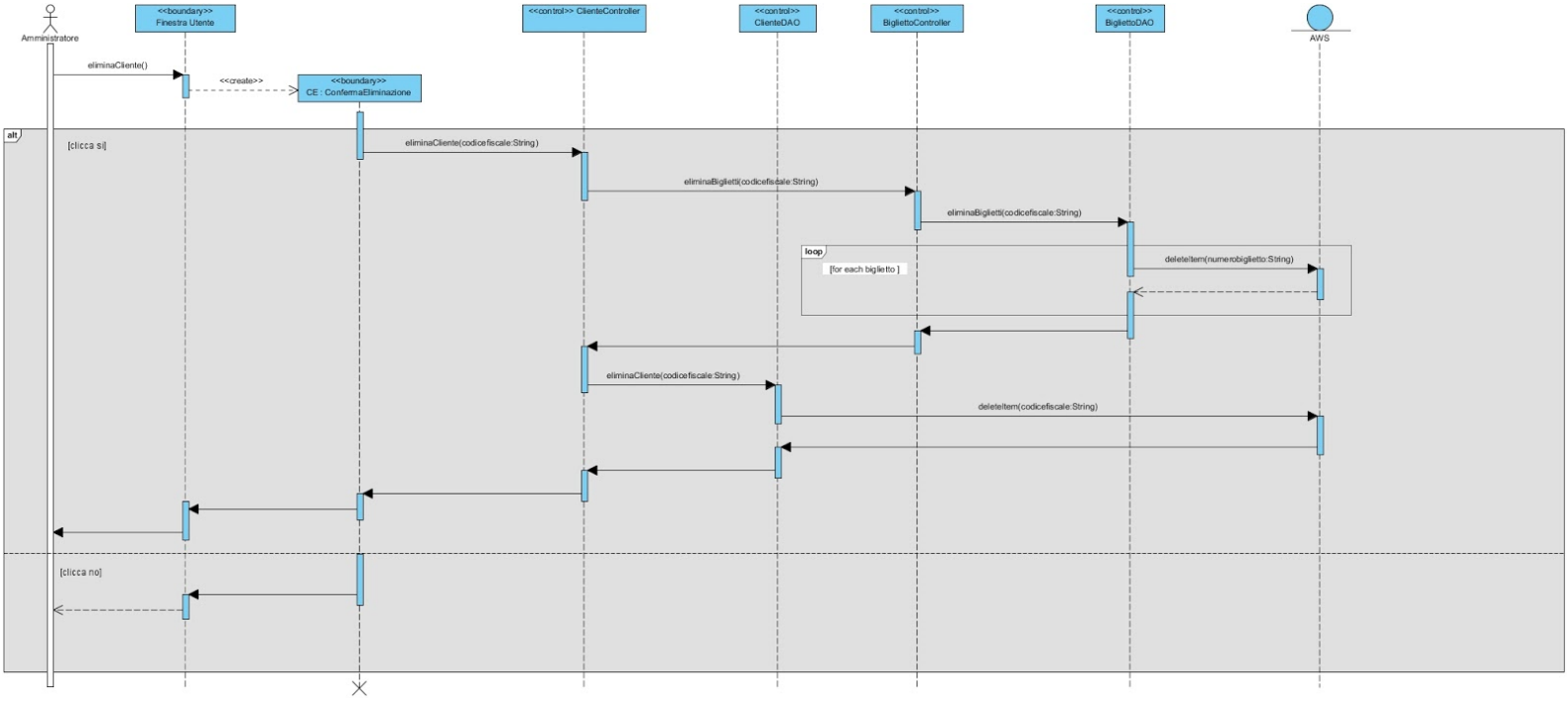
Inserisci cliente



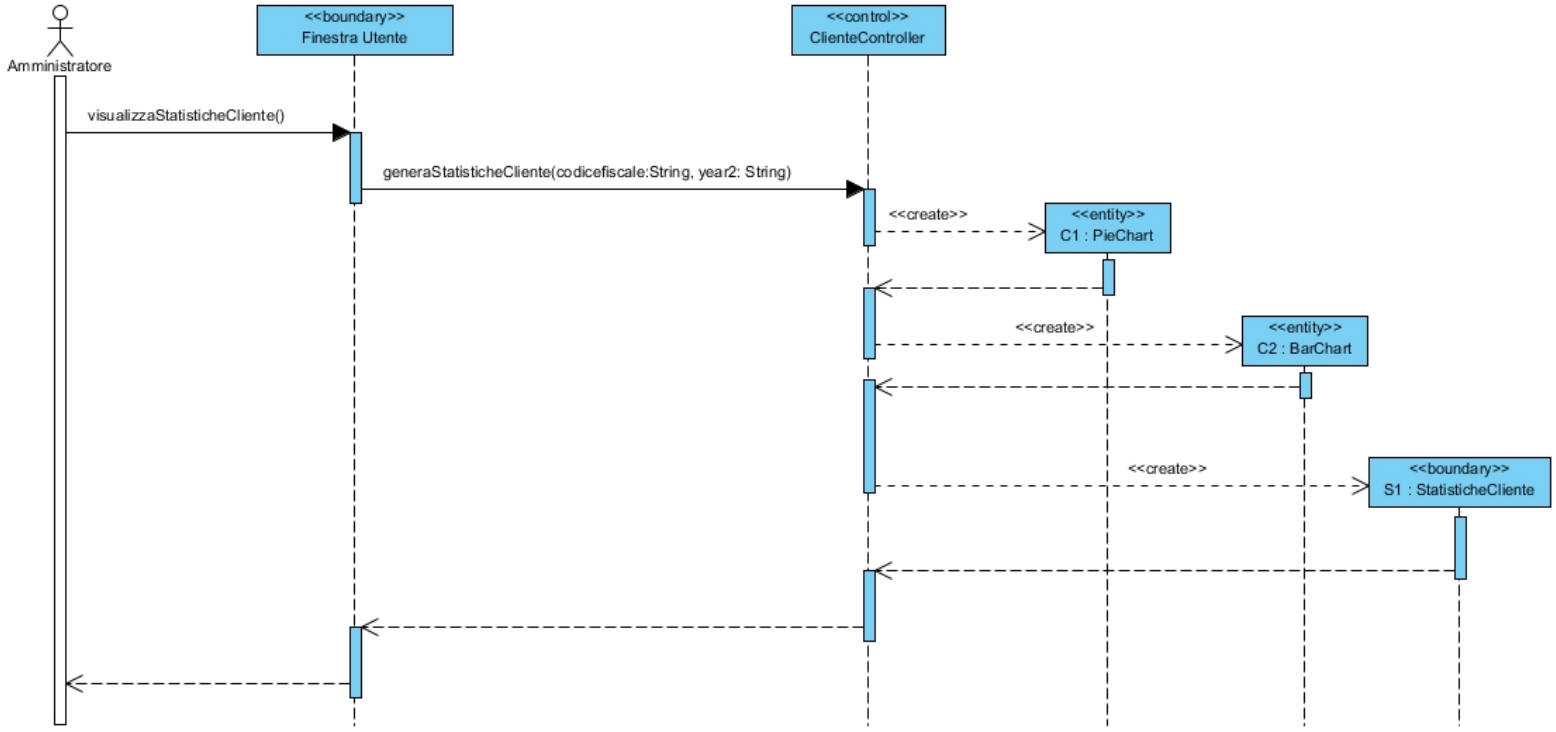
Modifica cliente



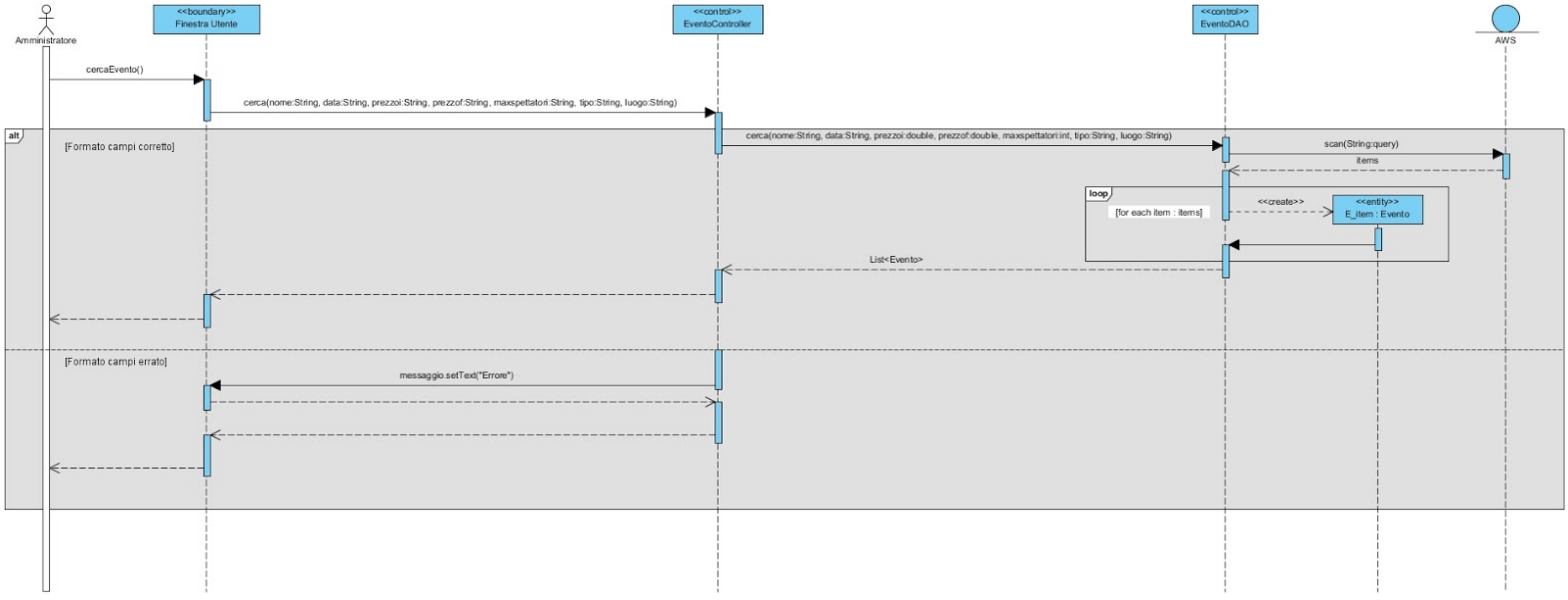
Elimina cliente



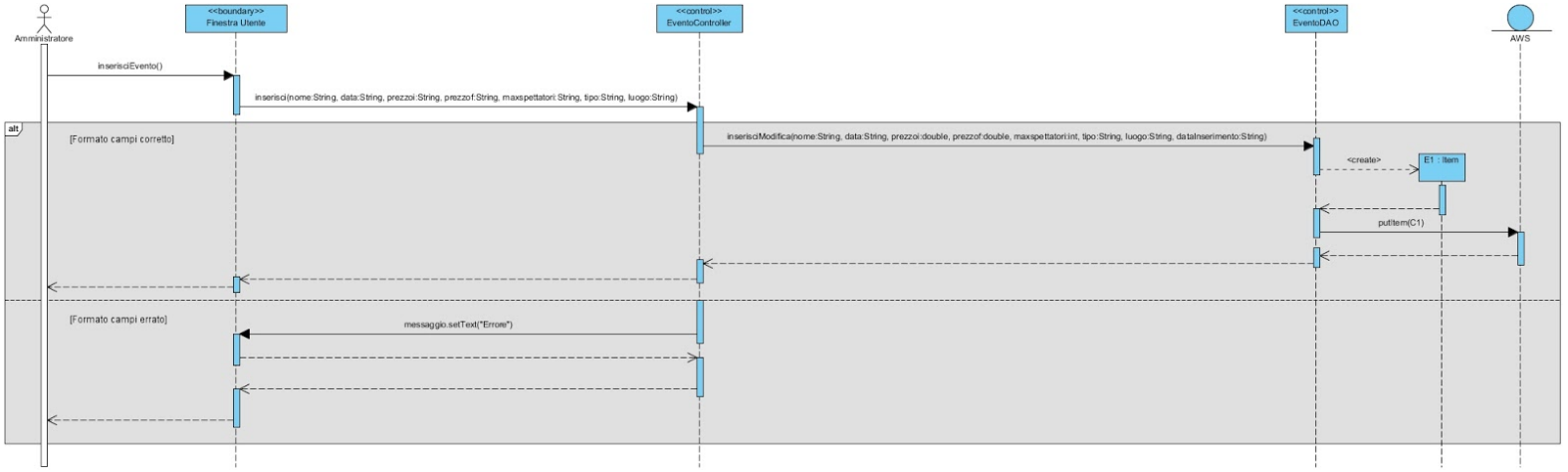
Visualizza statistiche cliente



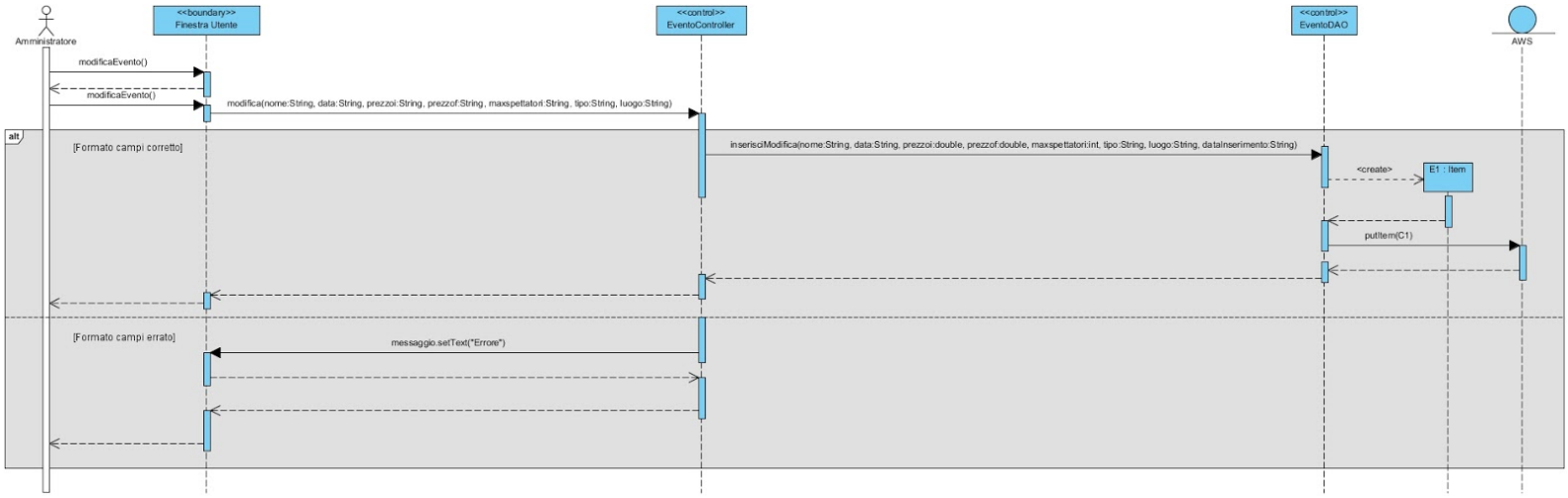
Cerca evento



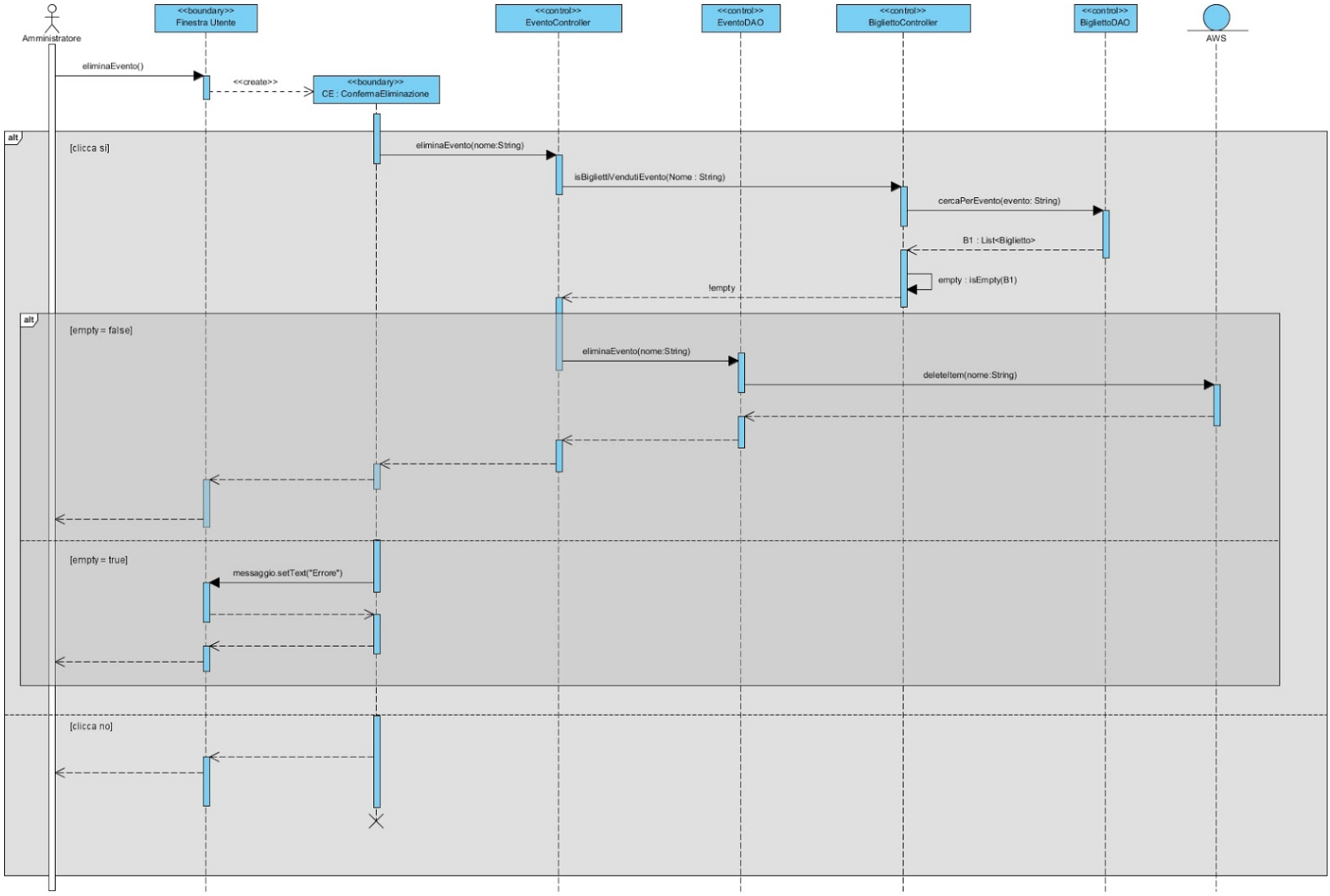
Inserisci evento



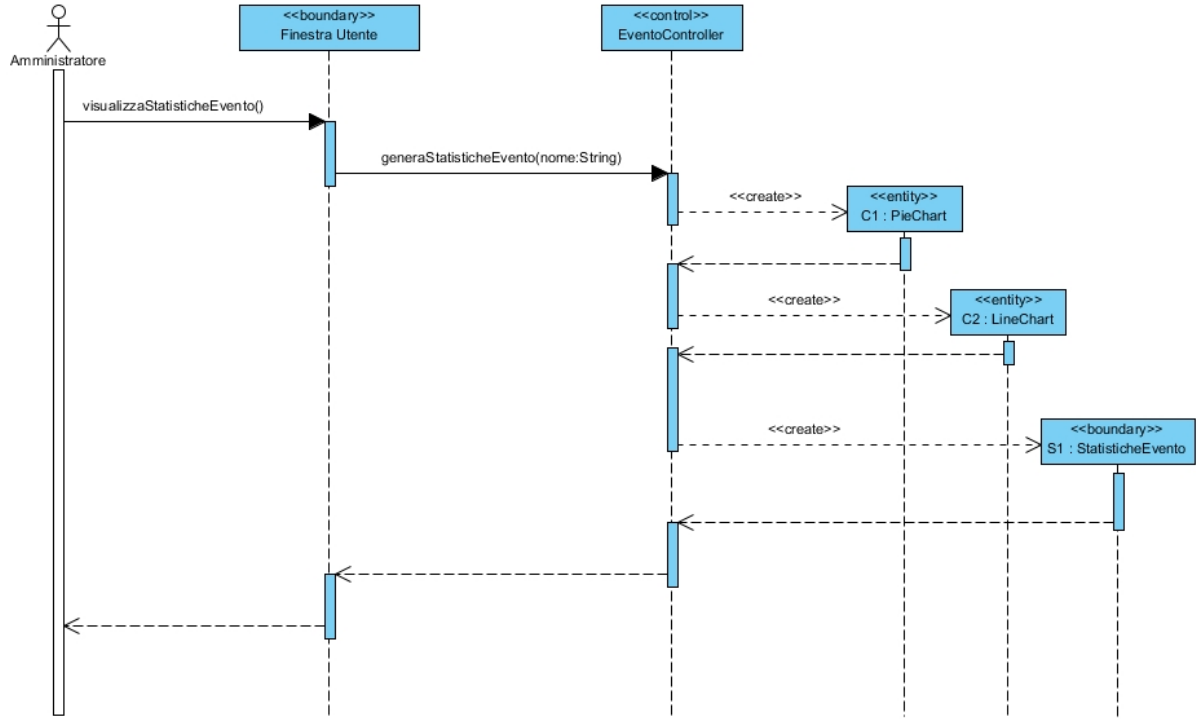
Modifica evento



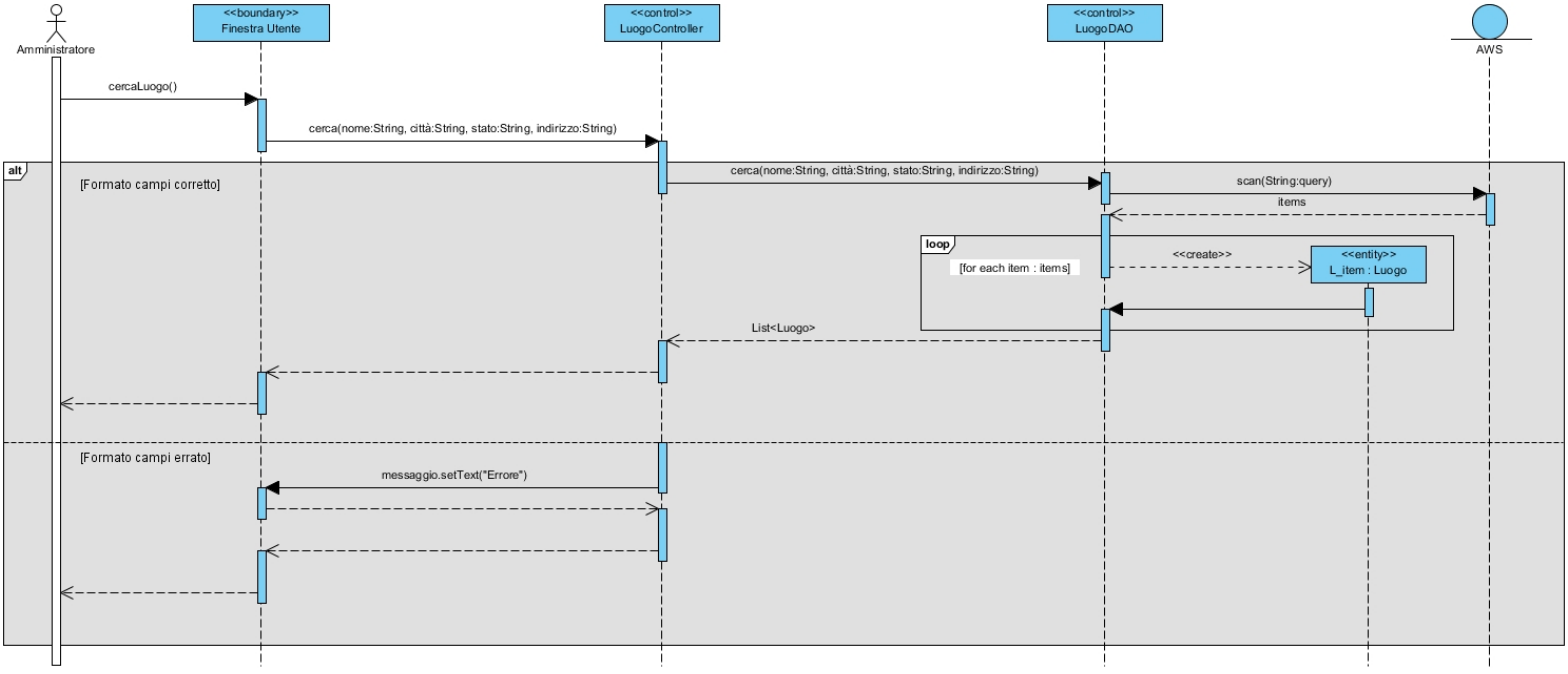
Elimina evento



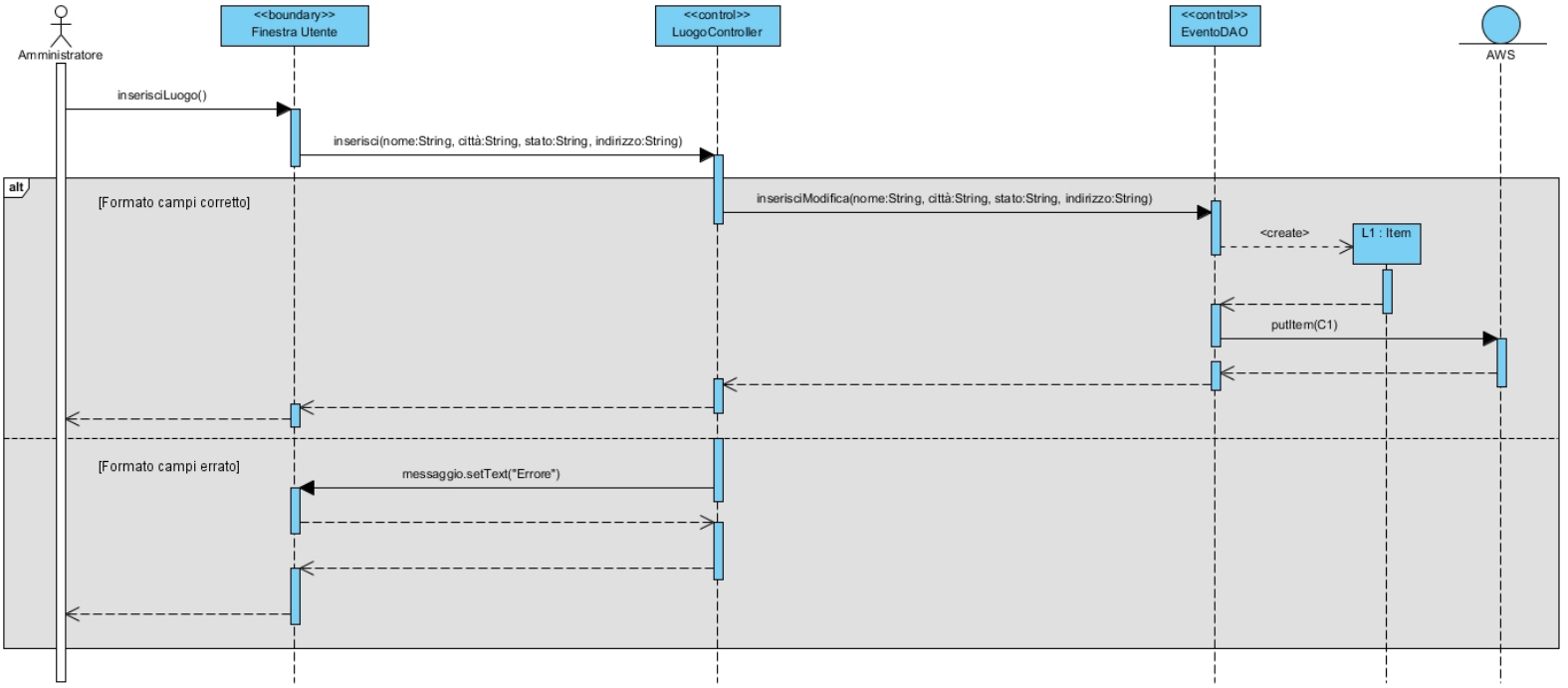
Visualizza statistiche evento



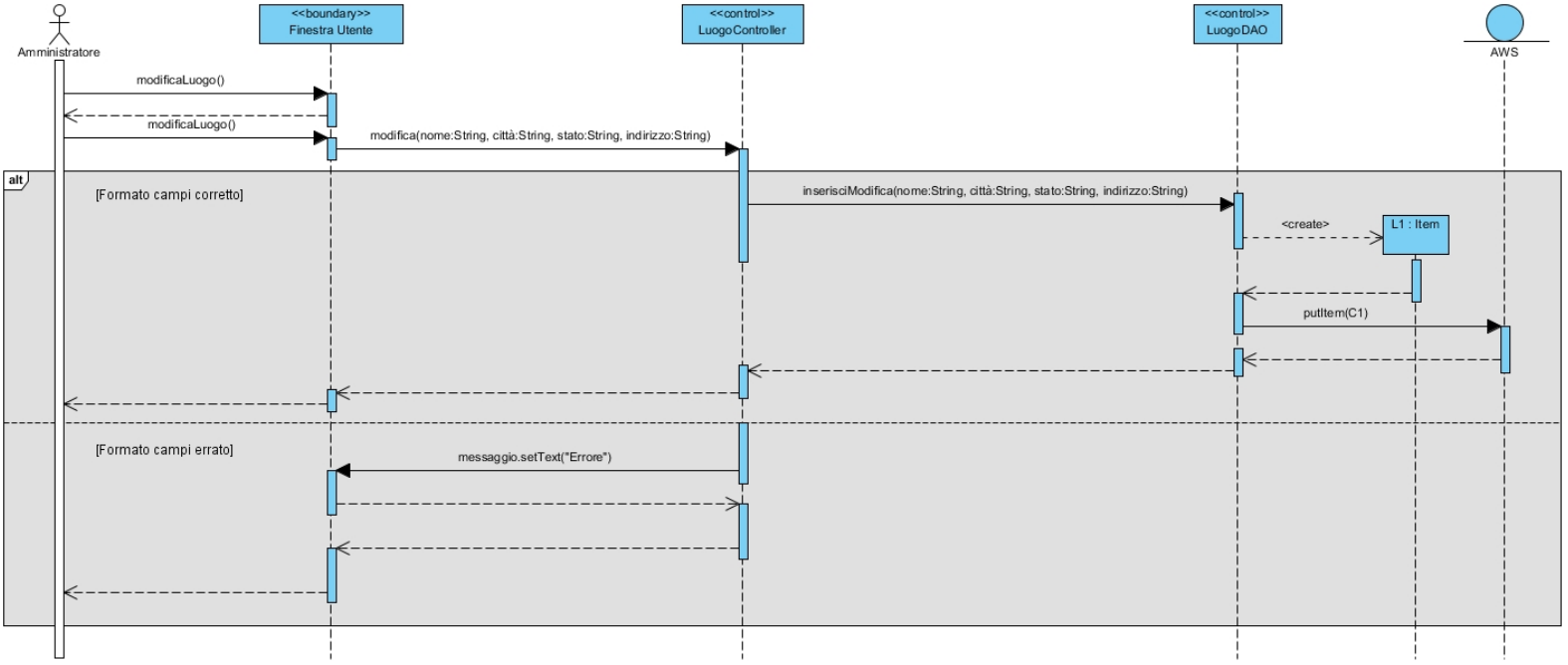
Cerca luogo



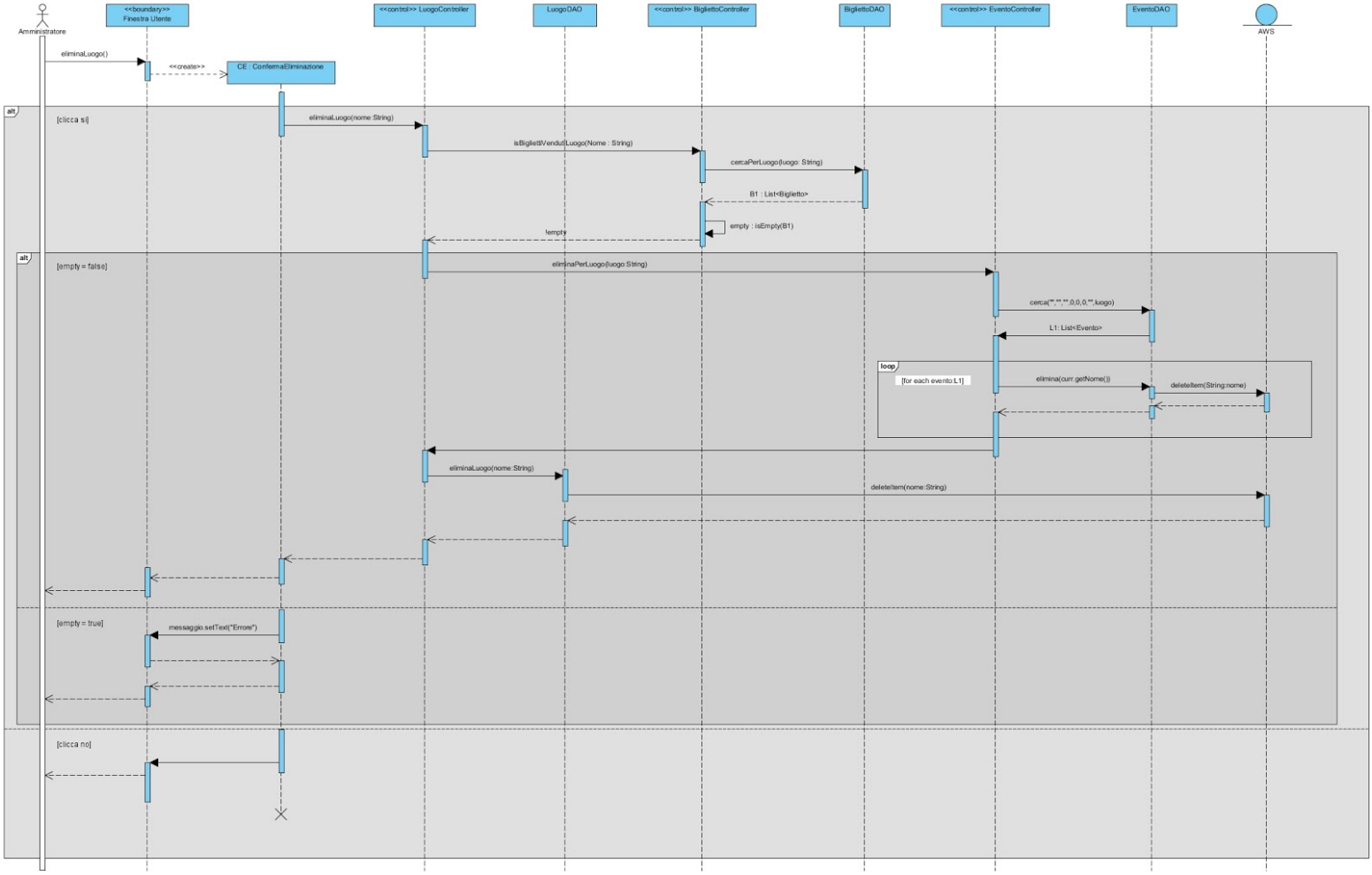
Inserisci luogo



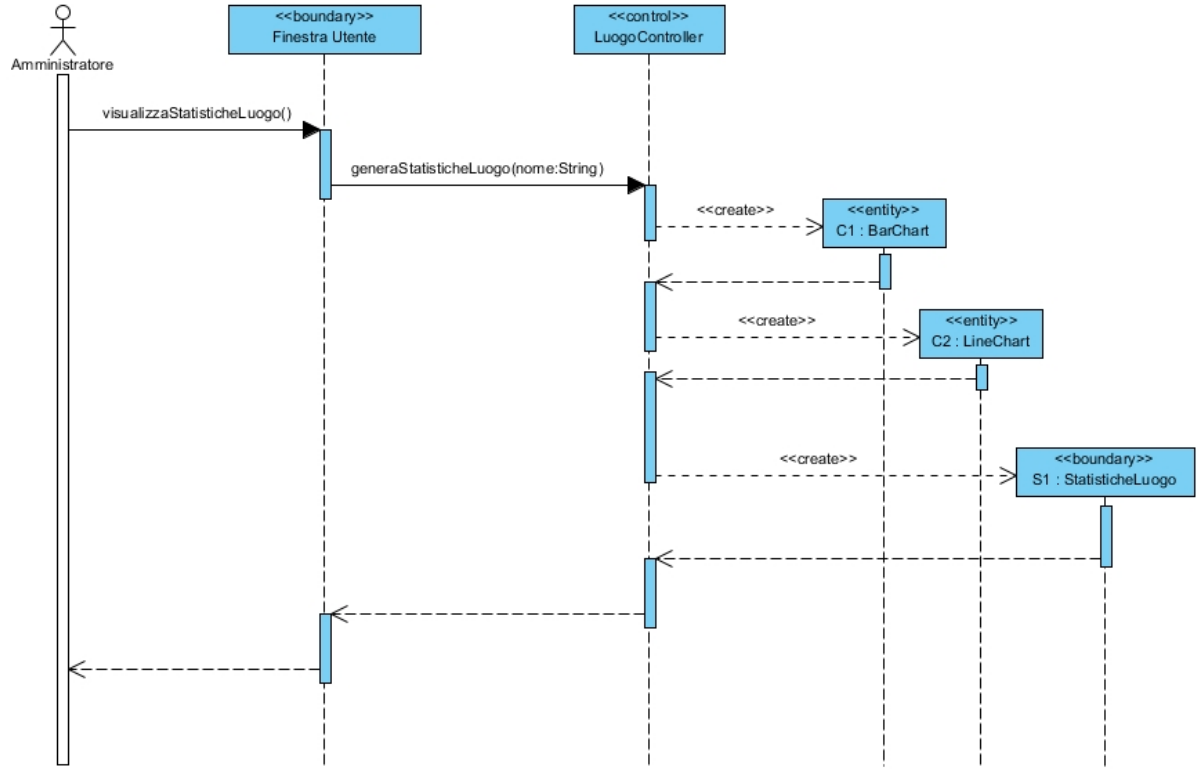
Modifica luogo



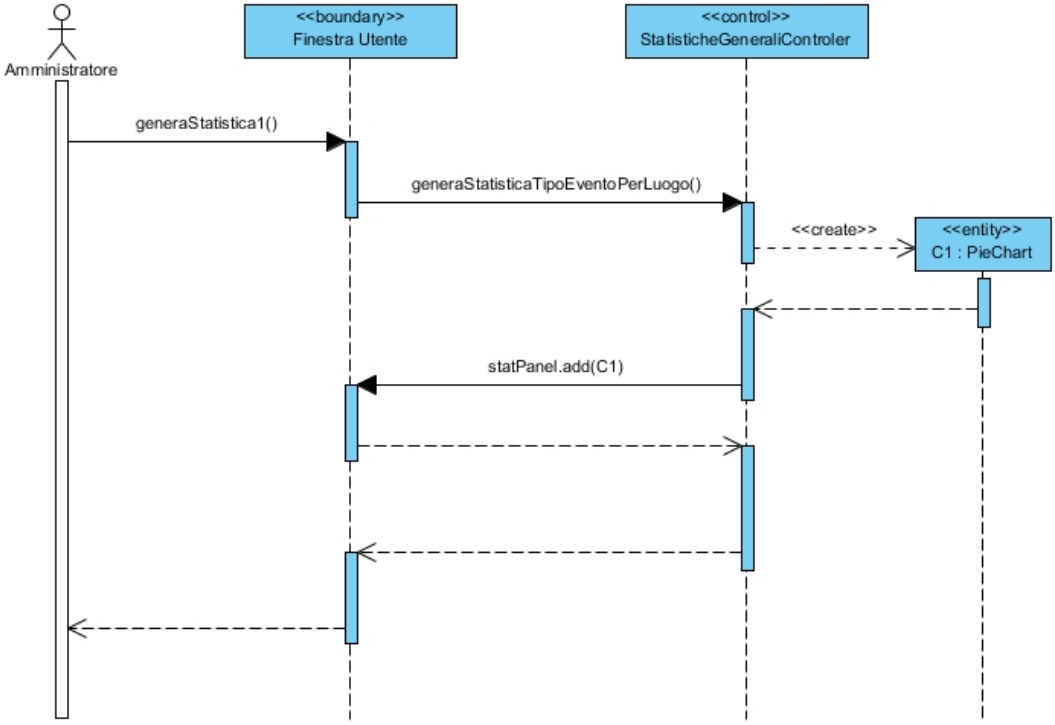
Elimina luogo



Visualizza statistiche luogo



Visualizza statistiche generali



generaStatisticaEventiRicavati, generaStatisticaNumeroEventiLuogo e generaStatisticaFasceEtaClienti sono del tutto analoghi (invocazione al metodo di nome diverso e a anziché creare un PieChart viene creato rispettivamente BarChart, BarChart, LineChart)