

Documentazione di progetto

-
Design UML



POLITECNICO
MILANO 1863

Tema C

Gruppo 1

El Abiad Riad Eddin

Randazzo Lorenzo

Terraneo Andrea

Class Diagram

All'interno del Class Diagram vengono evidenziati i tipi che vengono utilizzati nel progetto, i loro attributi, metodi e relazioni tra essi.

Le generalizzazioni osservabili avvengono tra Utente, superclasse di Dipendente e Cliente; Dipendente, superclasse di Manutentore e ImpiegatoAmministrativo; ComponenteRete, superclasse di Contatore, PannelloSolare, LineaElettrica, ComponenteGenerico, CentraleManutenzione.

Vengono mostrate le seguenti relazioni:

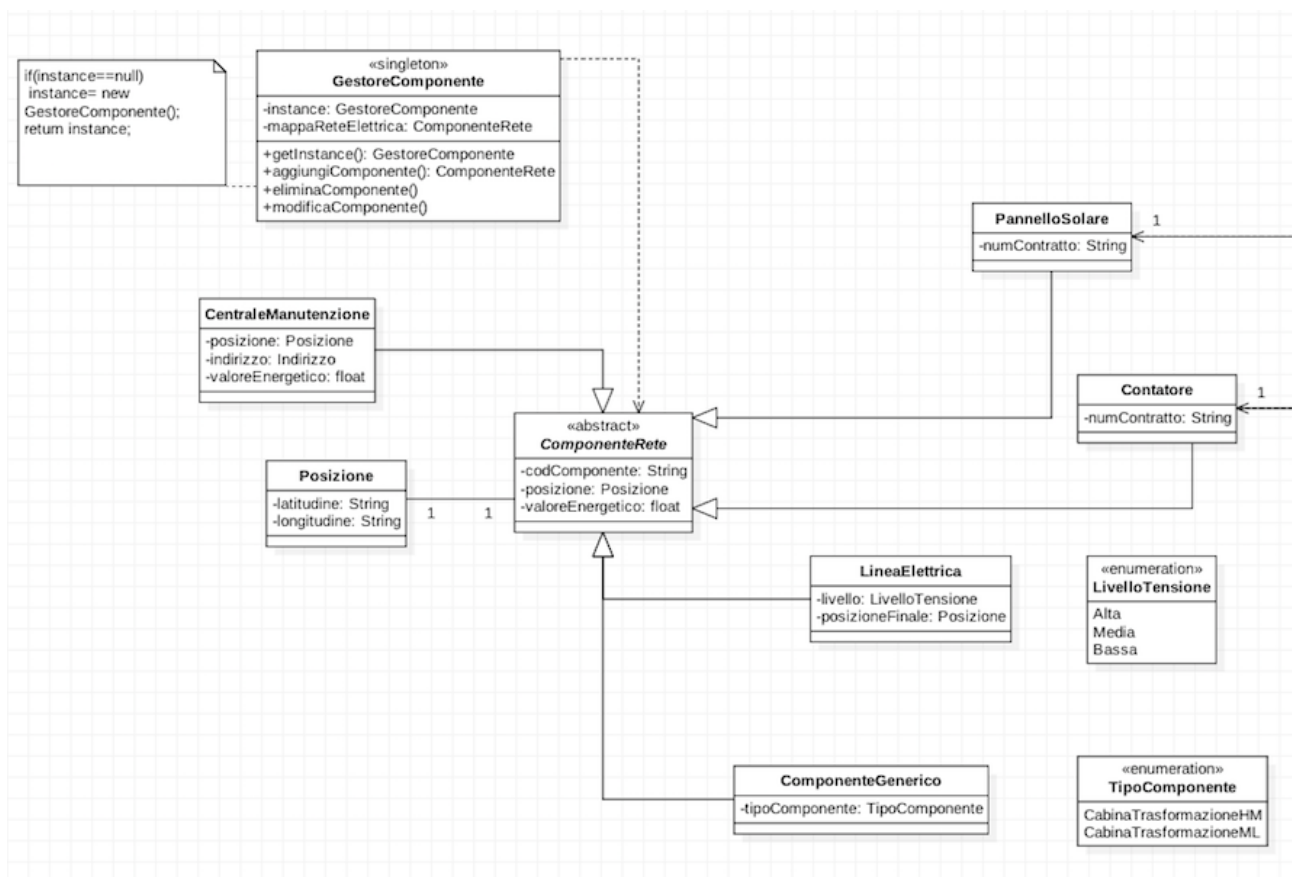
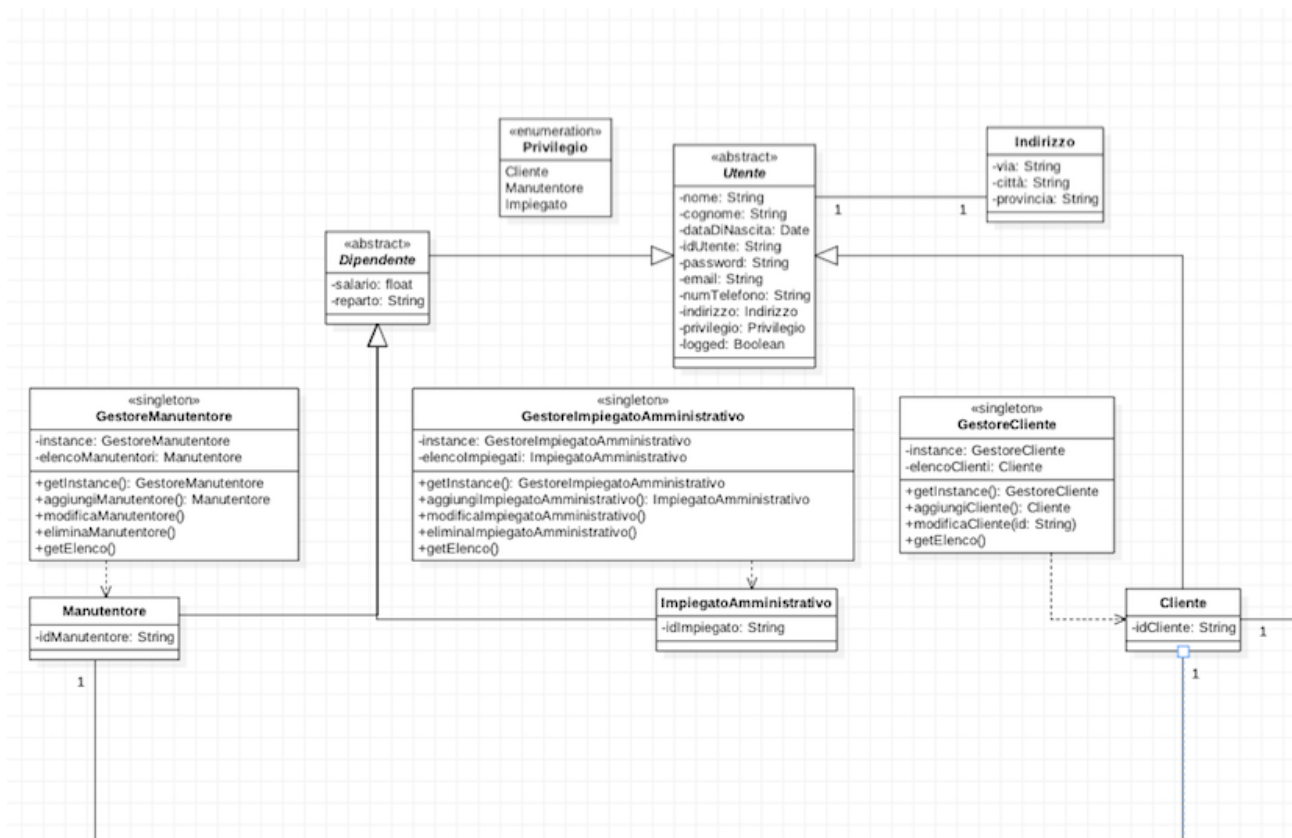
- Ad ogni utente corrisponde un solo indirizzo, ad ogni indirizzo corrisponde un solo utente.
- Ogni contratto ha un solo intestatario di tipo Cliente, ogni cliente può avere minimo un contratto.
- Una fattura può essere relativa ad un unico contratto, un contratto può avere minimo una fattura.
- Ad ogni contratto corrisponde un contatore, ogni contatore fa riferimento ad un unico contratto.
- Ad ogni contratto può corrispondere un solo pannello solare, un pannello solare corrisponde ad un solo contratto.
- Ad ogni contratto corrispondono più consumi mensili, un consumo corrisponde ad un unico contratto.
- Ogni consumo mensile fa riferimento ad un solo oggetto di tipo EnergiaConsumata e può far riferimento ad un solo oggetto di tipo EnergiaProdotta, EnergiaProdotta ed EnergiaConsumata concorrono a formare un solo consumo mensile.
- Ogni componente della rete ha una posizione, ad ogni posizione corrisponde un solo componente della rete.
- Ogni cliente può aver segnalato più di un guasto, ogni segnalazione di guasto fa riferimento ad un solo cliente.
- Un manutentore può gestire più guasti, un guasto è gestito da un unico manutentore.

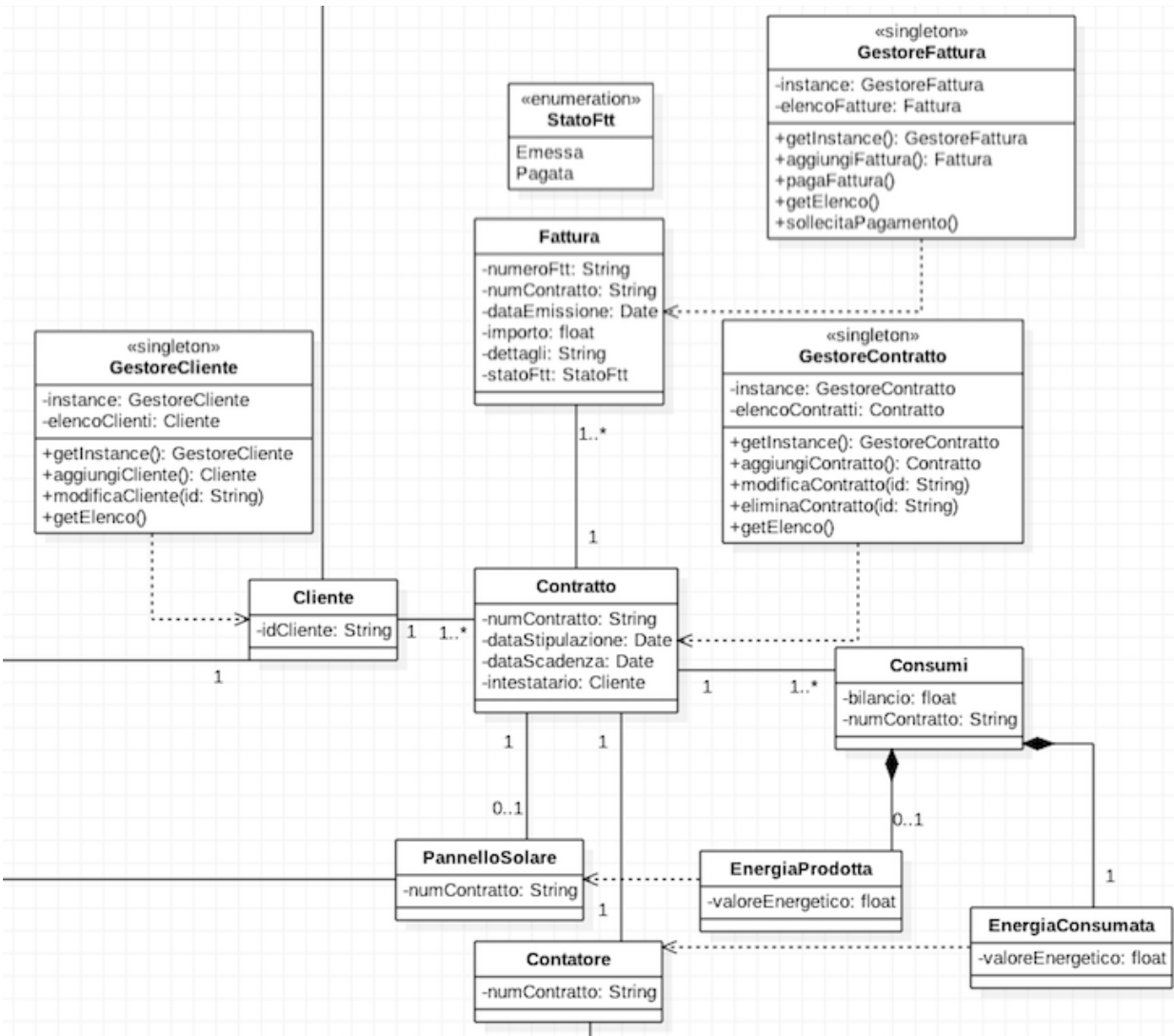
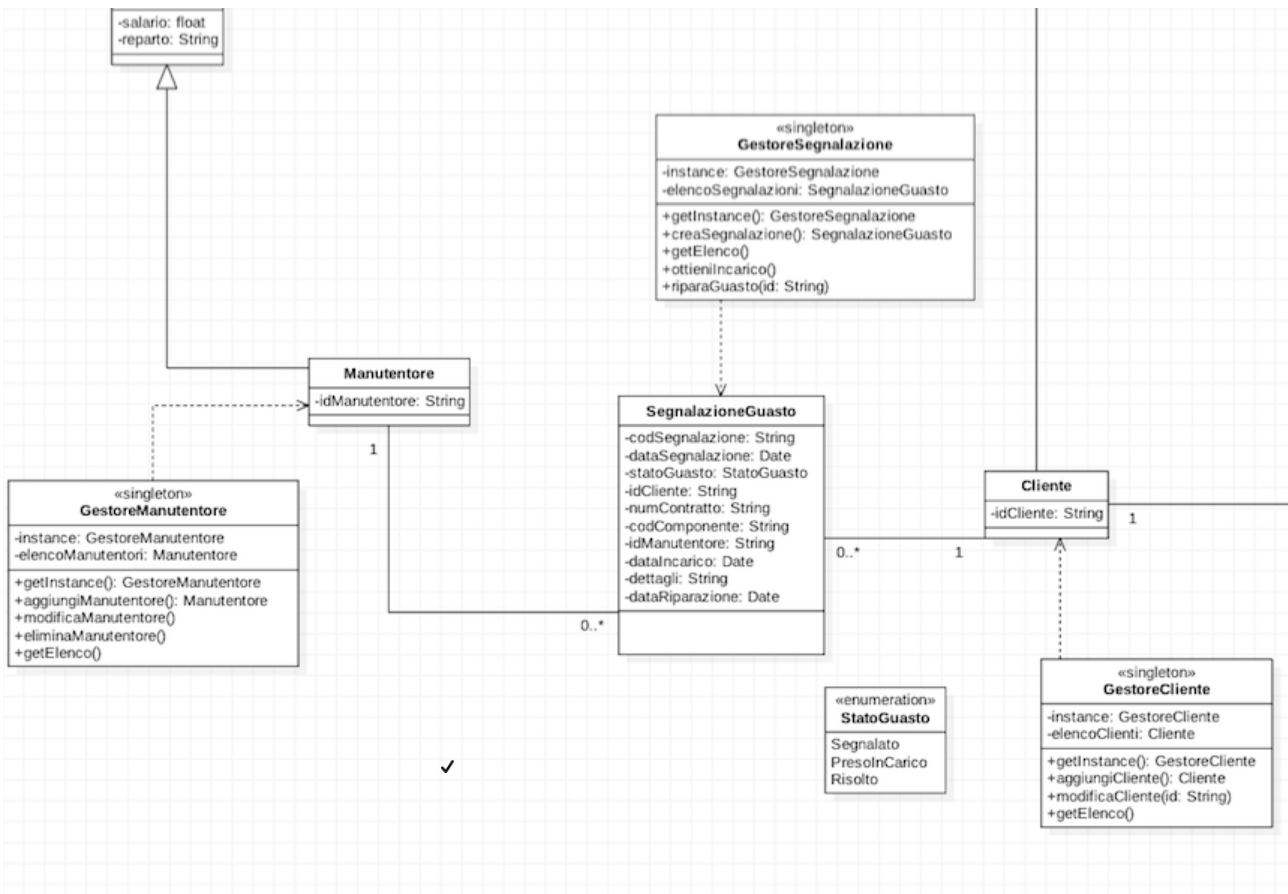
Sono presenti classi astratte, che non vengono mai istanziate, come Utente, Dipendente e ComponenteRete.

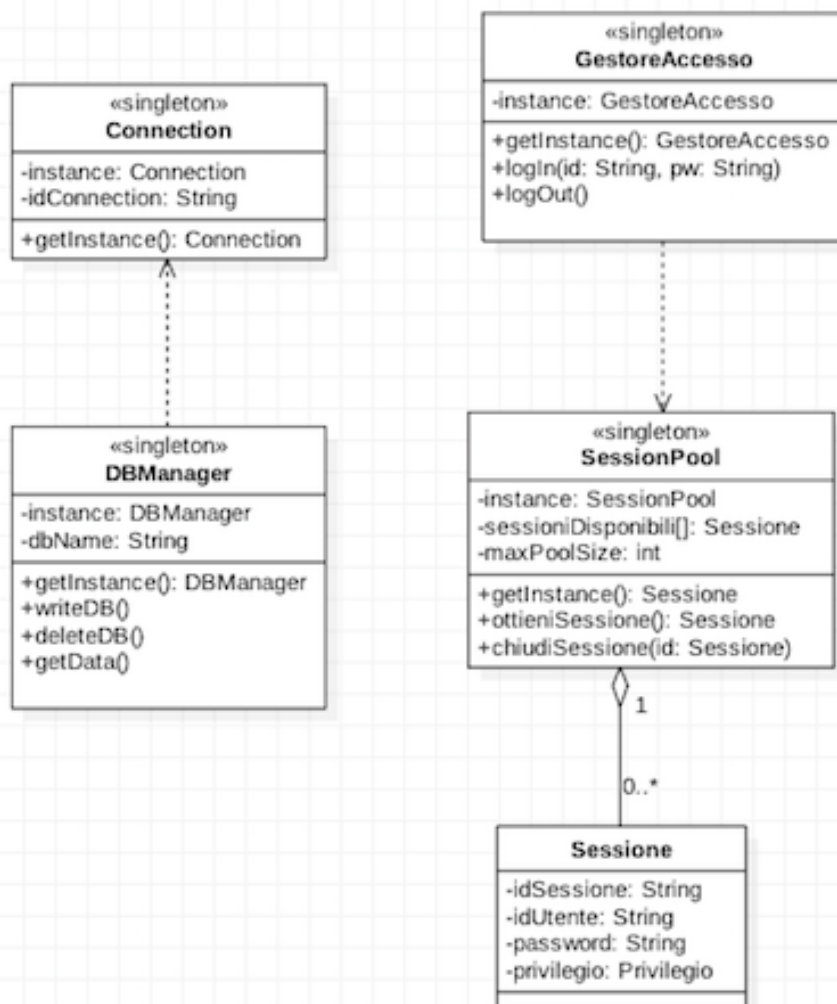
Le classi di tipo Gestore vengono istanziate una volta sola, queste permettono l'utilizzo e la gestione delle classi a cui fanno riferimento, tra cui anche l'interfaccia con il DB.

Questa avviene tramite un'unica Connessione, gestita dal DBManager che permette la consultazione del DB ogni qualvolta qualcuno voglia aggiornarlo o consultarlo.

Ogni classe contiene i metodi set e get, omissi la maggior parte delle volte se non per evidenziare i Singleton oppure metodi di consultazione DB. Ogni Gestore inoltre contiene gli array scaricabili in memoria dal DB.



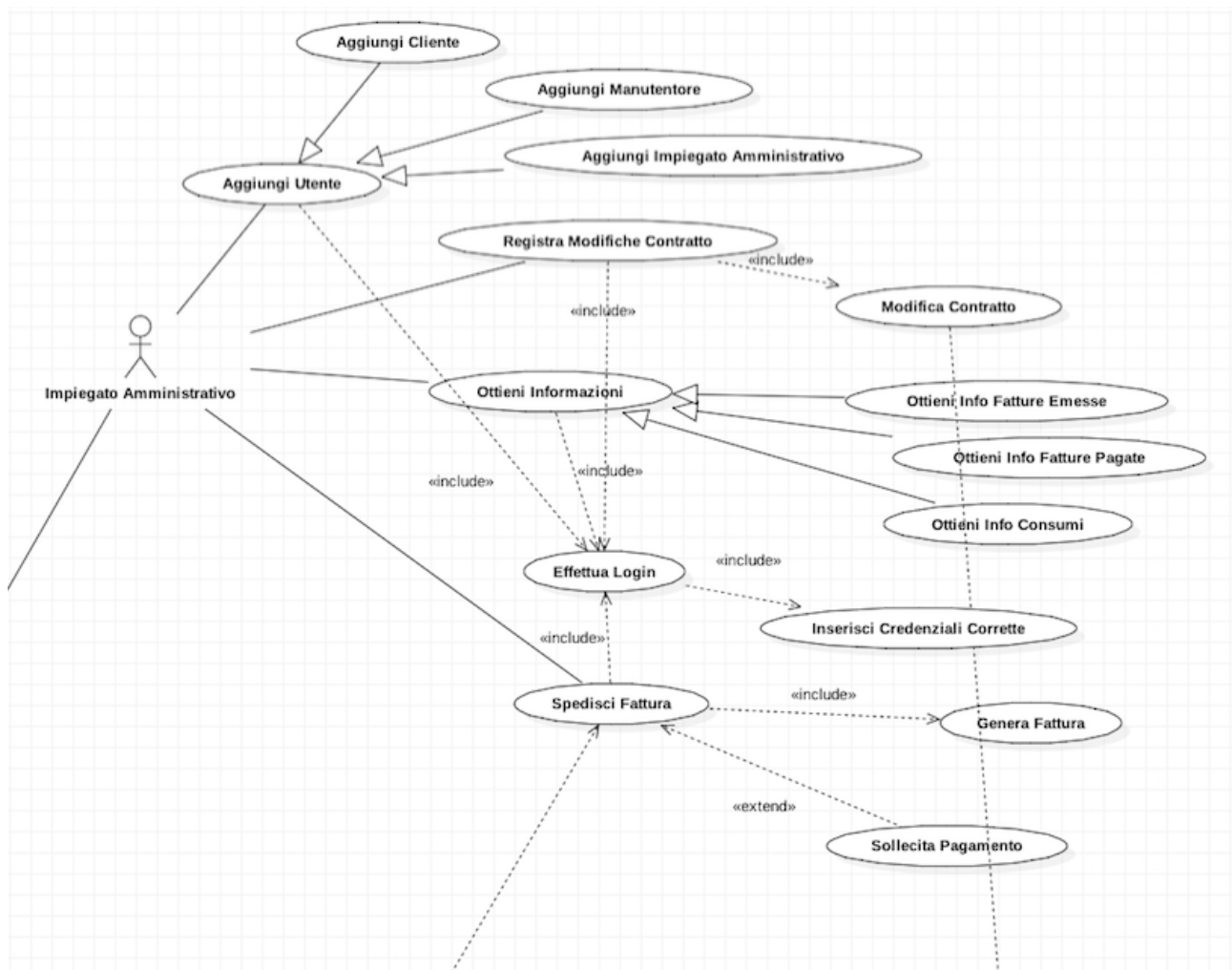




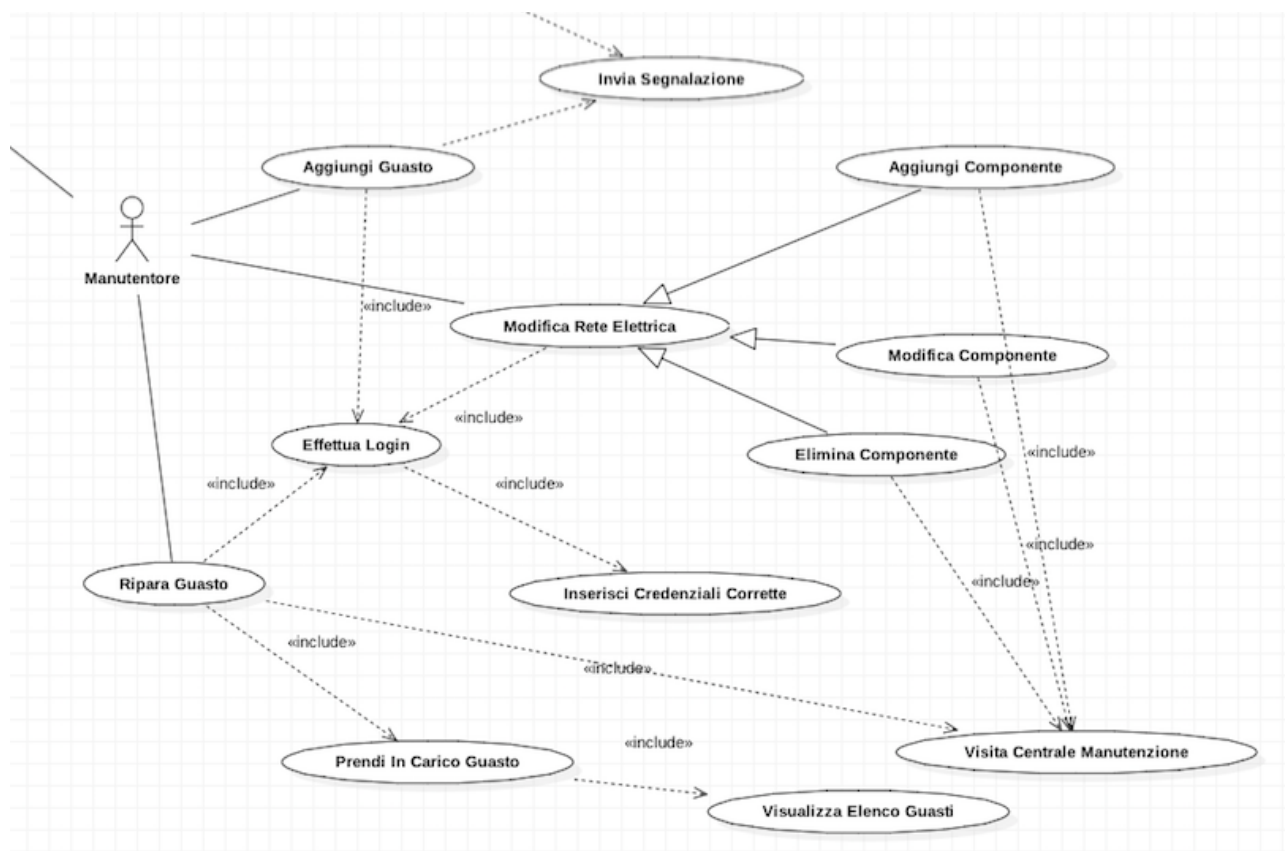
Use Case Diagram

Nello Use Case Diagram vengono evidenziati i casi d'uso, le funzioni e i servizi per come sono offerti e come vengono utilizzati dagli attori.

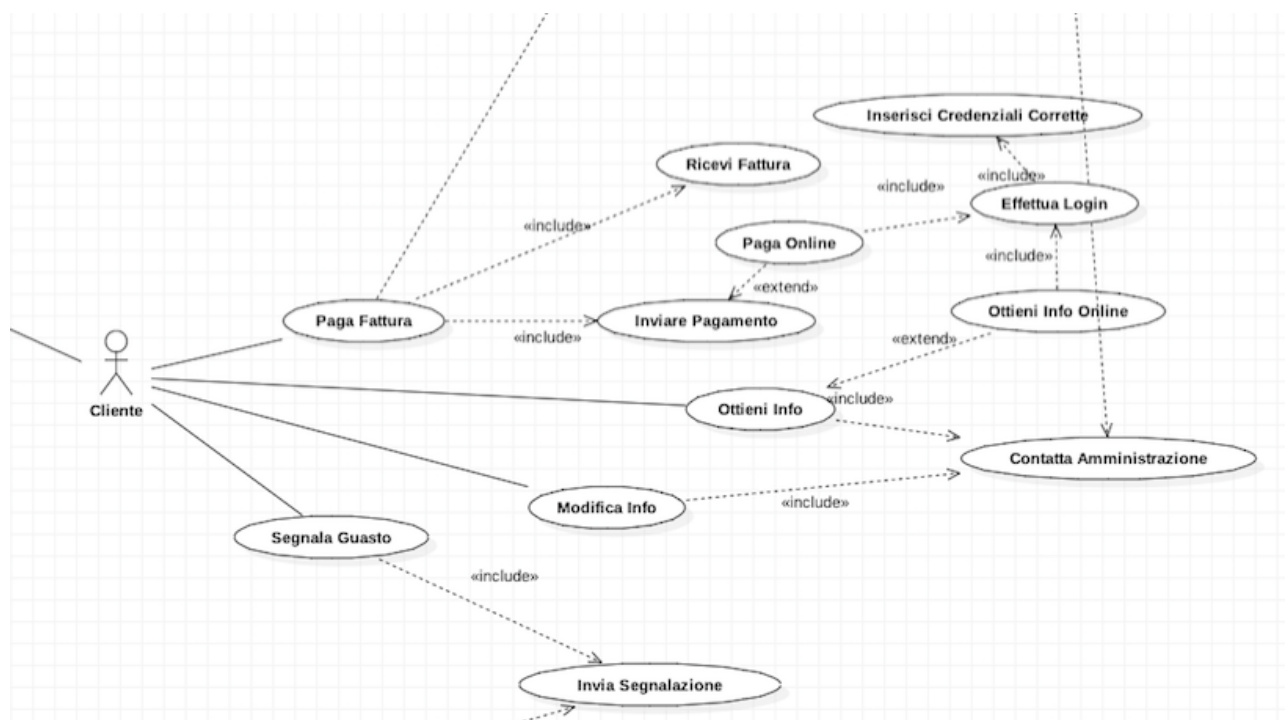
Nel nostro caso si possono notare le procedure con cui compie le proprie funzioni, tutte passando per il login nel sistema, come aggiungere nuovi utenti, modificare informazioni contrattuali, ottenere informazioni dal DB e inoltrare fatture.



Vengono portate all'attenzione le metodologie con cui un manutentore compie le proprie funzioni tra cui aggiungere un guasto al DB, riparare un guasto e modificare la rete elettrica.

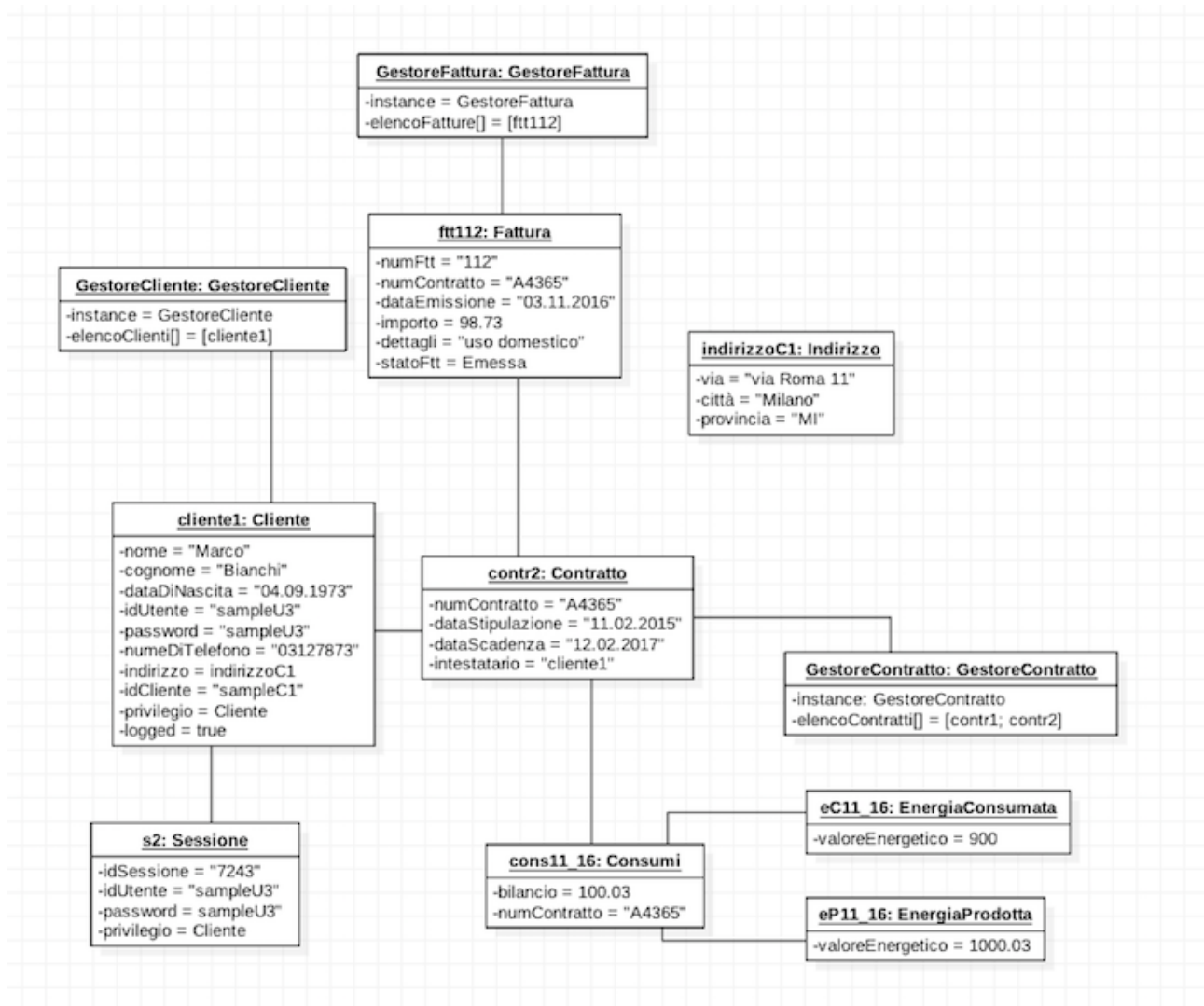


Per quanto riguarda il cliente, il diagramma mostra come egli possa segnalare un guasto, richiedere e/o modificare le proprie informazioni e pagare le fatture, alcune di queste possono essere compiute effettuando il login.

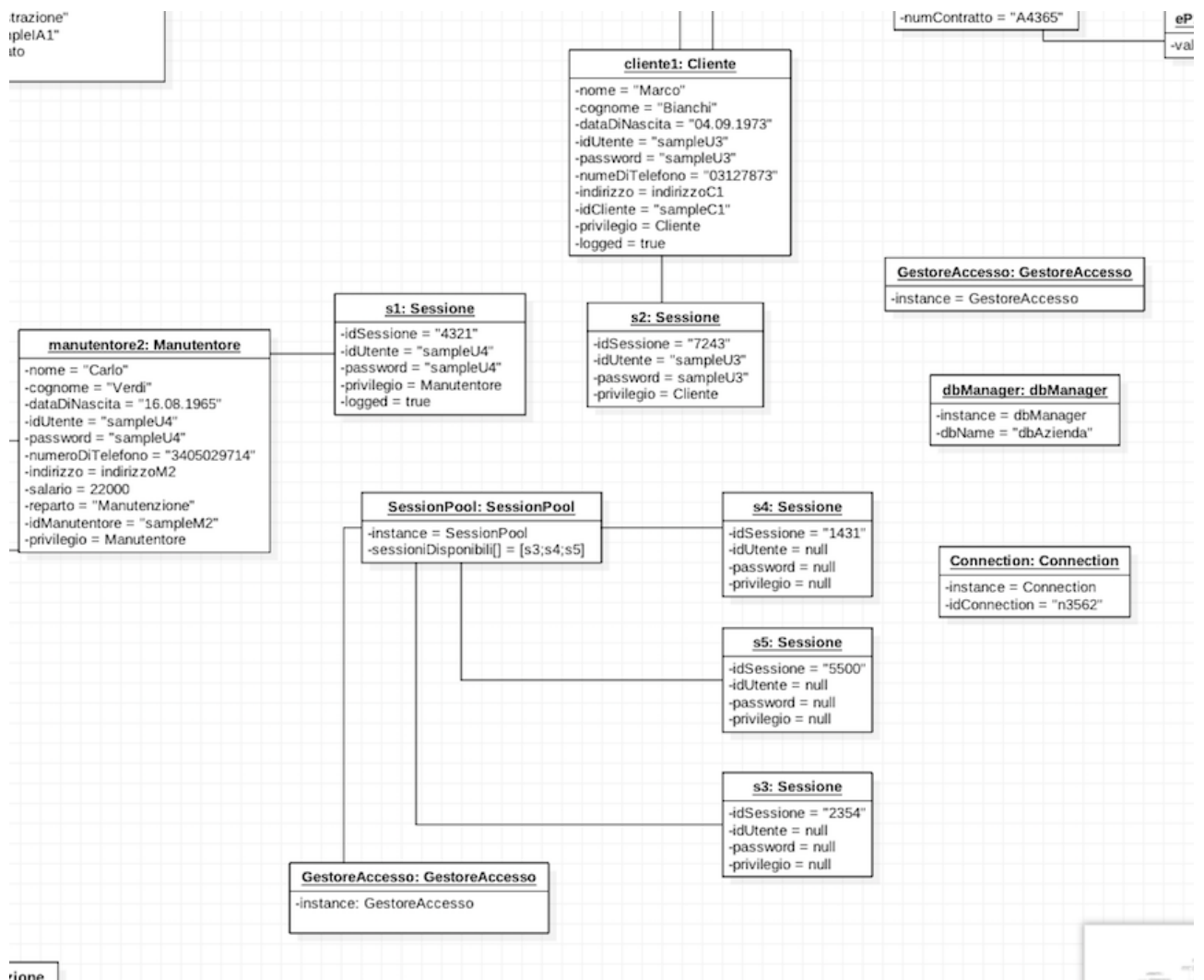


Object Diagram

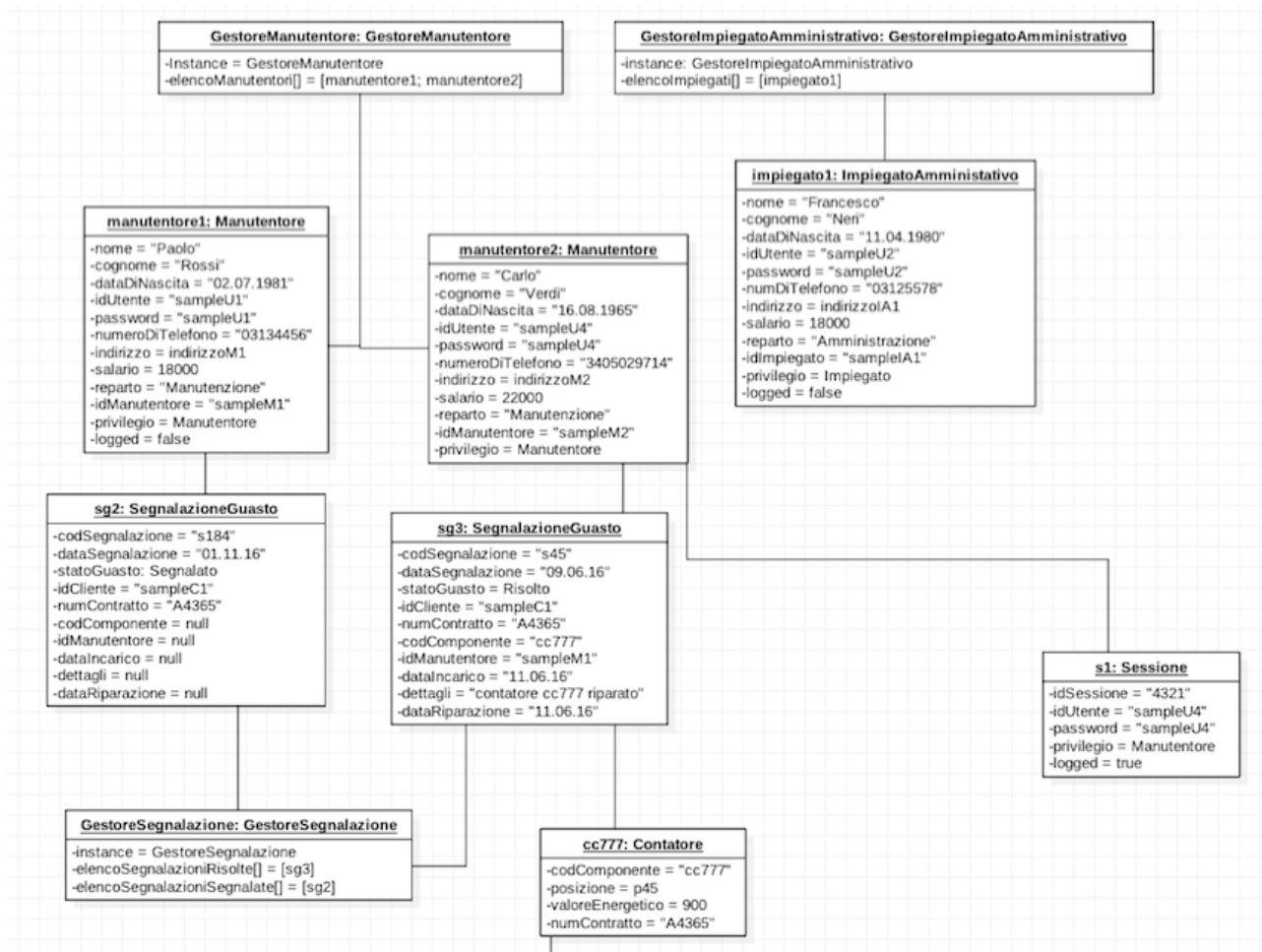
L'Object Diagram rappresenta una situazione statica una volta istanziate determinate classi. Di seguito saranno rappresentate diverse situazioni.



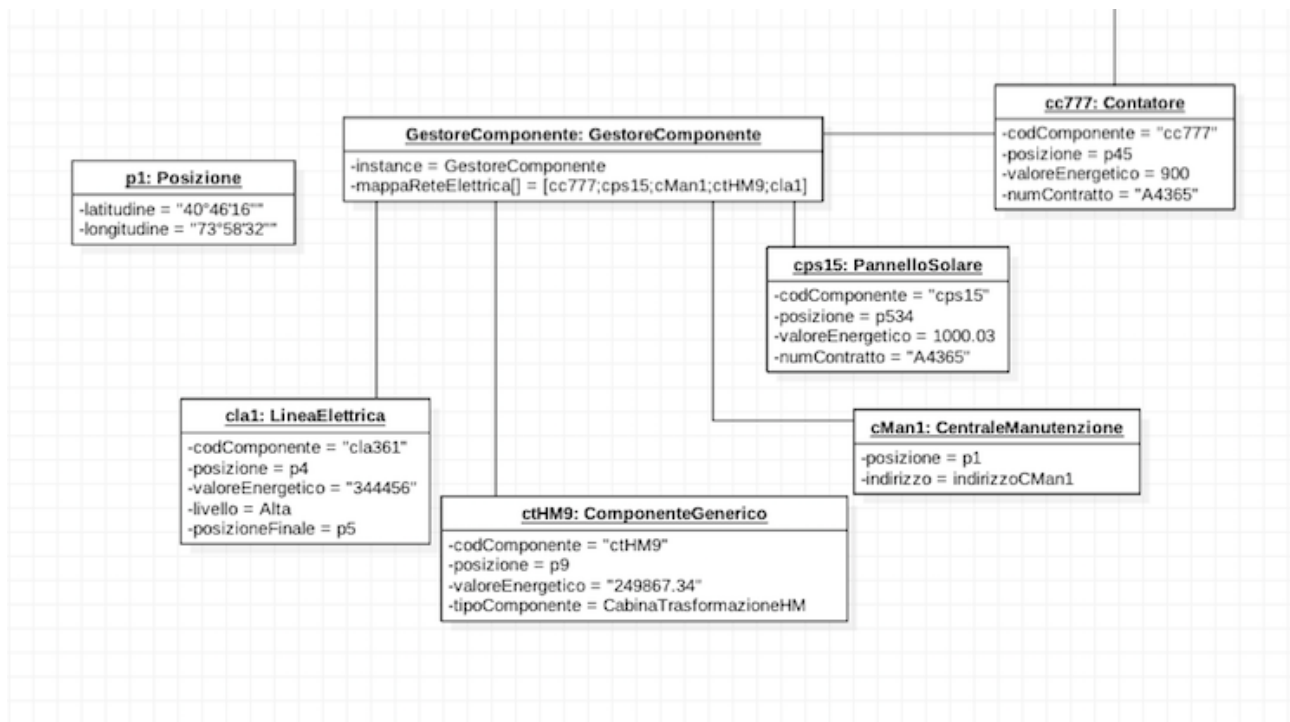
Sono rappresentate le istanze di una situazione in cui un cliente voglia visualizzare le sue fatture e i suoi contratti.



In questa figura è rappresentata la situazione in cui alcuni utenti stiano consultando informazioni online, è presente la pool di sessioni disponibili e si possono notare quelle occupate.



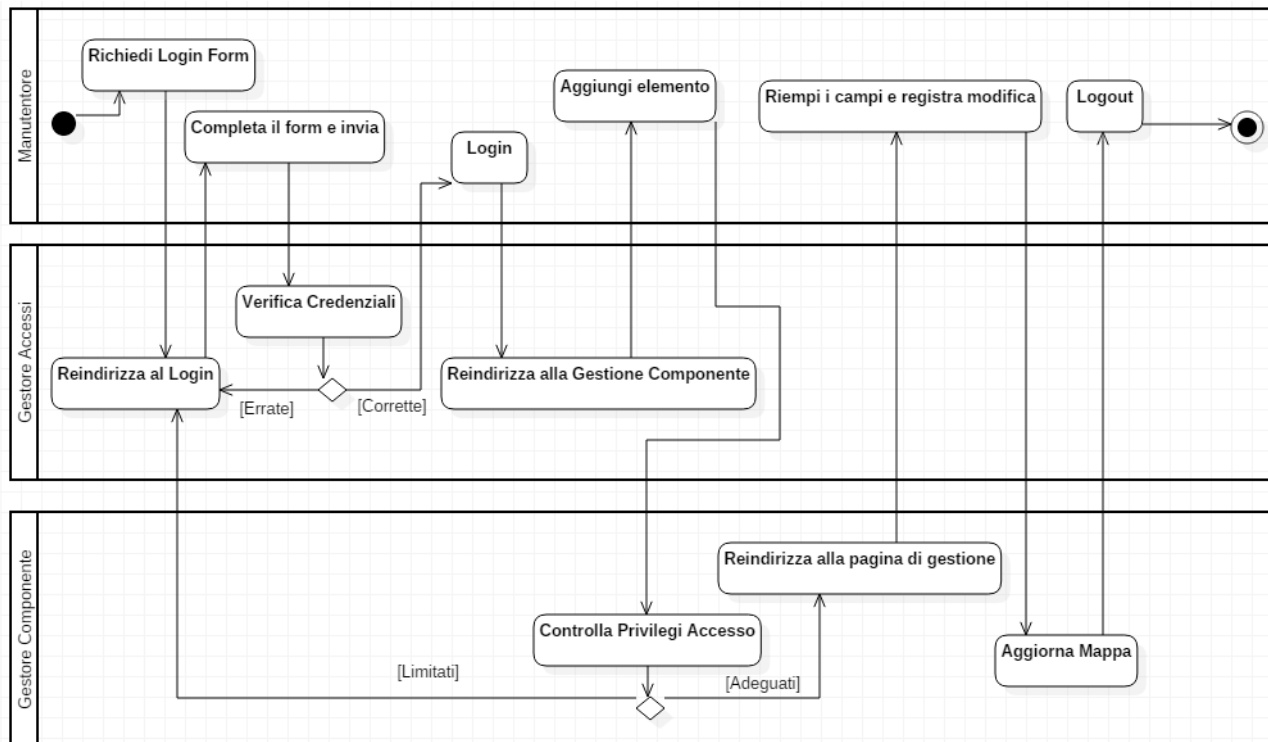
Vengono rappresentate le situazioni in cui un manutentore sta aggiungendo un guasto segnalato al DB e in cui un manutentore sta consultando l'elenco dei guasti risolti.



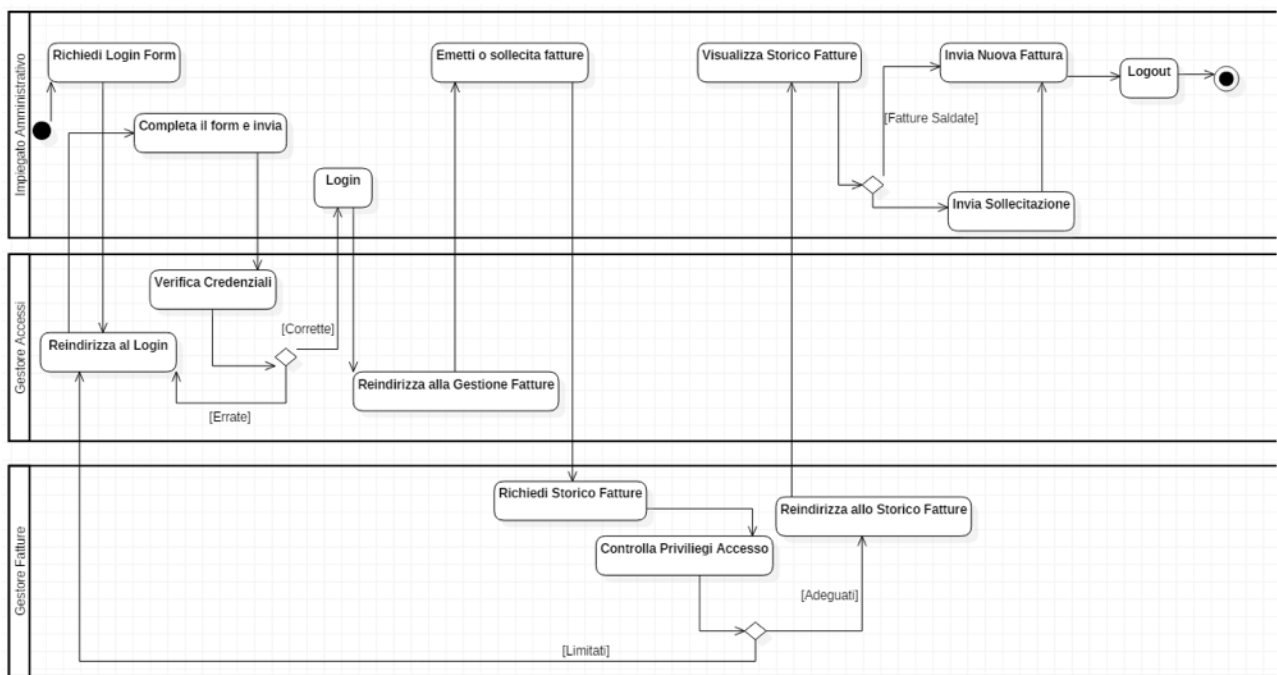
Rappresentazione della composizione della mappa della rete elettrica.

Activity Diagram

L'Activity Diagram mostra i passi (chiamati, propriamente, attività), i punti decisionali e i rami che intervengono nel flusso di un programma. E' utile per mostrare cosa accade in un'operazione ed è considerato come parte integrante dell'analisi di sistema.



In questo Activity Diagram abbiamo modellizzato la procedura di aggiunta di un componente effettuata dal manutentore. Dapprima il Manutentore “Richiede la Login Form”, il Gestore Accessi allora lo “Reindirizza al Login”, il Manutentore quindi “Completa il form e lo invia”; il controllo passa di nuovo al Gestore Accessi che “Verifica le Credenziali”: se queste sono errate “Reindirizza nuovamente al Login” altrimenti porta il Manutentore al “Login”. Il Gestore Accessi “Reindirizza alla Gestione Componente”, il Manutentore seleziona quindi “Aggiungi Elemento” e il controllo passa perciò al Gestore Componente il quale “Controlla i Privilegi Accesso”: se questi sono limitati il Gestore Accessi ti “Reindirizza al Login”, se questi sono invece adeguati il controllo rimane al Gestore Componente il quale “Reindirizza alla pagina di gestione”. Il Manutentore dunque “Riempie i campi e registra la modifica”, il Gestore Componente “Aggiorna Mappa” e il Manutentore effettua “Logout” e ciò ci porta all’attività finale.



In questo Activity Diagram abbiamo modellizzato la gestione delle fatture da parte dell'Impiegato Amministrativo. La procedura di Login è del tutto simile a quella dell'Activity "Aggiunta nuovo Componente" da parte del Manutentore e perciò verrà omessa tale parte. Dopo la procedura di Login il Gestore Accessi "Reindirizza alla Gestione Fatture", l'Impiegato Amministrativo "Emette o sollecita fatture"; ciò porta il controllo al Gestore Fatture che "Richiede lo storico Fatture" e "Controlla i Privilegi di Accesso": se sono limitati il Gestore Accessi "Reindirizza al Login", altrimenti il Gestore Fatture "Reindirizza allo Storico Fatture". L'Impiegato Amministrativo "Visualizza lo Storico Fatture" e, se questo sono tutte saldate "Invia una Nuova Fattura", altrimenti "Invia Sollecitazione" e poi "Invia una Nuova Fattura". L'impiegato poi si presuppone faccia "Logout" e quindi ciò ci porta all'Attività finale.

Sequence Diagram

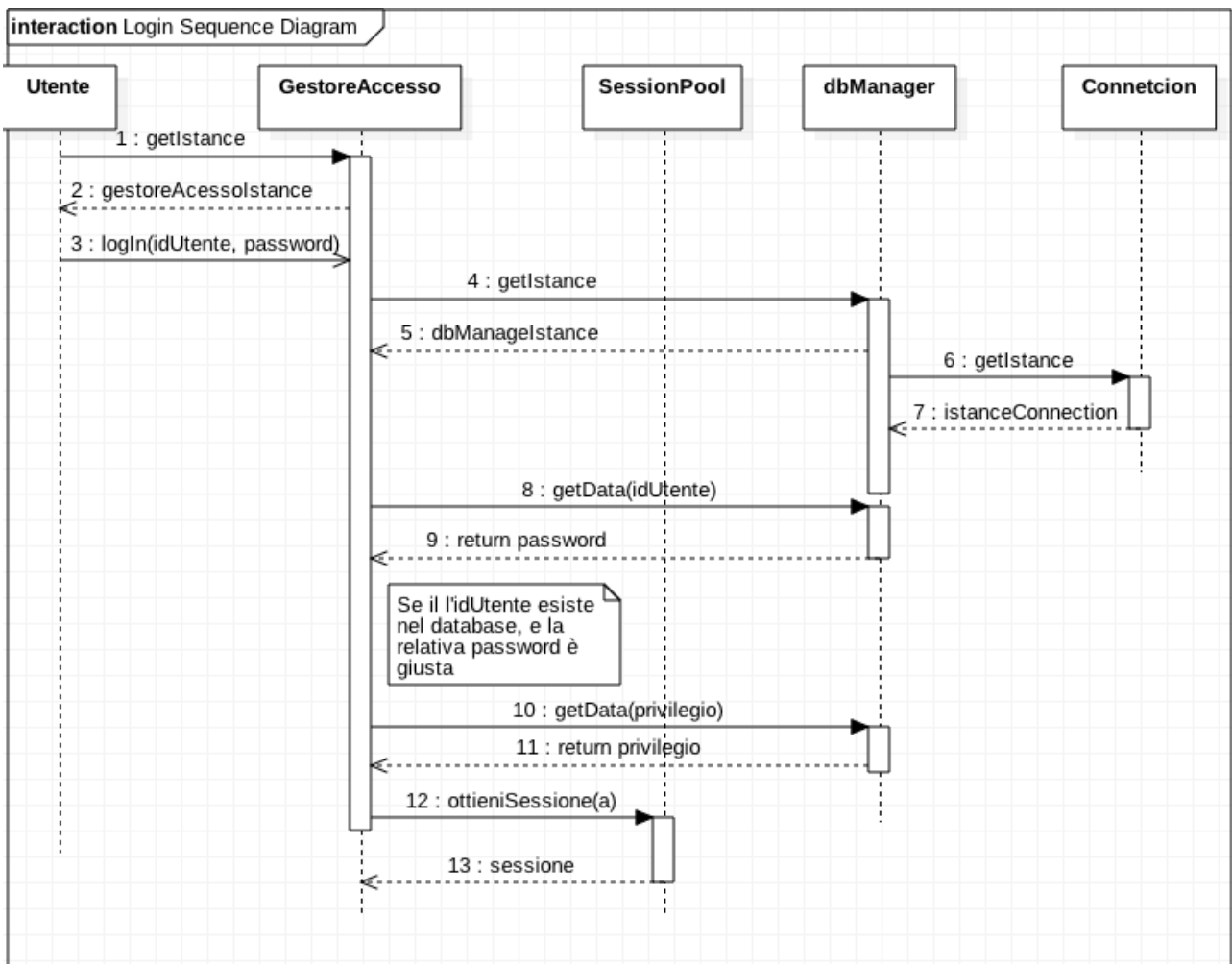
Il sequence diagram è un diagramma che permette di modellare la comunicazione tra un oggetto ed un altro in relazione al trascorrere del tempo.

Nel nostro caso abbiamo rappresentato il sequence diagram di un login di un utente generico, che può essere quindi sia un cliente, che un manutentore che un amministratore; e il sequence diagram di una richiesta di segnalazione da parte di un cliente.

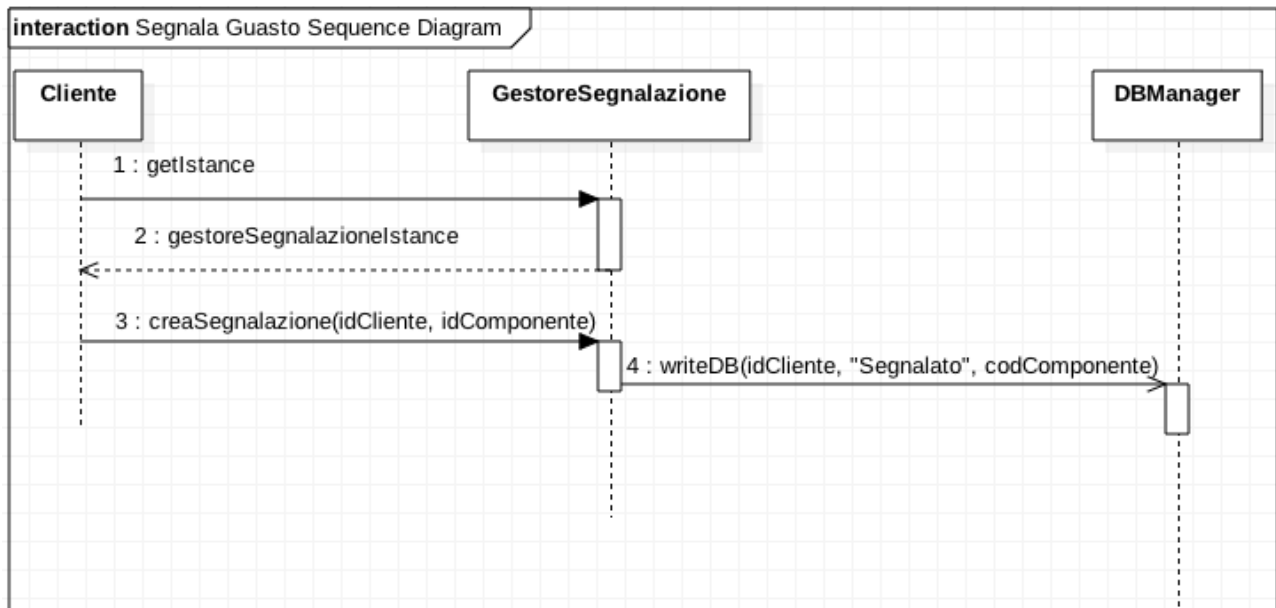
Per quanto riguarda il sequence del login da parte di utente, prima di tutto viene istanziato un'oggetto della classe GestoreAccesso, se non ancora fatto, dopodiché viene chiamato il metodo login passandogli i parametri idUtente e la sua relativa password.

A questo punto vengono eseguiti i metodi necessari per collegarsi al database tramite la classe Connection, e i metodi per leggere dal database tramite la classe DBManager.

Innanzitutto si richiede dal database la password del relativo idUtente, sempre se l'idUtente è presente nel database, una volta ottenuta la password, se è uguale a quella passata come parametro nel metodo login, allora procede alla lettura dal database che privilegi ha tale utente e il tutto viene salvato. Infine si procede a creare una sessione attraverso la classe SessionPool.

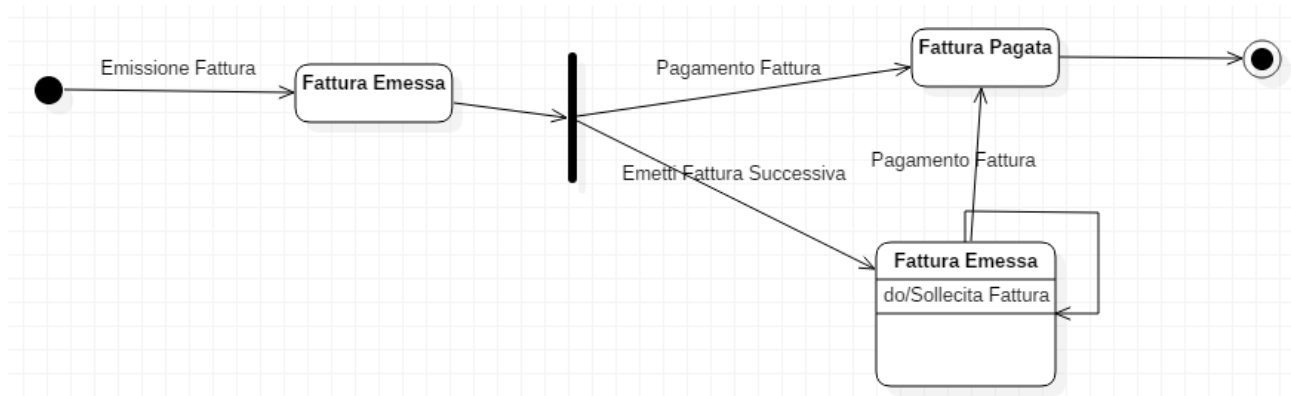


Il sequence diagram invece della richiesta di segnalazione da parte del cliente è molto semplice: viene istanziato un oggetto della classe GestoreSegnalazione nel caso non fosse ancora istanziato, dopodichè viene chiamato il metodo creaSegnalazione passandogli come parametri l'id del cliente e l'id del relativo componente dove crede che il cliente ci sia il guasto (in quanto un cliente puo' avere piu di un componente/contatore). Infine tale metodo accede al database e registra le informazioni riguardanti il guasto chiamando il metodo writeDB e passandogli come parametro il codiceSegnalazione, la data della segnalazione, la stringa "Segnalato", l'id del cliente, il numero del relativo contratto e il codice del componente.



Statechart Diagram

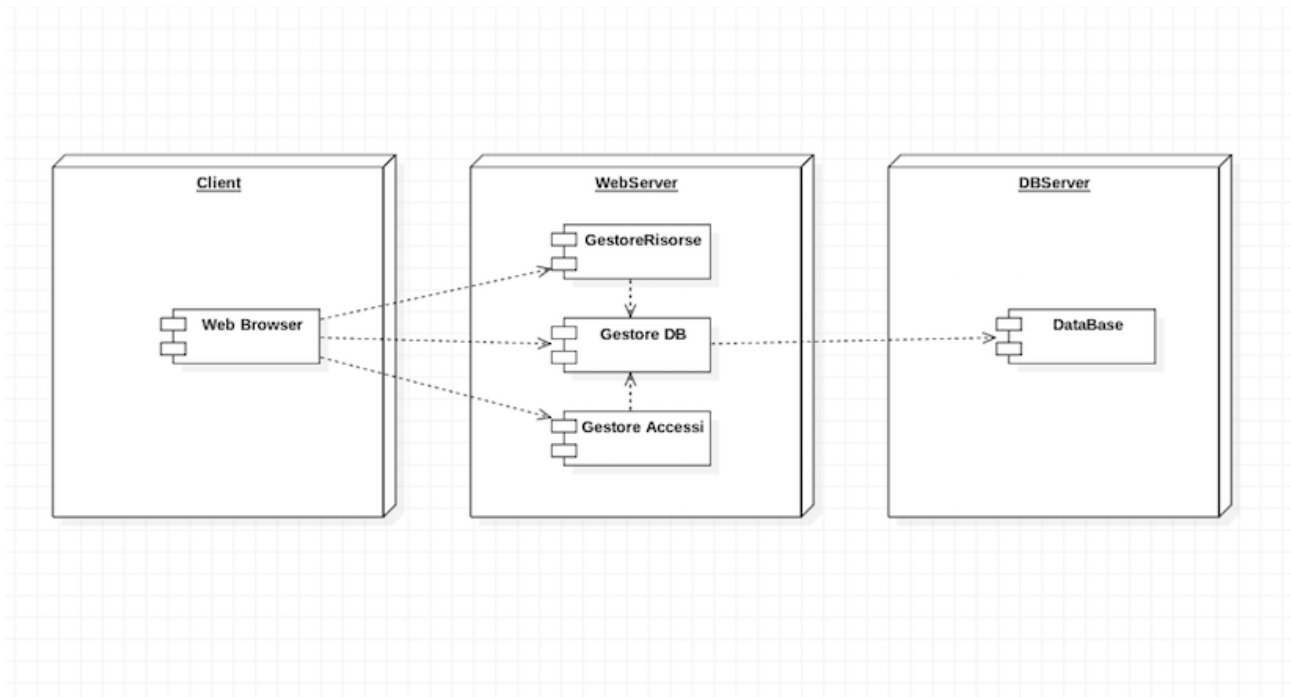
Lo State Diagram è un Diagramma di UML che rappresenta al meglio il cambiamento di stato di un singolo oggetto e le transizioni che portano a tale variazione di stato.



Il primo State Diagram che abbiamo rappresentato modella il cambiamento di stato di una generica fattura durante tutto il suo periodo di vita. Si comincia con l'emissione della fattura generata che parte con lo stato di "Fattura Emessa"; dopo di ciò si ha una fork, ossia un bivio (rappresentato in programmazione da un if-else) con due condizioni. La prima condizione "Pagamento Fattura" porta allo stato "Fattura Pagata" e allo stato finale, mentre la seconda (essendo passato un mese dall'emissione della fattura e non essendo ancora stata pagata) comporta l'emissione della fattura successiva che ci porta in un nuovo stato "Fattura Emessa" e l'azione "Sollecita Fattura". Questo ultimo stato riporta a se stesso in caso di nuovo mancato pagamento, mentre nel caso del pagamento porta nello stato "Fattura Pagata" e nel conseguente stato finale.

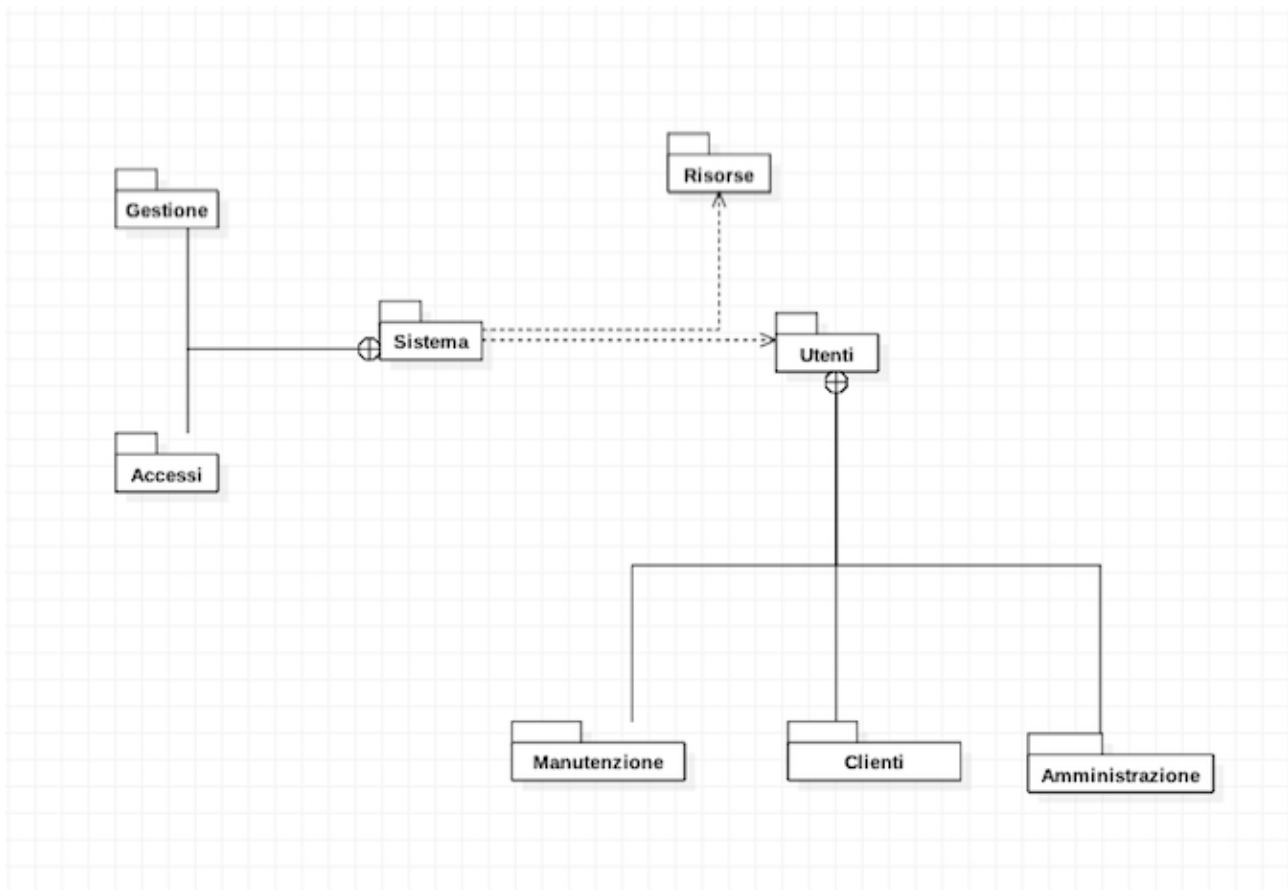
Deployment Diagram

Nel Deployment Diagram vengono evidenziati tramite i nodi i componenti hardware e le relazioni che intercorrono tra loro.



Package Diagram

All'interno del Package Diagram vengono raggruppati in un unico pacchetto elementi che hanno parti comuni al fine di chiarificare quali sono gli aspetti coperti dal programma.

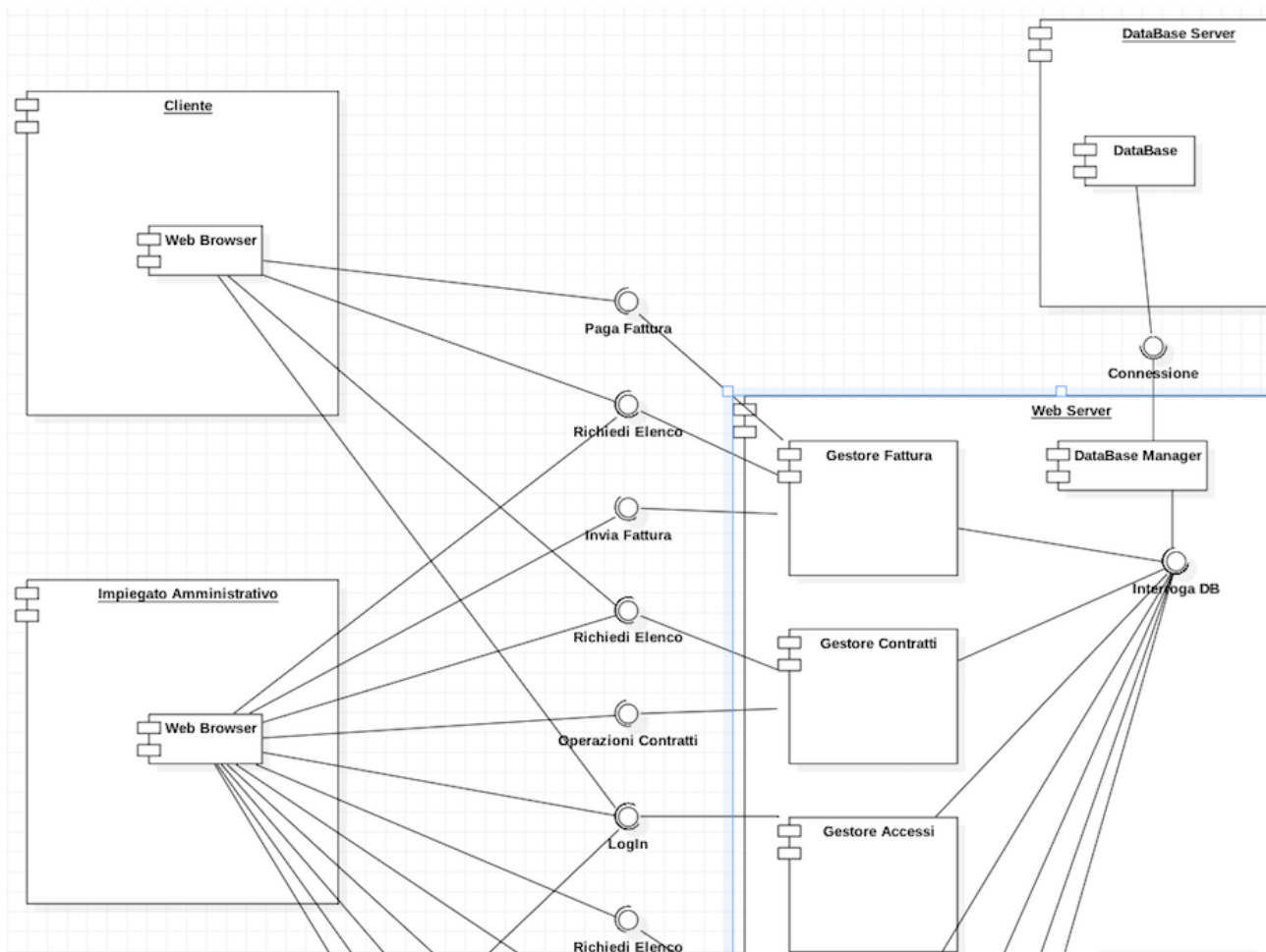


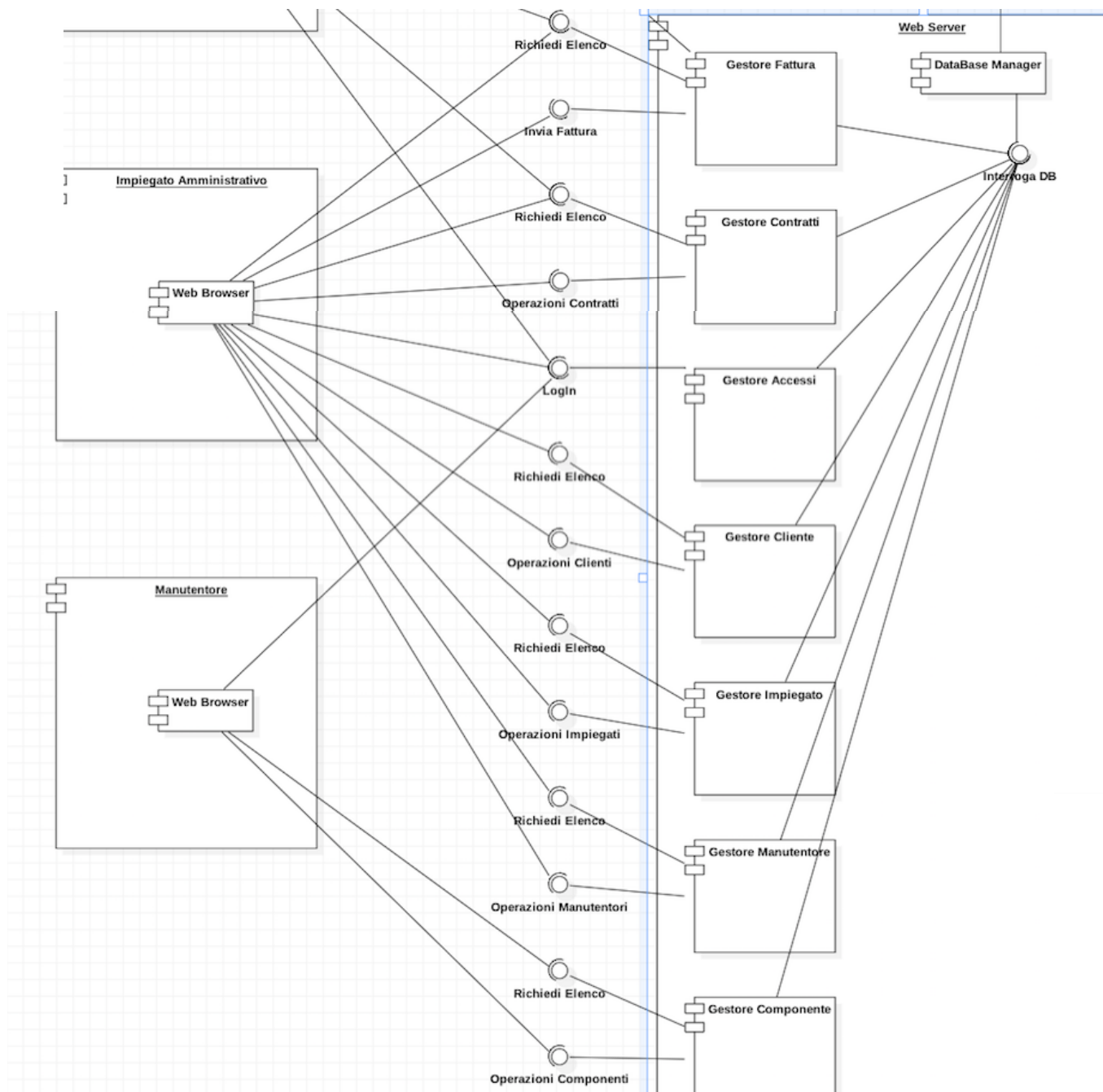
Component Diagram

All'interno del Component Diagram viene mostrata la composizione interna del software in termini di componenti.

Nel nostro diagramma si può notare come ogni client utente abbia accesso solo a determinate aree del web server, come si può notare il cliente può accedere solo alla gestione contratti, diversamente da impiegato amministrativo che ha più libertà.

E' importante sottolineare di come ogni informazione richiesta passa dal DB Manager, unico componente che si interfaccia al DB tramite una connessione.





Collaboration Diagram

Il collaboration diagram evidenzia come il sequence diagram le varie interazione tra i vari oggetti nel modello, solo che non vengono rappresentati nel tempo, ma più nello spazio.

Il collaboration diagram che abbiamo rappresentato riguarda la fase successiva a quella della richiesta di segnalazione, ovvero a quella di presa in carico della segnalazione e la riparazione del guasto da parte del manutentore. Questo avviene in modo molto semplice, la prima fase è la presa in carico, ove viene chiamato il metodo appunto `ottieniIncarico` il quale provvede ad ottenere l'elenco dei guasti con lo stato "Segnalato", e tramite il metodo `scegliGuasto` prende quello con la data più vecchia, memorizza i vari dettagli del guasto scelto, e scrive nel database del relativo guasto l'id del manutentore che l'ha preso in carico, gli passa la stringa "PresoInCarico", la data in cui ha preso l'incarico e vari dettagli se necessari. Dopodichè quando il manutentore ha risolto il guasto, chiama il metodo `riparaGuasto` passandogli l'id del guasto, e scrive nel database tramite il metodo `writeDB`, che lo stato del guasto è "Risolto" e aggiunge la data del giorno in cui è stato riparato il guasto.

