

Corso di Ingegneria del Software Deliverable di progetto	2021-2022
---	-----------

“Ingegneria del Software” 2023-2024

Docente: Prof. Angelo Furfaro

<Organigramma Aziendale >

Data	<5/11/2024>
Documento	Relazione progetto “Organigramma Aziendale”

Team Members		
Mattia Gravina	231188	Grvmitt03a13l049h@studenti.unical.it

Sommario

Un organigramma è uno schema grafico che rappresenta la struttura organizzativa di un'azienda o di un ente, mostrando le relazioni gerarchiche tra le diverse posizioni lavorative e reparti. Solitamente, gli organigrammi sono rappresentati attraverso forme geometriche, come rettangoli o cerchi, collegati da linee che indicano i rapporti di subordinazione o collaborazione tra i vari livelli. Questo strumento permette di visualizzare chiaramente la distribuzione delle responsabilità, i flussi di comunicazione e i ruoli di ogni membro all'interno dell'organizzazione.

È uno strumento utile sia per i dipendenti che per i manager poichè l'organigramma facilita la comprensione della struttura aziendale, promuove la chiarezza nei ruoli e nelle responsabilità e migliora la comunicazione interna, contribuendo all'efficacia operativa dell'organizzazione.

List of Challenging/Risky Requirements or Tasks

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Creazione di un modello per la gestione della logica dell'organigramma.	25/10	28/10	Impiego del pattern Composite al fine di creare un modello per la gestione di Unità, Impiegati e Ruoli definiti nell'organigramma.
Creazione di un'interfaccia grafica apposita finalizzata all'interazione utente-applicazione.	28/10	29/10	Utilizzo di librerie apposite al fine di creare una barra degli strumenti attraverso la quale l'utente sia in grado di modificare lo stato corrente dell'organigramma
Creazione di un modello di visualizzazione dinamico in grado di aggiornarsi al variare dello stato interno dell'organigramma	28/10	29/10	Impiego del pattern Observer al fine di monitorare eventuali cambiamenti dello stato e mostrarli a schermo
Creazione di un modello finalizzato alla persistenza dei dati, in grado di consentire salvataggio e caricamento dei dati da memoria secondaria	28/10	29/10	Ho utilizzato la Java Object Serialization al fine di consentire la memorizzazione dello stato interno corrente dell'organigramma.
Stesura di progetto	25/10	29/10	Realizzazione relazione inerente al progetto.

A. Stato dell'Arte

L'organigramma è un diagramma che visualizza la struttura di un'organizzazione, evidenziando gli enti che la compongono e le relazioni tra di essi. I rettangoli rappresentano gli organi aziendali, mentre le linee mostrano come questi si collegano. Esistono numerosi software progettati specificamente per la creazione di organigrammi aziendali.

Di seguito verranno esaminate le funzionalità più diffuse di queste applicazioni, con particolare attenzione alla visualizzazione della struttura organizzativa e alle operazioni fondamentali che una buona applicazione dovrebbe offrire.

1. Visualizzazione dell'organigramma: Le applicazioni moderne per organigrammi aziendali offrono varie opzioni grafiche. Tra queste ci sono diagrammi gerarchici, grafici a blocchi, alberi genealogici e organigrammi interattivi. Alcuni strumenti consentono anche di personalizzare l'aspetto dell'organigramma,

modificando stile, colori, icone e collegamenti ipertestuali.

2. Interfaccia user-friendly: Le applicazioni dedicate agli organigrammi puntano sempre più a un'esperienza utente intuitiva. L'interfaccia dovrebbe facilitare la navigazione e consentire interazioni dirette con il diagramma. Funzionalità come il drag-and-drop, il ridimensionamento degli elementi, l'aggiunta di collegamenti e l'organizzazione dinamica dei nodi sono ormai standard.

3. Collaborazione e condivisione: Le soluzioni moderne supportano la collaborazione in tempo reale. Le piattaforme cloud permettono a più utenti di modificare simultaneamente lo stesso organigramma, con funzionalità di annotazione e commento. Inoltre, è possibile condividere il diagramma tramite link, inviti o esportazioni in vari formati, come PDF o immagini.

4. Integrazione con altri sistemi: Le applicazioni possono integrarsi con altri software aziendali. Ad esempio, possono sincronizzarsi con i sistemi di gestione delle risorse umane (HRM) per importare automaticamente dati relativi a dipendenti, ruoli e struttura organizzativa. Anche l'integrazione con strumenti di produttività può facilitare la creazione e l'aggiornamento dell'organigramma.

5. Funzionalità avanzate: Alcuni software offrono opzioni avanzate, come la generazione automatica dell'organigramma da dati strutturati, la visualizzazione di diverse prospettive (per dipartimenti o posizioni) e l'analisi dei dati aziendali tramite grafici e statistiche. Queste applicazioni possono anche integrarsi con strumenti di business intelligence.

6. Accessibilità e mobilità: Un'applicazione moderna dovrebbe essere utilizzabile su diverse piattaforme, compresi smartphone e tablet. Le soluzioni web o le app mobili consentono agli utenti di accedere all'organigramma da qualsiasi luogo e in qualsiasi momento.

7. Sicurezza e privacy: È fondamentale che le applicazioni per organigrammi garantiscano la sicurezza e la riservatezza dei dati trattati.

B. Raffinamento dei Requisiti

L'organigramma fornisce una rappresentazione della struttura gerarchica dell'azienda, delineando le diverse unità organizzative e i relativi ruoli dei dipendenti.

Il principale obiettivo è trasformare le esigenze e le funzionalità espresse dagli utenti in requisiti specifici, che permettano una gestione e una manipolazione efficiente dell'organigramma aziendale.

Questo diagramma può essere visto come una struttura dati gerarchica, dove le unità organizzative sono rappresentate come nodi e le relazioni tra di esse sono illustrate tramite collegamenti.

È fondamentale avere una comprensione chiara delle operazioni necessarie, come la modifica dei nomi delle unità, l'aggiunta o la rimozione di unità e dipendenti, nonché l'assegnazione di ruoli ai vari impiegati.

È essenziale identificare e definire in modo dettagliato i requisiti funzionali, che includono operazioni specifiche che l'applicazione deve supportare, come l'aggiunta di sotto-unità, la rimozione di unità esistenti o l'assegnazione di dipendenti a determinate unità con ruoli specifici.

In aggiunta, è importante considerare i requisiti non funzionali da implementare nell'applicazione, come la sicurezza dei dati, la scalabilità, l'usabilità e l'efficienza operativa.

A.1 Servizi (con prioritizzazione)

L'applicazione offre una serie di servizi che permettono di interagire con l'applicazione al fine di creare un organigramma aziendale.

1. Modifiche allo stato dell'organigramma (alta priorità, media complessità): è necessario realizzare una struttura interconnessa tra le varie entità dell'organigramma in modo tale che la modifica di una si ripercuota sullo stato delle altre.

a. Modifica Nome Unità:

- i. L'utente deve essere in grado di selezionare un'unità esistente.
- ii. L'utente deve poter inserire un nuovo nome per l'unità selezionata.
- iii. L'applicazione deve salvare il nuovo nome dell'unità.

- b. Aggiungi Sotto Unità:
 - i. L'utente deve essere in grado di selezionare un'unità esistente a cui aggiungere una sotto unità.
 - ii. L'utente deve poter inserire i dettagli della nuova sotto unità, come il nome.
 - iii. L'applicazione deve creare una nuova sotto unità collegata all'unità selezionata.

- c. Rimuovi Unità:
 - i. L'utente deve essere in grado di selezionare un'unità esistente da rimuovere.
 - ii. L'applicazione deve verificare se l'unità selezionata ha sotto unità.
 - iii. Se l'unità ha sotto unità, l'applicazione rimuove anche le sue sottounità.
 - iv. Se l'unità non ha sotto unità, l'applicazione deve eliminare l'unità selezionata.

- d. Aggiungi Impiegato all'Unità:
 - i. L'utente deve poter inserire i dettagli del nuovo impiegato, come il nome e le informazioni personali.
 - ii. L'applicazione deve creare un nuovo impiegato all'interno dell'unità.
 - iii. L'utente deve poter selezionare un'unità esistente a cui assegnare l'impiegato.
 - iv. L'utente deve poter selezionare un ruolo consentito per l'unità selezionata.
 - v. L'applicazione deve assegnare l'impiegato all'unità con il ruolo selezionato.

- e. Rimuovi Impiegato dall'Unità:
 - i. L'utente deve poter selezionare un'unità esistente dalla quale rimuovere l'assegnamento di un impiegato.
 - ii. L'utente deve poter selezionare l'impiegato da rimuovere dall'unità.
 - iii. L'applicazione deve rimuovere l'assegnamento dell'impiegato dall'unità.

- f. Crea Ruolo:
 - g. L'utente deve poter inserire i dettagli del nuovo ruolo, come il nome e la descrizione.
 - ii. L'applicazione deve creare un nuovo ruolo ammissibile per quell'unità.
- i. Rimuovi Ruolo:
 - i. L'utente deve poter selezionare un ruolo esistente da eliminare.
 - ii. L'applicazione deve verificare se ci sono impiegati assegnati a quell'unità con il ruolo selezionato.
 - iii. Se ci sono impiegati assegnati solo a quel ruolo nell'unità, l'applicazione deve rimuovere anche gli impiegati poiché il ruolo per quell'unità non è più ammissibile
 - iv. L'applicazione deve eliminare il ruolo selezionato.
- h. Assegna Ruolo ad Unità:
 - i. L'utente deve poter selezionare un ruolo esistente.
 - ii. L'utente deve poter selezionare un'unità esistente a cui assegnare il ruolo.
 - iii. L'applicazione deve assegnare il ruolo all'unità selezionata.
- i. Rimuovi Ruolo da Unità:
 - i. L'utente deve poter selezionare un'unità esistente dalla quale cambiare l'assegnamento di un ruolo.
 - ii. L'utente deve poter selezionare il ruolo da rimuovere dall'unità.
 - iii. L'applicazione deve rimuovere l'assegnamento del ruolo dall'unità.

2. Visualizzazione grafica dello stato corrente dell'organigramma

- a. Visualizzazione dell'organigramma con rettangoli (media priorità, alta complessità): *è necessario realizzare una struttura grafica in grado di adattarsi all'inserimento di entità nell'organigramma, in modo tale che continui a risultare chiara e leggibile da parte dell'utente, occorre evitare che vi siano sovrapposizioni.*
 - i. Le unità sono rappresentate come rettangoli
 - ii. Le unità sono collegate alle relative sotto unità mediante segmenti.
 - iii. In questa visualizzazione vengono mostrate solo le unità, non gli impiegati. Vi è un bottone che mostra tutte le informazioni relative all'unità selezionata.
 - iv. La struttura grafica dell'organigramma si adatta dinamicamente all'inserimento e all'eliminazione delle unità.
- b. Rappresentazione degli impiegati e dei loro ruoli (alta priorità, bassa complessità): *se le funzionalità associate alla modifica dell'organigramma risultano ben implementate, non diviene complicato realizzare un meccanismo di visualizzazione per la visualizzazione di impiegati e ruoli.*
 - i. Un impiegato all'interno di un'unità viene visualizzato come figlio dell'unità nella gerarchia.
 - ii. Nelle informazioni dell'impiegato sono indicati i ruoli che egli ricopre all'interno dell'unità.

3. **Memorizzazione su memoria secondaria** (media priorità, alta complessità): è necessario gestire le dipendenze che intercorrono tra le varie entità facente parti della struttura interna dell'organigramma, al fine di garantire una corretta memorizzazione di esse, e di conseguenza un corretto ripristino degli stati precedenti.

- a. Salvataggio dell'organigramma:

- i. Si consente di salvare lo stato corrente dell'organigramma in un file in formato valido persistente su memoria secondaria.
 - ii. Il file salvato può successivamente essere ricaricato nell'applicazione.
 - b. Caricamento dell'organigramma:
 - i. Si consente di caricare un organigramma precedentemente salvato in un valido formato su memoria secondaria, precedentemente salvato dall'apposita funzione offerta dall'applicazione
 - c. Ripristino dell'organigramma allo stato iniziale:
 - i. Consente di azzera l'organigramma, ripristinandolo allo stato iniziale.
-
-
-

A.2 Requisiti non Funzionali

1. **Prestazioni:** L'applicazione deve gestire efficacemente grandi volumi di dati e garantire risposte rapide per gli utenti. È fondamentale che l'interfaccia utente sia reattiva e che l'organigramma possa essere aggiornato in tempo reale, anche in presenza di un numero elevato di dipendenti.
2. **Affidabilità:** La stabilità dell'applicazione è cruciale. Deve funzionare senza malfunzionamenti frequenti e gestire eventuali errori in modo elegante, consentendo il ripristino senza perdita di dati importanti.
3. **Scalabilità:** L'applicazione deve essere in grado di gestire un aumento nel numero di dipendenti e adattarsi a eventuali espansioni future senza compromettere le prestazioni. È importante che sia possibile aggiungere o rimuovere dipendenti senza influire sulla stabilità e sull'usabilità.
4. **Usabilità:** L'interfaccia deve essere intuitiva e accessibile, facilitando la navigazione nell'organigramma aziendale anche per utenti meno esperti. Gli utenti dovrebbero essere in grado di visualizzare informazioni dettagliate sui dipendenti e apportare modifiche con facilità.
5. **Manutenibilità:** Il codice dell'applicazione deve essere strutturato in modo chiaro e modulare, rendendo le modifiche, le correzioni e le estensioni semplici e senza effetti collaterali su altre parti del sistema.
6. **Portabilità:** L'applicazione deve funzionare su diverse piattaforme, come Windows, macOS e Linux, e non deve dipendere da specifiche configurazioni hardware o ambienti di esecuzione.

A.3 Scenari d'uso dettagliati

Di seguito vengono presentati i principali scenari d'uso per l'applicazione in questione. È importante notare che ulteriori casi d'uso relativi ad altre funzionalità possono essere facilmente adattati a questi esempi con piccole modifiche.

Scenario 1: Creazione di una nuova unità

Descrizione: Un utente ha la necessità di aggiungere una nuova unità all'interno del sistema di gestione aziendale. Accedendo alla piattaforma, seleziona l'opzione "Aggiungi Sotto Unità". L'utente è quindi invitato a fornire i dettagli dell'unità, come il suo nome. Dopo aver completato tutti i campi richiesti, conferma la creazione dell'unità. Il sistema controlla la validità delle informazioni e procede con la creazione della nuova unità aziendale con annessa notifica di avvenuta creazione in basso all'organigramma. Una volta completata l'operazione, l'utente può visualizzare l'organigramma aggiornato.

Scenario 2: Creazione di un nuovo ruolo ammissibile

Descrizione: Un utente intende creare un nuovo ruolo per definire le responsabilità e le autorizzazioni dei dipendenti all'interno di una determinata unità. Selezionando l'opzione "Crea Ruolo Ammissibile", l'utente deve inserire i dettagli necessari, come il nome del ruolo, la descrizione, i requisiti ed il salario. Dopo aver completato il modulo, conferma la creazione. Il sistema registra il nuovo ruolo e fornisce una conferma visiva dell'operazione completata tramite uno stato in basso all'organigramma. Viene quindi aggiornato l'elenco dei ruoli disponibili per l'unità selezionata, visualizzabile in "Visualizza info Unità" per includere quello appena creato.

Scenario 3: Creazione di un nuovo dipendente

Descrizione: Un utente desidera inserire un nuovo impiegato in un'unità con i ruoli pertinenti. Selezionando l'opzione "Aggiungi Dipendente ad Unità", l'utente è invitato a fornire i dettagli

dell'impiegato, come il nome, il cognome, l'indirizzo, l'età. Dopo aver completato tutti i campi necessari, conferma la creazione del profilo. Il sistema aggiunge il nuovo impiegato all'unità ed invia una conferma visiva in basso all'organigramma, aggiornando l'elenco degli impiegati di quell'unità per includere il nuovo inserimento.

Scenario 4: Assegnazione di un ruolo a un'unità

Descrizione: Un utente intende assegnare un ruolo a una specifica unità all'interno dell'organizzazione. Alla creazione del dipendente, l'utente accede a un elenco di ruoli disponibili per l'unità selezionata. Il sistema procede con l'assegnazione e fornisce una conferma visiva dell'operazione completata. L'elenco di dipendenti e dei ruoli per quella unità si trova in "Visualizza info Unità".

A.4 Excluded Requirements

Nella definizione dei requisiti per un'applicazione di organigramma, alcuni requisiti possono essere considerati escludibili a seconda delle specifiche dell'applicazione e delle esigenze degli utenti. Di seguito sono elencati i requisiti ritenuti escludibili per l'applicazione in esame:

1. **Collaborazione in tempo reale:** L'applicazione è progettata per un utilizzo individuale e non prevede la necessità di collaborazione simultanea tra più utenti. Pertanto, la funzionalità di collaborazione in tempo reale non è necessaria e può essere esclusa.
2. **Integrazione con altri software aziendali:** Poiché l'applicazione non è destinata a interagire con altri sistemi aziendali o software, non è necessario considerare requisiti specifici di integrazione con altre piattaforme.
3. **Funzionalità avanzate di formattazione:** L'applicazione si concentra principalmente sulla creazione e visualizzazione di organigrammi. Di conseguenza, funzionalità di formattazione avanzate, come stili di testo personalizzati, tabelle o colori personalizzati per le unità, non sono necessarie e possono essere escluse.
4. **Monitoraggio delle modifiche e tracciabilità:** L'applicazione non richiede un registro dettagliato delle modifiche apportate all'organigramma o la tracciabilità delle attività degli utenti. Pertanto, i requisiti relativi a questa funzionalità possono essere considerati escludibili.
5. **Funzionalità di esportazione in formati specifici:** Non è prevista l'esportazione dell'organigramma in formati specifici, come PowerPoint o Visio. I requisiti associati a tali formati possono quindi essere esclusi.
6. **Funzionalità di generazione automatica dell'organigramma:** L'applicazione si focalizza sulla creazione e modifica manuale dell'organigramma da parte degli utenti, rendendo superflua l'implementazione di funzionalità di generazione automatica basata su criteri specifici.
7. **Supporto per reti aziendali complesse:** L'applicazione è destinata a un uso individuale o a piccoli gruppi di utenti e non richiede supporto per reti aziendali complesse, come sistemi di autenticazione. Di conseguenza, i requisiti relativi a questo aspetto possono essere considerati escludibili.

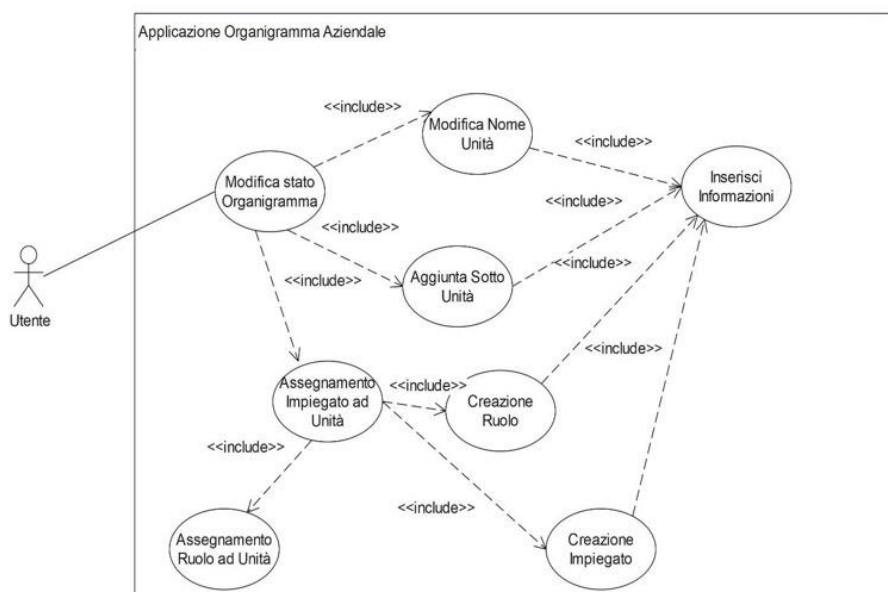
A.5 Assunzioni

Nel corso dello sviluppo dell'applicazione per l'organigramma aziendale, sono state fatte diverse scelte e assunzioni relative ai requisiti più rilevanti. Di seguito si presentano le principali decisioni adottate durante il progetto:

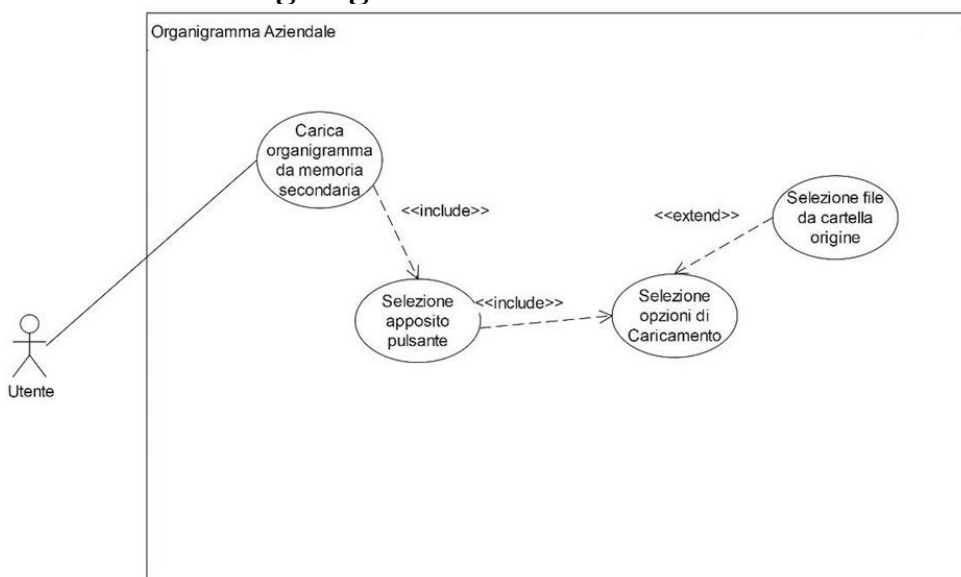
1. **Modalità di rappresentazione:** Una delle scelte fondamentali ha riguardato come visualizzare l'organigramma nell'applicazione. Dopo un'attenta valutazione, è stato deciso di utilizzare una rappresentazione grafica caratterizzata da rettangoli e linee di collegamento tra le unità. Questo metodo visivo permette di comprendere facilmente la gerarchia e le relazioni esistenti tra le diverse sezioni dell'azienda.
2. **Gestione delle unità organizzative:** È stata presa la decisione di permettere agli utenti di aggiungere, modificare e rimuovere unità all'interno dell'organigramma. Inoltre, si è stabilito di consentire l'aggiunta di sotto-unità per creare una gerarchia multilivello. Questa scelta mira a garantire flessibilità e a consentire una rappresentazione più accurata della struttura organizzativa.
3. **Gestione del personale:** È stato deciso di dare la possibilità agli utenti di aggiungere e rimuovere dipendenti, nonché di assegnarli alle varie unità. È stata inoltre prevista l'associazione di ruoli specifici a ciascun impiegato all'interno delle unità. Questo approccio consente una gestione più dettagliata delle responsabilità e dei ruoli all'interno dell'organigramma.
4. **Interfaccia e barra degli strumenti:** È stata decisa l'integrazione di una barra degli strumenti contenente vari comandi per semplificare l'interazione con l'applicazione. Tra le funzionalità incluse vi sono la modifica dei nomi delle unità, l'aggiunta di sotto-unità, la rimozione di unità, e l'assegnazione di impiegati e ruoli. Questa decisione mira a creare un'interfaccia utente chiara e intuitiva, facilitando la gestione dell'organigramma.
5. **Funzionalità di salvataggio e caricamento:** È stata prevista la possibilità di salvare l'organigramma attuale e di caricare versioni precedentemente salvate attraverso la serializzazione. Questo permette agli utenti di mantenere le proprie configurazioni personalizzate e di ripristinare facilmente versioni passate dell'organigramma.
6. **Regole per la rimozione delle unità:** È stato stabilito che le unità possono essere rimosse arbitrariamente, per dare la completa gestione dell'organigramma all'utente.

A.6 Use Case Diagrams

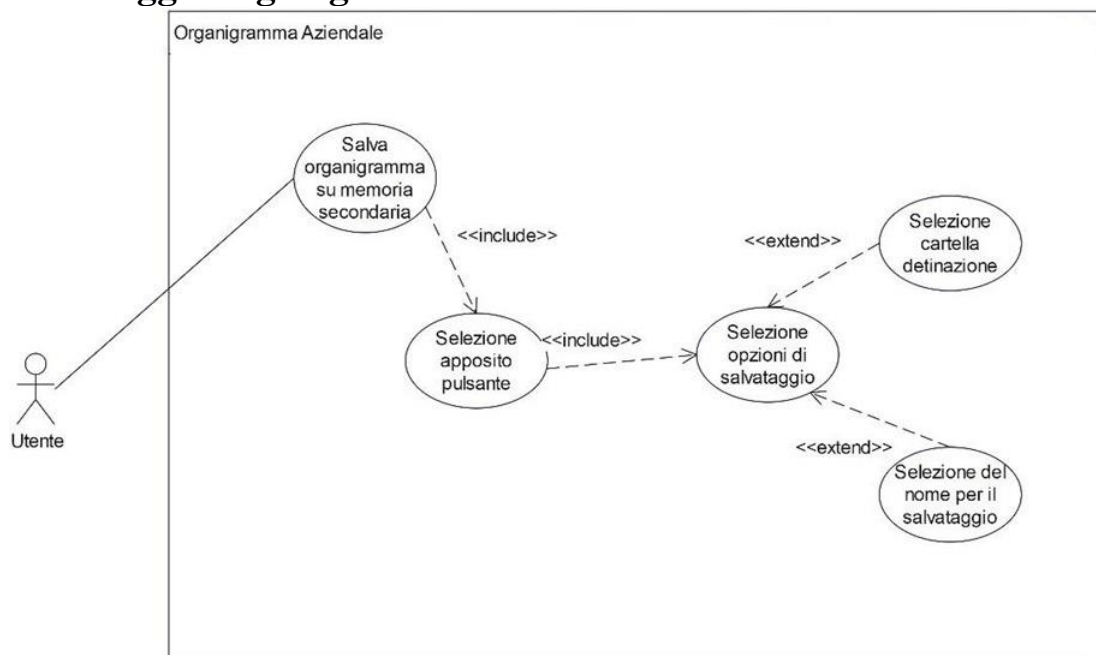
Modifica stato organigramma



Caricamento organigramma da memoria secondaria



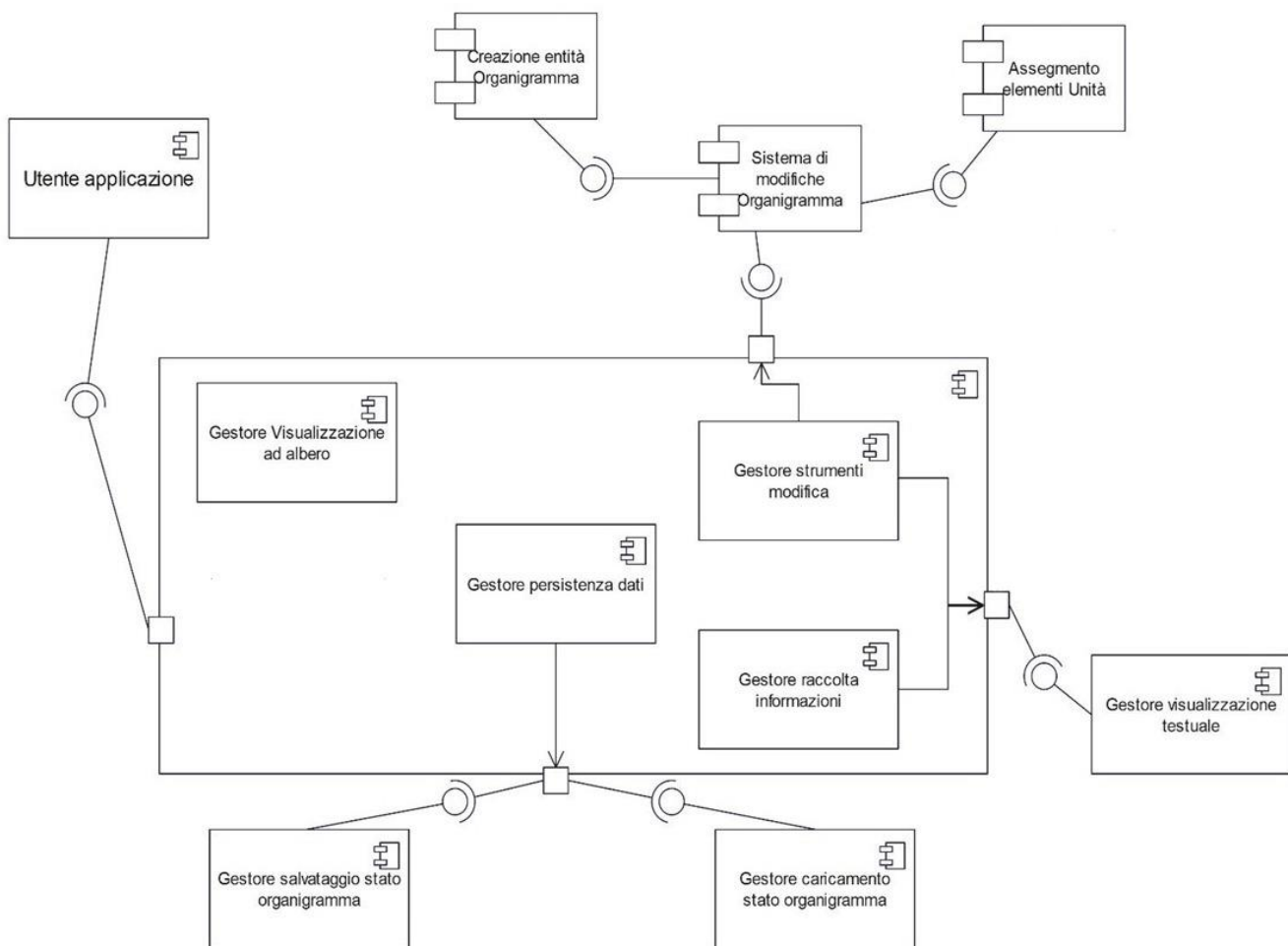
Salvataggio organigramma su memoria secondaria



C. Architettura Software

C.1 The static view of the system: Component Diagram

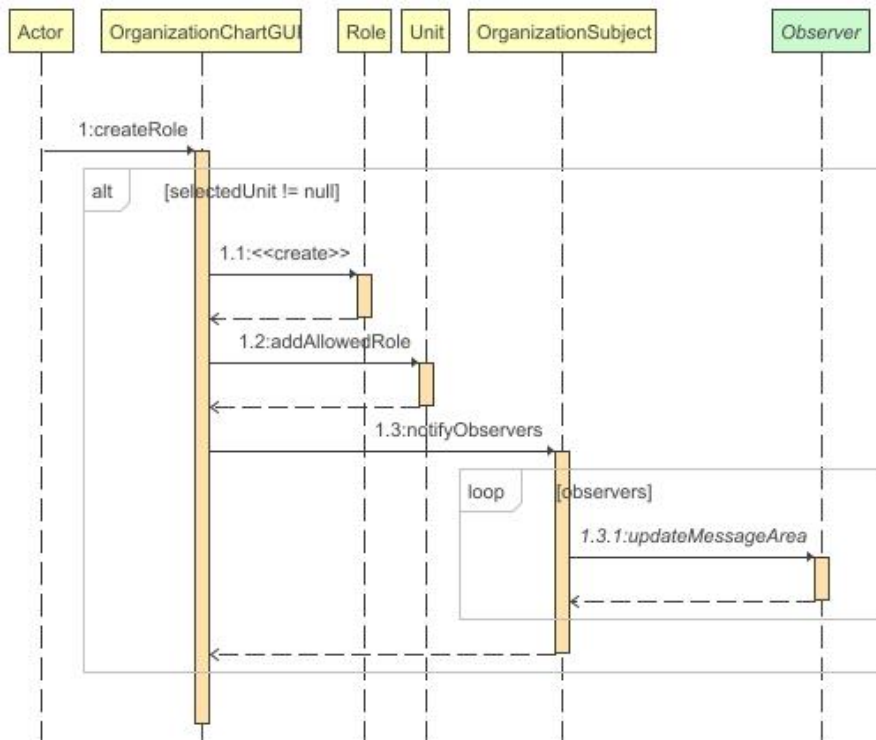
Visione statica della struttura dell'applicazione



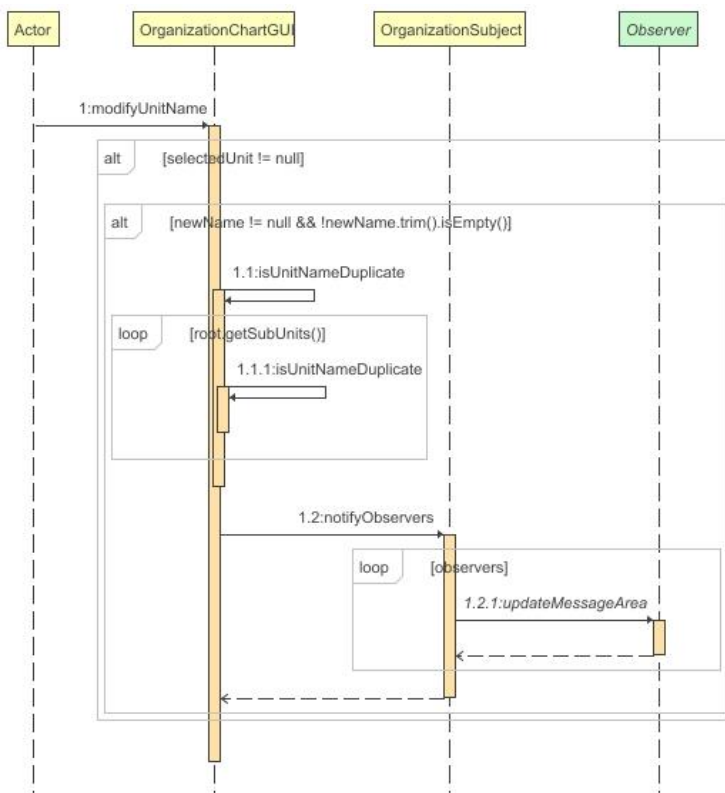
C.2 The dynamic view of the software architecture: Sequence Diagram

Visione dinamica dell'applicazione

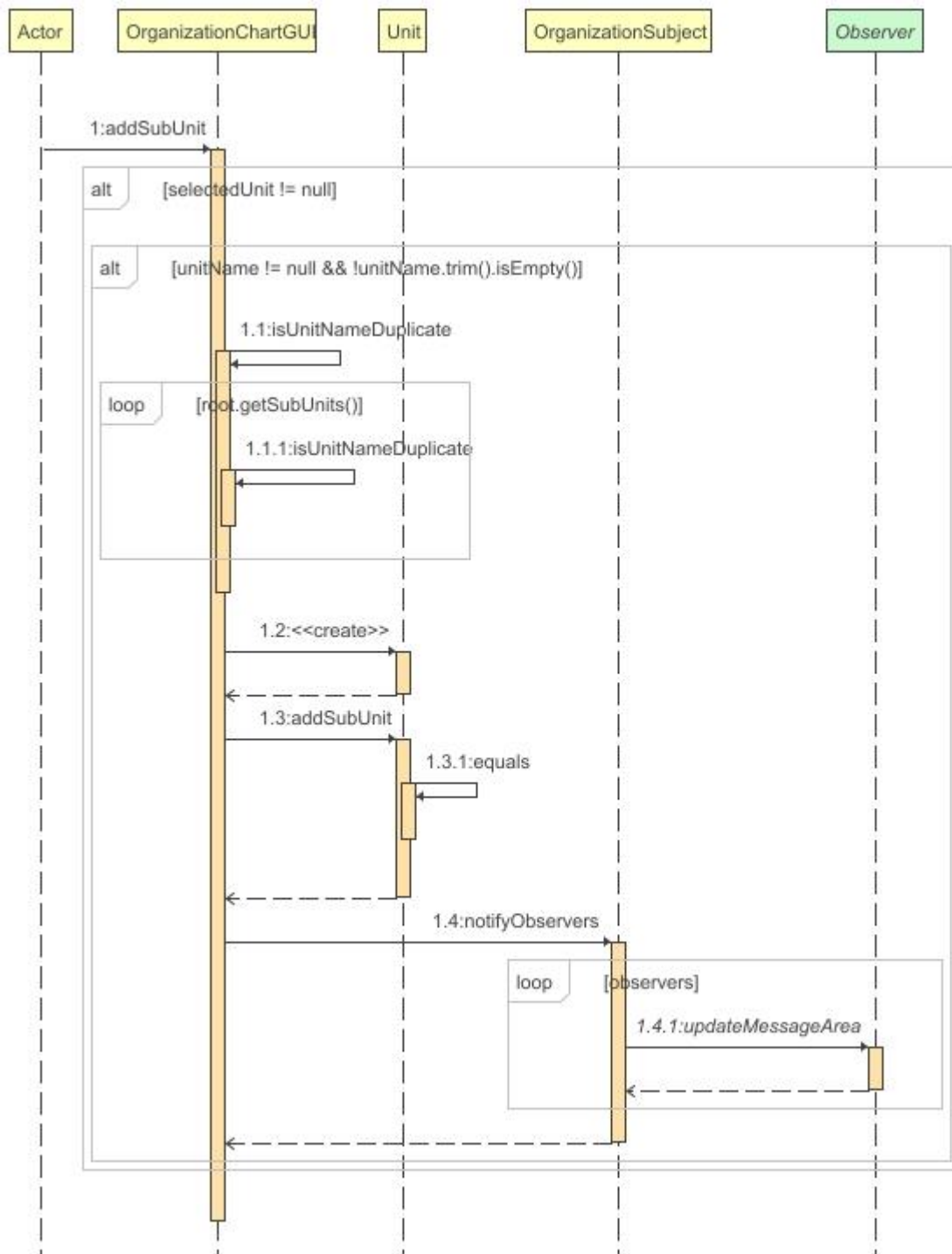
Creazione di un ruolo



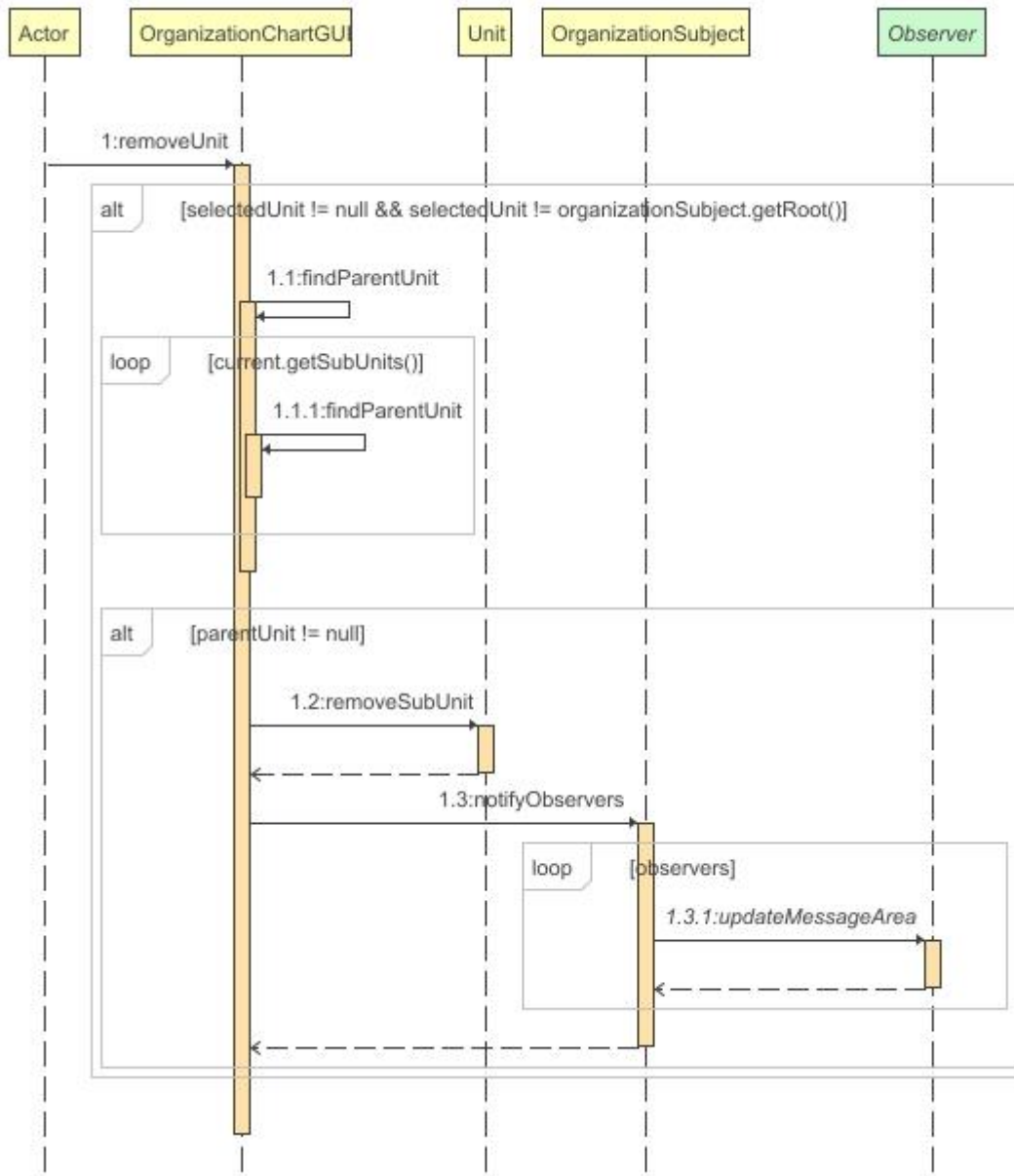
Modifica nome di una unità esistente



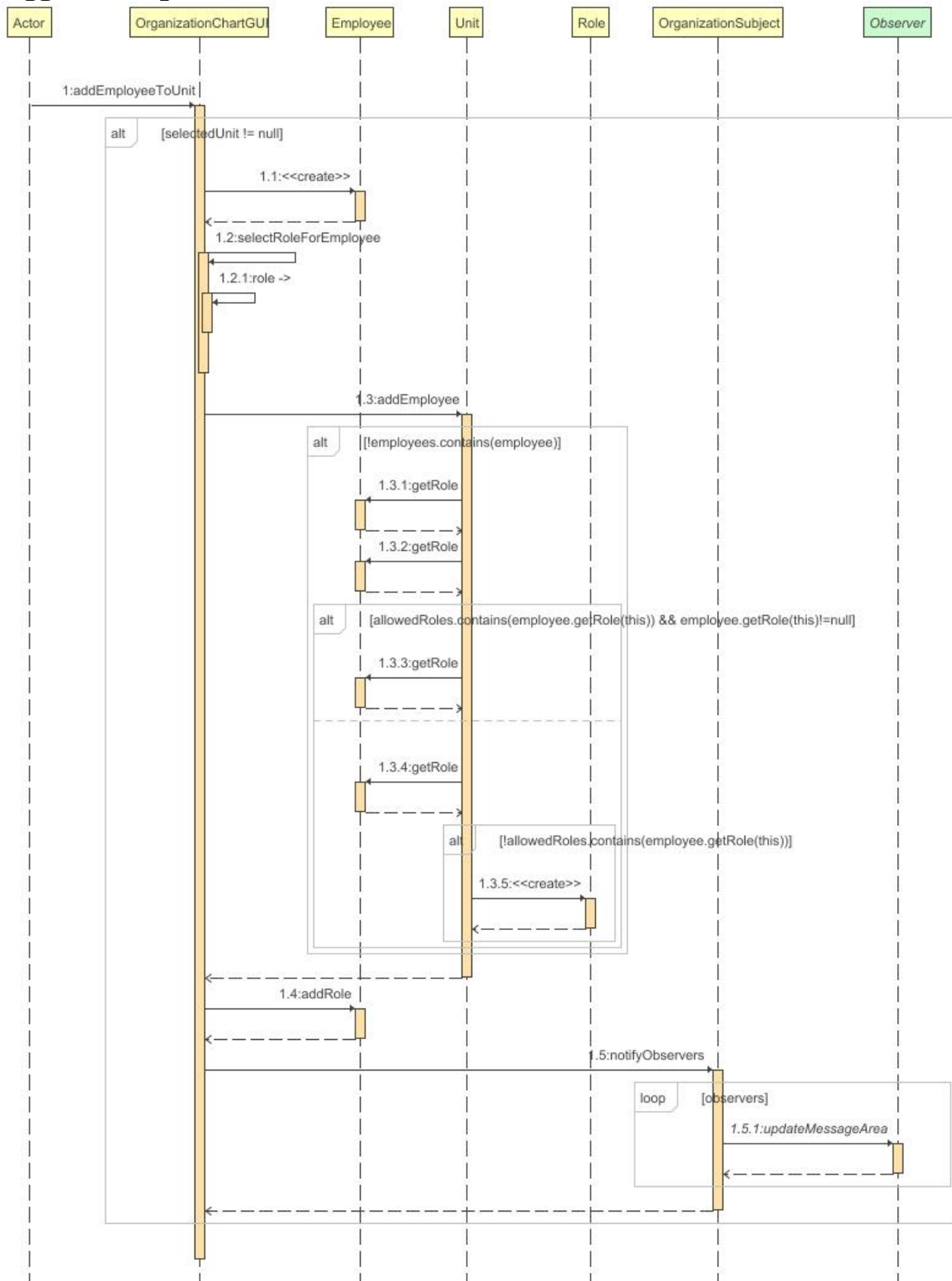
Aggiungi sotto unità



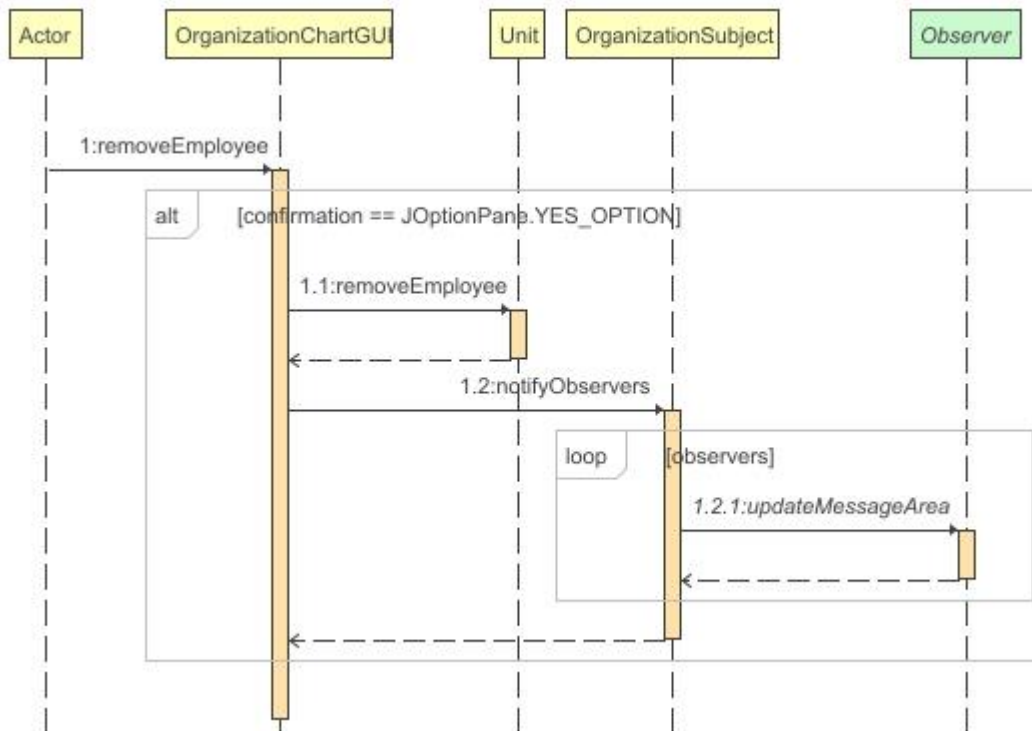
Rimozione unità



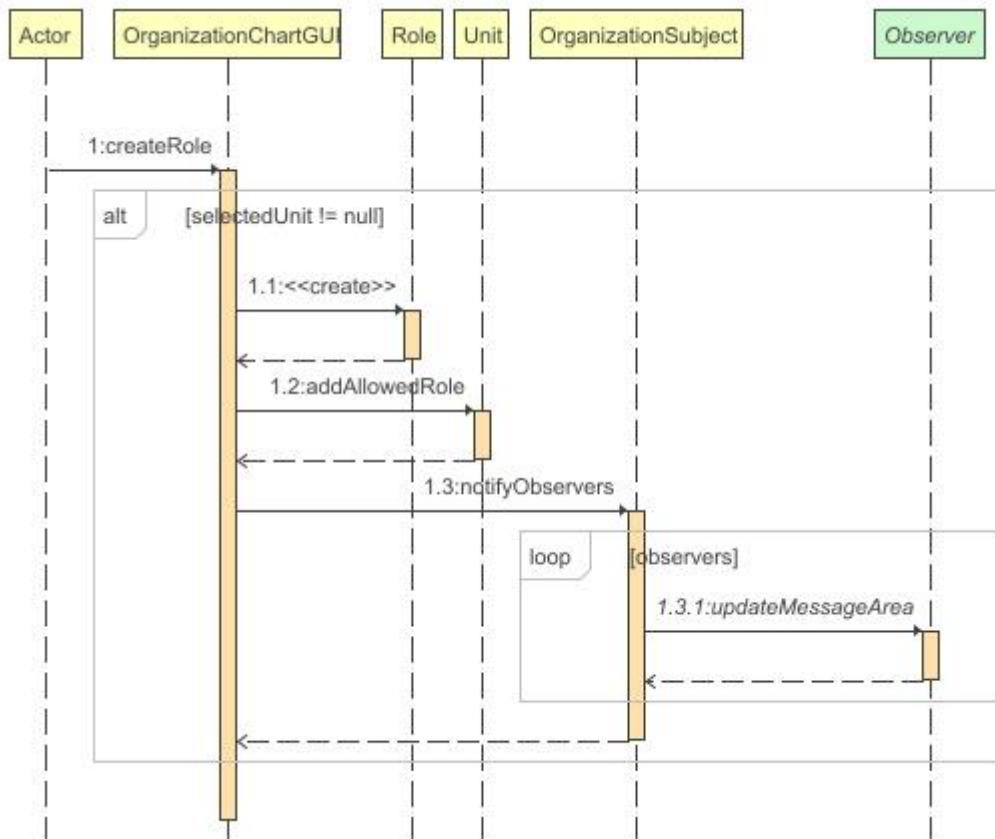
Aggiunta Dipendente ad Unità



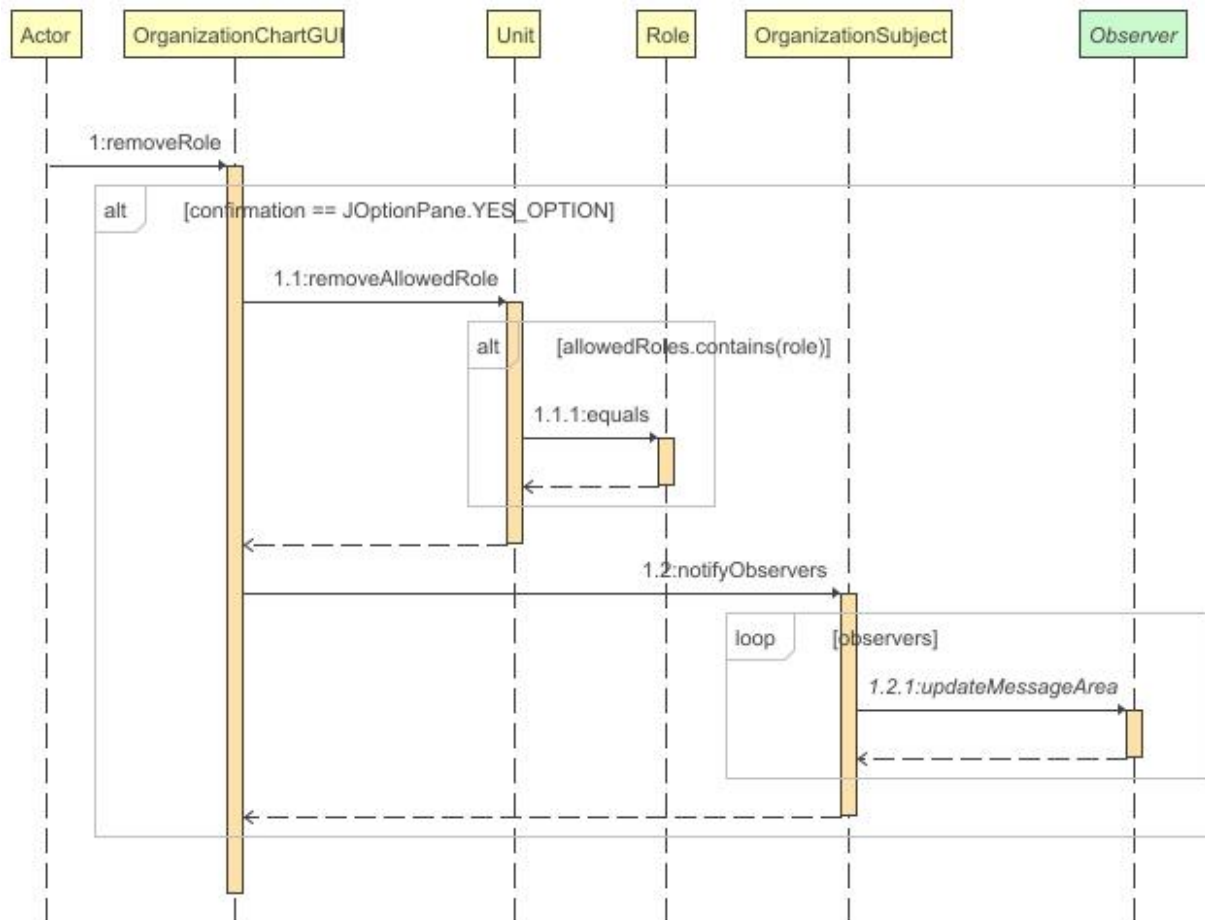
Rimozione Dipendente da Unità



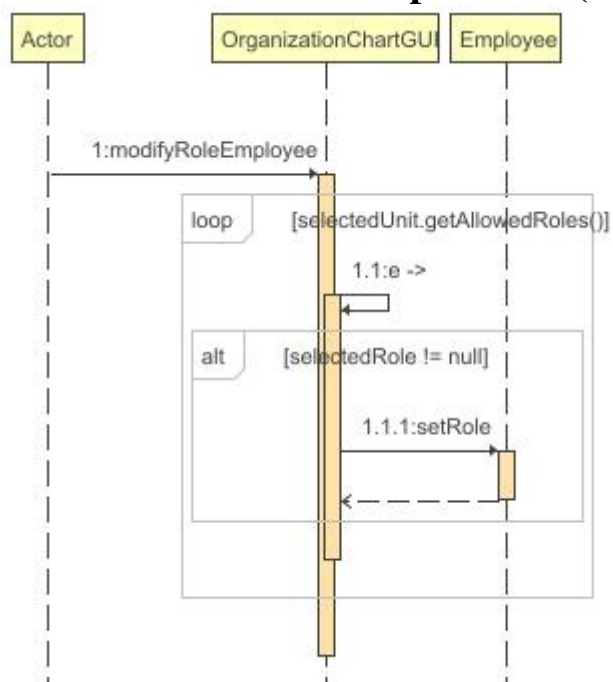
Aggiunta Ruolo ammissibile



Rimozione Ruolo ammissibile



Modifica Ruolo di un dipendente (tra i ruoli ammissibili)



D. Dati e loro modellazione (se il sistema si interfaccia con un DBMS)

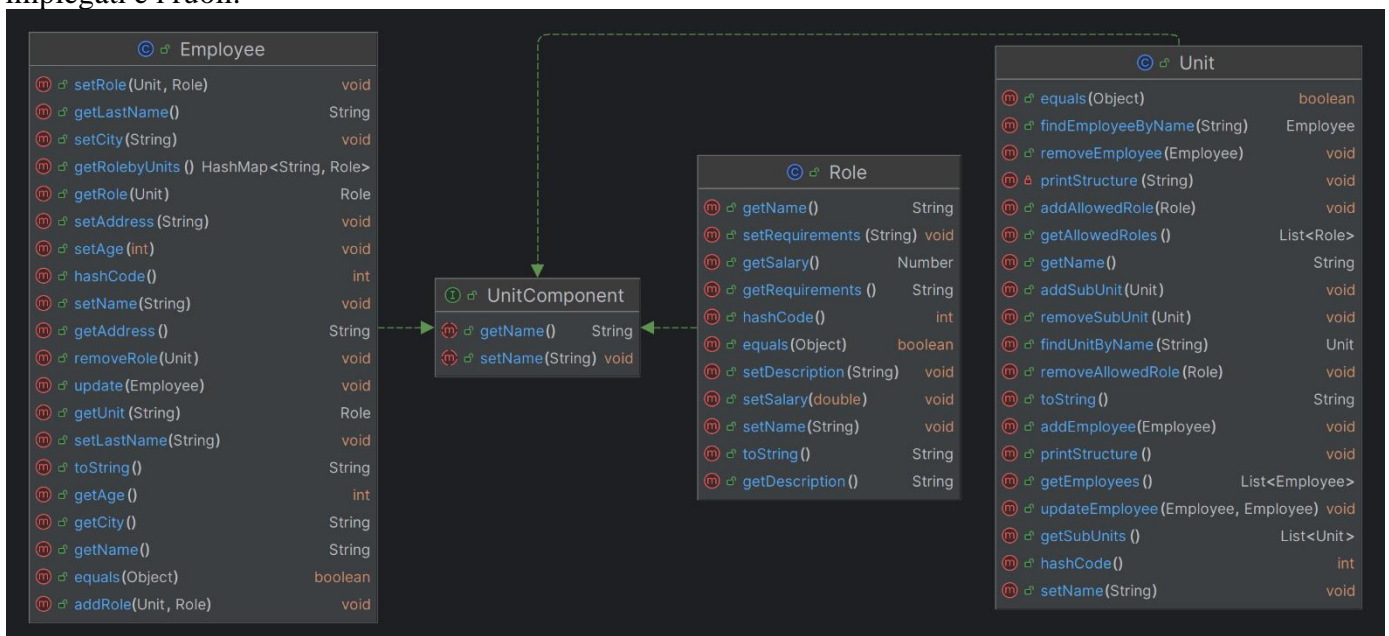
Il sistema non si interfaccia con un DBMS, salva lo stato dell'organigramma utilizzando la serializzazione. Deserializzando è possibile ripristinare lo stato precedentemente salvato.

E. Scelte Progettuali (Design Decisions)

Di seguito saranno riportare le 5 scelte progettuali più significative finalizzate alla realizzazione del sistema software in esame:

Struttura Gerarchica Organigramma: Composite

Nella creazione della struttura ad albero finalizzata alla rappresentazione dell'organigramma aziendale, si è scelto di impiegare il pattern Composite, con lo scopo di gestire in maniera uniforme le unità, gli impiegati e i ruoli.

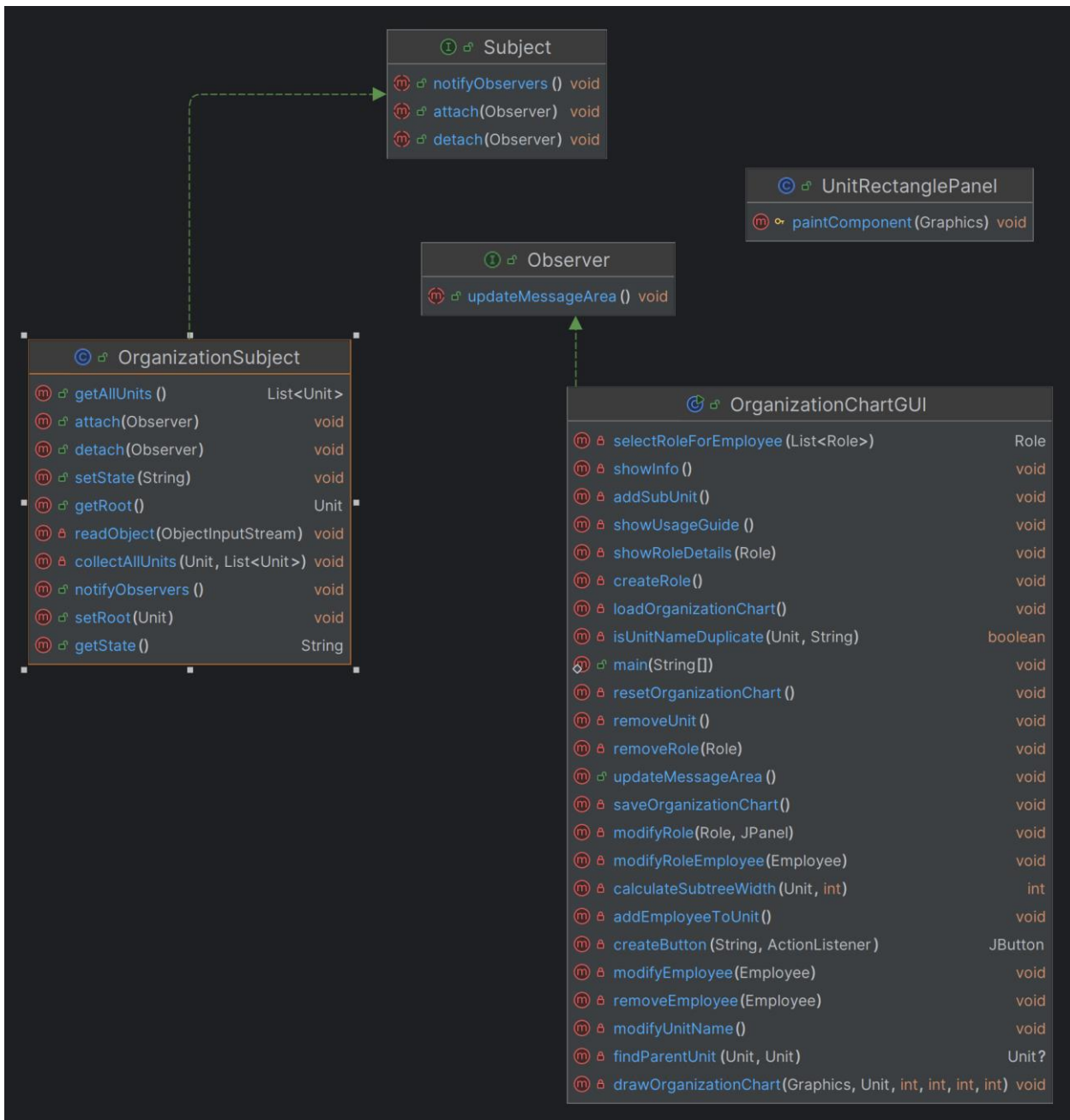


Interfaccia Grafica: Observer

L'interfaccia grafica è stata realizzata impiegando il design pattern Observer al fine di monitorare le operazioni dell'organigramma attraverso la gestione degli stati.

La classe OrganizationSubject rappresenta il soggetto osservato mentre la classe OrganizationChartGUI rappresenta l'osservatore ed è anche responsabile della visualizzazione ad albero dell'organigramma.

La classe UnitRectanglePanel è responsabile della creazione dei rettangoli utilizzati per la rappresentazione grafica dell'organigramma.



Persistenza dello stato dell'organigramma (Serializable):

La classe OrganizationSubject implementa Serializable, consentendo la persistenza dello stato dell'organigramma su file. Questo approccio permette di salvare e ricaricare lo stato dell'organizzazione, utile per mantenere i dati tra le sessioni. La gestione della serializzazione coinvolge anche il ripristino degli osservatori, che vengono reinizializzati per assicurare l'aggiornamento dell'interfaccia grafica.

Implementazione della struttura modulare per la separazione dei componenti:

Il progetto è organizzato in pacchetti distinti, come composite per la gestione della struttura aziendale e ObserverGUI per la gestione dell'interfaccia grafica. Questa modularità facilita la manutenibilità e l'estendibilità del software, in quanto ogni pacchetto è responsabile di un aspetto specifico del sistema. In caso di aggiornamenti o cambiamenti, è possibile lavorare su un singolo modulo senza influenzare gli altri, migliorando la coesione interna e riducendo il rischio di introduzione di bug dovuti a dipendenze non necessarie tra le parti del sistema.

Event-driven GUI con Listener:

La classe OrganizationChartGUI utilizza Listener per rispondere a eventi dell'utente (come MouseAdapter per rilevare selezioni e pulsanti per gestire le azioni sull'organigramma). Questo approccio semplifica l'interazione utente e garantisce che l'organigramma si aggiorni in tempo reale, seguendo i cambiamenti apportati agli oggetti dell'organizzazione.

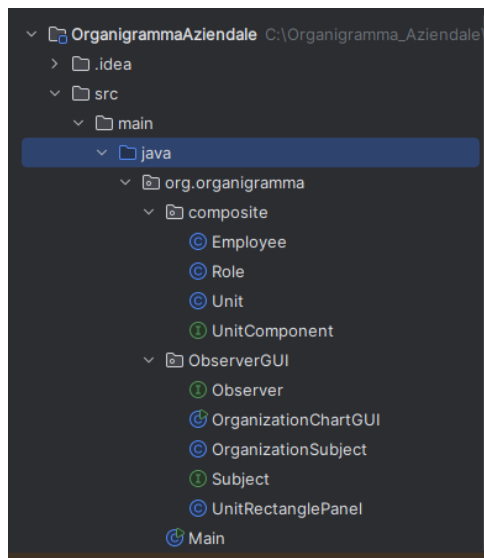
F. Progettazione di Basso Livello

Si descrivono i dettagli specifici dell'implementazione comprese le strutture dati, le interfacce, le classi ed i metodi principali.

L'applicazione finalizzata alla gestione ed organizzazione di un organigramma aziendale è stata realizzata in linguaggio Java sfruttando l'ambiente di sviluppo integrato IntelliJ IDEA.

Struttura codice Sorgente

Per quanto riguarda l'organizzazione del codice sorgente, si è scelto di dividere in package la seguente gerarchia riportata in figura:



OrganizationSubject (Classe)

- **Ruolo:** Rappresenta la struttura principale dell'organigramma aziendale e notifica gli osservatori di eventuali cambiamenti.
- **Strutture Dati:**

- List<Observer> observers: Lista di osservatori che vengono notificati quando lo stato dell'organigramma cambia.
- Unit rootNode: Nodo radice dell'organigramma.
- String state: Stato attuale dell'organizzazione, utilizzato per monitorare e notificare cambiamenti.
- long serialVersionUID: Versione della classe per la serializzazione.
- **Metodi:**
 - getAllUnits(): Ritorna una lista di tutte le unità presenti nell'organigramma.
 - collectAllUnits(Unit unit, List<Unit> allUnits): Metodo ricorsivo per raccogliere tutte le unità discendenti a partire da unit.
 - getState(): Restituisce lo stato corrente dell'organizzazione.
 - setState(String state): Imposta un nuovo stato per l'organizzazione.
 - getRoot(): Ritorna il nodo radice dell'organigramma.
 - setRoot(Unit root): Imposta un nuovo nodo radice per la struttura organizzativa.
 - attach(Observer observer): Aggiunge un osservatore alla lista observers.
 - detach(Observer observer): Rimuove un osservatore dalla lista.
 - notifyObservers(): Notifica tutti gli osservatori di un cambiamento nello stato dell'organigramma.

OrganizationChartGUI (Classe)

- **Ruolo:** Fornisce l'interfaccia grafica per visualizzare e interagire con l'organigramma aziendale.
- **Strutture Dati:**
 - OrganizationSubject organizationSubject: Riferimento alla struttura organizzativa principale.
 - JTextArea messageArea: Area di testo per mostrare messaggi e notifiche all'utente.
 - JPanel chartPanel: Pannello principale per la visualizzazione grafica dell'organigramma.
 - Unit selectedUnit: Unità aziendale attualmente selezionata dall'utente.
 - JScrollPane chartScrollPane: Pannello di scorrimento per visualizzare sezioni dell'organigramma.
 - Vari JButton: Pulsanti per azioni come aggiungere impiegati, modificare nomi, rimuovere unità e visualizzare informazioni, inclusi addEmployeeButton, modifyUnitNameButton, removeUnitButton, showInfoButton, resetOrganizationChartButton, saveOrganizationChartButton e loadOrganizationChartButton.
- **Metodi:**
 - paintComponent(Graphics g): Disegna l'organigramma all'interno del pannello grafico.
 - createButton(String text, ActionListener actionListener): Metodo ausiliario per creare pulsanti con testo e listener specifici.
 - drawOrganizationChart(Graphics g, Unit unit, int x, int y, int yOffset, int minSpacing): Disegna ricorsivamente l'organigramma a partire da una specifica unità.
 - mouseClicked(MouseEvent e): Gestisce i click del mouse per selezionare unità o impiegati.
 - calculateSubtreeWidth(Unit unit, int minSpacing): Calcola la larghezza di un sottoalbero dell'organigramma, per una visualizzazione spazialmente coerente.
 - updateMessageArea(): Aggiorna l'area di messaggi per mostrare le notifiche all'utente.
 - modifyUnitName(): Modifica il nome dell'unità attualmente selezionata.
 - addSubUnit(): Aggiunge una nuova sottounità alla struttura aziendale.
 - removeUnit(): Rimuove l'unità selezionata dall'organigramma.
 - findParentUnit(Unit current, Unit target): Trova e restituisce l'unità genitore di target partendo dall'unità current.
 - addEmployeeToUnit(): Aggiunge un impiegato a una specifica unità aziendale.
 - selectRoleForEmployee(List<Role> allowedRoles): Consente la selezione di un ruolo ammissibile per un impiegato in una determinata unità.
 - showInfo(): Visualizza informazioni dettagliate sull'unità o sull'impiegato selezionato.

- `resetOrganizationChart()`: Ripristina l'organigramma alla configurazione iniziale.
- `isUnitNameDuplicate(Unit root, String name)`: Verifica la presenza di un nome duplicato all'interno dell'organigramma.
- `saveOrganizationChart()`: Salva lo stato dell'organigramma su file per conservare le modifiche tra sessioni diverse.
- `loadOrganizationChart()`: Carica l'organigramma da un file salvato in precedenza, ripristinando lo stato salvato.

Observer (Interfaccia)

- **Ruolo:** Definisce il comportamento di un osservatore nel pattern Observer.
 - **Metodi:**
 - `updateMessageArea()`: Metodo che ogni osservatore deve implementare per aggiornare la propria area dei messaggi quando il soggetto invia una notifica.
-

Subject (Interfaccia)

- **Ruolo:** Definisce il comportamento del soggetto osservato nel pattern Observer.
 - **Metodi:**
 - `attach(Observer observer)`: Aggiunge un osservatore alla lista di quelli che devono essere notificati.
 - `detach(Observer observer)`: Rimuove un osservatore dalla lista.
 - `notifyObservers()`: Notifica tutti gli osservatori di un cambiamento.
-

UnitRectanglePanel (Classe)

- **Ruolo:** Rappresenta graficamente un'unità aziendale come un rettangolo nell'interfaccia grafica.
 - **Strutture Dati:**
 - `Unit unit`: Riferimento all'unità aziendale che viene rappresentata graficamente.
 - **Metodi:**
 - `UnitRectanglePanel(Unit unit, int x, int y, int width, int height)`: Costruttore che imposta l'unità da rappresentare e le proprietà grafiche del pannello, tra cui posizione, dimensioni e colore di sfondo. Configura inoltre un `MouseAdapter` per la gestione delle interazioni con il mouse.
 - `paintComponent(Graphics g)`: Sovrascrive il metodo `paintComponent` di `JPanel` per disegnare il nome dell'unità al centro del rettangolo, calcolando la posizione del testo in modo che sia centrato graficamente.
-

UnitComponent (Interfaccia)

- **Ruolo:** Interfaccia per le componenti dell'organigramma.
- **Metodi:**
 - `getName()`: Restituisce il nome della componente.

- setName(String name): Imposta il nome della componente.
-

Unit (Classe)

- **Ruolo:** Rappresenta un'unità aziendale che può contenere altre unità, impiegati e ruoli consentiti.
 - **Strutture Dati:**
 - String name: Nome dell'unità aziendale.
 - List<Employee> employees: Lista degli impiegati associati a questa unità.
 - List<Unit> subUnits: Lista delle sottounità aziendali contenute in questa unità.
 - List<Role> allowedRoles: Lista dei ruoli consentiti per questa unità.
 - long serialVersionUID: Versione della classe per la serializzazione.
 - **Metodi:**
 - addEmployee(Employee employee): Aggiunge un impiegato all'unità, verificando che il ruolo dell'impiegato sia tra quelli consentiti.
 - removeEmployee(Employee employee): Rimuove un impiegato dall'unità e aggiorna il ruolo associato.
 - addSubUnit(Unit subUnit): Aggiunge una sottounità, garantendo che non sia già presente.
 - updateEmployee(Employee oldEmp, Employee newEmp): Aggiorna le informazioni di un impiegato esistente con nuovi dati.
 - removeSubUnit(Unit subUnit): Rimuove una sottounità dall'unità corrente, compresi gli impiegati associati.
 - addAllowedRole(Role role): Aggiunge un ruolo alla lista dei ruoli consentiti per l'unità.
 - removeAllowedRole(Role role): Rimuove un ruolo dalla lista dei ruoli consentiti e aggiorna la lista degli impiegati.
 - findUnitByName(String unitName): Cerca e restituisce un'unità per nome all'interno della gerarchia.
 - getName(): Restituisce il nome dell'unità.
 - setName(String name): Imposta il nome dell'unità.
 - findEmployeeByName(String name): Cerca e restituisce un impiegato per nome.
 - getEmployees(): Ritorna la lista degli impiegati associati all'unità.
 - getSubUnits(): Ritorna la lista delle sottounità dell'unità corrente.
 - getAllowedRoles(): Ritorna la lista dei ruoli consentiti per l'unità.
 - equals(Object o): Confronta questa unità con un altro oggetto.
 - hashCode(): Calcola un valore hash per l'unità.
 - toString(): Rappresentazione testuale dell'unità.
-

Employee (Classe)

- **Ruolo:** Rappresenta un singolo impiegato all'interno dell'organigramma, con possibilità di avere ruoli diversi in unità diverse.
- **Strutture Dati:**
 - HashMap<String, Role> rolebyUnits: Mappa che associa ogni unità (identificata dal nome) a un ruolo specifico dell'impiegato in quella unità.
 - String name: Nome dell'impiegato.
 - String lastName: Cognome dell'impiegato.
 - String city: Città di residenza dell'impiegato.
 - String address: Indirizzo dell'impiegato.

- int age: Età dell'impiegato.
 - long serialVersionUID: Versione della classe per la serializzazione.
 - **Metodi:**
 - addRole(Unit unit, Role role): Aggiunge un ruolo all'impiegato per una specifica unità, se il ruolo è consentito dall'unità.
 - removeRole(Unit unit): Rimuove il ruolo dell'impiegato per l'unità specificata.
 - HashMap<String, Role> getRolebyUnits(): Ritorna la mappa di tutti i ruoli dell'impiegato suddivisi per unità.
 - Role getRole(Unit unit): Restituisce il ruolo dell'impiegato per una specifica unità.
 - Role getUnit(String name): Ritorna il ruolo associato all'unità specificata per nome.
 - setRole(Unit unit, Role role): Imposta il ruolo dell'impiegato in una specifica unità, verificando che il ruolo sia consentito.
 - getName(): Restituisce il nome dell'impiegato.
 - setName(String name): Imposta il nome dell'impiegato.
 - getLastName(): Restituisce il cognome dell'impiegato.
 - setLastName(String lastName): Imposta il cognome dell'impiegato.
 - getCity(): Restituisce la città dell'impiegato.
 - setCity(String city): Imposta la città di residenza dell'impiegato.
 - getAddress(): Restituisce l'indirizzo dell'impiegato.
 - setAddress(String address): Imposta l'indirizzo dell'impiegato.
 - getAge(): Restituisce l'età dell'impiegato.
 - setAge(int age): Imposta l'età dell'impiegato.
 - equals(Object o): Confronta l'impiegato con un altro oggetto per verificarne l'uguaglianza.
 - hashCode(): Calcola il valore hash dell'impiegato.
 - update(Employee emp): Aggiorna i dettagli dell'impiegato con i dati di un altro oggetto Employee.
 - toString(): Restituisce una rappresentazione testuale dell'impiegato.
-

Role (Classe)

- **Ruolo:** Definisce un ruolo aziendale.
- **Strutture Dati:**
 - String title: Titolo del ruolo.
 - String description: Descrizione del ruolo.
 - String requirements: Requisiti per il ruolo.
 - double salary: Salario associato al ruolo.
 - long serialVersionUID: Versione della classe per la serializzazione.
- **Metodi:**
 - getName(): Restituisce il titolo del ruolo.
 - setName(String name): Imposta il titolo.
 - getDescription(): Restituisce la descrizione.
 - setDescription(String description): Imposta la descrizione.
 - getRequirements(): Restituisce i requisiti.
 - setRequirements(String requirements): Imposta i requisiti.
 - getSalary(): Restituisce il salario.
 - setSalary(double salary): Imposta il salario.
 - equals(Object o): Confronta il ruolo con un altro oggetto per verificarne l'uguaglianza.
 - hashCode(): Calcola un valore hash per il ruolo.
 - toString(): Rappresentazione testuale del ruolo.

G. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs)

Di seguito sono descritti gli aspetti implementativi del progetto, che rispondono ai requisiti funzionali e non funzionali definiti durante la fase di analisi dei requisiti. Maggiori dettagli sull'implementazione specifica delle componenti dell'applicazione sono forniti nella sezione dedicata alla Progettazione di Basso Livello.

Requisiti Funzionali

Il progetto soddisfa i seguenti requisiti funzionali:

1. Gestione delle modifiche allo stato dell'organigramma

Grazie all'uso del pattern Composite e all'implementazione delle classi `Unit`, `Employee` e `Role`, il progetto supporta un'agevole modifica della struttura dell'organigramma.

- Gli utenti possono aggiungere, rimuovere o aggiornare membri ruoli e unità dell'organigramma aziendale.
- Ogni modifica si riflette automaticamente sulla struttura gerarchica e sui ruoli associati a ogni membro.

2. Il progetto fornisce una rappresentazione grafica dell'organigramma corrente.

Il sistema fornisce una rappresentazione grafica aggiornata dell'organigramma, resa possibile grazie al pattern Observer e alle classi `OrganizationChartGUI`, `OrganizationSubject` e `UnitRectanglePanel`, nonché agli elementi di gestione interattiva come `messageArea` e `chartPanel`.

- La visualizzazione dell'organigramma è intuitiva e offre una chiara rappresentazione delle relazioni gerarchiche e dei ruoli aziendali.
- Gli utenti possono navigare e visualizzare le strutture e le gerarchie, favorendo una comprensione immediata della disposizione aziendale.

3. Memorizzazione su memoria secondaria

Il progetto permette il salvataggio persistente dell'organigramma su memoria secondaria attraverso la serializzazione e deserializzazione delle strutture dati.

- Le funzionalità `saveOrganizationChart` e `loadOrganizationChart` consentono di memorizzare l'organigramma e di ripristinarlo al bisogno, mantenendo tutte le modifiche in modo permanente.

Requisiti Non Funzionali

1. Prestazioni

Il progetto è ottimizzato per gestire efficientemente un ampio numero di unità, impiegati e ruoli, senza degrado delle prestazioni. L'organizzazione delle strutture dati e l'implementazione dei metodi garantiscono performance elevate durante l'accesso e la

modifica delle informazioni.

2. Scalabilità e Manutenibilità

Il progetto risulta essere progettato per essere scalabile, in modo che possa gestire l'aggiunta di nuove unità, impiegati e ruoli senza dover apportare modifiche significative al codice sorgente, l'impiegato dei design pattern ha garantito ordine nella stesura del codice, e maggiore facilità nella comprensione dello stesso.

3. Affidabilità

Il sistema è affidabile e in grado di gestire correttamente situazioni anomale grazie a un'efficace gestione degli errori e alla validazione degli input. L'interfaccia grafica riduce il rischio di errori da parte dell'utente, guidandolo nell'uso dell'applicazione.

4. Sicurezza

La persistenza dei dati è assicurata tramite meccanismi interni di salvataggio e caricamento:

- Gli utenti possono archiviare l'organigramma su memoria secondaria e ripristinarlo quando necessario, facendo riferimento al file salvato in modo sicuro.

5. Usabilità

L'interfaccia utente è progettata per essere intuitiva e facile da usare, con una struttura che agevola la navigazione e l'esplorazione delle funzionalità disponibili. Il pattern Observer facilita la gestione ordinata e reattiva dell'interfaccia grafica, migliorando l'esperienza d'uso.

6. Portabilità

Implementato in Java, il progetto è portabile e può essere eseguito su qualsiasi piattaforma compatibile con la Java Virtual Machine (JVM), rendendo l'applicazione accessibile su diversi dispositivi e sistemi operativi.

Appendix. Prototype

Di seguito viene fornita una descrizione del prototipo realizzato, che costituisce un'applicazione completa per la creazione e gestione di un organigramma aziendale. Le funzionalità descritte sono progettate per rispondere ai requisiti funzionali e non funzionali definiti in fase di progettazione.

Descrizione

L'applicazione consente di caricare un organigramma di esempio tramite l'opzione **“Carica Organigramma”**, all'avvio viene mostrata una prima unità radice chiamata **“Azienda”**

Visualizzazione

Nel pannello centrale, le unità aziendali sono rappresentate come rettangoli, collegati graficamente alle relative sotto-unità con segmenti. Questa modalità mostra solo le unità, senza rappresentare gli impiegati

(visualizzabili cliccando su “visualizza info unità”), e si aggiorna automaticamente in base alle modifiche apportate (inserimento, rimozione di unità).

Barra degli Strumenti (a sinistra)

La barra degli strumenti offre i seguenti comandi di gestione dell’organigramma:

- Modifica Nome Unità: Modifica il nome di un'unità esistente.
- Aggiungi Sotto Unità: Definisce una nuova sotto-unità per un’unità selezionata.
- Rimuovi Unità: Rimuove un'unità selezionata (a meno che non sia la radice).
- Aggiungi Dipendente ad Unità: Consente di creare un nuovo impiegato da assegnare alle unità potendo scegliere ruoli ammissibili specifici
- Crea Ruolo Ammissibile: Definisce un nuovo ruolo ammissibile per l’unità selezionata, che può essere assegnato ai dipendenti.

Pannello Laterale (a destra)

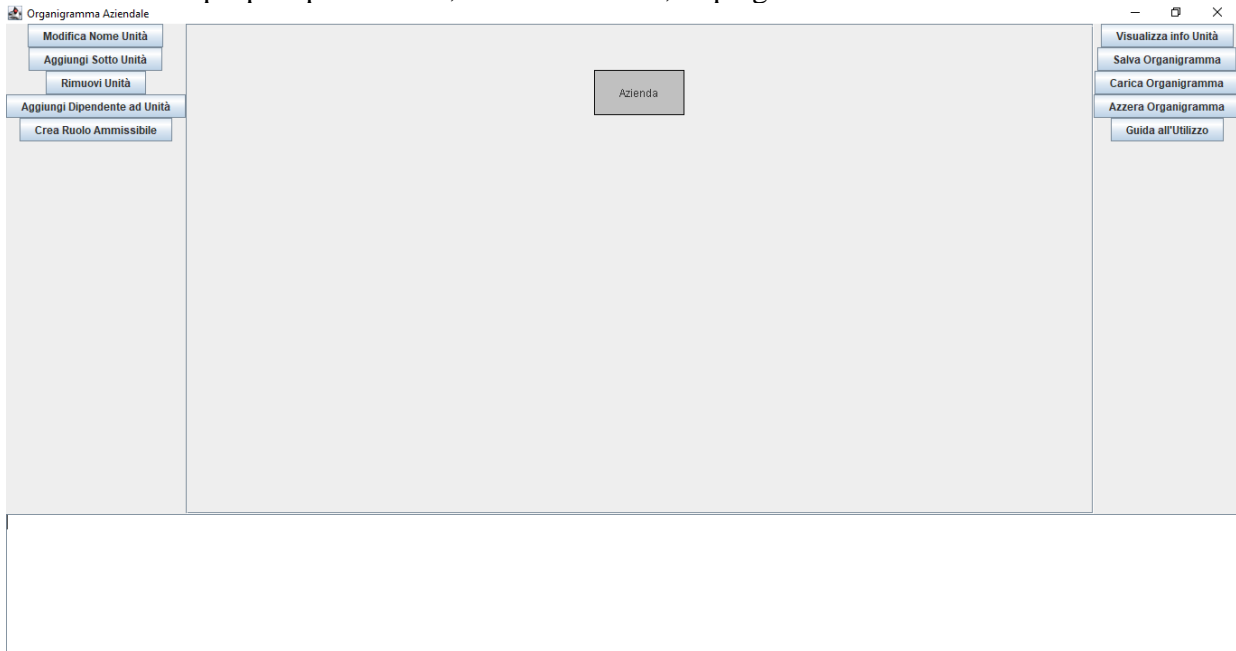
Il pannello laterale include ulteriori opzioni per la visualizzazione e la gestione avanzata dell’organigramma:

- Visualizza info Unità: Si apre un pannello in cui dipendentemente dall’unità selezionata sono disponibili tutti i dipendenti per quella specifica unità con i rispettivi ruoli ed anche una lista dei ruoli ammissibili per quella unità. Nota che di fianco ciascun dipendente sono presenti dei bottoni per la visualizzazione delle rispettive informazione, modifica ed eventuale rimozione. Discorso analogo per i ruoli ammissibili.
- Salva Organigramma: Permette di salvare l’organigramma corrente su, da poter ricaricare successivamente.
- Carica Organigramma: Consente di ricaricare un organigramma precedentemente salvato.
- Azzera Organigramma: Ripristina l’organigramma allo stato iniziale.

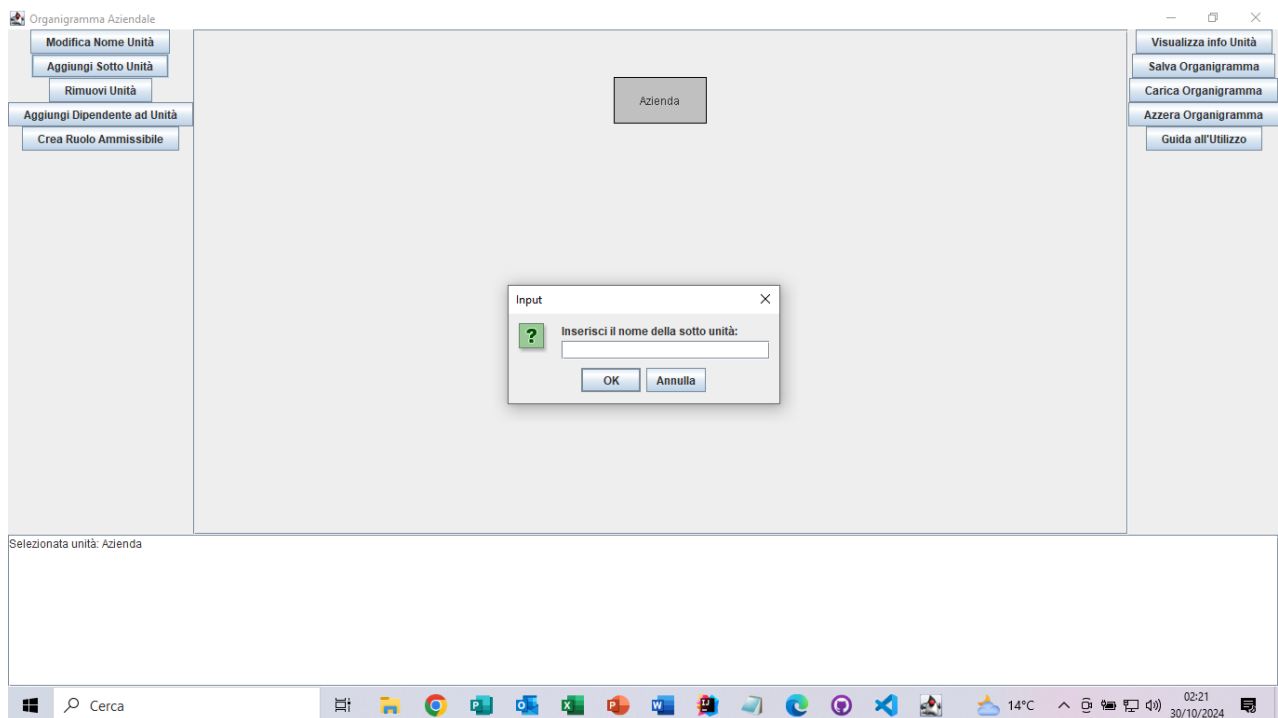
Avvio applicazione

All'avvio l'applicazione si presenterà con una sola Unità creata, Azienda, con nessun ruolo o impiegato definito all'interno della struttura dell'organigramma.

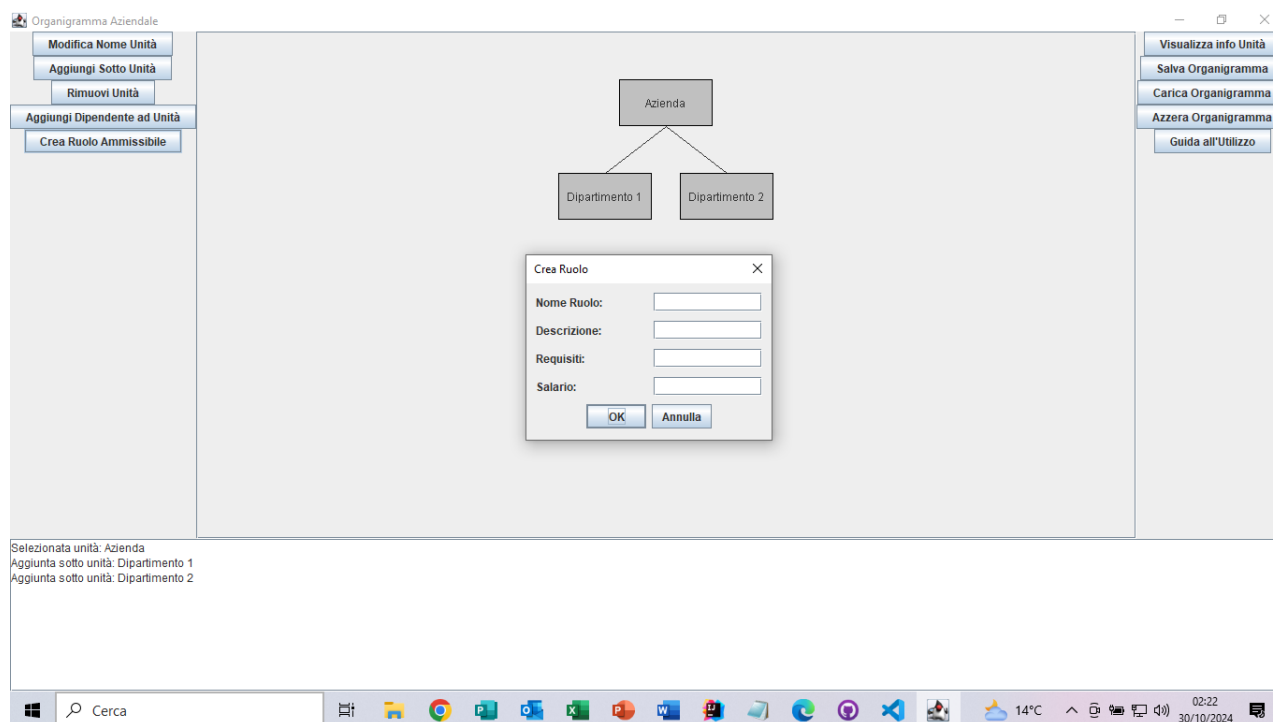
Per poter iniziare a lavorare è sufficiente modificare il nome dell'unica unità esistente e iniziare a definire a proprio piacimento, altre sotto unità, impiegati e ruoli.



Inserimento Unità

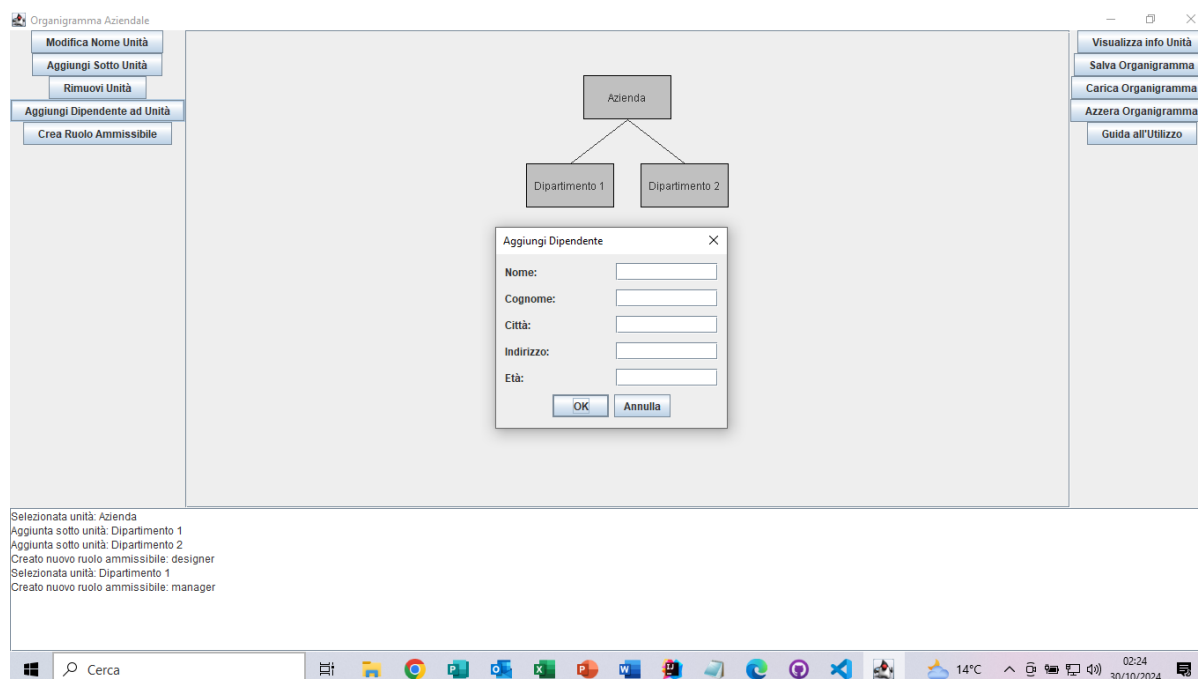


Creazione Ruolo Ammissibile per l'unità selezionata



Aggiunta Dipendente

Per aggiungere un dipendente ad una unità è necessario che essa contenga almeno un ruolo ammissibile.



Assegnazione Ruolo al dipendente

La scelta dei ruoli è tra i ruoli ammissibili dell'unità selezionata

The screenshot shows the 'Organigramma Aziendale' application interface. On the left, there is a sidebar with buttons: 'Modifica Nome Unità', 'Aggiungi Sotto Unità', 'Rimuovi Unità', 'Aggiungi Dipendente ad Unità', and 'Crea Ruolo Ammissibile'. On the right, there is another sidebar with buttons: 'Visualizza info Unità', 'Salva Organigramma', 'Carica Organigramma', 'Azzerà Organigramma', and 'Guida all'Utilizzo'. The main area displays a hierarchical tree structure with 'Azienda' at the top, branching into 'Dipartimento 1' and 'Dipartimento 2'. A dialog box titled 'Assegna Ruolo' is open in the center, showing a green question mark icon and the text 'Seleziona un ruolo per il dipendente:'. Below this text is a dropdown menu with 'manager' selected. At the bottom of the dialog are 'OK' and 'Annulla' buttons. Below the main area, there is a log of actions: 'Selezionata unità: Azienda', 'Aggiunta sotto unità: Dipartimento 1', 'Aggiunta sotto unità: Dipartimento 2', 'Creato nuovo ruolo ammissibile: designer', 'Selezionata unità: Dipartimento 1', and 'Creato nuovo ruolo ammissibile: manager'. The Windows taskbar at the bottom shows the date and time as 02:24 on 30/10/2024.

Visualizza info Unità

Cliccando il bottone “Visualizza info Unità” è possibile osservare tutte le informazioni sui dipendenti ed i ruoli ammissibili relativi all'unità selezionata.

The screenshot shows the 'Organigramma Aziendale' application interface with the 'Visualizza info Unità' button clicked. The main area now displays a dialog box titled 'Informazioni Unità Dipartimento 1'. The dialog box has two sections. The first section, 'Dipendente:', shows 'Mario - Ruolo: manager' with buttons 'Mostra/Modifica info personali', 'Modifica Ruolo', and 'Elimina'. The second section, 'Ruoli Ammissibili:', shows 'manager' with buttons 'Mostra Dettagli', 'Modifica', and 'Elimina'. The sidebar buttons remain the same as in the previous screenshot. The log of actions at the bottom now includes 'Aggiunto dipendente: Mario con ruolo manager'. The Windows taskbar at the bottom shows the date and time as 02:25 on 30/10/2024.