

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

“Ingegneria del Software” 2021-2022

Docente: Prof. Angelo Furfaro

Organigramma Aziendale

Data	<08/09/2022>
Documento	Documento Finale – D3

Team Members		
Nome e Cognome	Matricola	E-mail address
Michele Purrone	201181	michelegpurrone@gmail.com

Sommario

1	
---	--

List of Challenging/Risky Requirements or Tasks.....	3
A. Stato dell'Arte	4
B. Raffinamento dei Requisiti.....	5
B.1 Servizi (con prioritizzazione)	6
B.2 Requisiti non funzionali	7
B.2 Scenari d'uso dettagliati	8
B.3 Assunzioni	11
B.4 Use Case Diagrams	11
C. Architettura Software	12
C.1 The static view of the system: Component Diagram	12
C.2 The dynamic view of the software architecture: Sequence Diagram	13
D. Scelte Progettuali (Design Decisions)	17
E. Progettazione di Basso Livello.....	19
G. Soddisfacimento requisiti.....	20
Appendix. Prototype.....	22

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
Definire le entità di un organigramma. Come si possono rappresentare?	23-08-2022	26-08-2022	Considerando la struttura interna di un organigramma, viene naturale adottare il pattern <i>Composite</i> come la scelta migliore. Inoltre, il pattern ben si presta a poter introdurre nuovi tipi di entità da aggiungere alle unità, sottounità e organi di gestione richiesti dalla traccia.
Associazione tra unità e dipendenti. Come si può realizzare?	27-08-2022	27-08-2022	Dopo aver definito le unità e creato i dipendenti, si è scelto di associarli tramite una mappa <i><Unità, Ruolo></i> nella classe <i>Employee</i> . In questo modo è possibile compiere agevolmente varie operazioni di modifica (ad esempio è possibile aggiungere nuovi ruoli a un'unità già esistente).
Rappresentazione grafica delle entità e delle relazioni gerarchiche.	28-08-2022	31-08-2022	La scelta è ricaduta sul pattern architetturale <i>MVC</i> . Esso consente di rappresentare le entità come box e le relazioni gerarchiche come linee.
Implementazione GUI	31-08-2022	05-09-2022	La GUI principale è stata ottenuta combinando opportunamente delle GUI per ogni funzione richiesta dalla traccia.

A. Stato dell'Arte

Un organigramma è la rappresentazione grafica della struttura di una determinata organizzazione in un dato momento storico. L'organigramma aziendale illustra in genere le relazioni tra le varie entità (eventualmente anche tra i componenti) che compongono un'organizzazione. Ad esempio, si potrebbero includere relazioni tra i direttori di un'unità con gli amministratori delegati, gli amministratori delegati con i vari dipartimenti, i dipartimenti con le sottounità organizzative e così via. I diversi tipi di organigrammi aziendali possono essere strutturati come:

- *Gerarchia*
- *Matrice*
- *Piatto (o orizzontale)*

Dai primi prototipi di gerarchie organizzative dell'antichità rappresentate artisticamente come dipinti e sculture, si arriva al 1855, quando la Erie Railroad Company americana realizzò il primo organigramma moderno per poter facilmente visualizzare e catalogare la loro grande azienda sia per funzioni che per aree geografiche (<https://www.loc.gov/item/2017586274>). Ad oggi, con una storia che parte dalle rappresentazioni cartacee e arriva nell'era della digitalizzazione, sono disponibili tantissimi prodotti software in grado di rappresentare un organigramma aziendale. Tra i software attuali troviamo:

- *Microsoft Visio* (<https://www.microsoft.com/it-it/microsoft-365/visio/flowchart-software>): prodotto dalla Microsoft e venduto in aggiunta al pacchetto Office, Visio consente di disegnare vari diagrammi (di flusso, organigrammi, piante, planimetrie, modellazione di processi aziendali, mappe 3D e molti altri).
 - *Lucidchart* (<https://www.lucidchart.com>): è un'applicazione web per la creazione di diagrammi che consente agli utenti di collaborare visivamente per disegnare, rivedere e condividere grafici e diagrammi e migliorare processi, sistemi e strutture organizzative.
 - *Pingboard* (<https://pingboard.com>): è un software progettato per creare organigrammi in tempo reale. Sincronizza automaticamente gli organigrammi con il software HRMS, sostituendo la necessità di creare organigrammi manualmente utilizzando applicazioni di diagrammi di flusso come i precedenti *Microsoft Visio* o *Lucidchart*.
-

B. Raffinamento dei Requisiti

Il sistema software “Organigramma Aziendale” deve consentire di creare e gestire un organigramma, in particolare:

- L'applicazione deve consentire di creare e aggiornare dinamicamente la struttura di una determinata organizzazione. Quando vengono definite le entità e le relazioni gerarchiche, è possibile scegliere le tipologie di *unità* (ad es. dipartimenti), *sottounità organizzative* (ad es. gruppi di lavoro) e/o di *organi di gestione*.
 - Quando viene creata un'entità, di qualunque tipo essa sia, è possibile definire dei ruoli possibili per ciascuna di essa (ad es. l'unità “*Consiglio di Amministrazione*” può avere un ruolo “*Direttore*” disponibile; una sottounità potrebbe avere sia “*Caporeparto*” che “*Impiegato*” e così via...).
 - L'applicazione deve consentire di aggiungere i dipendenti dell'organizzazione e di definire i ruoli che essi possono ricoprire. Ciascun dipendente può avere uno o più ruoli (ad es. “*Marco Rossi*” può svolgere contemporaneamente il ruolo di “*Direttore*” e di “*Consigliere*”).
 - Dopo aver definito un dipendente dell'azienda, se i ruoli che può ricoprire sono compatibili con quelli disponibili in una determinata unità organizzativa, deve essere possibile associarli. Chiaramente, i ruoli introdotti possono avere senso in alcune unità e non in altre (ad es. il dipendente “*Mario Verdi*”, candidato come “*Impiegato*” non può svolgere il ruolo di “*Direttore*” nel “*Consiglio di Amministrazione*”).
 - L'applicazione deve consentire di memorizzare su memoria secondaria gli organigrammi che gestisce.
 - L'applicazione deve avere un'interfaccia utente grafica per poter manipolare gli organigrammi che gestisce.
-

A.1 Servizi (con prioritizzazione)

ID	SERVIZIO	IMPORTANZA	COMPLESSITA'
#1	<i>Creazione organigramma e definizione entità</i>	<i>Alta</i>	<i>Media</i>

È possibile definire in maniera dinamica e interattiva le entità e le relazioni dell'organigramma che si vuole creare. Si può scegliere la categoria di unità, la tipologia e i ruoli disponibili per ciascuna. Inoltre, per rappresentare le relazioni gerarchiche, è possibile definire l'unità genitore a cui sono legate una o più unità figlie.

#2	<i>Aggiunta dipendenti e definizione dei ruoli che possono ricoprire</i>	<i>Alta</i>	<i>Media</i>
----	--------------------------------------------------------------------------	-------------	--------------

È possibile aggiungere dinamicamente nuovi dipendenti all'organigramma definito. Si può specificare il nome del dipendente, l'unità per la quale il dipendente si candida e i ruoli che egli è in grado di ricoprire.

#3	<i>Associazione dipendente-unità e soddisfacimento vincoli</i>	<i>Alta</i>	<i>Bassa</i>
----	----------------------------------------------------------------	-------------	--------------

Quando vengono create delle unità e quando vengono inseriti nuovi dipendenti è possibile definire i ruoli disponibili nelle prime e ricopribili dai secondi. Se l'unità ha ruoli disponibili che il dipendente può ricoprire è possibile definire un'associazione. Non è possibile assegnare dipendenti a unità non compatibili e viceversa.

#4	<i>Salvataggio organigramma</i>	<i>Media</i>	<i>Bassa</i>
----	---------------------------------	--------------	--------------

È stato definito un sistema di salvataggio su file system dell'organigramma creato. I file vengono memorizzati in un formato e in una cartella di salvataggio scelti a piacere dall'utente.

ID	SERVIZIO	IMPORTANZA	COMPLESSITA'
#5	Interfaccia utente grafica	Media	Media
È stata creata un'interfaccia utente grafica per migliorare l'esperienza utente e l'interazione col sistema. Tramite essa è possibile svolgere tutte le operazioni offerte dall'applicativo.			

A.2 Requisiti non Funzionali

L'input che fornisce l'utente non è sempre valido: non è possibile creare unità senza definire i ruoli disponibili o l'eventuale unità genitore; oppure aggiungere dipendenti senza un ruolo da ricoprire o un'unità rispetto alla quale candidarsi

3 Scenari d'uso dettagliati

Creazione organigramma	
Tipo	Primario
Svolgimento normale	<ol style="list-style-type: none"> 1. L'utente avvia il sistema. 2. L'utente <u>crea nuove unità</u>. 3. L'utente <u>aggiunge nuovi dipendenti</u>. 4. L'utente <u>salva l'organigramma</u> ottenuto. 5. L'utente <u>ottiene informazioni</u> sul sistema. 6. L'utente <u>termina l'esecuzione</u> del sistema.
Svolgimento alternativo	6a. L'utente non termina correttamente il programma.
Descrizione	L'utente è in grado creare e gestire un organigramma aziendale partendo dalla creazione fino al salvataggio sul disco.

Creazione nuova unità	
Tipo	Primario
Svolgimento normale	<ol style="list-style-type: none"> 1. L'utente seleziona il punto in cui desidera visualizzare l'unità da creare. 2. L'utente seleziona la voce "<i>Create new Unit</i>" dalla barra degli strumenti per la creazione di una nuova unità. 3. L'utente sceglie una categoria tra le disponibili. 4. L'utente inserisce la tipologia di unità che desidera creare. 5. L'utente conferma le scelte fatte con un apposito pulsante. 6. L'utente inserisce il ruolo che desidera rendere disponibile. 7. L'utente conferma la scelta fatta con un apposito pulsante. 8. L'utente inserisce l'unità genitore per creare e visualizzare una relazione gerarchica. 9. L'utente conferma la scelta fatta con un apposito pulsante. 10. L'utente conferma la creazione: il sistema visualizza l'unità creata.
Svolgimento alternativo	<ol style="list-style-type: none"> 1a. La prima unità da creare ha una posizione di default. 4a. L'utente non inserisce la tipologia di unità: il sistema segnala l'errore ed esegue nuovamente la richiesta. 6a. L'utente non inserisce il ruolo: il sistema segnala l'errore ed esegue nuovamente la richiesta. 7b. L'utente inserisce e conferma un altro ruolo per l'unità che vuole creare. 6c. L'utente ripete il passo 7b più volte. 8a. La prima unità da creare non può avere un'unità genitore: il

	sistema non consente l'inserimento. 8b. L'utente non inserisce l'unità genitore o ne inserisce una che non esiste: il sistema segnala l'errore ed esegue nuovamente la richiesta.
Descrizione	Questa operazione consente di creare e visualizzare una o più unità.

Aggiunta di un nuovo dipendente	
Tipo	Primario
Svolgimento normale	<ol style="list-style-type: none"> 1. L'utente seleziona la voce "<i>Add new Employee</i>" dalla barra degli strumenti per l'aggiunta di un nuovo dipendente. 2. L'utente inserisce il nome del dipendente che desidera aggiungere. 3. L'utente conferma la scelta fatta con un apposito pulsante. 4. L'utente inserisce la tipologia di unità che desidera associare al dipendente. 5. L'utente inserisce il ruolo che il dipendente può ricoprire. 6. L'utente conferma le scelte fatte con un apposito pulsante. 7. L'utente conferma la creazione: il sistema aggiunge un dipendente e lo abbina a un'unità.
Svolgimento alternativo	<ol style="list-style-type: none"> 2a. L'utente non inserisce il nome del dipendente: il sistema segnala l'errore ed esegue nuovamente la richiesta. 4a. L'utente non inserisce la tipologia di unità o ne inserisce una che non esiste: il sistema segnala l'errore ed esegue nuovamente la richiesta. 5a. L'utente non inserisce un ruolo: il sistema segnala l'errore ed esegue nuovamente la richiesta. 6a. L'utente sta provando ad associare un dipendente con un ruolo che però non è disponibile per l'unità inserita: il sistema segnala l'errore ed esegue nuovamente la richiesta. 6b. L'utente può associare uno stesso dipendente a più unità ripetendo i passi 4-6.
Descrizione	Questa operazione consente di aggiungere uno o più dipendenti.

Salvataggio dell'organigramma	
Tipo	Primario

Svolgimento normale	<ol style="list-style-type: none"> 1. L'utente seleziona la voce "Save" dalla barra degli strumenti per il salvataggio dell'organigramma. 2. L'utente seleziona una directory e sceglie il nome e il formato del file in cui memorizzare l'organigramma. 3. L'utente conferma il salvataggio con un apposito pulsante. 4. Il sistema conferma l'avvenuto salvataggio.
Svolgimento alternativo	<ol style="list-style-type: none"> 1a. L'utente tenta di salvare nuovamente. <ol style="list-style-type: none"> .1 Il sistema non ha un file precedentemente salvato. <ol style="list-style-type: none"> .1 Il sistema riporta l'utente al passo 2. .2 Il sistema ha già un file precedentemente salvato. <ol style="list-style-type: none"> .1 Il sistema propone di sovrascrivere il file già salvato. <ol style="list-style-type: none"> .1 Risposta affermativa, si torna al passo 2a.1. .2 Risposta negativa, si torna al passo 2. 2a. L'utente annulla l'operazione di salvataggio <ol style="list-style-type: none"> .1 Il sistema riporta l'utente alla schermata precedente
Descrizione	Questa operazione permette all'utente di memorizzare su file system un organigramma definito precedentemente, unità e dipendenti compresi.

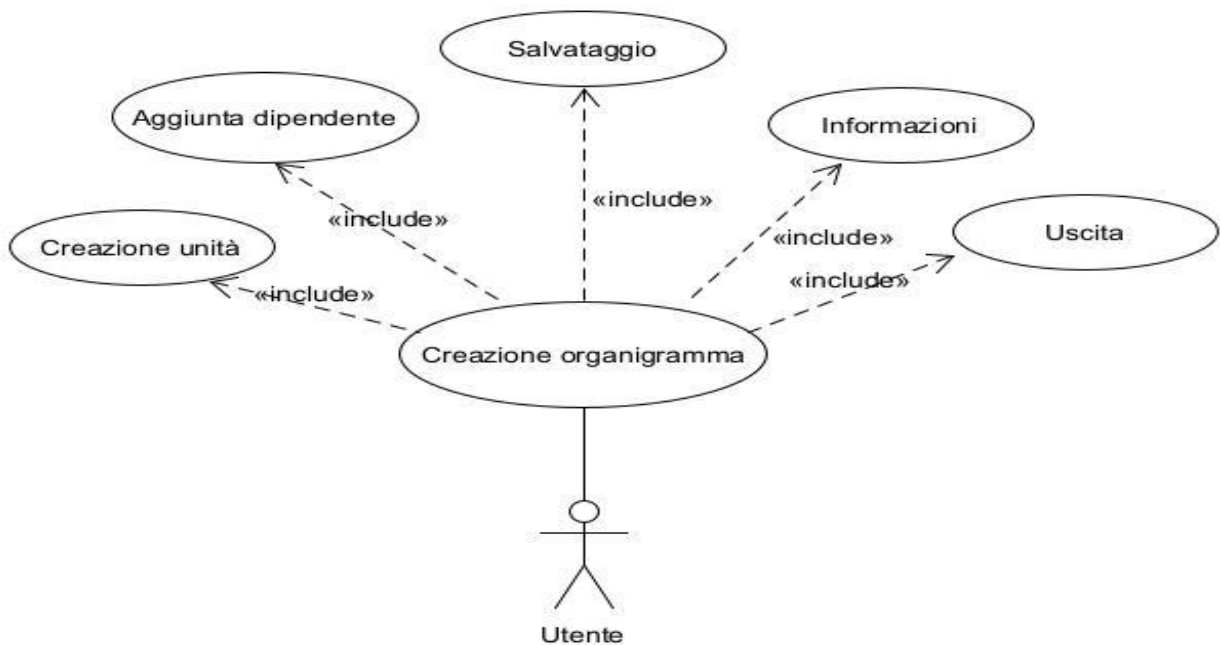
Informazioni sul sistema	
Tipo	Secondario
Svolgimento normale	<ol style="list-style-type: none"> 1. L'utente seleziona la voce "Info" dalla barra degli strumenti. 2. Il sistema mostra all'utente informazioni su di esso.
Descrizione	Questa operazione permette all'utente di ottenere informazioni sul sistema.

Uscita	
Tipo	Secondario
Svolgimento normale	<ol style="list-style-type: none"> 1. L'utente seleziona la voce "Exit" dalla barra degli strumenti. 2. Il sistema chiede all'utente una ulteriore conferma. 3. In caso di risposta affermativa, il sistema termina l'esecuzione.
Svolgimento alternativo	3a. In caso di risposta negativa, il sistema riporta l'utente alla finestra principale.
Descrizione	Questa operazione permette all'utente di terminare l'esecuzione dell'applicazione.

A.5 Assunzioni

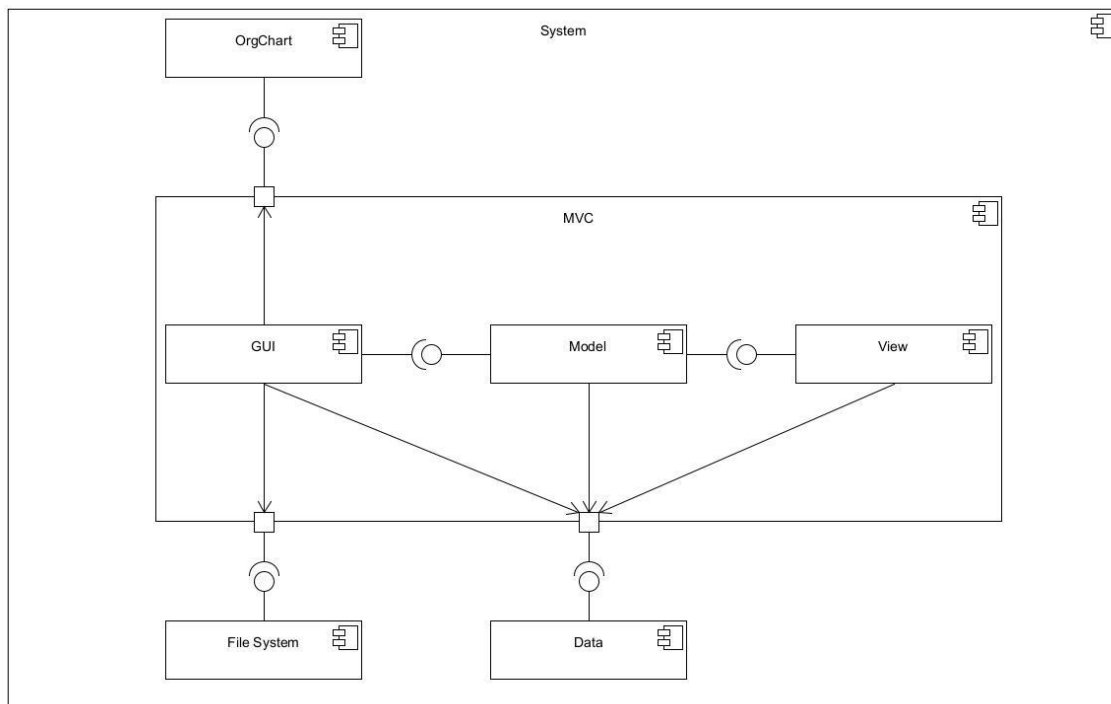
- L'utente utilizza la tastiera per inserire gli input.
- Una volta creata un'unità, non è possibile modificarla.
- Una volta aggiunto un dipendente, non è possibile modificarlo.
- Non è consentito modificare la gerarchia di un organigramma già visualizzato.

.6 Use Case Diagrams



C. Architettura Software

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

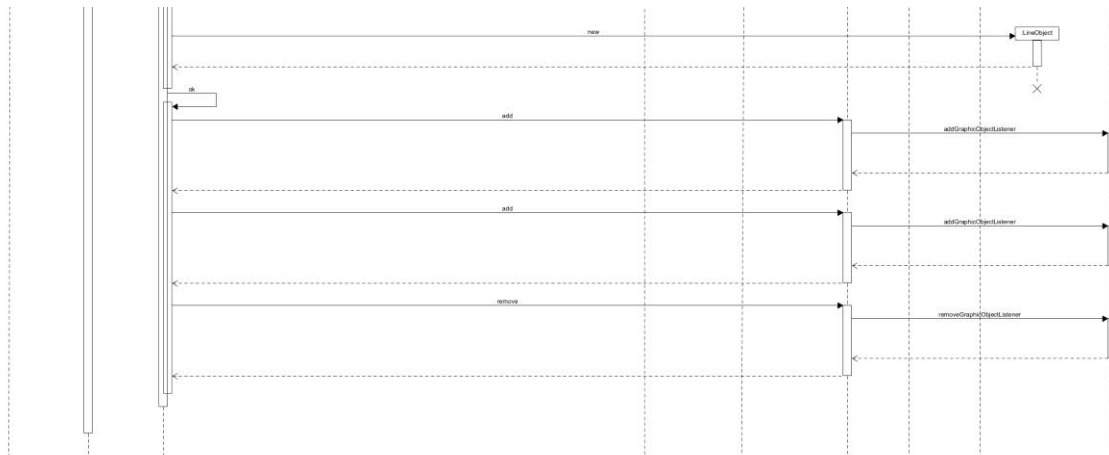


C.2 The dynamic view of the software architecture: Sequence Diagram

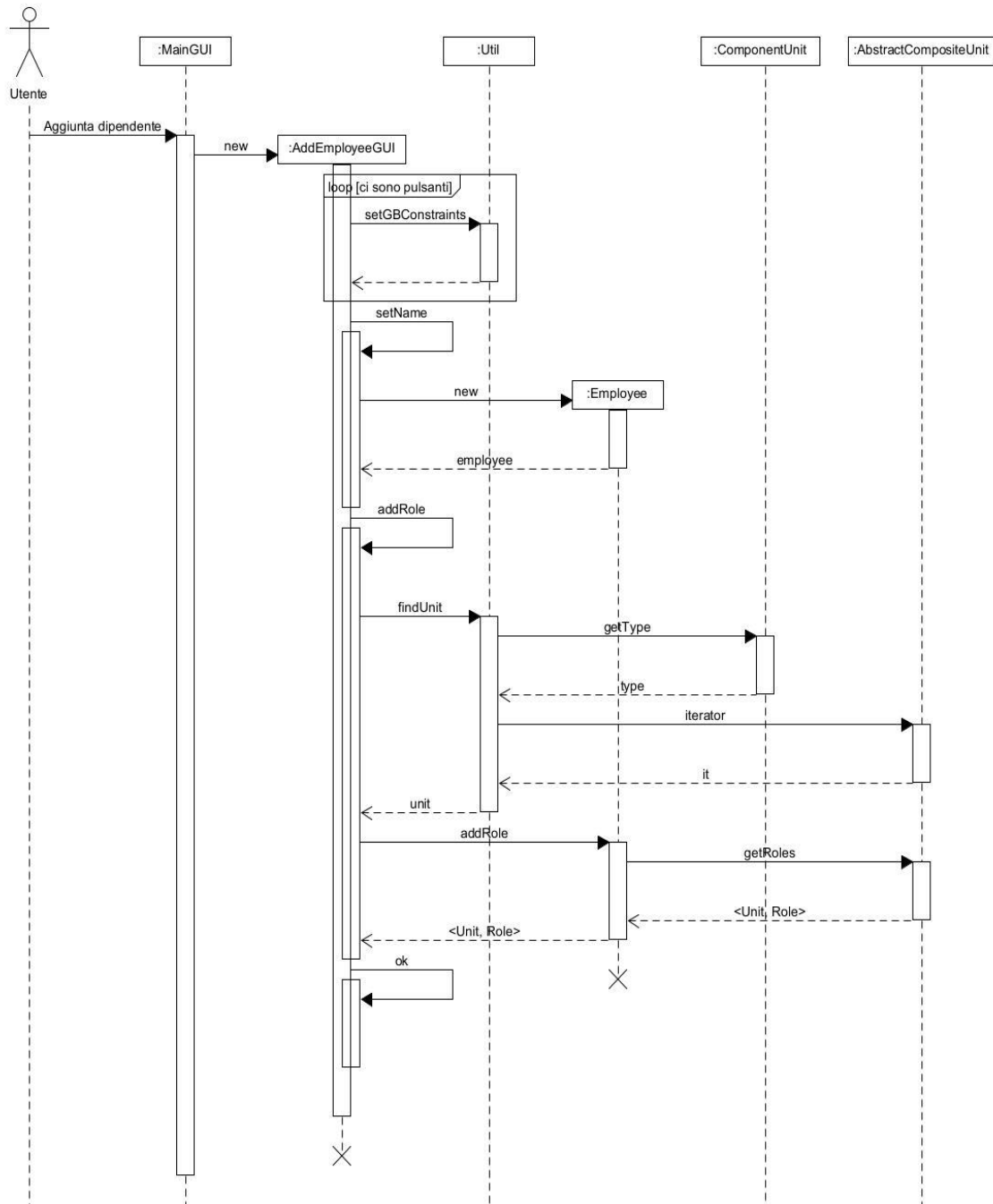
```

sequenceDiagram
    participant User
    participant MainGUI
    participant LMI
    participant AbstractGraphicObject
    participant GraphicObjectListener
    participant GraphicObjectPanel
    participant ComponentList
    participant AbstractComponentList
    participant GraphicObject

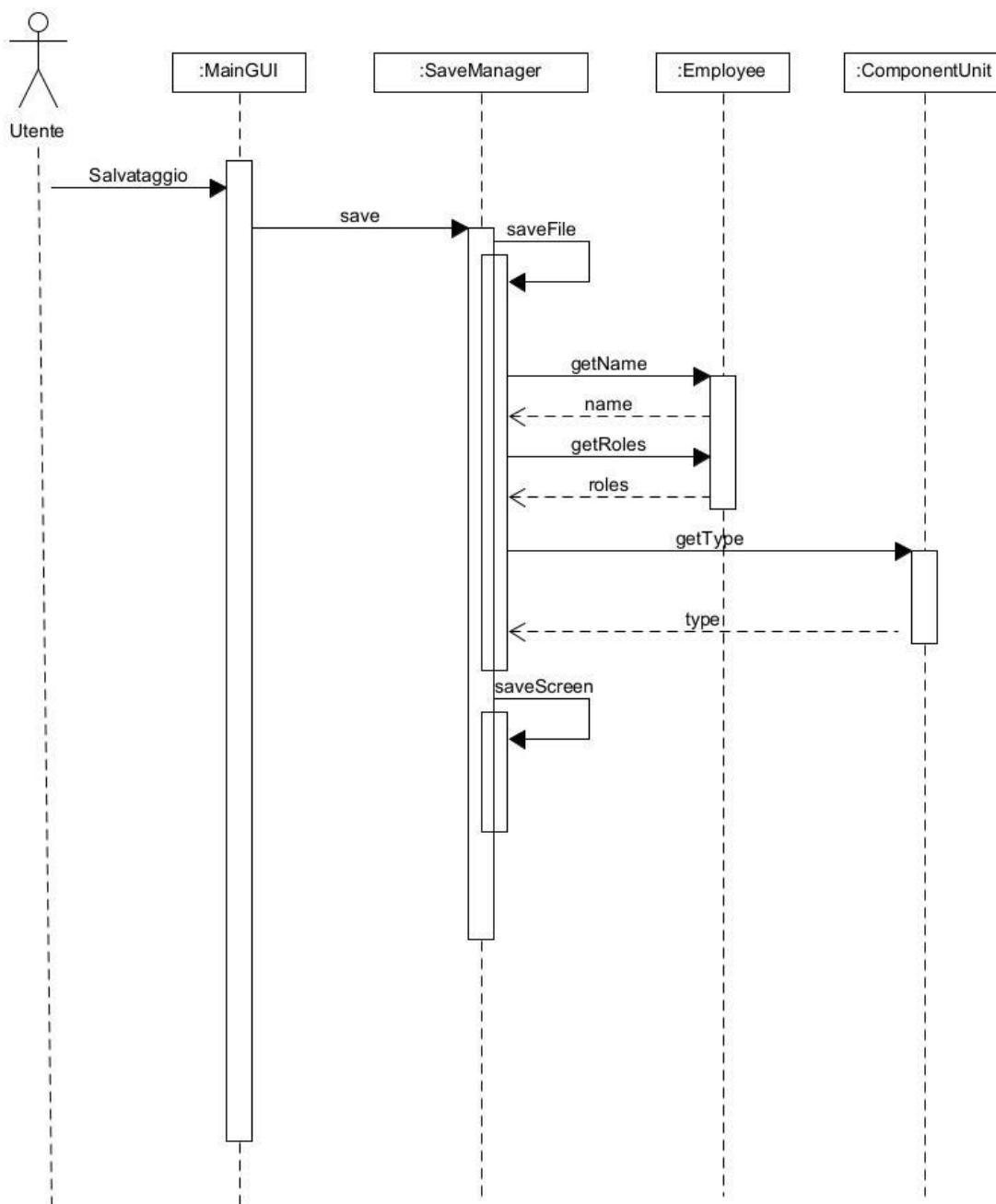
    User->>MainGUI: Create new units
    activate MainGUI
    MainGUI->>LMI: new
    activate LMI
    LMI->>AbstractGraphicObject: new (ComponentPanel)
    activate AbstractGraphicObject
    AbstractGraphicObject->>GraphicObjectListener: setGraphicListener
    deactivate AbstractGraphicObject
    LMI->>ComponentList: new
    activate ComponentList
    ComponentList->>AbstractComponentList: new
    activate AbstractComponentList
    AbstractComponentList->>GraphicObject: new
    activate GraphicObject
    GraphicObject->>GraphicObjectPanel: new
    activate GraphicObjectPanel
    GraphicObjectPanel->>GraphicObject: new
    activate GraphicObject
    GraphicObject->>GraphicObjectPanel: setUnit
    deactivate GraphicObject
    GraphicObjectPanel->>GraphicObject: new (GraphicEvent)
    activate GraphicObject
    GraphicObject->>GraphicObjectListener: notifyListeners
    activate GraphicObjectListener
    GraphicObjectListener->>GraphicObjectPanel: graphicChanged
    deactivate GraphicObjectListener
    GraphicObjectPanel->>GraphicObject: graphicChanged
    deactivate GraphicObjectPanel
    deactivate GraphicObject
    deactivate AbstractComponentList
    deactivate ComponentList
    deactivate LMI
    LMI->>AbstractGraphicObject: setType
    activate AbstractGraphicObject
    AbstractGraphicObject->>GraphicObjectListener: setType
    deactivate AbstractGraphicObject
    LMI->>AbstractGraphicObject: setUnit
    activate AbstractGraphicObject
    AbstractGraphicObject->>GraphicObjectListener: new (GraphicEvent)
    activate GraphicObject
    GraphicObject->>GraphicObjectPanel: notifyListeners
    activate GraphicObjectPanel
    GraphicObjectPanel->>GraphicObject: graphicChanged
    deactivate GraphicObjectPanel
    GraphicObject->>GraphicObjectListener: graphicChanged
    deactivate GraphicObject
    deactivate AbstractGraphicObject
    LMI->>AbstractGraphicObject: addStyle
    activate AbstractGraphicObject
    AbstractGraphicObject->>GraphicObjectListener: addStyle
    deactivate AbstractGraphicObject
    LMI->>AbstractGraphicObject: setParent
    activate AbstractGraphicObject
    AbstractGraphicObject->>GraphicObjectListener: setParent
    deactivate AbstractGraphicObject
    LMI->>AbstractGraphicObject: findUnit
    activate AbstractGraphicObject
    AbstractGraphicObject->>GraphicObjectListener: getType
    activate GraphicObjectListener
    GraphicObjectListener->>GraphicObjectPanel: type
    deactivate GraphicObjectListener
    GraphicObjectPanel->>GraphicObject: type
    deactivate GraphicObjectPanel
    GraphicObject->>GraphicObjectListener: iter
    activate GraphicObjectListener
    GraphicObjectListener->>GraphicObjectPanel: iter
    deactivate GraphicObjectListener
    GraphicObjectPanel->>GraphicObject: iter
    deactivate GraphicObjectPanel
    deactivate AbstractGraphicObject
    LMI->>AbstractGraphicObject: addChild
    activate AbstractGraphicObject
    AbstractGraphicObject->>GraphicObjectListener: addChild
    activate GraphicObjectListener
    GraphicObjectListener->>GraphicObjectPanel: addChild
    activate GraphicObjectPanel
    GraphicObjectPanel->>GraphicObject: addChild
    activate GraphicObject
    GraphicObject->>GraphicObjectPanel: setParent
    deactivate GraphicObject
    GraphicObjectPanel->>GraphicObject: setParent
    deactivate GraphicObjectPanel
    deactivate GraphicObjectListener
    deactivate AbstractGraphicObject
    deactivate LMI
    deactivate MainGUI
  
```



Aggiunta dipendente



Salvataggio

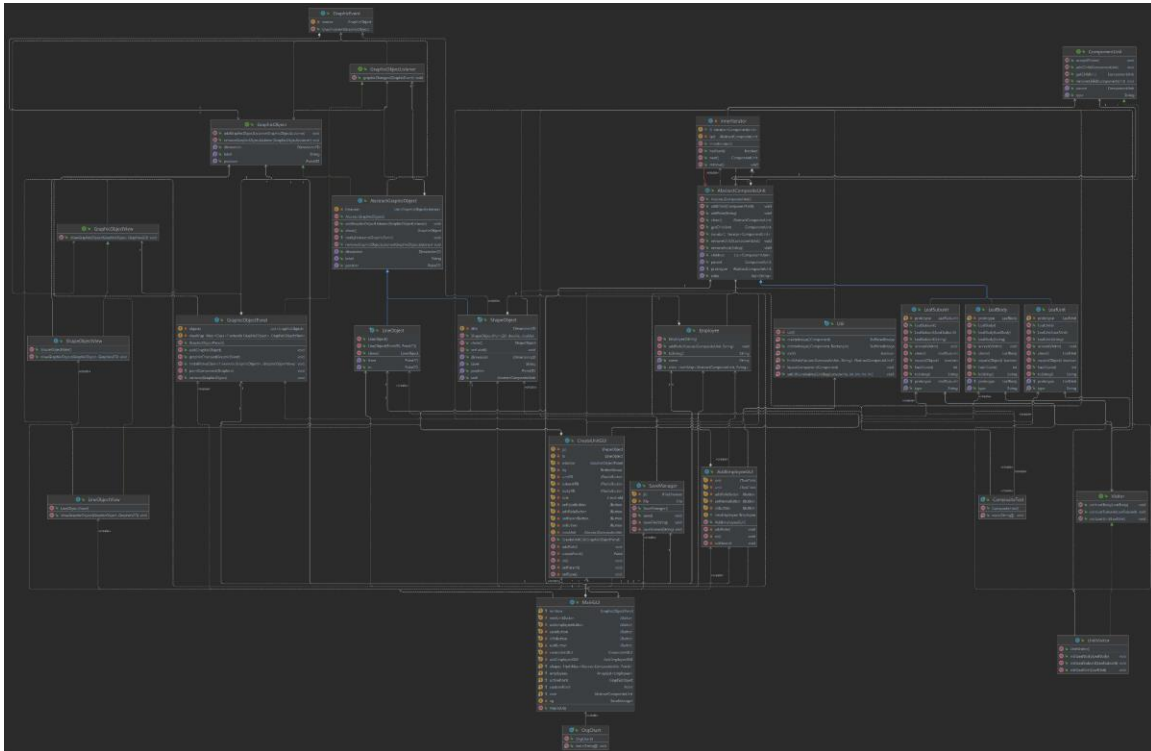


E. Scelte Progettuali (Design Decisions)

N°	SCELTA PROGETTUALE
1	<p>Architettura <i>MVC</i> con supporto al <i>design pattern Observer</i></p> <p>Per gestire al meglio l'interazione dell'utente con il sistema si è scelto di utilizzare il pattern architetturale <i>MVC (Model-View-Controller)</i>. L'applicazione è dotata di tre package: <i>gui</i>, <i>model</i> e <i>view</i>. Le classi del package <i>gui</i> ricevono i comandi dell'utente e li attuano modificando lo stato degli altri due componenti. Il package si occupa di presentare l'interfaccia grafica all'utente e di gestire operazioni come la creazione di una nuova unità, l'aggiunta di un nuovo dipendente o il salvataggio di un organigramma. Il package <i>model</i> fornisce i metodi per accedere ai dati utili all'applicazione: contiene inoltre le classi <i>LineObject</i> e <i>ShapeObject</i>, rappresentanti le relazioni gerarchiche e le entità di un organigramma. Il package <i>view</i> visualizza i dati contenuti nel <i>model</i> e si occupa dell'interazione con utenti e agenti: contiene il pannello <i>GraphicObjectPanel</i>, la "lavagna" su cui visualizzare l'organigramma e le classi vista di <i>Line</i> e <i>ShapeObject</i>. È stato inoltre implementato il pattern <i>Observer</i> per gestire il meccanismo di notifiche e cambiamento di stato: <i>GraphicObjectPanel</i> svolge il ruolo di osservatore, mentre <i>AbstractGraphicObject</i> il ruolo di <i>Subject</i>.</p>
2	<p><i>Design pattern Composite</i></p> <p>Considerando la struttura interna di un organigramma, viene naturale adottare il pattern <i>Composite</i> come la scelta migliore. Inoltre, il pattern ben si presta a poter introdurre nuovi tipi di entità da aggiungere alle unità, sottounità e organi di gestione richiesti dalla traccia. È stato creato un package contenente le interfacce e le classi essenziali all'implementazione del pattern: "<i>ComponentUnit</i>" dichiara l'interfaccia per gli oggetti componibili; "<i>AbstractCompositeUnit</i>" definisce il comportamento dei componenti che hanno figli, inoltre memorizza i componenti figli e implementa i relativi metodi introdotti dall'interfaccia "<i>ComponentUnit</i>". Le classi "<i>Leaf</i>" rappresentano gli oggetti primitivi che non hanno figli.</p>
3	<p><i>Design pattern Iterator</i></p> <p>Dalla motivazione del pattern <i>Iterator</i>: "fornisce un'interfaccia di accesso sequenziale agli elementi di un oggetto composito (ad esempio una collezione di oggetti) senza esporne la struttura interna". Il pattern, in sinergia con il <i>Composite</i> ben si presta a scorrere gli elementi della classe "<i>AbstractCompositeUnit</i>", all'interno della quale è stato implementato. Inoltre, per la sua versatilità, viene</p>

	usato nel metodo di supporto <i>findUnit</i> per verificare l'esistenza e restituire una determinata entità a partire da una radice.
4	<p><i>Design pattern Visitor</i></p> <p>Il design pattern “<i>Visitor</i>” rappresenta un'operazione da eseguire sugli oggetti di una struttura. Anche questo pattern si presta a una sinergia con il pattern <i>Composite</i>: come detto sopra l'operazione può essere eseguita su un intero albero composto. Inoltre, esso consente di gestire specifiche operazioni solo su determinate entità, capacità che lo rende adatto a svolgere compiti mirati. È stato implementato nel package <i>visitor</i>, L'interfaccia "<i>Visitor</i>" dichiara un metodo di visita per ogni classe concreta della struttura. "<i>UnitVisitor</i>" implementa i metodi definiti da <i>Visitor</i> e consente di personalizzarli. Le classi “<i>Leaf</i>” del package <i>composite</i> implementano il metodo <i>accept()</i> che riceve un <i>Visitor</i>.</p>
5	<p><i>Design pattern Prototype</i></p> <p>Il design pattern creazionale “<i>Prototype</i>” consente di creare nuovi oggetti clonando un oggetto iniziale, chiamato prototipo. È stato implementato in sinergia con il pattern <i>Composite</i>, in quanto l'applicazione del <i>Prototype</i> consente di clonare strutture complesse invece di ricostruirle da zero, prestandosi bene a copiare complessi organigrammi. Inoltre, esso è stato applicato alle classi del package <i>model</i>, per consentire facilmente la copia.</p>

F. Progettazione di Basso Livello



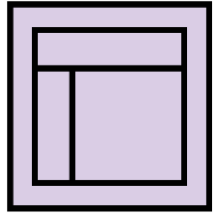
G. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs)

Il sistema software “*Organigramma Aziendale*” soddisfa i requisiti e le specifiche richieste.

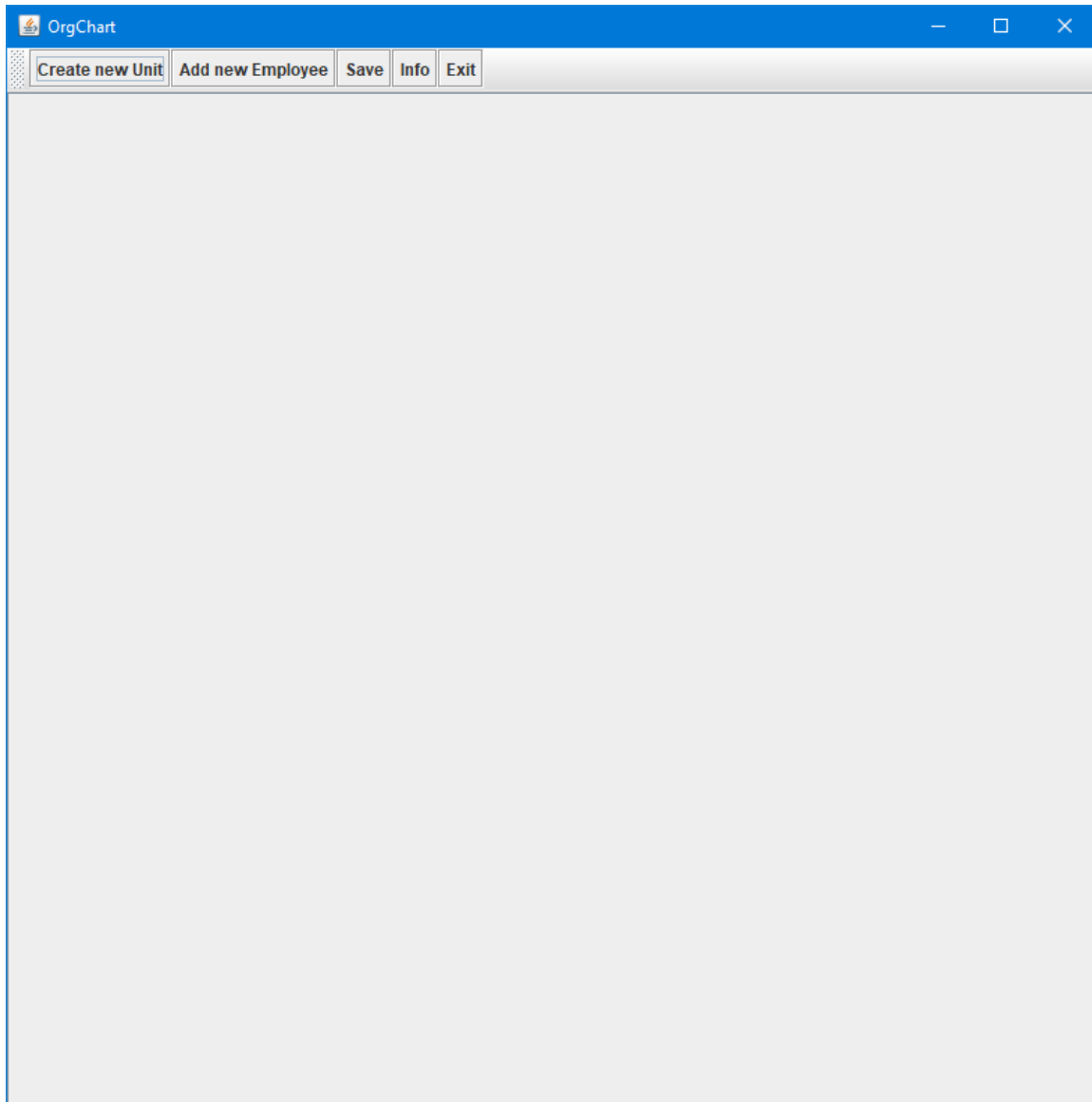
L'applicazione si presenta all'utente con un'interfaccia grafica semplice e intuitiva (#5). Essa consente all'utente di usufruire di tutti i servizi offerti dal sistema. La possibilità di rappresentare graficamente l'organigramma e di poterlo gestire è data dall'adozione del pattern architetturale *MVC*. I package *model* e *view* si interfacciano con il controllo che avviene nel package *gui*. La creazione di una nuova entità (servizio #1) e la costruzione delle relazioni gerarchiche avviene tramite il pattern *Composite*: l'utente seleziona una categoria di entità, che può essere una *LeafUnit* (tipologie di unità come i dipartimenti), una *LeafSubunit* (sottounità organizzative come i gruppi di lavoro) o una *LeafBody* (organi di gestione). Eventualmente, è possibile aggiungere più categorie grazie alla flessibilità del pattern *Composite*. All'entità così definita viene assegnato dall'utente un tipo (una denominazione) e dei ruoli disponibili. È anche possibile, se non si sta rappresentando la prima entità dell'organigramma, specificare l'entità genitore per instaurare delle relazioni gerarchiche: il pattern *Iterator* consente di scorrere l'organigramma e di individuare l'entità genitore a partire dalla radice. Alla flessibilità del pattern *Composite* si affianca quella del pattern *Visitor*: nell'ottica dell'evolvibilità del software e seguendo il principio dell'anticipazione del cambiamento, il pattern *Visitor* consente di definire nuove operazioni (ad esempio sarebbe possibile aggiungere dei ruoli particolari solo a determinate categorie di entità). L'aggiunta di un nuovo dipendente (#2) si lega alla classe *Employee*. Un dipendente possiede un nome e un ruolo che è in grado di ricoprire, inoltre è in grado di candidarsi a un'unità dell'organigramma (#3). Come già detto, un'unità ha determinati ruoli disponibili: se il candidato non rispetta il requisito (non ha un ruolo abbinato al ruolo disponibile nell'unità selezionata), non sarà possibile definire l'associazione. Il salvataggio di un organigramma (servizio #5) è selezionabile dalla finestra principale

dell'applicazione. Esso fa uso della classe *SaveManager*, che fornisce i metodi necessari a effettuare il servizio. L'utente può selezionare un qualsiasi formato per il file di salvataggio: il sistema effettuerà un salvataggio dell'organigramma che sta gestendo in forma testuale e un salvataggio di una rappresentazione grafica dello stesso in formato .jpeg. Sempre dalla finestra principale è possibile selezionare un pulsante per ottenere informazioni sul sistema e un pulsante per uscire in qualunque momento. Qualora si tentasse di uscire, il sistema ricorderà all'utente di salvare.

Appendix. Prototype

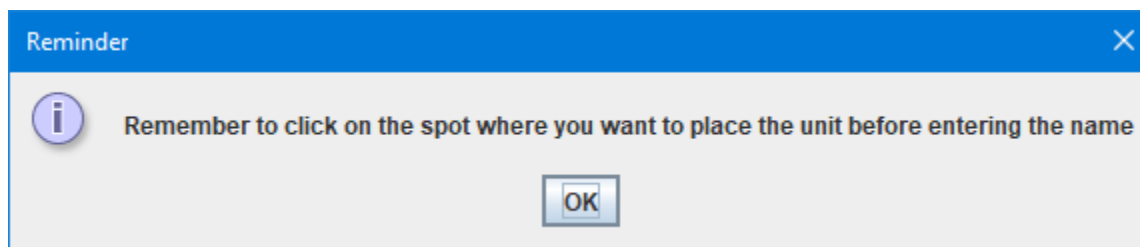


Il sistema software si presenta all'utente finale in veste grafica:

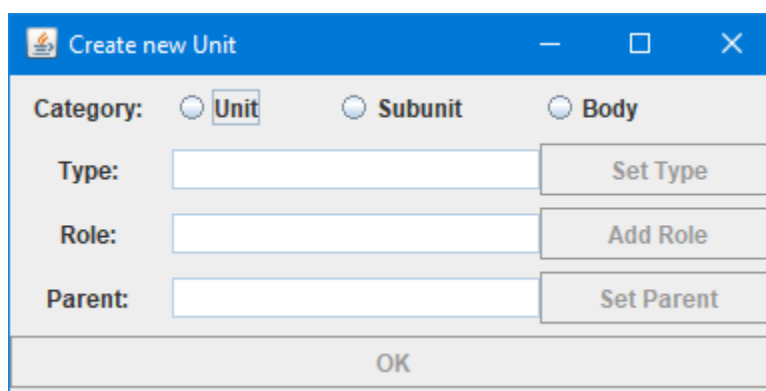


I vari pulsanti consentono all'utente di usufruire dei servizi che l'applicativo può offrire.

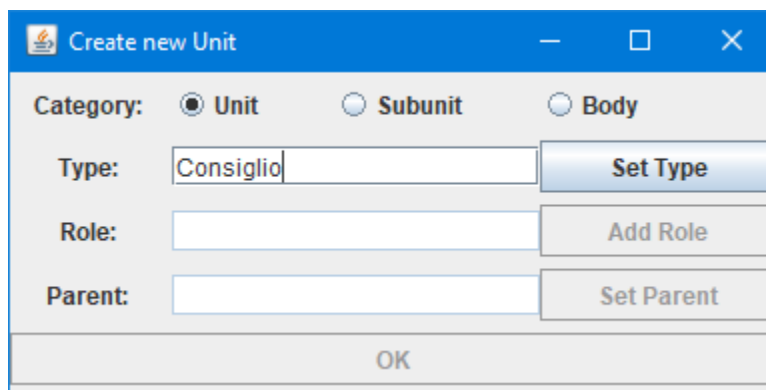
Il primo pulsante, “*Create new Unit*” consente di accedere al menù di creazione di una entità:



Il promemoria ricorda all'utente di selezionare un punto prima di creare l'entità.

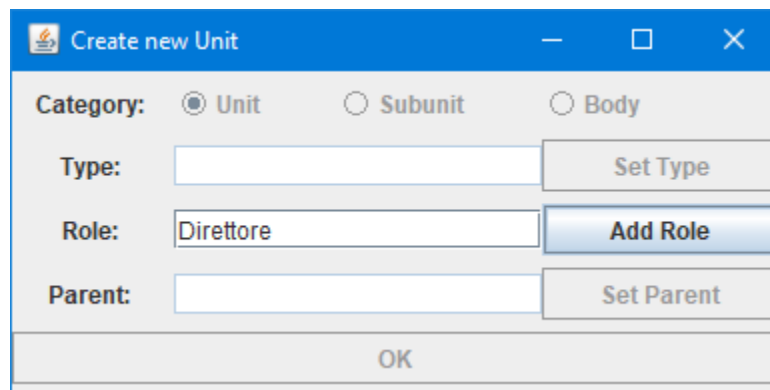
A "Create new Unit" dialog box with a blue title bar. It features three radio buttons for "Category": "Unit" (selected), "Subunit", and "Body". Below these are three input fields: "Type:", "Role:", and "Parent:". To the right of each field is a button: "Set Type", "Add Role", and "Set Parent". At the bottom is a large "OK" button.

Ecco come appare la finestra “*Create new Unit*”. I primi tre pulsanti permettono all'utente di selezionare la categoria di entità che desiderano creare. Appena selezionata, il campo e il bottone “*Set Type*” saranno disponibili.

The "Create new Unit" dialog box is shown with the "Unit" category selected. The "Type:" field now contains the text "Consiglio" and the "Set Type" button is highlighted in blue. The "Role:" and "Parent:" fields are empty, and their respective buttons are disabled. The "OK" button remains at the bottom.

L'utente inserisce i dati e preme sul pulsante:

1. Se l'entità è la radice dell'organigramma, saranno sbloccati solo il campo e il bottone “*Add Role*”, dove l'utente può inserire i ruoli che desidera assegnare all'unità.



Create new Unit

Category: ☒ Unit ☐ Subunit ☐ Body

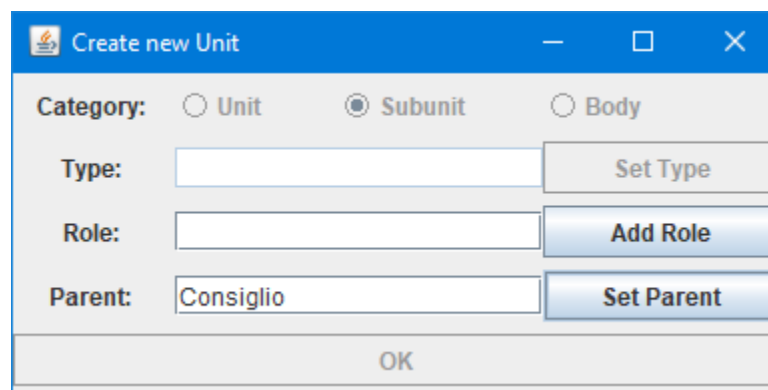
Type: Set Type

Role: Add Role

Parent: Set Parent

OK

2. Se l'entità è un nodo non radice dell'organigramma, anche il campo e il bottone "Set Parent" saranno sbloccati: l'utente può inserire l'entità genitore.



Create new Unit

Category: ☐ Unit ☒ Subunit ☐ Body

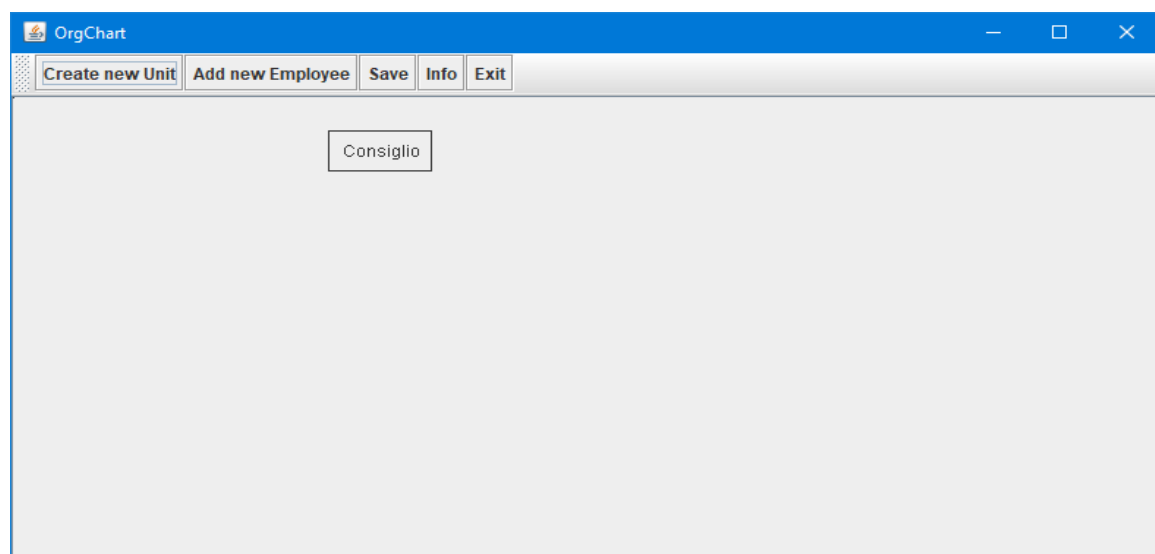
Type: Set Type

Role: Add Role

Parent: Set Parent

OK

Dando conferma l'unità sarà creata e il risultato sarà:





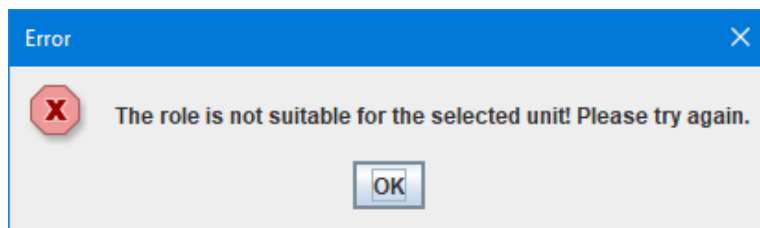
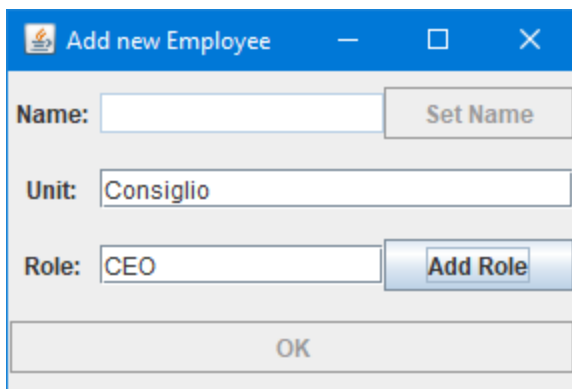
Il secondo pulsante “*Add new Employee*” consente di accedere al menù di aggiunta di un dipendente:

The image shows two side-by-side screenshots of the 'Add new Employee' dialog box. The left screenshot shows the dialog with empty input fields for Name, Unit, and Role. The right screenshot shows the dialog with 'Mario Verdi' entered in the Name field, and the Unit and Role fields are now enabled.

Inserendo il nome del dipendente, vengono sbloccati i due campi e il bottone “*Add Role*”:

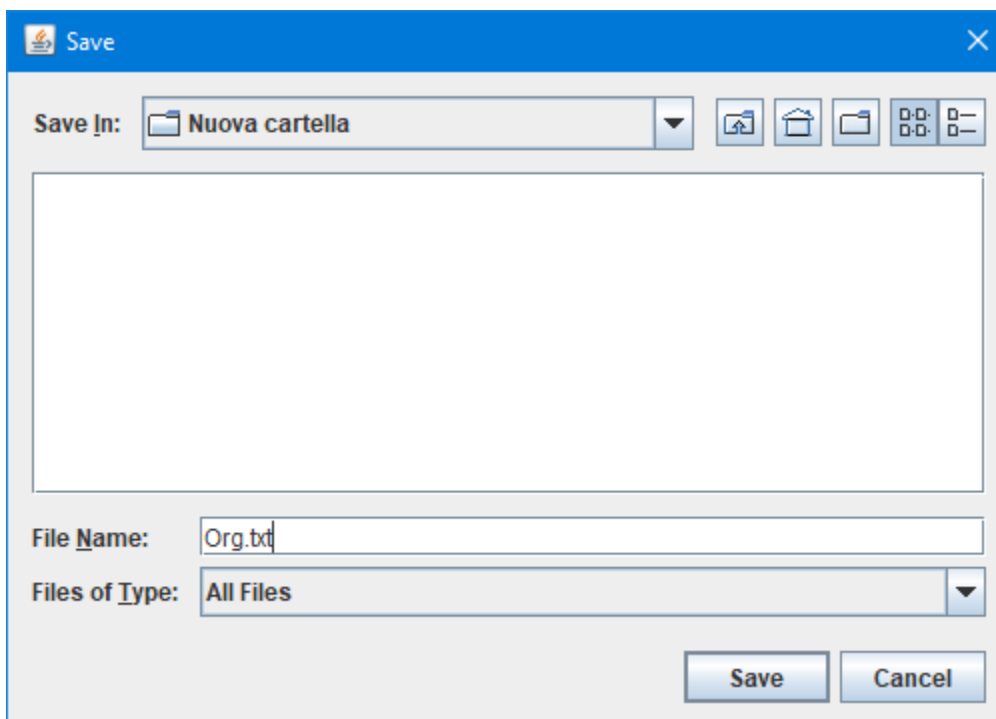
The screenshot shows the 'Add new Employee' dialog box with the following values: Name: Mario Verdi, Unit: Consiglio, and Role: Direttore. The 'Add Role' button is now enabled.

Una volta riempiti, se il ruolo è disponibile per quella determinata unità, il dipendente potrà essere associato. Nel caso dello screen il responso sarà positivo, dato che in precedenza è stata creata un'unità "*Consiglio*" con "*Direttore*" come ruolo disponibile. Se invece provassimo a inserire un ruolo errato:



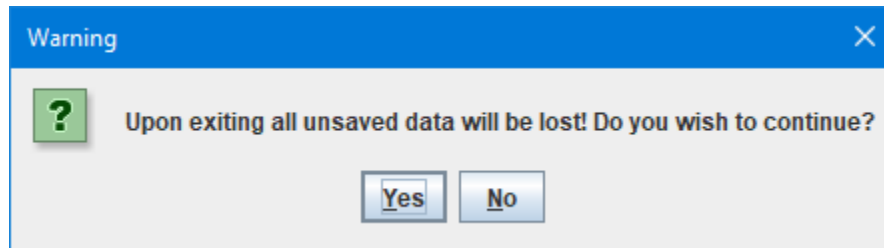
Otteniamo un errore! Il ruolo non è adatto all'unità selezionata.

Il terzo pulsante "Save" consente di accedere al menù di salvataggio:



In questo caso, l'organigramma è stato salvato come "*Org.txt*" e i suoi dati sono consultabili nel documento di testo o graficamente nel file .jpeg.

Il pulsante “*Exit*” consente di uscire dall’applicazione, ricordando di salvare i dati:



Per concludere, ecco una rappresentazione dell’organigramma proposto come esempio nella traccia:

