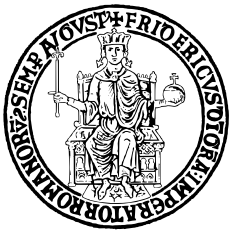
Università degli Studi di Napoli Federico II



Scuola Politecnica e delle Scienze di Base

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Corso di Laurea Magistrale in Informatica

Progetto d'esame di Object Orientation

Progettazione e sviluppo di un applicativo in Java per la gestione di conferenze scientifiche

Relatore
Professore Sergio Di Martino

Candidati
Antonio Caporaso
matr: N86003458
Giorgio Di Fusco
matr: N86004389

Anno Accademico 2022-2023

Questa pagina è stata lasciata intenzionalmente vuota.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 Definizione del problema

Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione di **conferenze scientifiche**.

Ogni conferenza ha una data di inizio e di fine, una collocazione (sede, indirizzo), uno o più enti che la organizzano, degli sponsor (che coprono in parte le spese), una descrizione, ed un gruppo di organizzatori, che può essere distinto in comitato scientifico e comitato locale (che si occupa cioè della logistica). Di ognuno degli organizzatori, così come di tutti i partecipanti, si riportano titolo, nome, cognome, email ed istituzione di afferenza.

Ogni conferenza può avere una o più sessioni, anche in parallelo fra loro. Ogni sessione ha una locazione all'interno della sede. Per ogni sessione c'è un programma, che prevede la presenza di un coordinatore (chair) che gestisce la sessione, ed eventualmente di un keynote speaker (un partecipante di particolare rilievo invitato dagli organizzatori). Ogni sessione avrà quindi una successione di interventi ad orari predefiniti e di specifici partecipanti. Per ogni intervento si conserva un abstract (un breve testo in cui viene spiegato il contenuto del lavoro presentato).

Si deve poter considerare la presenza di spazi di intervallo (coffee breaks, pranzo) ma anche la presenza di eventi sociali (cene, gite, etc).

1.2 Symposium: un applicativo per la gestione di conferenze scientifiche

Symposium è un applicativo per la gestione di conferenze scientifiche, sviluppato in JavaFX e basato su PostgreSQL15. Il sistema è progettato per fornire una piattaforma completa e intuitiva per gli organizzatori delle conferenze, consentendo loro di pianificare, gestire e vsualizzare le varie attività svolte durante le conferenze.

1.2.1 Caratteristiche principali di Symposium

All'interno di *Symposium* è possibile:

1. **Creare nuove conferenze:** Ogni conferenza viene registrata nel sistema con dettagli come la data di inizio e fine, la collocazione (sede e indirizzo), gli enti organizzatori, gli sponsor coinvolti e ogni eventuale sessione prevista durante la conferenza. Una descrizione dell'evento sarà disponibile per fornire informazioni generali.

2. **Gestione dei comitati:** Per ogni conferenza, sono registrati i dettagli dei membri del comitato scientifico e del comitato locale, che si occupano rispettivamente degli aspetti scientifici e logistici dell'evento.

3. **Gestione della conferenza:** Per ogni conferenza è possibile modificare i suoi dettagli generali (quali il titolo, l'inizio, la fine e la sede), modificare gli enti organizzatori e le varie sponsorizzazioni oppure slittare la conferenza.

4. **Gestione delle sessioni:** Per ogni sessione, è possibile creare un programma dettagliato con gli orari dei vari punti in programma.

1.2.2 Interfaccia Utente

L'interfaccia utente di Symposium sarà realizzata utilizzando JavaFX, fornendo un'esperienza utente intuitiva e piacevole. Gli organizzatori possono accedere alla piattaforma per registrarsi e visualizzare i dettagli delle conferenze già presenti nel database. Ogni utente avrà la possibilità di gestire le proprie conferenze, specificare il programma delle sessioni, specificare la nomina dei comitati e altro ancora.

In conclusione, Symposium è un sistema informativo embrionale per la gestione di conferenze scientifiche che offre funzionalità complete e una piattaforma intuitiva per organizzatori.

Capitolo 2

Progettazione del software: CRC card e class diagrams

2.1 Analisi delle entità

Le entità che possono essere individuate nel problema sono:

1. **Conferenza** : per le conferenze delle quali si vuole poter gestire le informazioni. Di ogni conferenza si conservano il *nome*, l'*inizio* e la *fine* e una *descrizione*.
2. **Ente**: per gli enti che organizzano le conferenze scientifiche. Di ogni ente si conserva il *nome* e la *sigla*.
3. **Sponsor** : per gli sponsor che coprono le spese della conferenza. Di ogni sponsor si conserva il *nome*.
4. **Comitato** : per i gruppi di organizzatori che si occupano della gestione della conferenza. Si distinguono in comitati *scientifici* e *locali*.
5. **Organizzatore** : per i membri dei comitati. Di ogni organizzatore si riportano *titolo, nome, cognome, email* ed *istituzione di afferenza*.
6. **Sede**: per descrivere il luogo dove si tengono le varie conferenze. Di ogni sede si conservano il *nome*, l'*indirizzo* e la *città*.
7. **Sala**: per tenere traccia dell'ubicazione delle varie sessioni. Di ogni sala si conserva il *nome della sala* e la sua *capacità*.
8. **Sessione**: per rappresentare le sessioni di una conferenza. Per ogni sessione si riporta il *titolo*, un *coordinatore*, data e orario d'*inizio* e di *fine*.
9. **Programma** : per il programma di ciascuna sessione. Ogni programma specifica la presenza di un *keynote speaker*, ovvero un partecipante di rilievo.
10. **Intervento** : per i vari interventi di una sessione. Per ogni intervento si conserva un *abstract* e l'*orario* dello stesso.
11. **Speaker**:per descrivere chi effettua un intervento.
12. **Partecipante**: per i partecipanti delle varie sessioni. Ogni partecipante ha gli stessi attributi degli organizzatori
13. **Intervallo**: per descrivere i vari intervalli presenti all'interno di una sessione. Questi possono essere di tue tipologie:
14. *coffee break* oppure dei *pranzi*. Per ogni intervallo si riporta l'*orario*.

15. **Evento sociale**: per i vari eventi sociali previsti all'interno di una sessione. Questi possono essere di varia natura. Come per gli intervalli se ne riporta l'*orario*.

16. **Utente**: per i vari utenti che creano le conferenze all'interno di un applicativo.

2.2 L'architettura

La realizzazione di Symposium ha richiesto un'approfondita considerazione dell'architettura software. Al fine di garantire una struttura modulare, manutenibile ed estensibile, è stato scelto di utilizzare JavaFX come framework di sviluppo per l'interfaccia utente e di suddividere la gestione delle funzionalità in varie singole classi controller.

Per mantenere un codice ben strutturato e facilmente manutenibile, le varie classi di Symposium sono state suddivisi in package in base alle loro responsabilità. Questa organizzazione modulare permette di isolare le diverse funzionalità dell'applicativo, semplificando lo sviluppo parallelo e la manutenzione continua.

I package dell'intero progetto sono stati progettati seguendo le linee guida della progettazione software quali il pattern MVC, la coesione e il principio di singola responsabilità. Ciò contribuisce a garantire una chiara separazione delle responsabilità all'interno dell'applicativo e facilita la comprensione del codice:

1. `Model`
    (a) `DAO`
    (b) `DbConfig`
    (c) `Entities`
    (d) `Utilities`
2. `View`
    (a) `FXML`
        i. `Create`
        ii. `Edit`
        iii. `View`
        iv. `Stats`
    (b) `CSS`
3. `Controller`
    (a) `Create`
    (b) `Edit`
    (c) `View`
    (d) `Stats`
4. `Exceptions`

All'interno del package `View` sono presenti tutti i file `.fxml` che descrivono le interfacce grafiche dell'applicazione e i file `.css` utilizzati per la loro personalizzazione, mentre all'interno del package `Controller` sono presenti le classi che implementano i controller delle interfacce grafiche.

All'interno del package `Model` sono presenti vari packages che implementano il modello del nostro progetto come presentato nel Diagramma 2.1. Nel sub-package `DAO` sono state inserite tutte le classi utilizzate per implementare il pattern DAO, mentre nel sub-package `DbConfig` è presente la classe per la configurazione del database. Infine, nel sub-package `Utilities` sono presenti le classi che ci sono state necessarie per la gestione di molteplici istanze delle classi di dominio.

2.3 Le CRC Cards

2.3.1 Il package Model

2.3.1.1 Model.Entities

Class name	Conferenza
Superclass	Nessuna
Subclasses	Nessuna
Responsabilities	Riunione tematica di rappresentanti di vari enti
Collaborations	Sessione Comitato Utente Sede Ente

Class name	EventoSociale
Superclass	ActivityModel
Subclasses	Nessuna
Responsabilities	Momento dedicato agli invitati della conferenza
Collaborations	Nessuna

Class name	Intervento
Superclass	ActivityModel
Subclasses	Nessuna
Responsabilities	Momento in cui un partecipante prende la parola
Collaborations	Nessuna

Class name	Programma
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Specifica i punti di una sessione
Collaborations	Sessione Intervento Intervallo EventoSociale

Class name	Intervallo
Superclass	ActivityModel
Subclasses	Nessuna
Responsabilità	Breve pausa all'interno di una sessione
Collaborazioni	Nessuna

Nella Figura 2.1 è presente il class diagram rappresentante il modello di dominio.

Class name	Sede
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Luogo in cui si svolge una conferenza
Collaborations	Conferenza Sala Indirizzo

Class name	Sala
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Luogo in cui si svolge una sessione
Collaborations	Sessione Sede

Class name	Sessione
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Parte di una conferenza
Collaborations	Conferenza Sala Programma Organizzatore

Class name	Comitato
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Insieme di organizzatori che si occupa della logistica e dell'organizzazione
Collaborations	Conferenza Organizzatore

Class name	Ente
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Istituzione che organizza una conferenza
Collaborations	Conferenza Speaker Partecipante Organizzatore

Class name	Indirizzo
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Luogo dove è presente la conferenza
Collaborations	Sede

Class name	Organizzatore
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Membro di un ente
Collaborations	Ente Sessione Comitato

Class name	Sponsor
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Azienda che sponsorizza una conferenza
Collaborations	Sponsorizzazione

Class name	Sponsorizzazione
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Sponsorizzazione di uno sponsor
Collaborations	Conferenza Sponsor

Class name	Speaker
Superclass	Nessuna
Subclasses	Nessuna
Responsabilità	Specifica i punti di una sessione
Collaborations	Intervento

2.3.1.2 Model.DAO

Nell'architettura dell'applicativo Symposium, le classi DAO (Data Access Object) svolgono un ruolo fondamentale nella gestione dell'accesso ai dati e nell'interazione con il sistema di archiviazione dei dati sottostante. Le classi DAO fungono da ponte tra il livello di business dell'applicativo e il database o qualsiasi altro meccanismo di persistenza utilizzato per memorizzare e recuperare i dati.

Le classi DAO sono state introdotte per affrontare diverse esigenze critiche all'interno dell'applicativo Symposium:

1. **Separazione delle responsabilità:** le classi DAO separano chiaramente la logica di accesso ai dati dalla logica di business dell'applicativo.
2. **Astrazione dal dettaglio di archiviazione:** le classi DAO nascondono i dettagli tecnici legati al meccanismo di persistenza sottostante. Questo significa che le modifiche alla tecnologia di archiviazione dei dati, come il passaggio da un database SQL a un database NoSQL, possono essere gestite senza impattare significativamente il codice di business.

Class name	ComitatoDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realative ai comitati.
Collaborations	Comitato Utente EnteDao DbConnection

Class name	ConferenzaDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realative alle conferenze.
Collaborations	Conferenza Utente Sede UtenteDao SedeDao ComitatoDao DbConnection

Class name	EnteDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realative all'Ente.
Collaborations	Conferenza Ente DbConnection

Class name	IndirizzoDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Indirizzi.
Collaborations	Indirizzo DbConnection

Class name	IntervalloDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Intervalli.
Collaborations	Intervallo Programma DbConnection

Class name	EventoSocialeDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Eventi Sociali.
Collaborations	EventoSociale ProgrammaDao DbConnection

Class name	ProgrammaDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi ai Programmi.
Collaborations	Programma Sessione SpeakerDao DbConnection

Class name	InterventoDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Interventi.
Collaborations	Intervento Programma Stats SpeakerDao DbConnection

Class name	OrganizzatoreDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Organizzatori.
Collaborations	Organizzatore EnteDao DbConnection

Class name	SalaDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi alle Sale.
Collaborations	Sala SedeDao DbConnection

Class name	SedeDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi alle Sedi.
Collaborations	Sede IndirizzoDao DbConnection

Class name	SessioneDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi alle Sessioni.
Collaborations	Sessione ConferenzaDao OrganizzatoreDao SalaDao ProgrammaDao DbConnection

Class name	SpeakerDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Speaker.
Collaborations	Speaker EnteDao DbConnection

Class name	SponsorDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Sponsor.
Collaborations	Sponsor DbConnection

Class name	SponsorizzazioneDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi alle Sponsorizzazioni.
Collaborations	Sponsorizzazione Conferenza SponsorDao DbConnection

Class name	UtenteDao
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Comunicare le oprazioni da eseguire al database realativi agli Utenti.
Collaborations	Utente UtenteDao DbConnection

2.3.1.3 Model.DbConfig

2.3.1.4 Model.Utilities

Le classi del dominio in Symposium rappresentano oggetti significativi all'interno del nostro sistema, ad esempio le conferenze, gli utenti e altri elementi chiave. Tuttavia, per garantire una gestione efficiente delle liste di questi oggetti e monitorare i cambiamenti in tempo reale, abbiamo introdotto le classi utilities.

Le classi utilities sono state introdotte per affrontare le seguenti esigenze:

1. **Monitoraggio delle modifiche:** attraverso queste classi abbiamo costruito liste dinamiche che possono essere osservate per rilevare automaticamente le modifiche apportate agli oggetti. Questo è particolarmente utile quando si gestiscono elenchi di oggetti che devono essere aggiornati in risposta a interazioni utente o modifiche nel sistema.

2. **Incapsulazione e modularità:** le classi utilities contribuiscono all'incapsulazione delle operazioni di gestione delle liste. Questo promuove una buona pratica di progettazione del software, consentendo di isolare la logica di gestione delle liste in classi dedicate, riducendo la complessità all'interno delle classi del dominio.

3. **Riutilizzo del codice:** la creazione di classi utilities per operazioni comuni di gestione delle liste consente il riutilizzo del codice in più parti dell'applicativo. Ciò porta a una maggiore coerenza e manutenibilità del codice.

Class name	ActivityModel
Superclass	Nessuna
Subclasses	Intervallo Intervento EventoSociale
Responsability	Classe astratta per la costruzione delle viste sul programma
Collaborations	Nessuna

Class name	Conferenze
Superclass	Nessuna
Subclasses	ConferenzeUtente
Responsability	Insieme delle conferenze presente nel sistema
Collaborations	Nessuna

Class name	ConferenzeUtente
Superclass	Conferenze
Subclasses	Nessuna
Responsability	Insieme delle conferenze create da un utente
Collaborations	Nessuna

Class name	Enti
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Insieme delle istituzioni presenti nel sistema
Collaborations	Nessuna

Class name	Sale
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Insieme delle sale disponibili presenti in una sede
Collaborations	Sede

Class name	Sedi
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Insieme delle sedi presenti nel sistema
Collaborations	Nessuna

Class name	Speakers
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Insieme degli speaker presenti nel sistema
Collaborations	Nessuna

Class name	Sponsors
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Insieme degli sponsor presenti nel sistema
Collaborations	Nessuna

Class name	Stats
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Classe per la costruzione dei grafici per le statistiche
Collaborations	Nessuna

2.3.2 Il package `Controller`

Nel contesto dello sviluppo di applicazioni JavaFX come Symposium, i controller svolgono un ruolo centrale nell'architettura dell'applicazione. I controller fungono da punto di collegamento tra l'interfaccia utente (View) e dati sottostante (Model), ma anche con le classi di utilities e le classi DAO. Questa relazione ben strutturata tra controller e altre componenti è essenziale per la realizzazione di applicazioni robuste ed estensibili.

I controller, infatti, gestiscono la business logic dell'intera applicazione essendo responsabili di:

1. **Gestire l'interfaccia utente**: i controller gestiscono gli elementi dell'interfaccia utente, come finestre, pulsanti, campi di testo e altro ancora. Rispondono agli eventi generati dagli utenti, come clic del mouse o pressione dei tasti, e reagiscono modificando la vista o invocando operazioni sul Model.
2. **Comunicare con il modello:** i controller accedono alle classi di modello per ottenere o modificare dati. Ad esempio, quando un utente inserisce dati in un campo di input, il controller può recuperare questi dati dal Model e inviarli per l'elaborazione.
3. **Gestire la logica di business:** i controller implementano la logica di business dell'applicazione. Questa logica può includere la validazione dei dati, il calcolo di risultati o qualsiasi altra elaborazione necessaria.

2.3.2.1 Controller.Edit

I controller presenti nel package `Controller.Edit` si occupano della gestione e modifica di tutto ciò che riguarda una conferenza. Dalla gestione della durata, alla gestione delle singole sessioni previste.

Class name	AddSessione_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Finestra per l'inserimento di una sessione
Collaborations	Sessione Sale Conferenza

Class name	ChooseKeynote_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Finestra di selezione del keynote
Collaborations	Programma

Class name	EditKeynote_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Finestra per la modifica del keynote di una sessione
Collaborations	Sessione Programma

Class name	ModificaConferenza_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Visualizzazione dettagli conferenza
Collaborations	Conferenza

Class name	ModificaConferenze_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Visualizzazioni conferenze dell'utente
Collaborations	ConferenzeUtente

Class name	ModificaDettagliConferenza_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Gestione informazioni conferenza
Collaborations	Conferenza Sedi

2.3.2.2 Controller.Create

2.3.2.3 Controller.Login

2.3.2.4 Controller.Stats

Una delle feature di Symposium è la visualizzazione delle statistiche. Il sistema infatti fornisce all'utente un resoconto delle conferenze organizzate, delle istituzioni ed aziende coinvolte nonché un calcolo su base mensile ed annuale del tasso di appartenenza dei vari speaker che partecipano alle varie conferenze.

Class name	ModificaDettagliSessione_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Modifica della durata, locazione e coordinatore di una sessione
Collaborations	Sessione Sala Conferenza

Class name	ModificaEntiOrganizzatori_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Gestione delle istituzioni organizzatrici
Collaborations	Enti Conferenza

Class name	ModificaProgrammaSessione_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Gestione dei punti presenti nel programma
Collaborations	Programma Sessione

Class name	ModificaSessione_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Gestione dei dettagli e del programma della sessione
Collaborations	Conferenza Programma

Class name	ModificaSessioni_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Finestra per aggiunta, rimozione e modifica delle sessioni presenti in una conferenza
Collaborations	Conferenza

Class name	ModificaSponsorizzazioni_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Aggiunta e rimozione di una sponsorizzazione
Collaborations	Conferenza Sponsors

2.3.2.5 Controller.View

Class name	MonthlyStatWindow_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Calcolo delle statistiche su base mensile
Collaborations	Stats

Class name	YearlyStatWindow_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Calcolo delle statistiche su base annuale
Collaborations	Stats

Class name	VisualizzaStatistiche_Controller
Superclass	Nessuna
Subclasses	Nessuna
Responsability	Finestra principale per la visualizzazione delle statistiche di sistema
Collaborations	Nessuna

2.4 I class diagram

2.4.1 Il modello di dominio

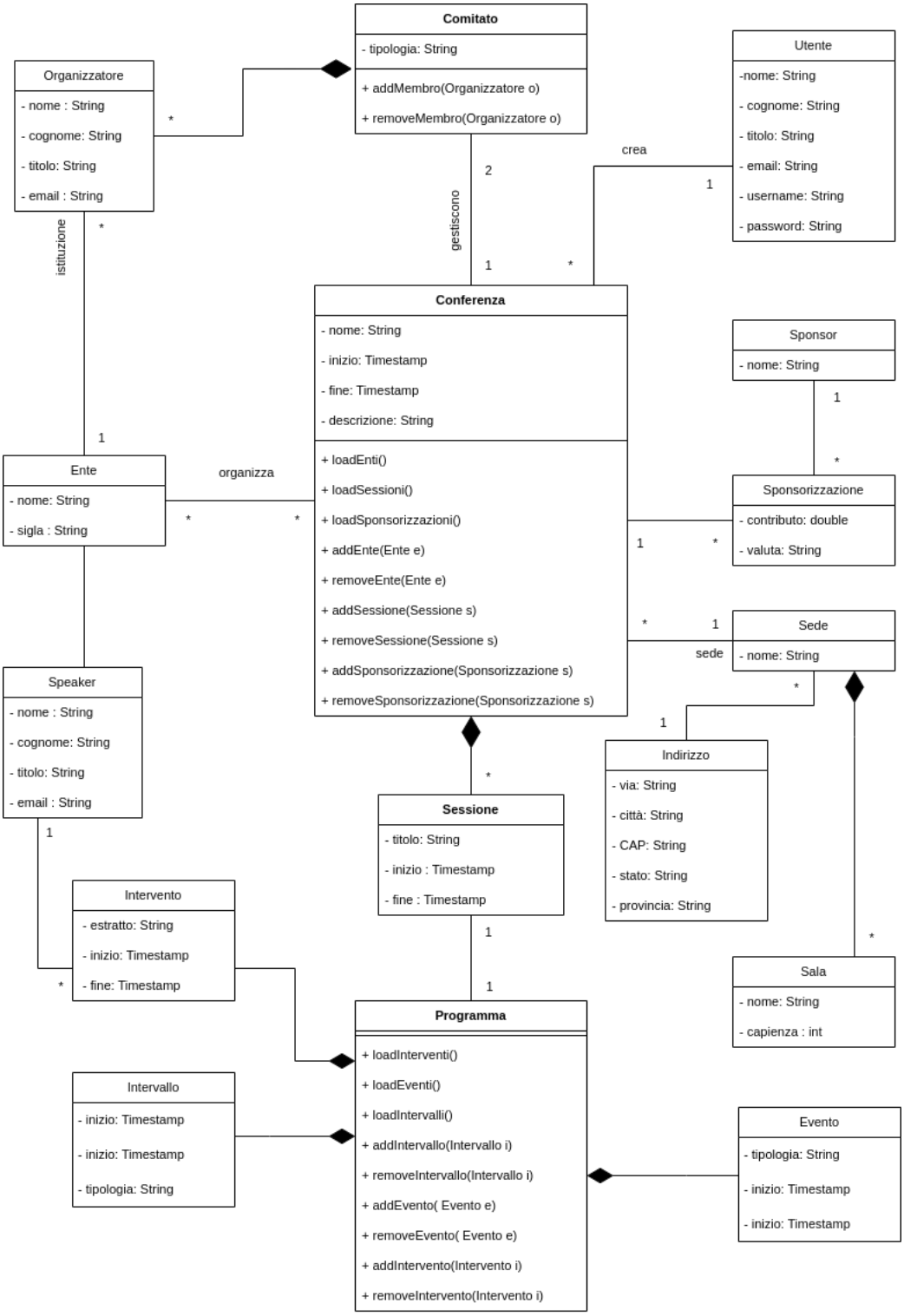


Figura 2.1: *Class Diagram del modello di dominio*

2.4.2 I controller

2.4.2.1 Controller.Create

Figura 2.2: *Controller per la fase di creazione*

2.4.2.2 Controller.Edit

2.4.2.3 Controller.Login

2.4.2.4 Controller.Stats

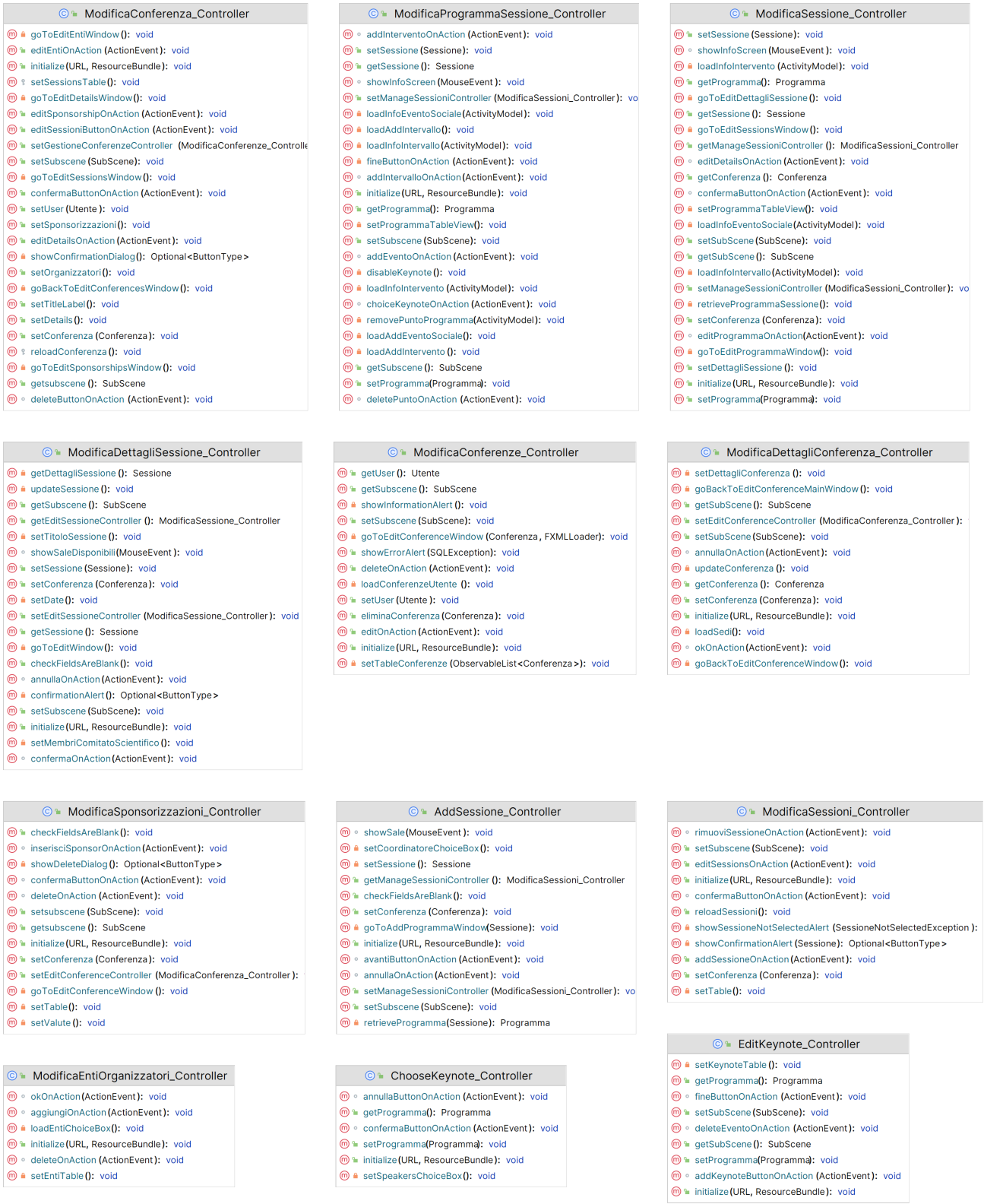


Figura 2.3: *Controller per la fase di modifica*

2.4.2.5 Controller.View

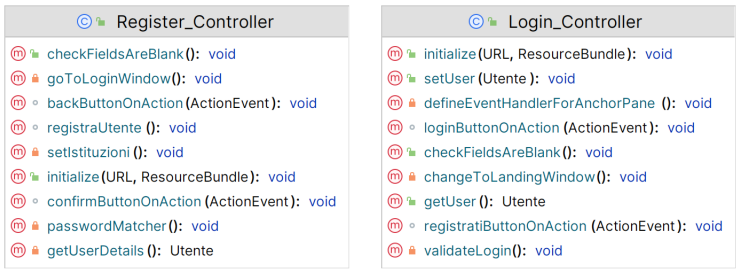


Figura 2.4: *Controller per la fase di login e registrazione*

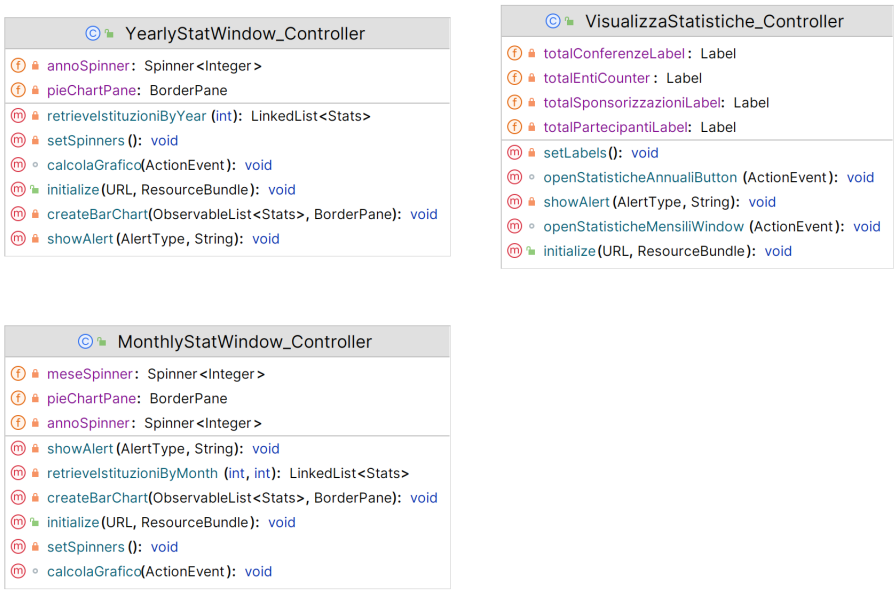


Figura 2.5: *Controller per la fase di visualizzazione delle statistiche*

2.4.3 Le classi DAO

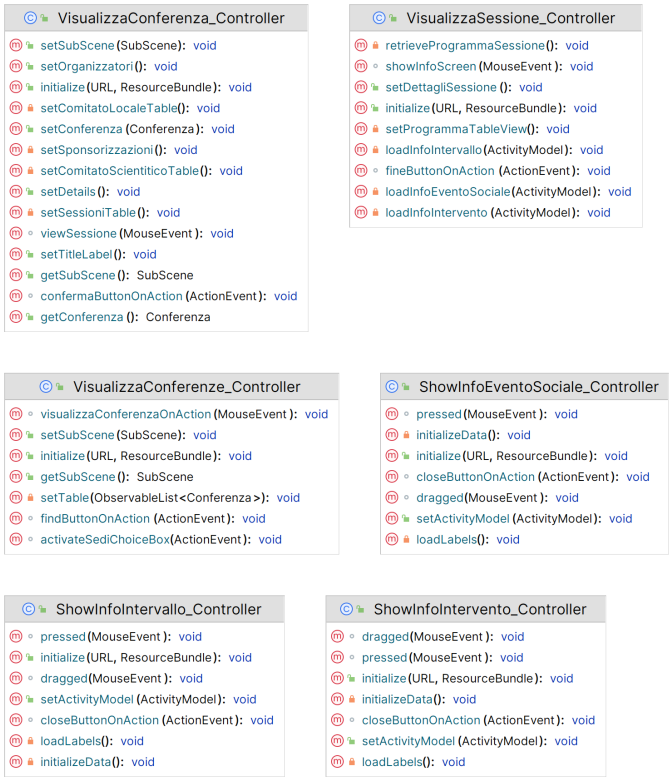


Figura 2.6: *Controller per la fase di visualizzazione delle conferenze*

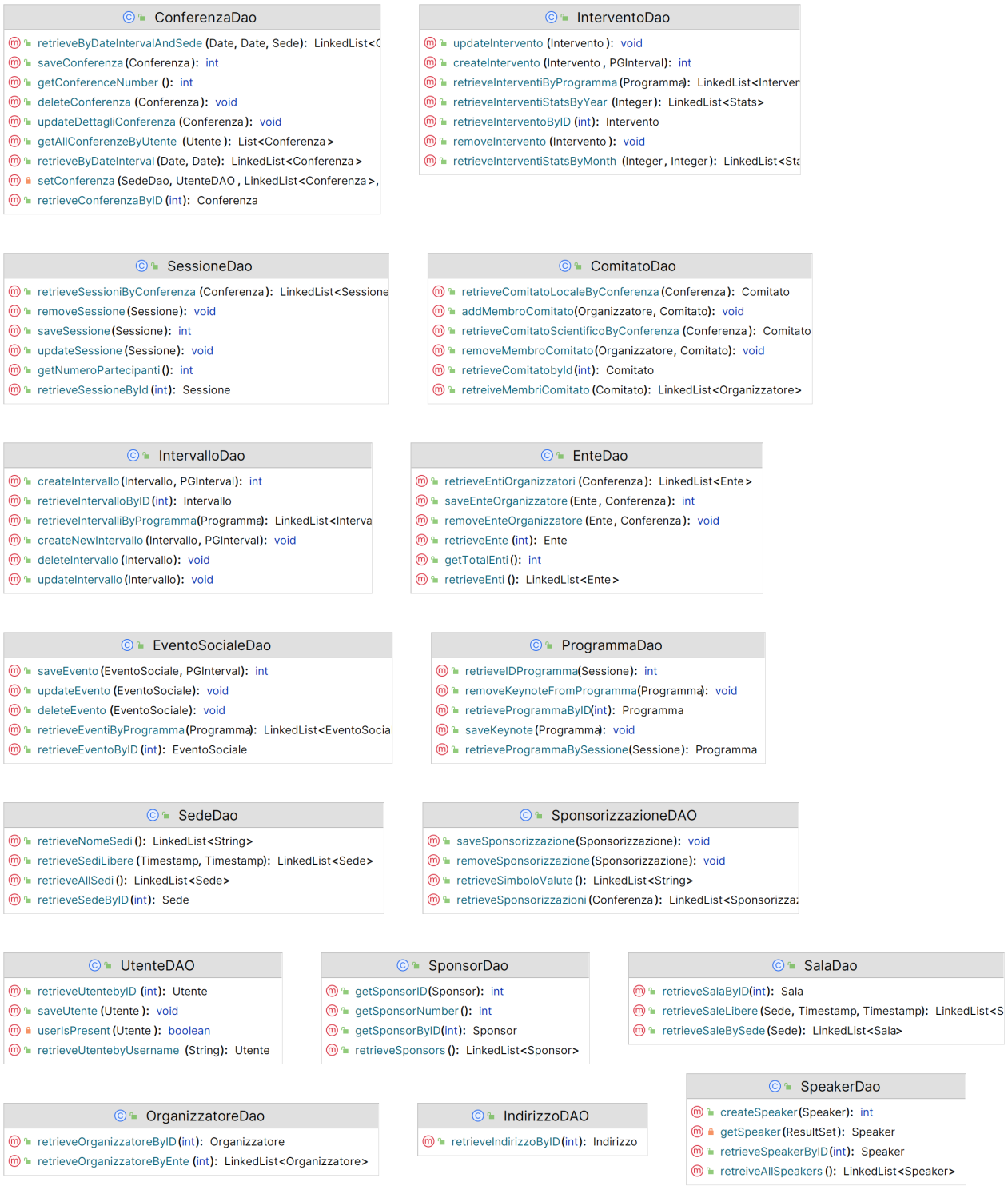


Figura 2.7: *Classi DAO*

Capitolo 3

Sequence Diagram e Mockup

3.1 I sequence diagram

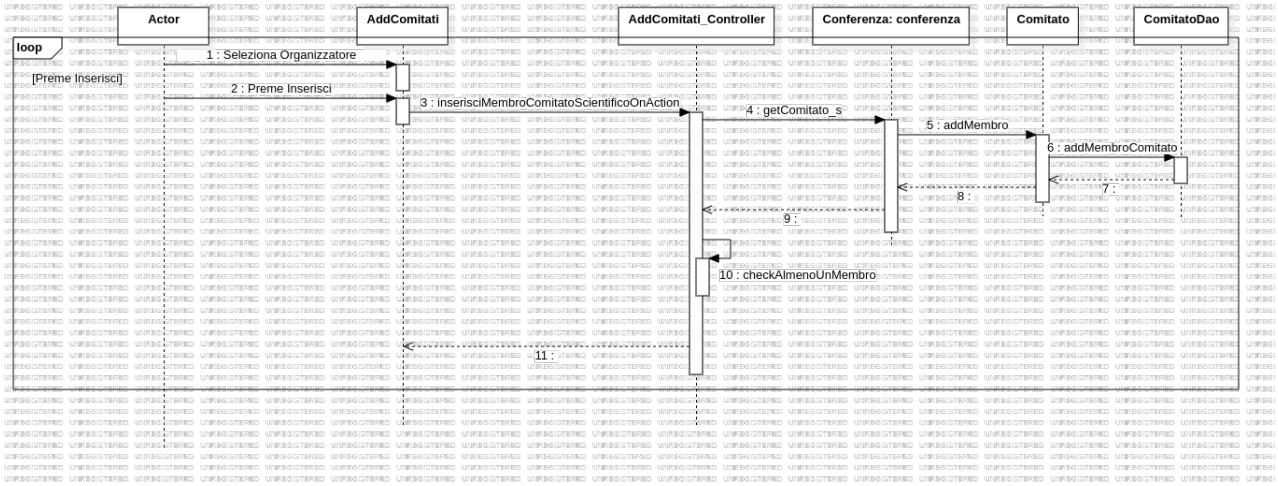


Figura 3.1: *Inserimento di un organizzatore in un comitato scientifico*