



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II
DOCUMENTAZIONE DEL PROGETTO DI BASI DI DATI
A.A. 2022-2023

Giulio Pianese Giuseppe Sindoni
N86004335 N86004107
ID GRUPPO: OODB_T1G25

Indice

1	Analisi della realtà di interesse	4
1.1	Dominio del problema generale	5
1.2	Dizionari	6
1.2.1	Dizionario delle entità	6
1.2.2	Dizionario delle associazioni	8
2	Model	9
2.1	Traduzione delle classi e delle relazioni	9
2.1.1	Package Model Diagram:	11
3	Comunicazione con il Database	12
3.0.1	Gestione della Connessione al Database	12
4	Package DAO	13
4.1	Interfaccia UtenteDAO	13
4.2	Interfaccia LibroDAO	13
4.3	Interfaccia PubblicazioneDAO	13
4.4	Interfaccia ConferenzaDAO	14
4.5	Interfaccia RivistaDAO	14
4.6	Interfaccia SerieDAO	14
4.7	Interfaccia PreferitiDAO	14
4.8	Interfaccia CanaleDistribuzioneDAO	14
4.9	Interfaccia CollanaDAO	14
5	Implementazioni Del Package DAO	15
6	Package Diagram DAO	16
7	Graphical User Interface	17
7.0.1	Package GUI diagram	18
8	Design patter BCED	19
8.0.1	Package Controller	19
9	Dominio della soluzione	21
9.1	Solution Diagram	21

10 Sequence Diagram	22
10.1 Sequence Diagram Aggiungi Serie preferita	22
10.2 Sequence Diagram Elimina libro	23

Capitolo 1

Analisi della realtà di interesse

La prima fase da affrontare è quella di esaminare le richieste da parte del cliente. La realtà che ci è stata chiesta di gestire è quella di una **Biblioteca digitale**.

"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 una biblioteca digitale. Gli elementi che possono essere inclusi nella biblioteca digitale sono di due tipi: articoli scientifici (o pubblicazioni) e libri (didattici o romanzi)."

Le entità principali del **Miniworld** in questione sono: **Pubblicazioni** (o Articoli scientifici) e **Libri**. Una prima precisazione è la suddivisione dei libri in **Libri didattici** e **Romanzi**. Non essendo specificato deduciamo che un libro può far parte strettamente ad una delle due categorie.

Un libro può anche far parte di una collana, la quale può raggruppare tutte le pubblicazioni che condividono una determinata caratteristica (non tutti i libri fanno parte di collane).

Qui ci viene specificata l'esistenza di **Collane**, che possiamo vedere come un raggruppamento di libri secondo un particolare criterio. Come specificato, un libro può non far parte di alcuna collana e nel caso opposto deduciamo possa far parte di una o più di quest'ultime.

Per le pubblicazioni, andare a definire in quale rivista (nome, argomento, anno di pubblicazione, responsabile della rivista) o in quale conferenza (luogo della conferenza, data di inizio e data fine conferenza, struttura organizzatrice e responsabile) è stato pubblicato.

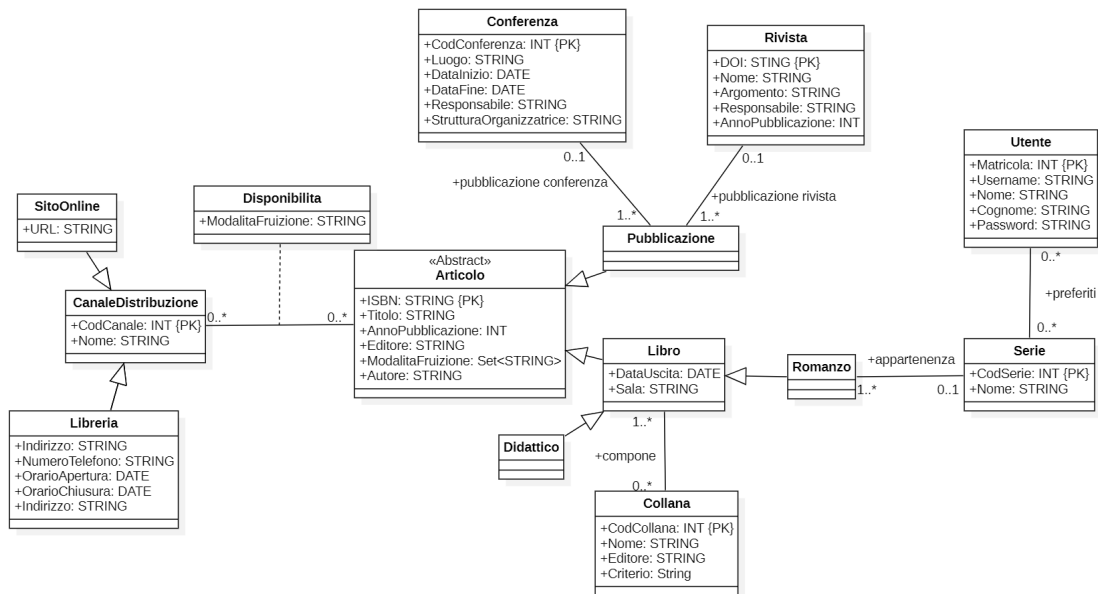
Altra informazione importante relativa ad una **Pubblicazione** è la sua relativa divulgazione, che può avvenire esclusivamente tramite una **Conferenza** o tramite una **Rivista**.

Per ogni libro, è necessario specificare dove può essere acquistato (libreria, online). Un romanzo può avere anche uno o più seguiti. In tal caso, è importante prevedere un'interrogazione che permetta di recuperare tutte le librerie (o siti internet) dai quali è possibile acquistare l'intera serie dei libri.

Oltre alla '**Disponibilità**' di un libro mediante **Siti** o **Librerie**, ci viene specificato che un '**Romanzo**' può far parte di una '**Serie**'. La richiesta prevede che il sistema notifichi l'**utente** ogni qualvolta un sito o una libreria dispongano di un'intera serie di libri (più precisamente Romanzi). Per soddisfare a pieno la richiesta definiamo la lista dei **Preferiti**, che non è altro un elenco di Serie per il quale un utente è interessato.

1.1 Dominio del problema generale

Dopo aver analizzato quanto detto nella parte precedente, è stato elaborato il seguente schema concettuale rappresentato tramite diagramma UML.



1.2 Dizionari

1.2.1 Dizionario delle entità

Entità	Descrizione	Attributi
Articolo	Rappresenta un'opera generica presente nella biblioteca digitale.	ISBN: codice identificante dell'articolo. Titolo: titolo dell'articolo. Autore: autori che hanno contribuito alla scrittura dell'articolo. Anno pubblicazione: anno in cui è stato pubblicato l'articolo. Modalità di fruizione: rappresenta i formati nei quali l'articolo esiste Editore: l'ente o l'organizzazione responsabile della pubblicazione
Libro	Rappresentazione specifica di un articolo.	Data uscita: Si riferisce alla data di pubblicazione del libro. Sala presentazione: Sala o libreria in cui è stata effettuata l'eventuale presentazione del libro.
Pubblicazione	Rappresentazione specifica di un Articolo.	Nessun attributo specifico.
Didattico	Genere specifico di Libro.	Nessun attributo specifico.
Romanzo	Genere specifico di Libro.	Nessun attributo specifico.
Collana	Insieme di volumi accomunati da particolari caratteristiche.	CodCollana: Codice identificante della collana. Nome: Nome della collana. Editore: Editore o l'organizzazione responsabile della pubblicazione. Criterio: rappresenta il criterio che accomuna tutti i libri appartenenti alla collana.
Serie	Composizione di Romanzi correlati che condividono personaggi, ambientazioni o una trama continua.	CodSerie: Codice identificante della serie. Nome: Nome della serie.
Utente	Rappresenta l'utente che interagisce con il sistema.	Matricola: Codice identificante dell'utente. Username: Identificatore univoco o nome scelto dall'utente per identificarsi. Nome: Nome dell'utente. Cognome: Cognome dell'utente. Password: Password dell'utente utilizzata per effettuare il login.

Entità	Descrizione	Attributi
Conferenza	Rappresenta una conferenza in cui una o più pubblicazioni scientifiche sono state presentate.	CodConferenza: Codice identificante della conferenza. Luogo: Luogo in cui si è svolta la conferenza. Responsabile: La persona o l'ente responsabile della gestione e supervisione generale della conferenza. Struttura organizzatrice: Struttura che ha ospitato la conferenza. Data inizio: Data di inizio della conferenza. Data fine: Data fine della conferenza.
Conferenza	Rappresenta una rivista in cui una pubblicazione scientifica è stata presentata.	DOI: codice identificante di una Rivista. Nome: nome della rivista. Argomento: l'argomento principale o l'area di interesse della rivista. Responsabile: la persona o l'ente responsabile della gestione e supervisione generale della rivista. Anno pubblicazione: anno in cui è stata pubblicata la rivista.
Canale distribuzione	Piattaforma generica che si occupa della vendita di articoli.	CodCanale: codice identificante del canale. Nome: nome del canale di distribuzione.
Sito online	Piattaforma o entità virtuale accessibile tramite Internet.	URL: indirizzo web del sito online.
Libreria	Luogo dedicato alla raccolta e fruizione di Articoli.	Indirizzo: specifica la posizione geografica della libreria. Numero telefono: numero di recapito telefonico della Libreria. Orario apertura: orario di apertura della libreria al pubblico. Orario chiusura: orario di chiusura della libreria al pubblico.

Tabella 1.1: Tabella delle entità con descrizione e attributi

1.2.2 Dizionario delle associazioni

Associazione	Descrizione	Cardinalità
Disponibilità	Associazione tra Articolo e Canale distribuzione. Rappresenta la disponibilità di un articolo, mediante un canale di distribuzione, in una specifica modalità di fruizione.	M:N Un articolo può essere disponibile presso zero o più canali, mentre un canale può disporre di zero o più articoli.
Pubblicazione rivista	Associazione tra Pubblicazione e Rivista. Rappresenta la presentazione di una pubblicazione tramite una rivista.	1:N Una pubblicazione può essere pubblicata al più da una rivista, mentre una rivista deve presentare una o più pubblicazioni.
Pubblicazione conferenza	Associazione tra Pubblicazione e Conferenza. Rappresenta la presentazione di una pubblicazione mediante una conferenza.	1:N Una pubblicazione può essere presentata al più da una conferenza, mentre una conferenza deve presentare una o più pubblicazioni.
Compone	Associazione tra Libro e Collana. Rappresenta l'appartenenza di un libro ad una collana.	N:M Un libro può far parte di zero o più collane, mentre ad una collana devono far parte uno o più libri.
Appartenenza	Associazione tra Romanzo e Serie. Rappresenta l'appartenenza di un Romanzo ad una serie.	N:M Un libro può appartenere o meno ad una serie, mentre ad una serie devono appartenere uno o più libri.
Preferiti	Associazione tra Utente e Serie. Rappresenta l'interesse di un utente verso quella serie.	N:M Un utente può avere zero o più serie nei preferiti e allo stesso tempo una serie può essere "preferita" di zero o più utenti.

Tabella 1.2: Tabella delle associazioni con descrizione e cardinalità

Capitolo 2

Model

2.1 Traduzione delle classi e delle relazioni

Una volta completata l'analisi del dominio del problema, possiamo avanzare con la mappatura delle classi e delle associazioni per creare il Model del nostro problema, delineando il metodo con cui gestiamo le diverse associazioni.

- **Disponibilità:** Essendo disponibilità una relazione *multi-a-multi* parziale, abbiamo deciso di gestire tale associazione attraverso delle hashmap, una all'interno della classe Articolo (CanaleDistribuzione, Set<String>) e una all'interno della classe CanaleDistribuzione (Articolo, Set<String>). Rispettivamente, andranno a specificare la disponibilità di un articolo presso zero o più canali, in una o più modalità di fruizione (sottoinsieme proprio delle modalità di fruizione per cui l'articolo esiste), e la disponibilità di zero o più articoli presso un canale di distribuzione rispettivamente. Nell'ambito dell'analisi dei requisiti, ci è stato richiesto di tracciare solamente la disponibilità degli articoli presso i canali di distribuzione. Tuttavia seguendo le regole di mapping delle associazioni, abbiamo esteso questa funzionalità per includere anche la tracciabilità della disponibilità dei vari articoli presso ogni canale di distribuzione. In questo modo, è possibile identificare quali articoli sono disponibili presso un determinato canale di distribuzione.

Nonostante questa estensione non fosse esplicitamente richiesta, abbiamo deciso di implementarla per migliorare l'efficienza del sistema nel caso in cui diventasse necessaria in futuro.

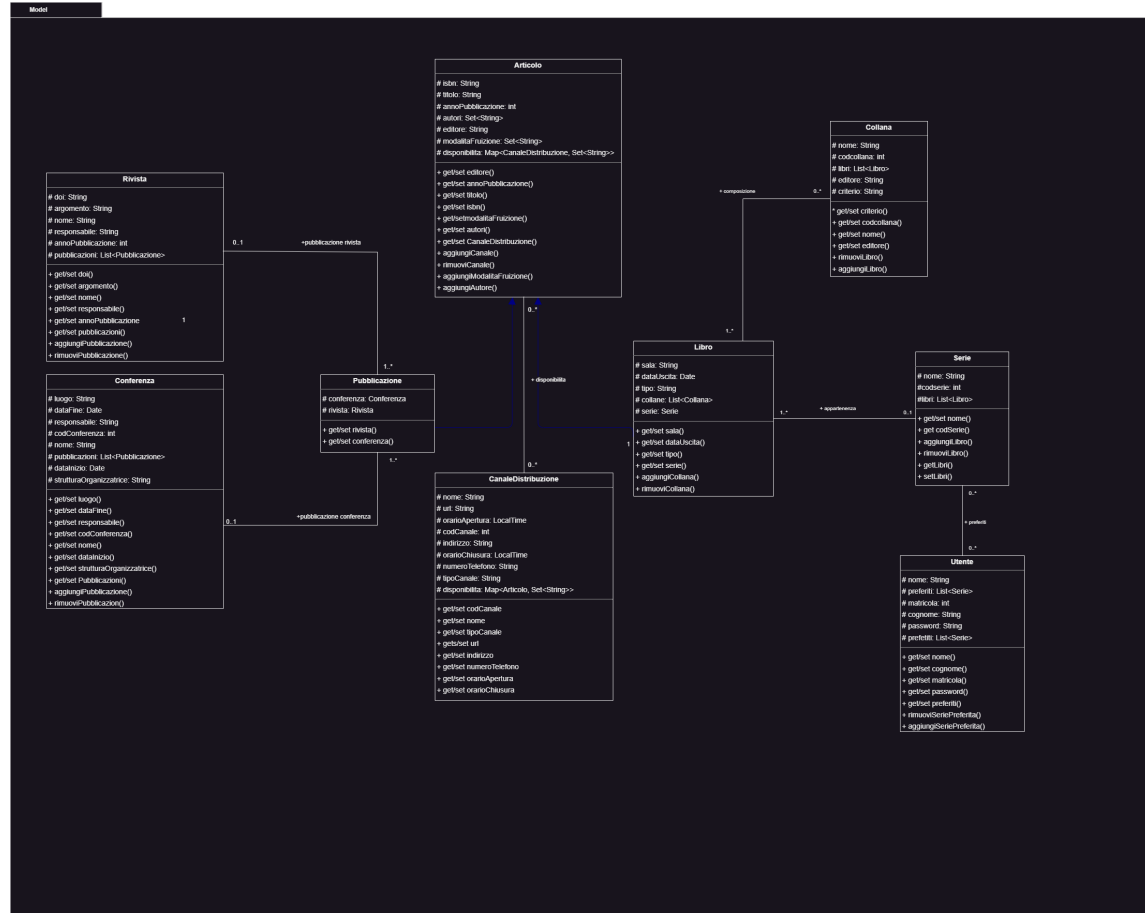
- **Pubblicazione conferenza:** Si tratta di un tipo di associazione *uno-a-multi* parziale. Per questo motivo, abbiamo deciso di mappare tale associazione utilizzando nella classe Pubblicazione il riferimento a una Conferenza e, per ogni Conferenza, una lista di riferimenti agli oggetti della classe Pubblicazione.
- **Pubblicazione rivista:** Si tratta di un tipo di associazione *uno-a-multi* parziale. Per questo motivo, abbiamo deciso di mappare tale associazione utilizzando nella classe Pubblicazione il riferimento a una Rivista e, per ogni Rivista, una lista di riferimenti agli oggetti della classe Pubblicazione.
- **Compone:** Si tratta di un tipo di associazione *multi-a-multi* parziale. Per questo motivo abbiamo deciso di mappare tale associazione utilizzando due liste all'interno delle rispettive

classi. Ogni oggetto della classe Libro contiene una lista di riferimenti alle Collane a cui appartiene, mentre la classe Collana ha una lista di riferimenti ai libri che la compongono.

- **Apparteneza:** Si tratta di una associazione di tipo *uno-a-molti* parziale. Per questo motivo abbiamo deciso di mappare tale associazione utilizzando nella classe Romanzo il riferimento a una Serie, mentre nella classe Serie una lista di riferimenti agli oggetti della classe Libro.
- **Preferiti:** Si tratta di un tipo di associazione di tipo *molti-a-molti* parziale, il che significa che un utente può avere zero o più serie preferite e una serie può essere preferita da zero o più utenti. Per mappare tale associazione abbiamo realizzato una lista di riferimenti ad oggetti della classe Serie all'interno della classe Utente, invece abbiamo ritenuto inutile tenere traccia all'interno della classe serie degli utenti che hanno "targato" quella rispettiva serie come preferita.

2.1.1 Package Model Diagram:

Il seguente diagramma rappresenta il Package model:



Capitolo 3

Comunicazione con il Database

Per stabilire la comunicazione con il database della biblioteca digitale, è stata adottata l'architettura **DAO** (Data Access Object). Questo pattern è scelto per gestire la persistenza dei dati e regolare gli accessi ad essi. L'obiettivo è fornire un'interfaccia uniforme per l'accesso ai dati, permettendo all'applicazione di non doversi preoccupare dei dettagli riguardanti l'archiviazione o il recupero dei dati stessi.

3.0.1 Gestione della Connessione al Database

La connessione al database della biblioteca digitale è gestita attraverso il pacchetto `Connessione`, che contiene la classe `ConnessioneDatabase`. Questa classe è responsabile della creazione di un'unica istanza di connessione al database Postgres, mantenuta attiva per l'intera durata dell'applicazione. Tale scelta consente all'applicativo di essere eseguito su più dispositivi senza la necessità di creare una copia locale del database.

Capitolo 4

Package DAO

4.1 Interfaccia UtenteDAO

L'interfaccia **UtenteDAO** fornisce le funzioni per eseguire il "Login" e la "Registrazione" di un utente.

4.2 Interfaccia LibroDAO

L'interfaccia **LibroDAO** fornisce le funzioni per:

- eseguire l'inserimento di un libro
- eseguire la cancellazione di un libro
- aggiungere la disponibilità di un libro presso un canale di distribuzione
- restituire i libri appartenenti ad una serie
- restituire i libri appartenenti ad una collana
- restituire tutti i libri presenti nella base di dati
- effettuare la ricerca di libri secondo uno specifico campo

4.3 Interfaccia PubblicazioneDAO

L'interfaccia **PubblicazioneDAO** fornisce le funzioni per:

- eseguire l'inserimento di una pubblicazione
- eseguire la cancellazione di una pubblicazione
- aggiungere la disponibilità di una pubblicazione presso un canale di distribuzione
- restituire tutte le pubblicazioni presenti nella base di dati
- effettuare la ricerca di pubblicazioni secondo uno specifico campo

4.4 Interfaccia ConferenzaDAO

L'interfaccia **ConferenzaDAO** fornisce la funzione per ricercare una conferenza nella base di dati.

4.5 Interfaccia RivistaDAO

L'interfaccia **RivistaDAO** fornisce la funzione per ricercare una rivista nella base di dati.

4.6 Interfaccia SerieDAO

L'interfaccia **SerieDAO** fornisce le funzioni per:

- restituire le serie presenti nella base di dati.
- effettuare la ricerca di serie secondo uno specifico campo

4.7 Interfaccia PreferitiDAO

L'interfaccia **PreferitiDAO** fornisce le funzioni per:

- inserire una serie nei preferiti di uno specifico utente.
- eliminare una serie nei preferiti di uno specifico utente.
- restituire l'elenco di serie preferite di uno specifico utente.
- determinare se una serie fa già parte dei preferiti di un utente.

4.8 Interfaccia CanaleDistribuzioneDAO

L'interfaccia **CanaleDistribuzioneDAO** fornisce le funzioni per:

- restituire i canali di distribuzione che hanno disponibile un'intera serie appartenente all'elenco dei preferiti di uno specifico utente.
- visualizzare tutti i canali di distribuzione presenti nella base di dati.
- recuperare un canale di distribuzione dalla base di dati secondo il nome
- restituire i canali che dispongono di uno specifico articolo

4.9 Interfaccia CollanaDAO

L'interfaccia **CollanaDAO** fornisce le funzioni per restituire tutte le collane e ricercare una collana nella base di dati tramite il nome.

Capitolo 5

Implementazioni Del Package DAO

Le interfacce presenti nel package DAO sono poi implementate per il database Postgres nel package implementazioniPostgresDAO. Le classi che implementano le interfacce definiscono il modo in cui vengono effettuate le varie query al database. Queste classi descrivono quindi la comunicazione tra il nostro software e il database utilizzato.

Package Diagram DAO

Il diagramma successivo offre una panoramica dell'architettura DAO impiegata nel software, delineando i package, le classi e i metodi utilizzati, nonché le loro relazioni.



Capitolo 7

Graphical User Interface

In questa sezione viene dettagliato il metodo di realizzazione dell'interfaccia grafica.

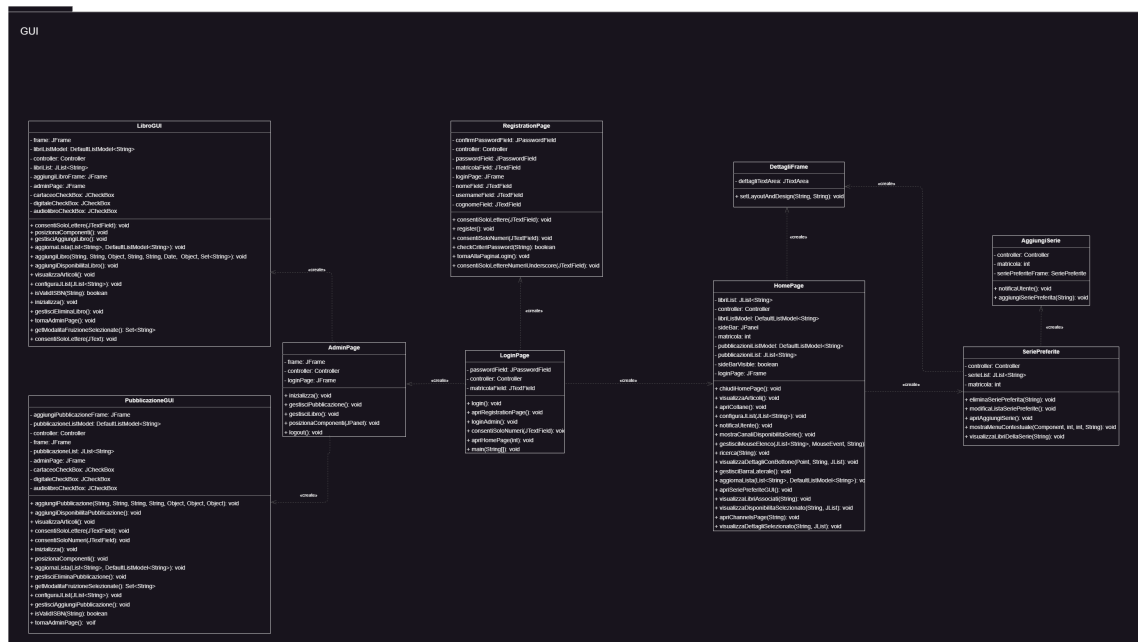
Abbiamo optato per il framework Swing. L'interfaccia utente è stata sviluppata all'interno del package GUI, con i file gestiti secondo il seguente pattern:

- **LoginPage**: è la prima pagina visualizzata all'avvio dell'applicativo. Qui un generico utente può accedere alla **RegistrationPage** per registrarsi ed effettuare il proprio login, oppure un admin può accedere e visualizzare l'**AdminPage**, inserendo specifiche credenziali (Matricola: -1, Password: admin).
- **RegistrationPage**: schermata dove l'utente inserisce le proprie informazioni per creare un proprio "profilo" per effettuare il login.
- **HomePage**: Schermata principale dell'applicativo "lato user". Presenta varie feature come:
 - visualizzare tutti gli articoli presenti nel database insieme ai loro dettagli e alla loro disponibilità presso i vari canali di distribuzione.
 - filtrare la ricerca degli articoli presenti nel database tramite una barra di ricerca
 - accedere alla pagina "Serie Preferite"
 - visualizzare i canali che dispongono, per intero, di una serie preferita dell'utente.
 - visualizzare le collane presenti nel database assieme ai libri che le compongono
 - effettuare il log out.
- **SeriePreferite**: schermata che permette all'utente di visualizzare le sue attuali serie preferite, visualizzare i libri che compongono una serie, selezionare una serie preferita dal proprio elenco e rimuoverla da quest'ultimo, accedere alla pagina "AggiungiSerie".
- **AggiungiSerie**: schermata che permette di selezionare una serie tra quelle presenti nel database ed aggiungerla ai propri "preferiti". Se una serie preferita è disponibile per intero presso uno o più canali di distribuzione l'utente viene notificato tramite un messaggio di avviso.
- **AdminPage**: pagina relativa all'accesso "lato admin". Permette di interfacciarsi alle schermate di inserimento-cancellazione delle varie entità

- **LibroGUI**: schermata relativa all'inserimento e cancellazione di libri, ed aggiunta di disponibilità di un libro presso un canale di distribuzione.
- **PubblicazioneGUI**: schermata relativa all'inserimento e cancellazione di pubblicazioni, ed aggiunta di disponibilità di una pubblicazione presso un canale di distribuzione.

7.0.1 Package GUI diagram

Il diagramma successivo fornisce una descrizione dell'implementazione nel package GUI.



Capitolo 8

Design patter BCED

Prima di procedere con la definizione del dominio della soluzione, è necessario identificare il pattern utilizzato per la creazione dell'applicativo Java. In questo contesto, è stato adottato un approccio BCED. Questo design pattern, abbreviato BCED, si riferisce a Entity-controller-boundary-database:

- **Entity:** rappresenta l'insieme di classi che riflettono il diagramma del dominio delle informazioni da gestire, corrispondente al nostro package MODEL.
- **Controller:** comprende le classi responsabili dell'implementazione delle operazioni algoritmiche, sollecitate direttamente dall'interfaccia utente (GUI), e gestiscono i dati presenti nel Model e nel database.
- **Boundary:** indica il package GUI, ovvero l'interfaccia grafica attraverso cui l'utente interagisce con l'applicazione.
- **Database:** comprende il nostro package DAO, dedicato alla gestione dei dati nel database.

8.0.1 Package Controller

Il seguente diagramma illustra il package Controller con la sua classe associata, che rappresenta il fulcro del nostro applicativo. In particolare, il controller si occupa di gestire qualsiasi interazione tra il database, il modello e l'interfaccia utente (GUI), implementando i metodi necessari per coordinare efficacemente le operazioni del sistema.

CONTROLLER

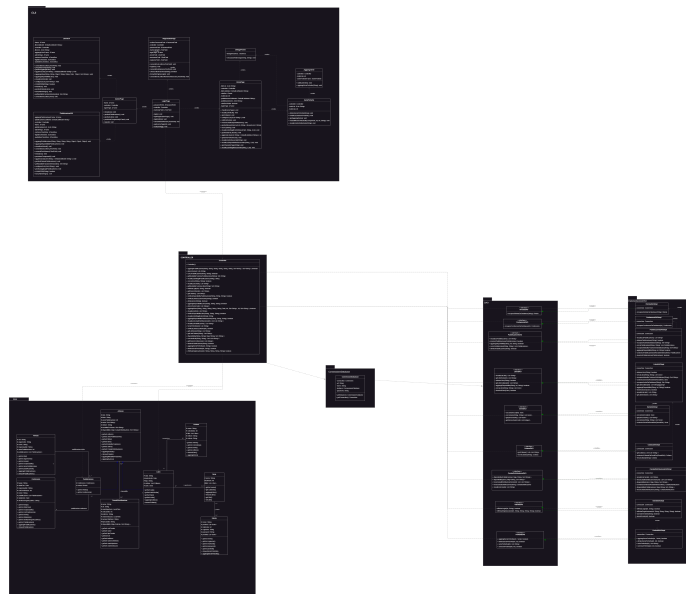
Controller
+ Controller():
+ aggiungiPubblicazione(String, String, String, String, String, String, Set<String>, Set<String>): boolean
+ elencoSerie(): List<String>
+ cercaPubblicazione(String, String): boolean
+ getModalitaFruizionePubblicazione(String): Set<String>
+ visualizzaDettagliPubblicazione(String): String
+ cercaLibro(String, String): boolean
+ visualizzaCanali(): List<String>
+ getModalitaFruizioneLibro(String): Set<String>
+ effettuaLogin(int, String): boolean
+ getElencoRiviste(): List<String>
+ getCollane(): List<String>
+ verificaDisponibilitaPubblicazione(String, String, String): boolean
+ verificaEsistenzaUsername(String): boolean
+ eliminaLibro(String): boolean
+ aggiungiDisponibilitaLibro(String, String, String): boolean
+ elencoSerieCod(): List<Integer>
+ aggiungiLibro(String, String, String, String, String, Date, int, Set<String>, int, Set<String>): boolean
+ visualizzaLibri(): List<String>
+ verificaDisponibilitaLibro(String, String, String): boolean
+ visualizzaDettagliLibro(String): String
+ aggiungiDisponibilitaPubblicazione(String, String, String): boolean
+ visualizzaCanaliDistribuzione(int): List<List<String>>
+ visualizzaPubblicazioni(): List<String>
+ SeriePreferite(int): List<String>
+ verificaEsistenzaUtente(int): boolean
+ getLibriSerie(String): List<String>
+ getLibriCollana(String): List<String>
+ disponibilita(String, String): Map<String, Set<String>>
+ cercaArticoli(String, String): List<List<String>>
+ getElencoConferenze(): List<Integer>
+ eliminaPubblicazione(String): boolean
+ aggiungiSeriePreferita(int, String): boolean
+ eliminaSeriePreferita(int, String): boolean
+ effettuaRegistrazione(int, String, String, String, String): boolean

Capitolo 9

Dominio della soluzione

In seguito a tutta l'implementazione descritta precedentemente, possiamo presentare il dominio della soluzione.

9.1 Solution Diagram



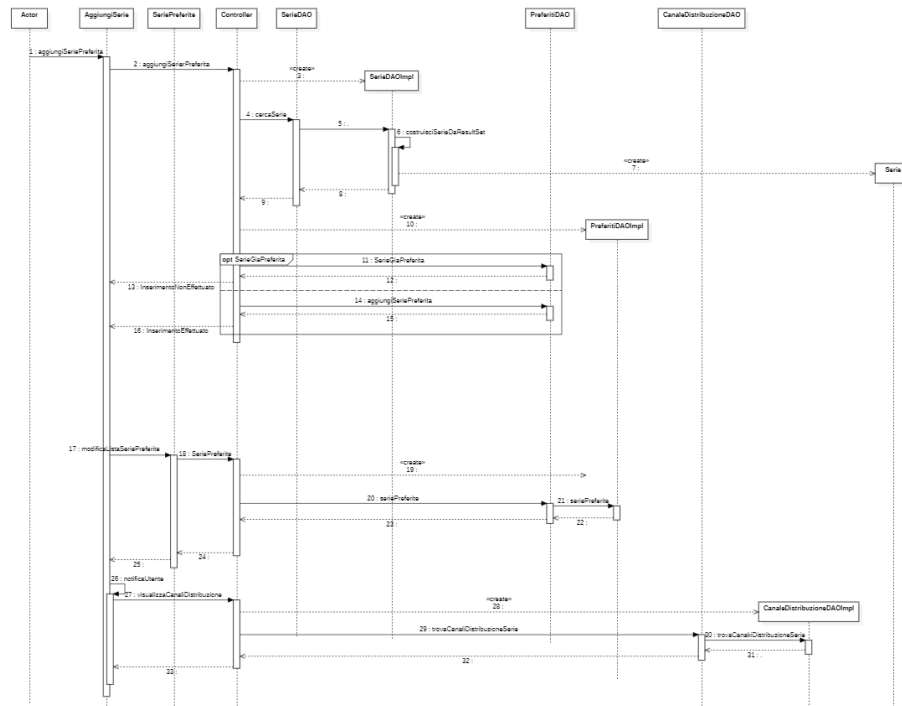
Capitolo 10

Sequence Diagram

In questa sezione presentiamo due diagrammi di sequenza che descrivono il flusso computazionale e il funzionamento effettivo di due operazioni in particolare.

10.1 Sequence Diagram Aggiungi Serie preferita

Il diagramma successivo fornisce un dettaglio completo del flusso del programma quando l'utente richiede di aggiungere una serie ai preferiti.



10.2 Sequence Diagram Elimina libro

Il diagramma successivo fornisce un dettaglio completo del flusso del programma quando l'admin richiede di eliminare un libro dalla base di dati.

