



UNIVERSITÀ DEGLI STUDI DI PALERMO
DIPARTIMENTO DI INGEGNERIA
CORSO DI LAUREA TRIENNALE IN
INGEGNERIA INFORMATICA

System Design Document

Allegra Luciano
Patti Francesco (referente)
Pecoraro Andrea
Scaglione Giulio

Anno Accademico 2024-2025

Sommario

1	OBIETTIVO DEL SISTEMA	2
2	ARCHITETTURA SOFTWARE ATTUALE.....	2
3	OBIETTIVI DI PROGETTAZIONE	3
4	ARCHITETTURA SOFTWARE PROPOSTA	4
4.1	PANORAMICA.....	4
4.2	REQUISITI MINIMI PER L'UTILIZZO DEL SOFTWARE PROPOSTO	4
4.3	SCOMPOSIZIONE IN SOTTOSISTEMI.....	5
4.3.1	<i>Mappatura degli oggetti nei sottosistemi</i>	<i>10</i>
4.4	MAPPATURA HARDWARE/SOFTWARE.....	18
5	GESTIONE DEI DATI PERSISTENTI	18
5.1	SCHEMA E-R	17
5.2	MODELLO RELAZIONALE.....	18
5.3	STRUTTURA DELLE TABELLE	19
5.3.1	<i>utenti</i>	<i>19</i>
5.3.2	<i>conferenze</i>	<i>19</i>
5.3.3	<i>keywords.....</i>	<i>21</i>
5.3.4	<i>articoli.....</i>	<i>21</i>
5.3.5	<i>notifiche.....</i>	<i>21</i>
5.3.6	<i>ruoli</i>	<i>22</i>
5.3.7	<i>specificInfoAggiuntive</i>	<i>23</i>
5.3.8	<i>revisione.....</i>	<i>23</i>
5.3.9	<i>sottomette.....</i>	<i>24</i>
5.3.10	<i>keywordsArticoli</i>	<i>24</i>
5.3.11	<i>competenzeRevisore</i>	<i>24</i>
5.3.12	<i>keywordsConferenza</i>	<i>25</i>

1 Obiettivo del sistema

Il sistema proposto ha lo scopo di supportare in modo completo l'organizzazione e la gestione delle conferenze scientifiche.

Il software è concepito per assistere tutte le fasi del ciclo di vita di una conferenza: dalla creazione dell'evento, alla raccolta delle sottomissioni da parte degli autori, alla gestione della fase di revisione tramite il Comitato di Programma (PC), fino alla pubblicazione degli atti finali.

Il sistema consentirà ai Chair di configurare i parametri della conferenza, selezionare e invitare i membri del PC, supervisionare l'assegnazione degli articoli ai revisori, monitorare il processo di revisione e gestire eventuali situazioni di stallo. Gli autori potranno caricare, modificare o ritirare le proprie sottomissioni entro i termini stabiliti, ricevere notifiche relative allo stato dei propri lavori e, in caso di accettazione, caricare la versione definitiva dei propri articoli. Il sistema garantirà il rispetto delle scadenze temporali, la gestione dei conflitti di interesse, la corretta distribuzione degli articoli fra i revisori nel rispetto dei limiti di carico massimo e l'integrità delle procedure di valutazione. Al termine del processo, il sistema provvederà alla raccolta delle versioni finali e alla trasmissione dei proceedings all'editore. Tutte le operazioni rilevanti saranno opportunamente tracciate mediante un sistema di logging, a supporto della trasparenza e della responsabilità gestionale.

2 Architettura software attuale

Si presume che, al momento, l'azienda non utilizzi alcun software con funzionalità analoghe e che le attività che il sistema mira ad automatizzare vengano attualmente eseguite manualmente dalle diverse figure aziendali.

3 Obiettivi di progettazione

- Il sistema e il DBMS devono risultare sempre attivi, permettendo agli utenti di accedere alle informazioni in qualunque momento. Tuttavia, sono previste delle manutenzioni giornaliere
- Le password degli account vengono salvate nel DBMS sotto forma di hash SHA-512
- La piattaforma deve risultare intuitiva, immediata e di facile utilizzo per la maggior parte della utenza.
- Il sistema deve poter fronteggiare una perdita di connessione segnalandola all'utente, assicurando la persistenza dei dati immagazzinati.

4 Architettura software proposta

4.1 Panoramica

L'architettura software adottata per il sistema è di tipo **ibrido**, combinando i benefici di due pattern architetturali a diversi livelli di astrazione per garantire modularità, manutenibilità e scalabilità.

A livello **macro-architetturale**, il sistema è basato sul pattern **Repository**. L'architettura è distribuita in componenti software autonomi, ciascuno responsabile di un dominio funzionale specifico (sottosistema). Questi componenti sono fortemente disaccoppiati e la loro interazione avviene esclusivamente in modo indiretto, attraverso la manipolazione di dati condivisi in un repository centrale, ospitato su un nodo dedicato (DBHost).

A livello **micro-architetturale**, ogni singolo sottosistema è strutturato internamente secondo il pattern **Three-Tier**. Questa architettura a tre livelli separa nettamente le responsabilità del componente in:

- ❖ **Interface Layer:** Gestisce l'interazione con l'utente (UI) e la visualizzazione dei dati.
- ❖ **Application Logic Layer (Controller):** Contiene la logica di business, le regole applicative e il coordinamento delle operazioni specifiche del sottosistema.
- ❖ **Storage Layer (DAO):** Incapsula tutta la comunicazione con il repository centrale, eseguendo le interrogazioni (query) necessarie per la persistenza dei dati.

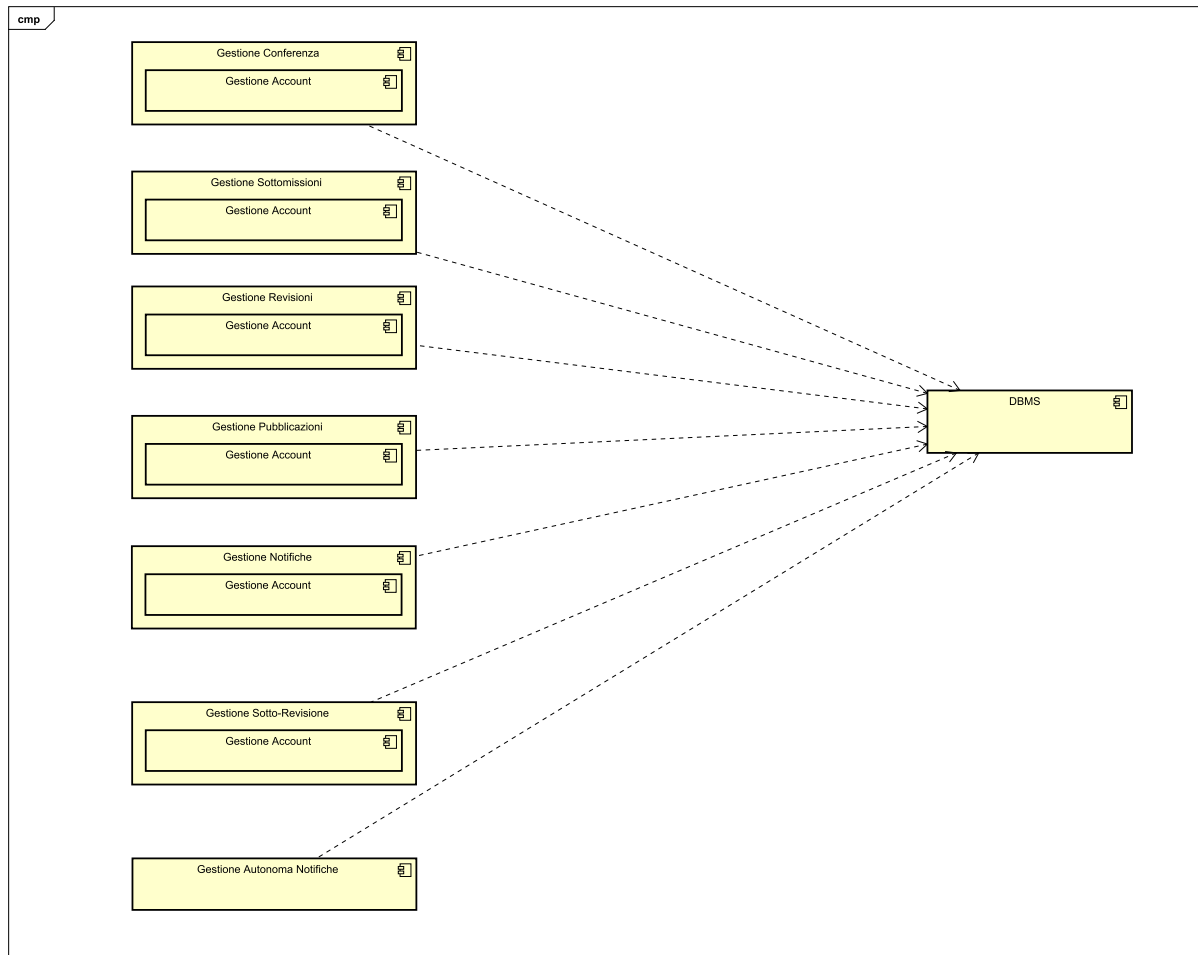
In termini di deployment, i tre livelli di ciascun sottosistema applicativo risiedono sullo stesso nodo (Application Server), mentre il repository (DBMS) è isolato su un nodo DBHost separato, come definito nel modello di distribuzione.

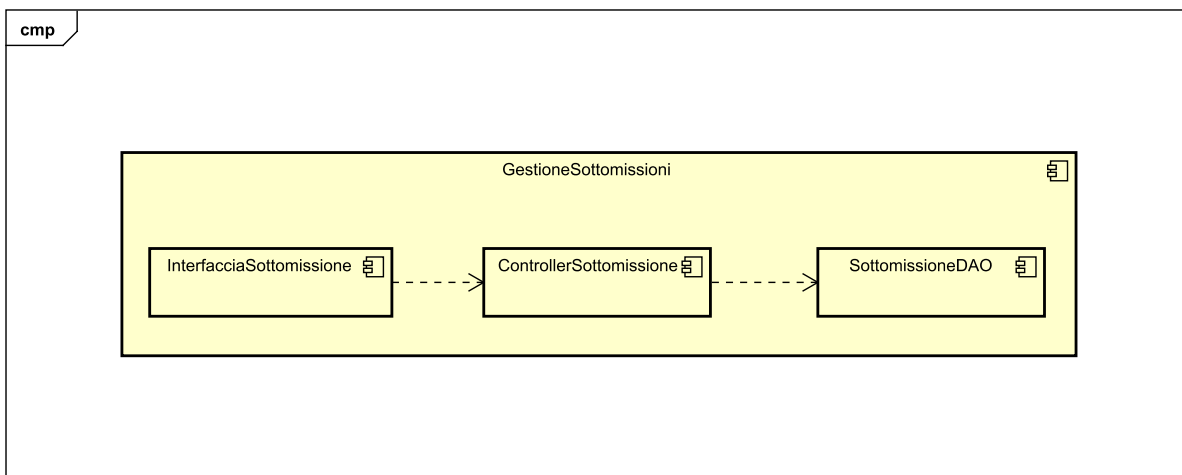
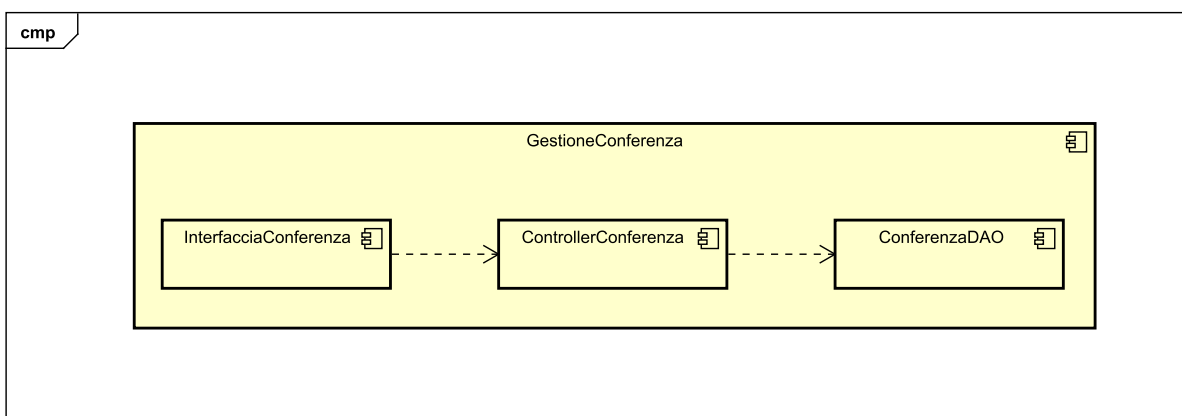
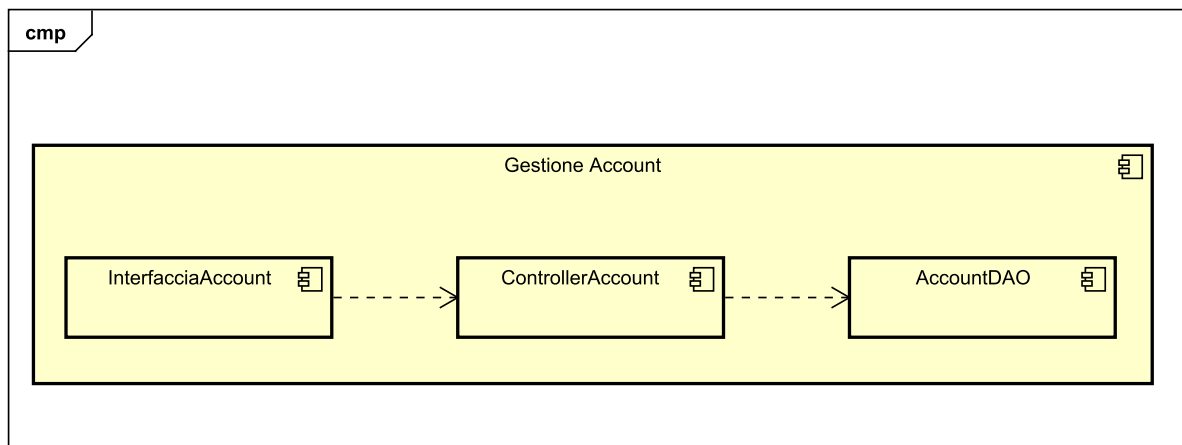
4.2 Requisiti minimi per l'utilizzo del software proposto

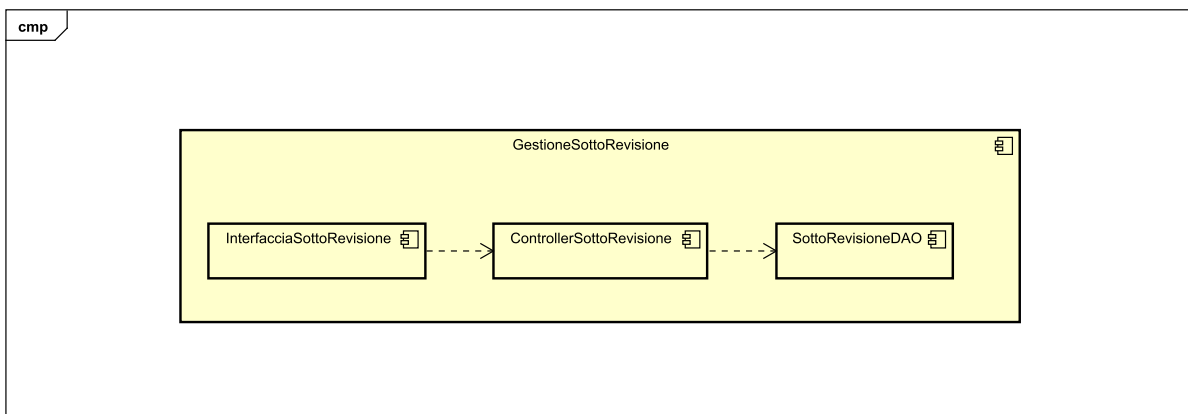
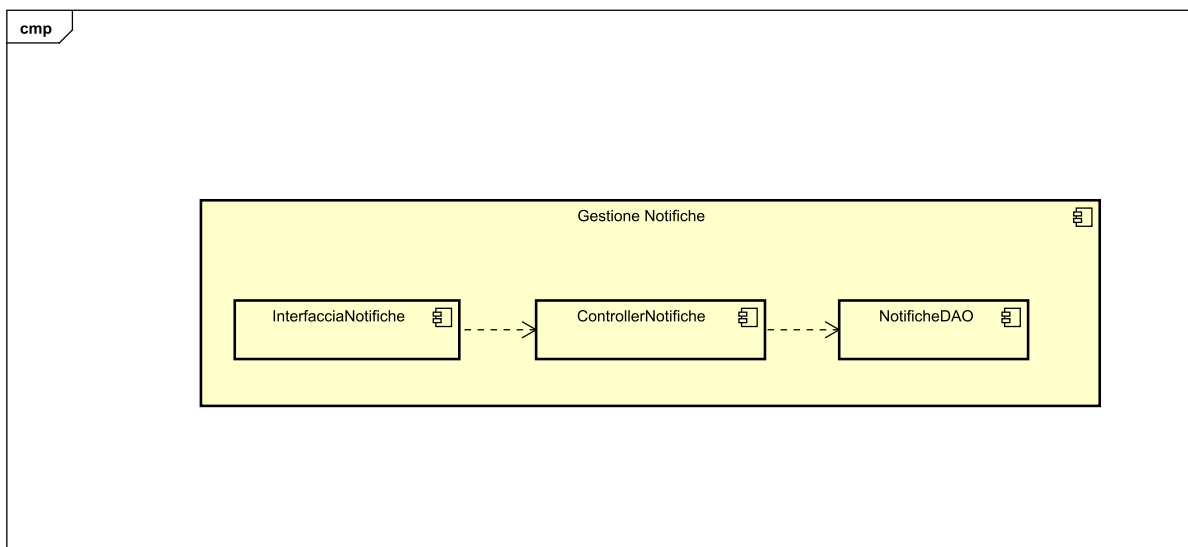
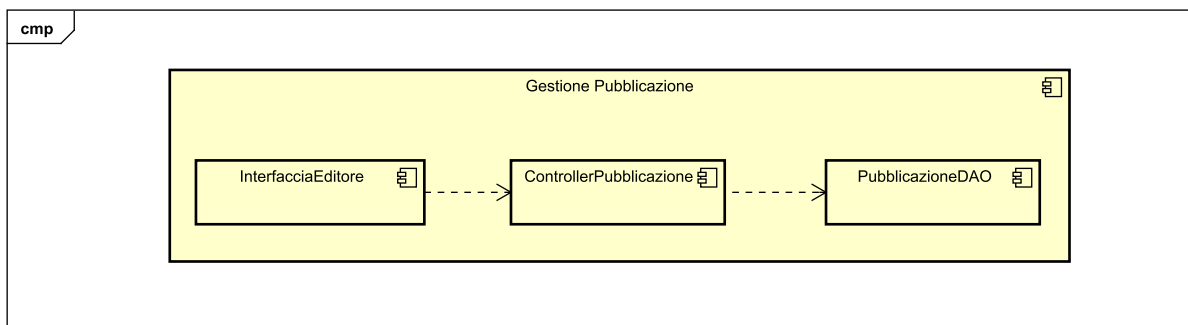
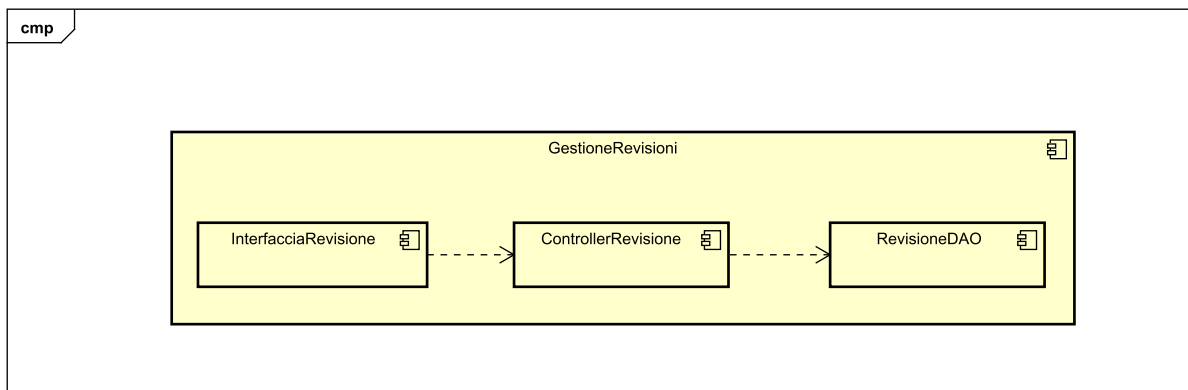
Il corretto funzionamento del software richiede una connessione a internet stabile, necessaria sia per consentire la comunicazione dei sottosistemi con il DBMS, sia per permettere al Sistema l'invio delle e-mail per il recupero delle credenziali e le notifiche; inoltre è necessaria una casella di posta elettronica per la ricezione delle suddette e-mail.

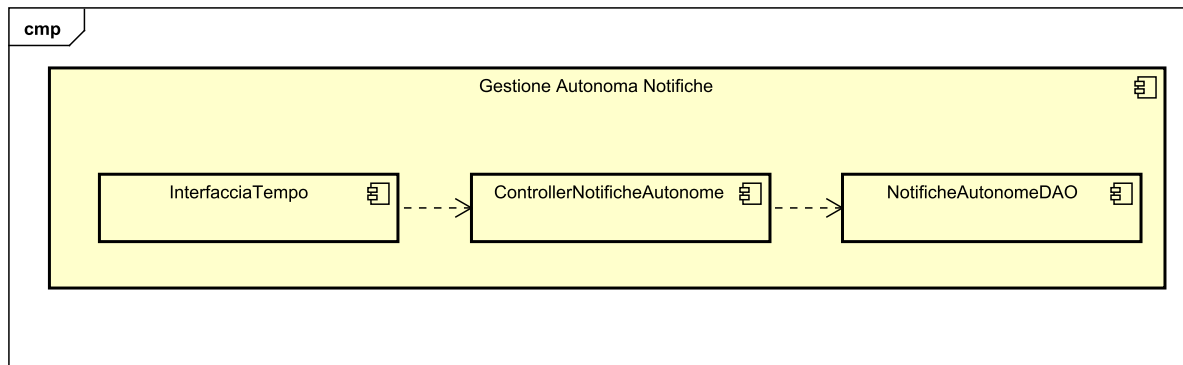
4.3 Scomposizione in sottosistemi

Di seguito viene rappresentata la suddivisione in sottosistemi del sistema proposto:









1. Sottosistema di Gestione Account

Il sottosistema Gestione Account contiene i componenti che permettono a qualsiasi utente di registrarsi al sistema, effettuare il login e il logout, gestire i dati del proprio profilo e recuperare le credenziali in caso di smarrimento. Questo sottosistema è responsabile della gestione centrale dei dati anagrafici, delle credenziali di accesso e dei ruoli degli utenti (es. Autore, Revisore, Chair). Tali dati sono resi disponibili nel repository centrale (DBMS) per essere utilizzati dagli altri sottosistemi al fine di autorizzare le operazioni specifiche per ogni ruolo.

2. Sottosistema di Gestione Conferenza

Il sottosistema Gestione Conferenza contiene la logica e le interfacce che permettono al Chair di una conferenza di gestire l'intero ciclo di vita dell'evento. Le funzionalità includono la creazione e configurazione di nuove conferenze, l'invito e la gestione dei membri del comitato di programma (Revisori), l'assegnazione degli articoli sottomessi ai revisori e il monitoraggio dello stato di avanzamento delle revisioni. Infine, in caso di necessità ha la possibilità di aggiungere una revisione per un qualunque articolo della conferenza.

Le assegnazioni e le configurazioni della conferenza vengono salvate nel DBMS, rendendole accessibili al sottosistema di Gestione Revisioni.

3. Sottosistema di Gestione Sottomissioni

Il sottosistema Gestione Sottomissioni contiene i componenti che permettono all'Autore di interagire con il sistema per la sottomissione dei propri lavori. L'autore può creare una nuova sottomissione caricandone il file, modificarla e ritirarla entro i termini stabiliti. Permette inoltre di visualizzare lo stato di avanzamento del proprio articolo nel processo di revisione e, al termine di esso, di scaricare le recensioni ricevute. I dati relativi agli articoli sottomessi vengono memorizzati nel DBMS, dove diventano visibili al sottosistema di Gestione Conferenze per il processo di assegnazione.

4. Sottosistema di Gestione Revisioni

Il sottosistema Gestione Revisioni rappresenta il cuore del processo di peer-review e fornisce gli strumenti necessari al Revisore per svolgere il proprio compito. Permette di visualizzare gli articoli assegnati, accettare o rinunciare all'incarico e sottomettere la propria valutazione. Include la logica per convocare un Sotto-revisore, delegando la

revisione, e per visualizzare e approvare il lavoro svolto da quest'ultimo. Permette inoltre ai Revisori di dichiarare le proprie aree di competenza e i conflitti d'interesse. Le revisioni completate vengono salvate nel DBMS, rendendole disponibili sia al sottosistema di Gestione Conferenze (per le decisioni finali del Chair) sia a quello di Gestione Sottomissioni (per la consultazione da parte dell'Autore).

5. Sottosistema di Gestione Pubblicazioni

Il sottosistema Gestione Pubblicazioni gestisce la fase finale del processo, fornendo all'Editore gli strumenti per la preparazione degli atti della conferenza. Permette di ottenere la lista di tutti gli articoli accettati e di scaricarne le versioni definitive singolarmente o in un unico file. Inoltre, consente all'Editore di notificare gli autori riguardo a eventuali problemi di formattazione grafica, scrivendo la relativa richiesta nel DBMS che verrà poi gestita dal sottosistema di Gestione Notifiche.

6. Sottosistema di Gestione Notifiche

Il sottosistema Gestione Notifiche fornisce all'utente l'interfaccia per interagire con le comunicazioni generate dal sistema. Permette a qualsiasi utente autenticato di visualizzare la lista delle proprie notifiche, leggerne il contenuto dettagliato e agire su di esse, ad esempio confermando la presa visione o accettando/rifiutando un invito (come quello a diventare revisore). Le notifiche visualizzate sono quelle generate e memorizzate nel repository centrale (DBMS) dagli altri sottosistemi.

7. Sottosistema di Gestione Sotto-Revisore

Il sottosistema Gestione Sotto-Revisore contiene i componenti dedicati a supportare il processo di revisione delegata. Questo sottosistema abilita il Sotto-revisore, dopo essere stato convocato da un Revisore principale, a partecipare al processo di peer-review. Nello specifico, permette al Sotto-revisore di visualizzare i dettagli dell'articolo che gli è stato assegnato, di compilare e sottomettere la propria recensione e, qualora non fosse in grado di svolgerla, di rinunciare all'incarico. La revisione completata viene salvata nel repository centrale (DBMS), rendendola così visibile al Revisore che ha effettuato la delega per la successiva validazione, la quale avviene all'interno del sottosistema di Gestione Revisioni.

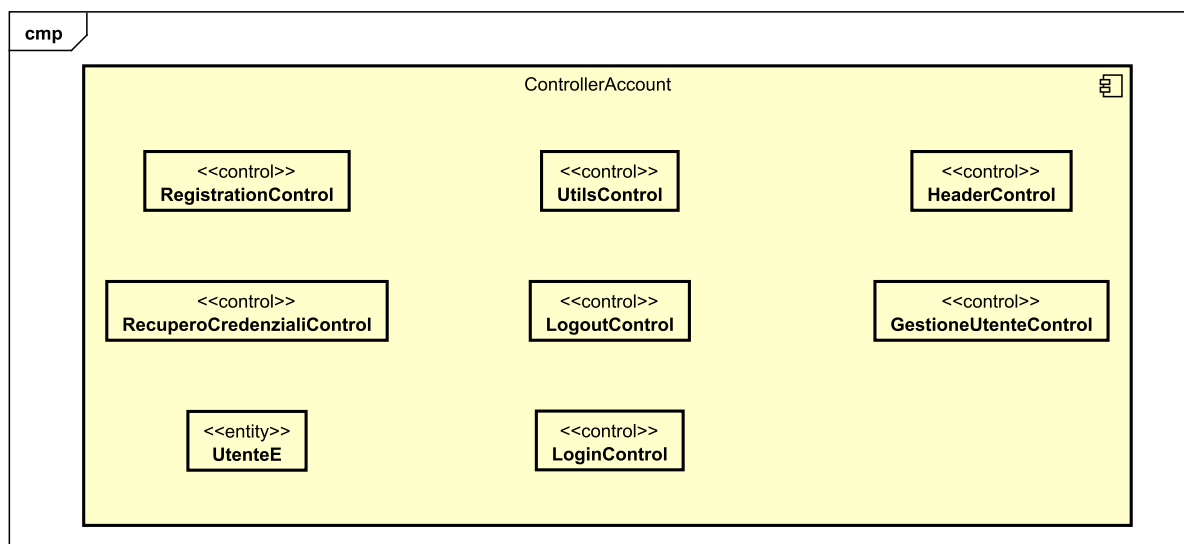
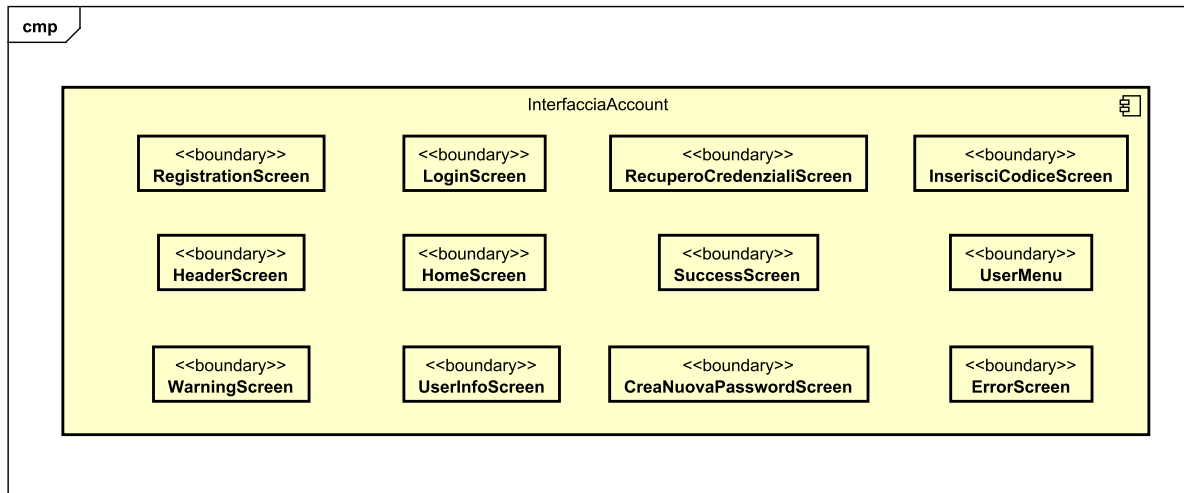
8. Sottosistema di Gestione Autonoma Notifiche

Il sottosistema Gestione Autonoma Notifiche contiene la logica che opera in modo indipendente dall'interazione diretta dell'utente, guidata unicamente dall'attore Tempo. La sua responsabilità è quella di monitorare lo stato del sistema e le scadenze temporali per generare automaticamente avvisi e promemoria. Ad esempio, crea notifiche per ricordare la scadenza di una sottomissione, per segnalare la necessità di assegnare revisori, o per avvisare di revisioni in ritardo. Questo sottosistema scrive le nuove notifiche direttamente nel repository (DBMS), rendendole così disponibili per la consultazione tramite il sottosistema di Gestione Notifiche.

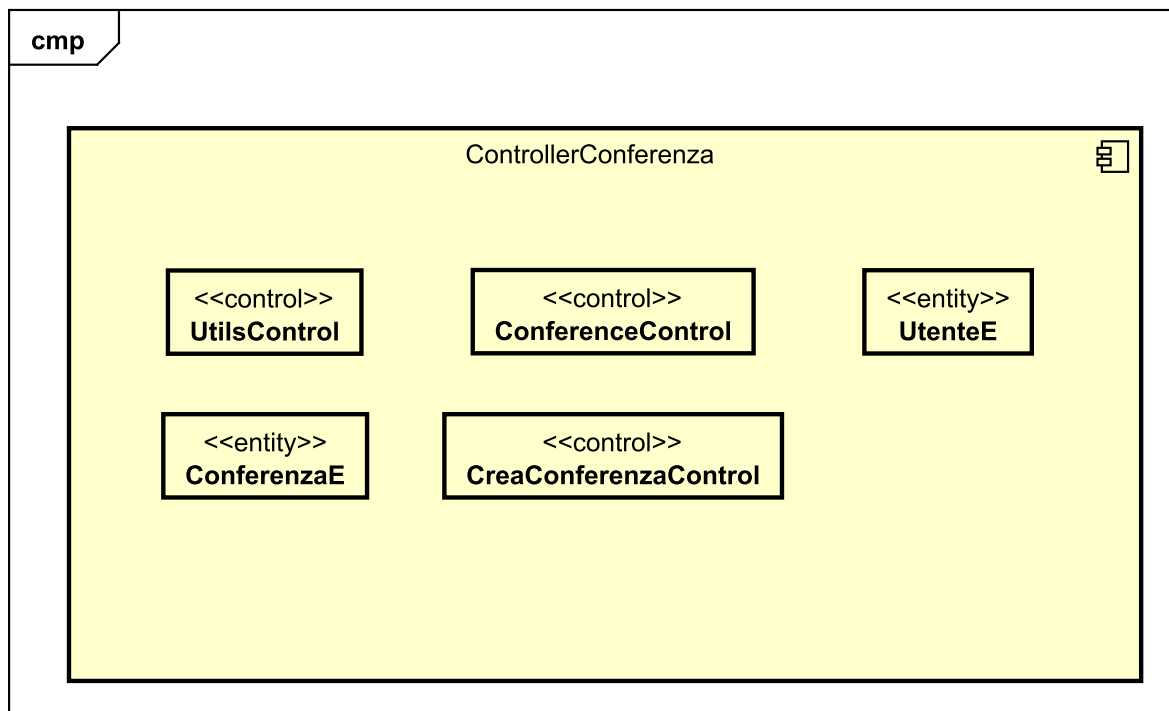
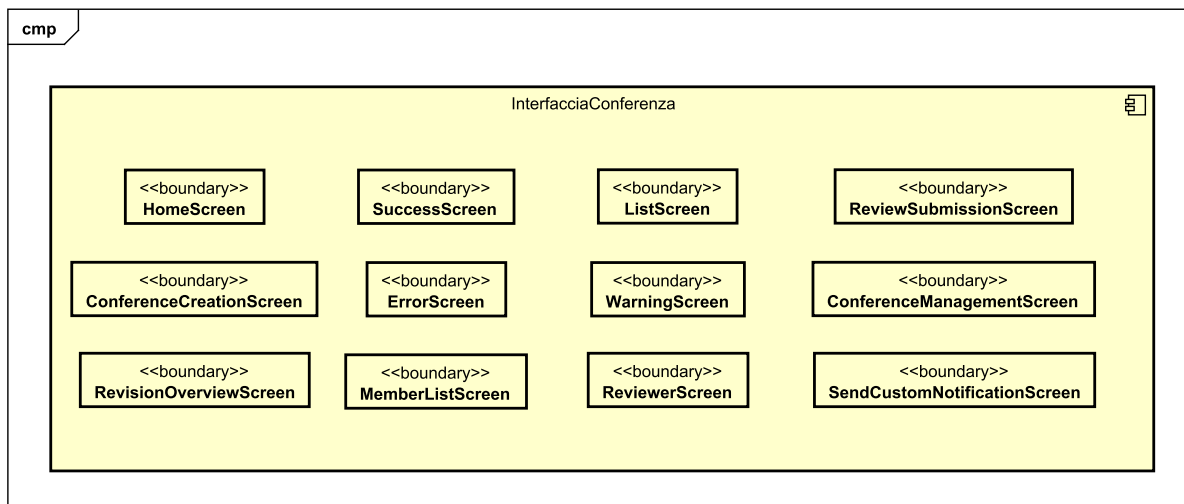
4.3.1 Mappatura degli oggetti nei sottosistemi

Di seguito sono riportate le funzionalità offerte dai sottosistemi

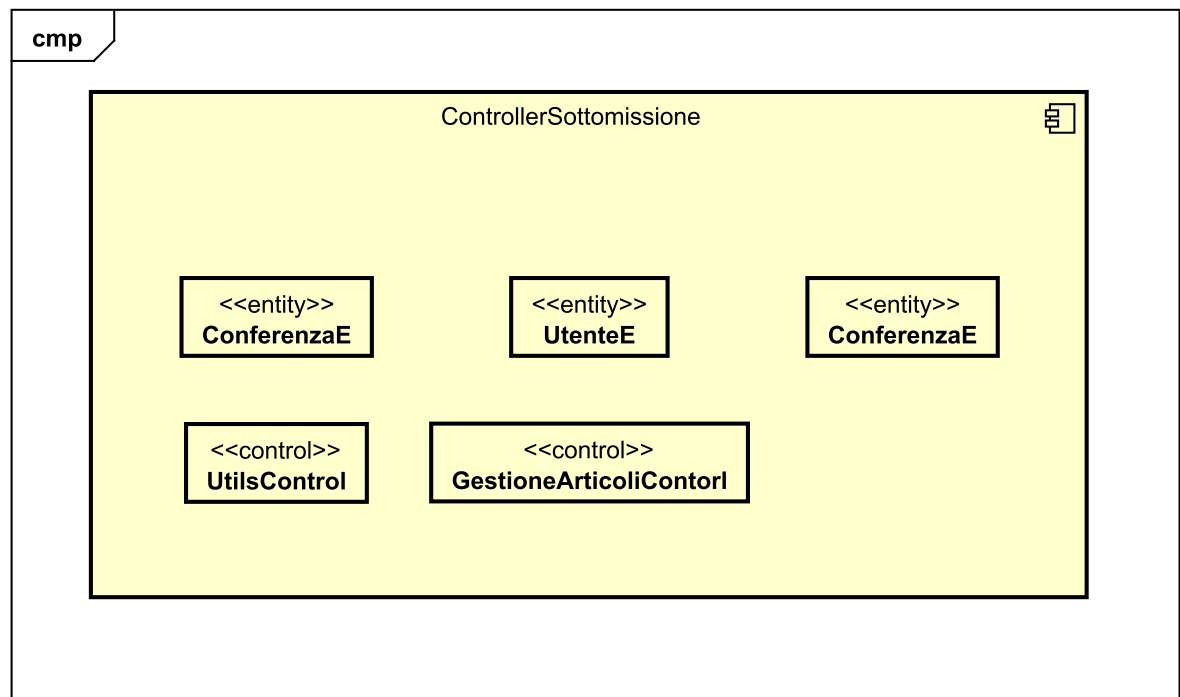
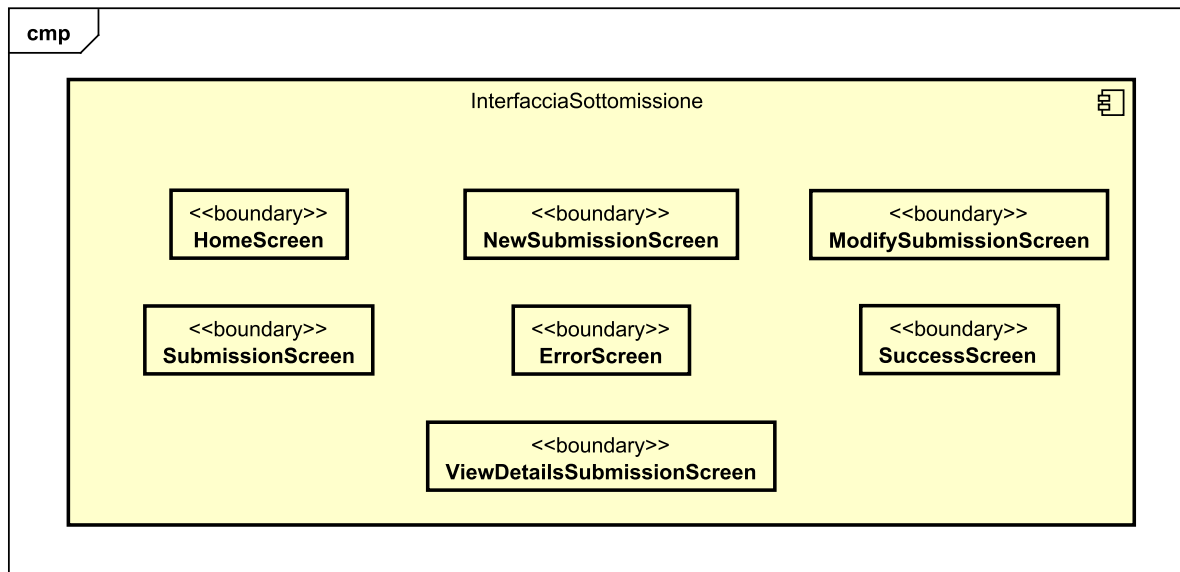
4.3.1.1 Gestione Account



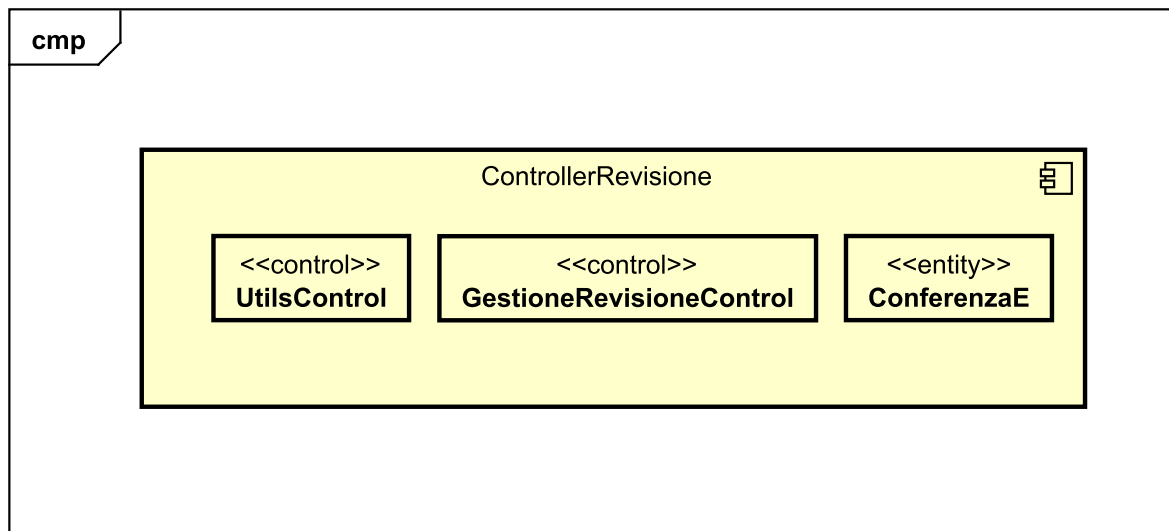
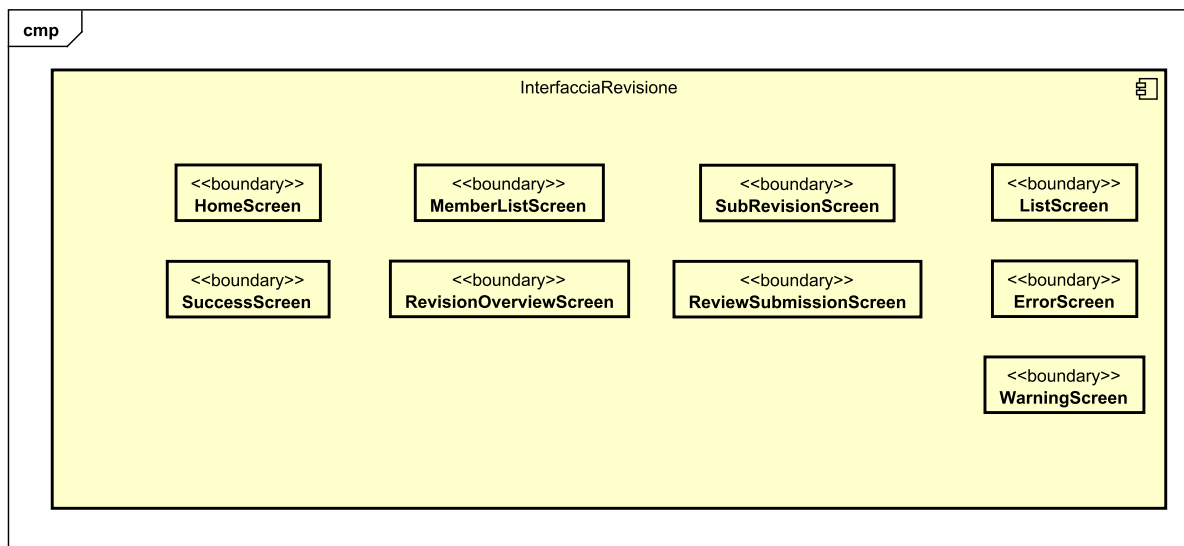
4.3.1.2 Gestione Conferenza



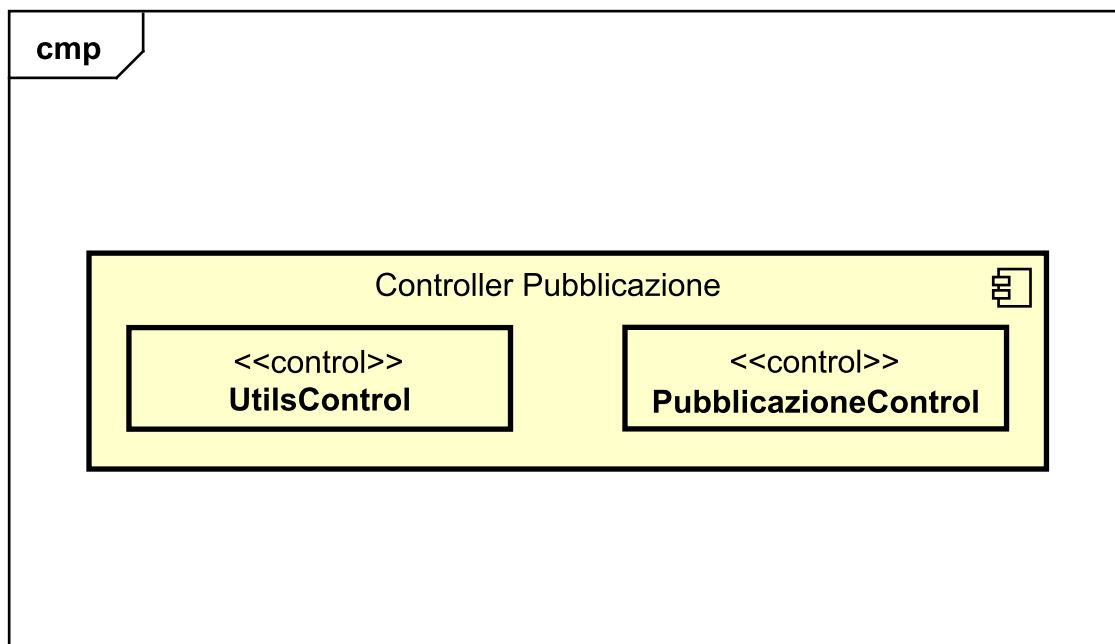
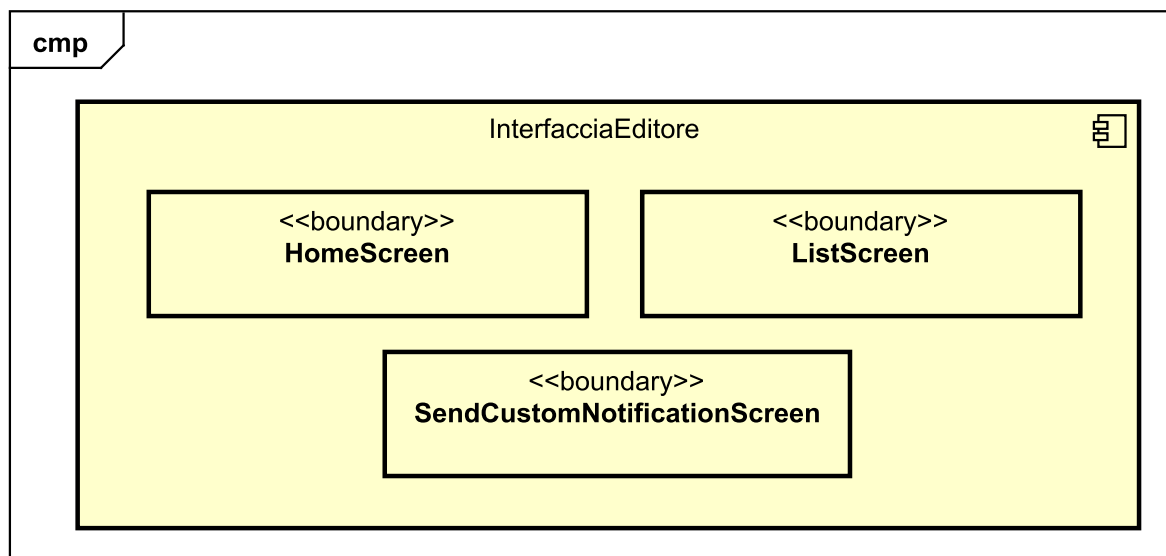
4.3.1.3 Gestione Sottomissioni



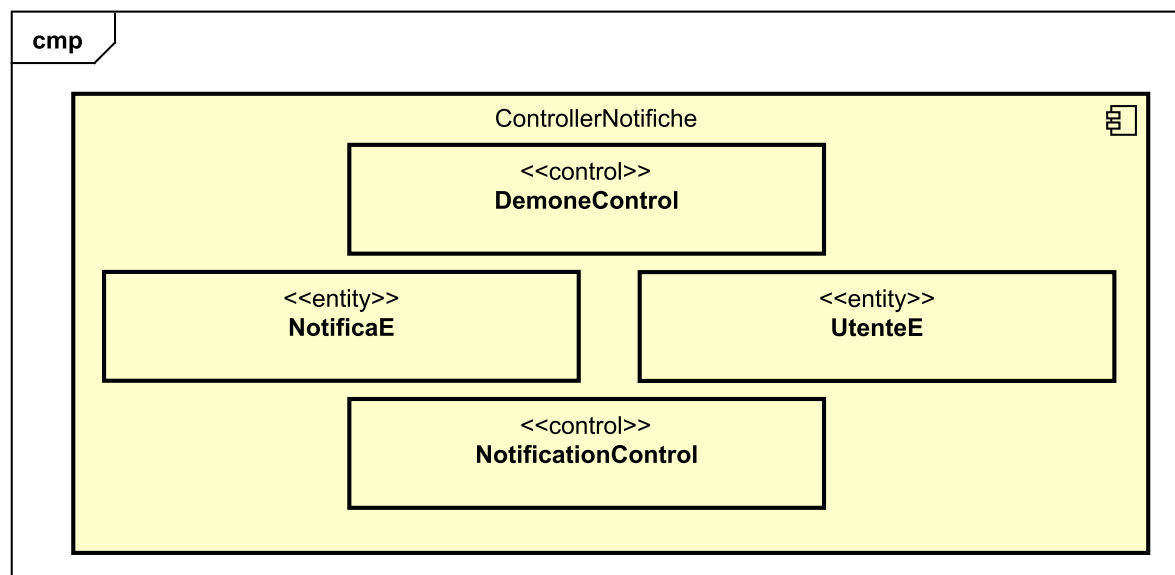
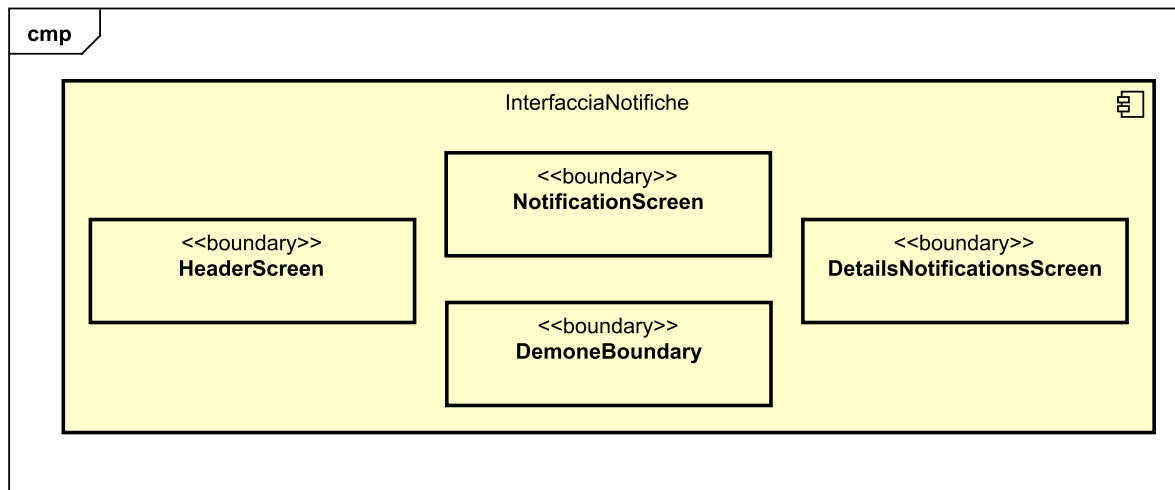
4.3.1.4 Gestione Revisioni



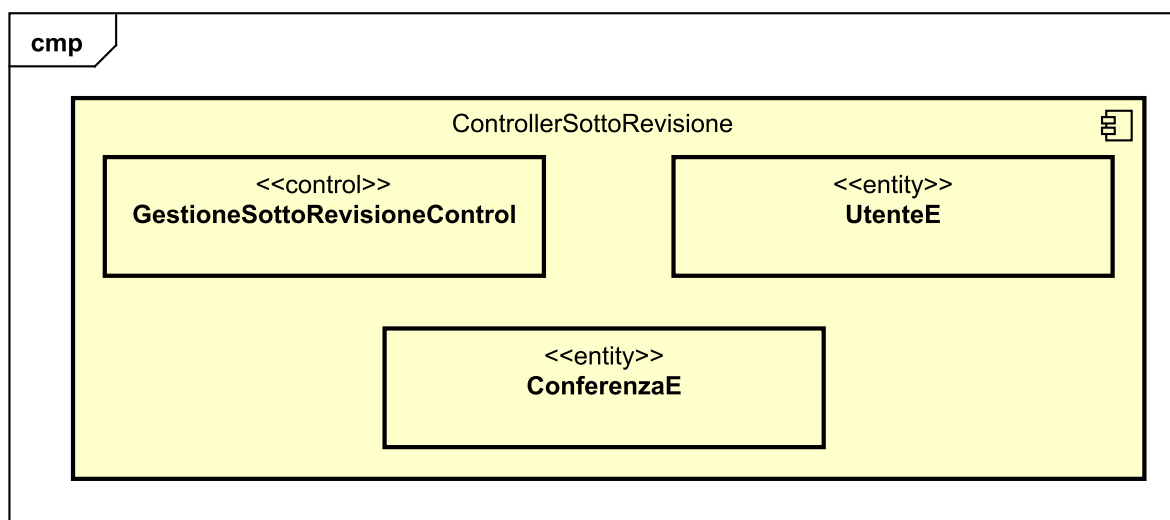
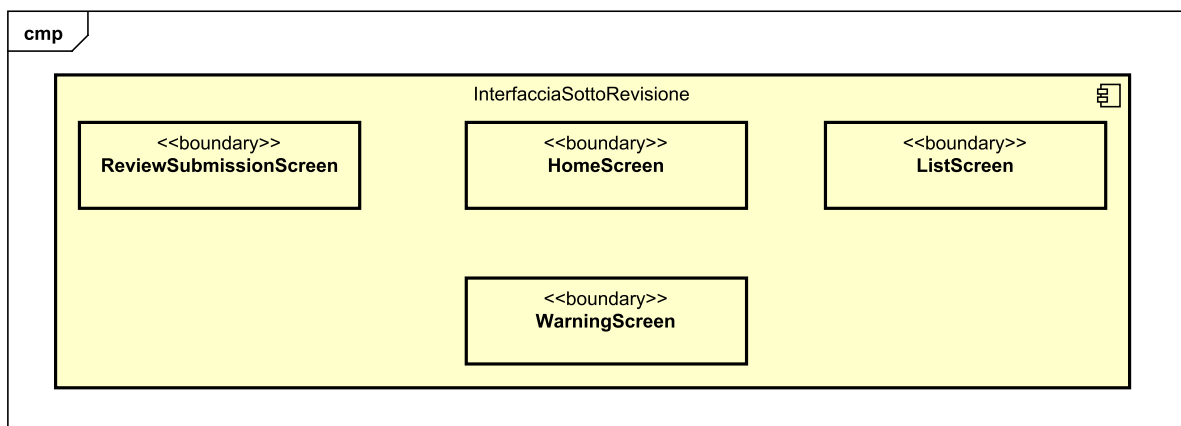
4.3.1.5 Gestione Pubblicazioni



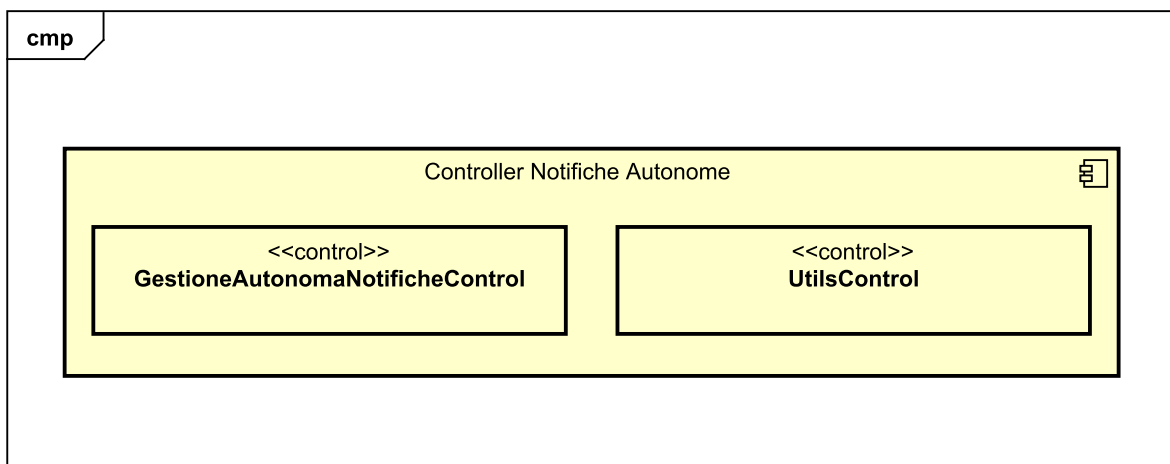
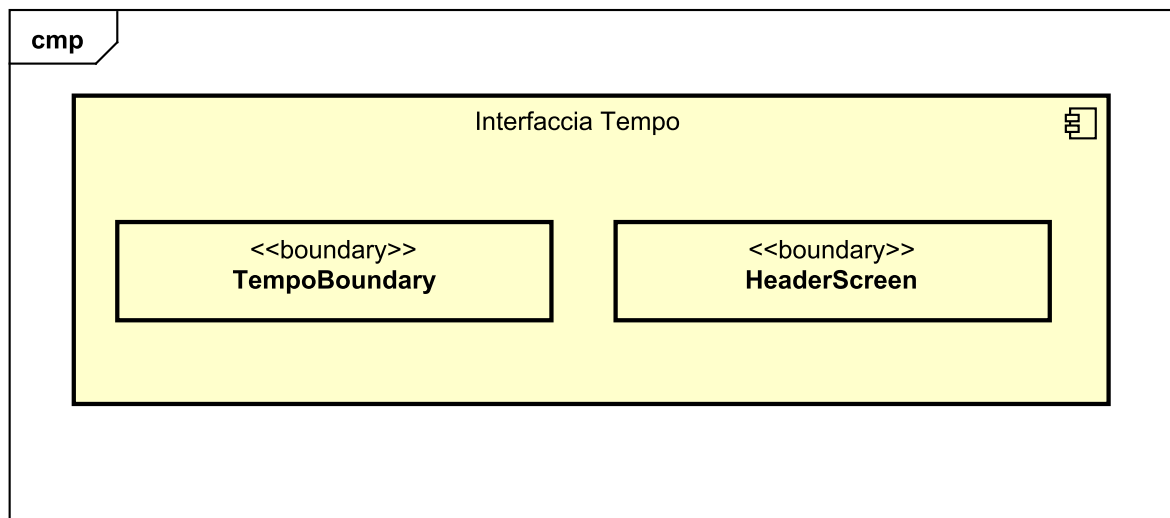
4.3.1.6 Gestione Notifiche



4.3.1.7 Gestione Sotto-Revisione



4.3.1.8 Gestione Autonoma Notifiche



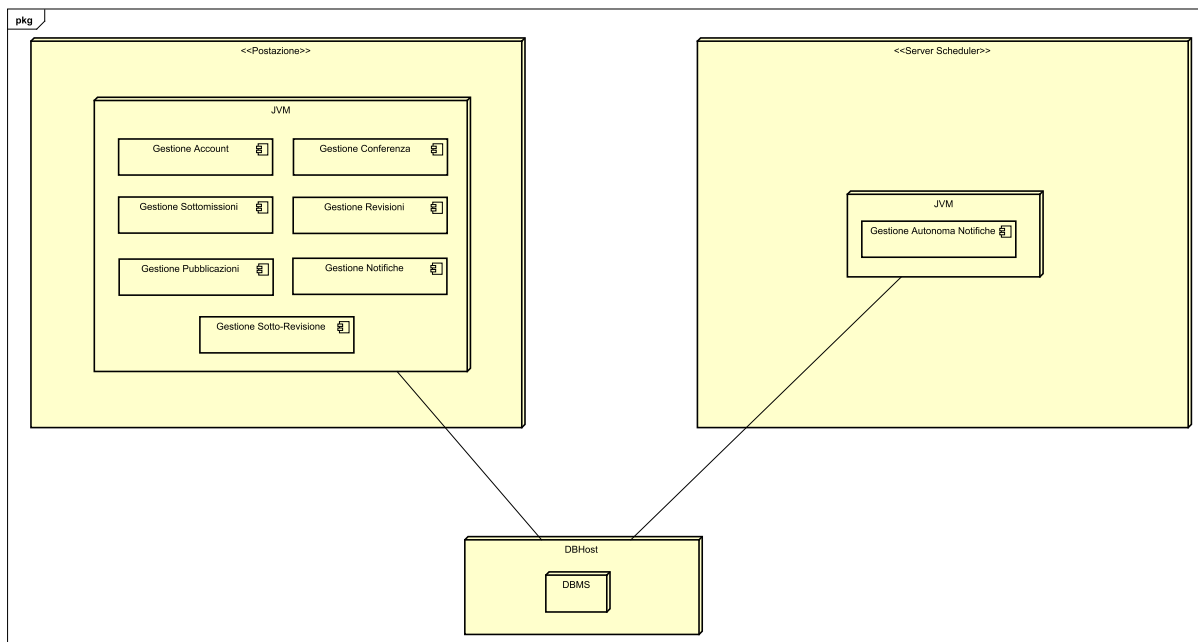
4.4 Mappatura Hardware/Software

La mappatura del sistema viene effettuata sul Modello architetturale Repository.

L'architettura è distribuita su diversi nodi fisica, che collaborano scambiando informazioni esclusivamente una repository di dati centrale (DBMS).

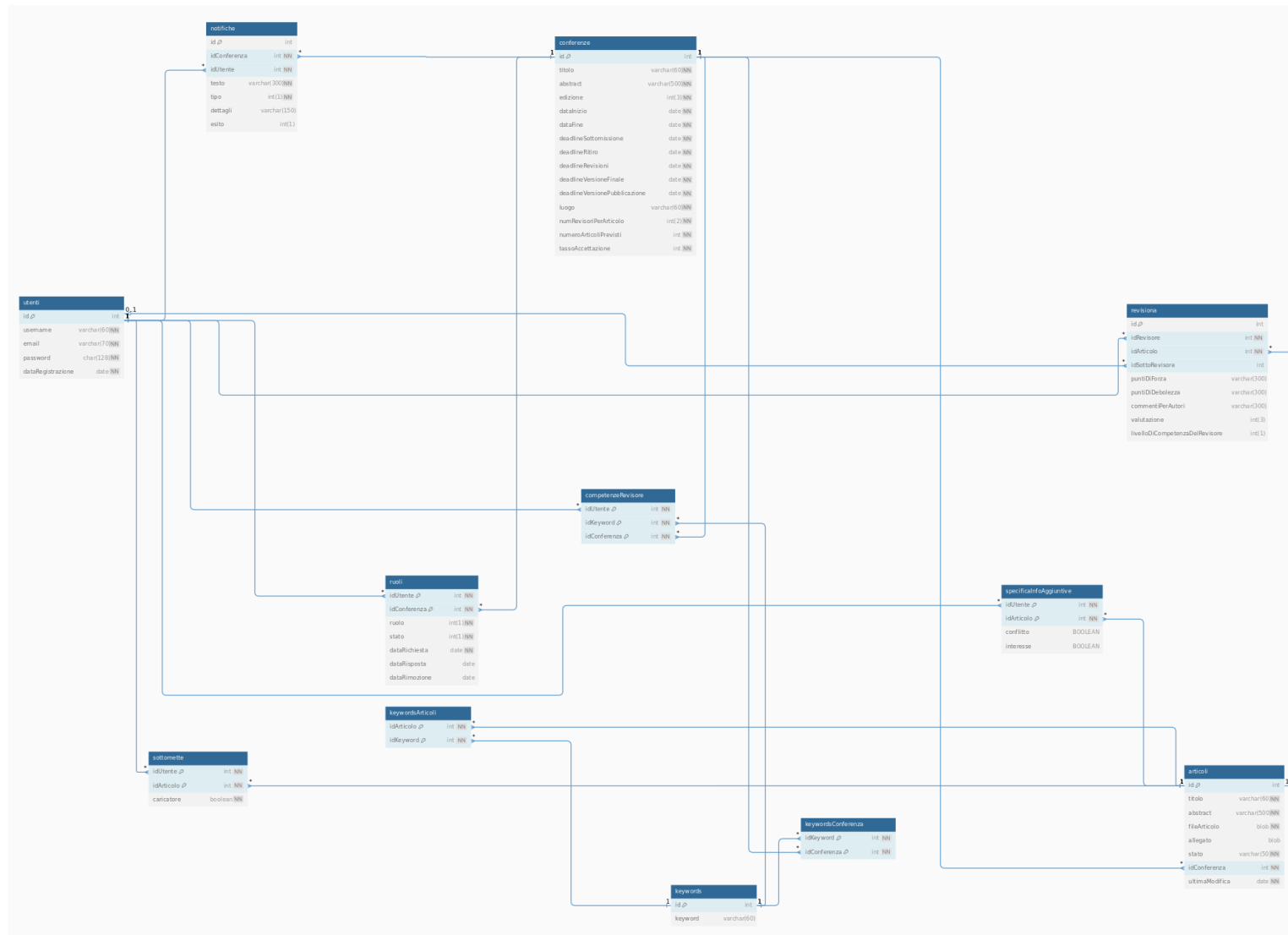
I nodi identificati sono:

- **Postazione Utente:** Rappresenta il nodo fisico su cui viene eseguita l'applicazione. Questo nodo ospita l'insieme dei sottosistemi che forniscono l'interfaccia e la logica per l'interazione dell'utente con i dati, permettendo agli utenti di leggere e modificare lo stato del sistema in base al proprio ruolo.
- **DBHost:** È il nodo che ospita fisicamente il Repository del sistema, ovvero il DBMS.
- **Server Scheduler:** È un nodo specializzato che ospita un ulteriore agente software che opera sul repository: il sottosistema di **Gestione autonoma notifiche**. Questo componente agisce in modo indipendente dalle postazioni utente per eseguire le logiche di invio notifiche/promemoria, accedendo ai dati del repository e modificandoli affinché gli altri agenti possano prenderne visione.



5 Gestione dei dati persistenti

5.1 Schema E-R



5.2 Modello relazionale

- ❖ **utenti** (id, username, email, password, dataRegistrazione)
PK: id
- ❖ **conferenze** (id, titolo, abstract, edizione, dataInizio, dataFine, deadlineSottomissione, deadlineRitiro, deadlineRevisioni, deadlineVersioneFinale, deadlineVersionePubblicazione, luogo, numRevisoriPerArticolo, numeroArticoliPrevisti, tassoAccettazione)
PK: id
- ❖ **keywords** (id, keyword)
PK: id
- ❖ **articoli** (id, titolo, abstract, fileArticolo, allegato, stato, idConferenza, ultimaModifica)
PK: id
FK: idConferenza references Conferenza(id)
- ❖ **notifiche** (id, idConferenza, idUtente, testo, tipo, dettagli, esito)
PK: id
FK: idConferenza references Conferenza(id), idUtente references Utente(id)
- ❖ **ruoli** (idUtente, idConferenza, ruolo, stato, dataRichiesta, dataRisposta, dataRimozione)
PK: idUtente, idConferenza
FK: idUtente references Utente(id), idConferenza references Conferenza(id)
- ❖ **specificInfoAggiuntive** (idUtente, idArticolo, conflitto, interesse)
PK: idUtente, idArticolo
FK: idUtente references Utente(id), idArticolo references Articolo(id)
- ❖ **revisiona** (id, idRevisore, idArticolo, idSottoRevisore, puntiDiForza, puntiDiDebolezza, commentiPerAutori, valutazione, livelloDiCompetenzaDelRevisore)
PK: id
FK: idRevisore references Utente(id), idArticolo references Articolo(id), idSottoRevisore references Utente(id)
- ❖ **sottomette** (idUtente, idArticolo, caricatore)
PK: idUtente, idArticolo
FK: idUtente references Utente(id), idArticolo references Articolo(id)
- ❖ **keywordsArticoli** (idArticolo, idKeyword)
PK: idArticolo, idKeyword
FK: idArticolo references Articolo(id), idKeyword references Keyword(id)

❖ **competenzeRevisore** (idUtente, idKeyword, idConferenza)

PK: idUtente, idKeyword, idConferenza

FK: idUtente references Utente(id), idKeyword references Keyword(id), idConferenza references Conferenza(id)

❖ **keywordsConferenza** (idKeyword, idConferenza)

PK: idKeyword, idConferenza

FK: idKeyword references keywords(id), idConferenza references conferenze(id)

5.3 Struttura delle tabelle

5.3.1 utenti

Nome attributo	Tipo	Vincoli	Descrizione
id	int	PK, Autoincrement	Codice identificativo univoco dell'utente.
username	varchar(60)	Not null, Unique	Nome utente univoco scelto per l'accesso al sistema.
email	varchar(70)	Not null	Indirizzo email dell'utente, usato per le comunicazioni.
password	char(128)	Not null	Password di accesso dell'utente (memorizzata come hash).
dataRegistrazione	date	Not null	Data in cui l'utente ha creato il proprio account.

5.3.2 conferenze

Nome attributo	Tipo	Vincoli	Descrizione
id	int	PK, Autoincrement	Identificativo univoco della conferenza.
titolo	varchar(60)	Not null, Unique (con edizione)	Il titolo ufficiale della conferenza.
abstract	varchar(500)	Not null	Breve sommario o descrizione della conferenza.
edizione	int(3)	Not null, Default 1, Unique (con titolo)	Numero dell'edizione della conferenza.
dataInizio	date	Not null, Check	Data di inizio effettivo della conferenza.
dataFine	date	Not null, Check	Data di fine della conferenza.

deadlineSottomissione	date	Not null, Check	Data ultima per la sottomissione degli articoli.
deadlineRitiro	date	Not null, Check	Data ultima per il ritiro di un articolo sottomesso.
deadlineRevisioni	date	Not null, Check	Data ultima per il completamento delle revisioni.
deadlineVersioneFinale	date	Not null, Check	Data ultima per la sottomissione della versione finale corretta.
deadlineVersionePubblicazione	date	Not null, Check	Data ultima per la sottomissione della versione per la pubblicazione.
luogo	varchar(60)	Not null	Luogo fisico o virtuale in cui si tiene la conferenza.
numRevisoriPerArticolo	int(2)	Not null	Numero di revisori assegnati a ciascun articolo.
numeroArticoliPrevisti	int	Not null	Numero di articoli che la conferenza prevede di gestire.
tassoAccettazione	int	Not null	Tasso di accettazione degli articoli, valore percentuale tra 1 e 100.
<i>(vincolo di tabella)</i>	-	CHECK	Le date devono seguire un ordine cronologico logico.

5.3.3 keywords

Nome attributo	Tipo	Vincoli	Descrizione
id	int	PK, Autoincrement	Identificativo univoco della keyword.
keyword	varchar(60)	Unique	La parola chiave o etichetta tematica.

5.3.4 articoli

Nome attributo	Tipo	Vincoli	Descrizione
id	int	PK, Autoincrement	Identificativo univoco dell'articolo.
titolo	varchar(60)	Not null	Titolo dell'articolo sottomesso.
abstract	varchar(500)	Not null	Breve sommario del contenuto dell'articolo.
fileArticolo	blob	Not null	Contenuto binario del file principale dell'articolo (es. PDF).
allegato	blob	-	Contenuto binario di un eventuale file allegato.
stato	varchar(50)	Not null, Default 'Inviato'	Stato attuale dell'articolo nel processo di revisione.
idConferenza	int	Not null, FK references conferenze(id)	Conferenza a cui l'articolo è stato sottomesso.
ultimaModifica	date	Not null	Data dell'ultima modifica effettuata sull'articolo.

5.3.5 notifiche

Nome attributo	Tipo	Vincoli	Descrizione
id	int	PK, Autoincrement	Identificativo univoco della notifica.
idConferenza	int	Not null, FK references conferenze(id)	Conferenza a cui si riferisce la notifica.
idUtente	int	Not null, FK references utenti(id)	Utente destinatario della notifica.

testo	varchar(300)	Not null	Contenuto testuale del messaggio di notifica.
tipo	int(1)	Not null, Default 0	Tipologia di notifica (0: presa visione, 1: accetta/rifiuta).
dettagli	varchar(150)	-	Informazioni aggiuntive sulle conseguenze di un'azione.
esito	int(1)	Default NULL	Esito dell'interazione dell'utente (NULL: nessuna azione, 1: accettato, 2: rifiutato, 3: presa visione).

5.3.6 ruoli

Nome attributo	Tipo	Vincoli	Descrizione
idUtente	int	Not null, PK, FK references utenti(id)	Identificativo dell'utente a cui è assegnato il ruolo.
idConferenza	int	Not null, PK, FK references conferenze(id)	Identificativo della conferenza in cui è valido il ruolo.
ruolo	int(1)	Not null	Ruolo ricoperto (1: chair, 2: revisore, 3: editore).
stato	int(1)	Not null, Default 0	Stato della richiesta di ruolo (0: Pending, 1: Accettato, 2: Rifiutato, 3: Rimosso).
dataRichiesta	date	Not null	Data in cui il ruolo è stato richiesto o assegnato.
dataRisposta	date	Default NULL	Data in cui l'utente ha accettato o rifiutato il ruolo.
dataRimozione	date	Default NULL	Data in cui il ruolo è stato revocato.

5.3.7 specificaInfoAggiuntive

Nome attributo	Tipo	Vincoli	Descrizione
idUtente	int	Not null, PK, FK references utenti(id)	Utente (revisore) che fornisce l'informazione.
idArticolo	int	Not null, PK, FK references articoli(id)	Articolo a cui si riferisce l'informazione.
conflitto	boolean	Default NULL	Indica se il revisore ha un conflitto di interessi con l'articolo.
interesse	boolean	Default NULL	Indica se il revisore è interessato a revisionare l'articolo.

5.3.8 revisiona

Nome attributo	Tipo	Vincoli	Descrizione
id	int	PK, Autoincrement	Identificativo univoco della revisione.
idRevisore	int	Not null, FK references utenti(id)	Utente che ha effettuato la revisione principale.
idArticolo	int	Not null, FK references articoli(id)	Articolo che è stato revisionato.
idSottoRevisore	int	Default NULL, FK references utenti(id)	Eventuale sotto-revisore a cui è stata delegata la revisione.
puntiDiForza	varchar(300)		Commenti sui punti di forza dell'articolo.
puntiDiDebolezza	varchar(300)		Commenti sui punti di debolezza dell'articolo.
commentiPerAutori	varchar(300)		Commenti e suggerimenti destinati agli autori.
valutazione	int(3)	Check (tra 1 e 100)	Punteggio complessivo della revisione, da 1 a 100.

livelloDiCompetenzaDelRevisore	int(1)	Default 1, Check (tra 1 e 10)	Livello di competenza del revisore sull'argomento, da 1 a 10.
---------------------------------------	--------	-------------------------------	---

5.3.9 sottomette

Nome attributo	Tipo	Vincoli	Descrizione
idUtente	int	Not null, PK, FK references utenti(id)	Utente (autore) che ha sottomesso l'articolo.
idArticolo	int	Not null, PK, FK references articoli(id)	Articolo sottomesso.
caricatore	boolean	Not null, Default FALSE	Indica se questo utente è colui che ha materialmente caricato il file (TRUE) o è solo un co-autore (FALSE).

5.3.10 keywordsArticoli

Nome attributo	Tipo	Vincoli	Descrizione
idArticolo	int	Not null, PK, FK references articoli(id)	Identificativo dell'articolo.
idKeyword	int	Not null, PK, FK references keywords(id)	Identificativo della keyword associata all'articolo.

5.3.11 competenzeRevisore

Nome attributo	Tipo	Vincoli	Descrizione
idUtente	int	Not null, PK, FK references utenti(id)	Utente (revisore) di cui si dichiara la competenza.
idKeyword	int	Not null, PK, FK references keywords(id)	Keyword che definisce l'area di competenza.
idConferenza	int	Not null, PK, FK references conferenze(id)	Conferenza per la quale viene dichiarata la competenza.

5.3.12 keywordsConferenza

Nome attributo	Tipo	Vincoli	Descrizione
idKeyword	int	Not null, PK, FK references keywords(id)	Id della keyword appartenente alla conferenza.
idConferenza	int	Not null, PK, FK, references conferenze(id)	Conferenza a cui appartiene la keyword specificata.