



**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

# **Audire**

## **System Design Document**

Versione 1.5



25 Novembre 2025

Gabriele Malanga - 0512119344  
Gennaro Carmine Tozza - 0512120382

Tabella 1: Revision History

| <b>Data</b> | <b>Versione</b> | <b>Descrizione</b>       | <b>Autore</b>                                 |
|-------------|-----------------|--------------------------|---|
| 13/11/2025  | 1.0             | Creazione documento      | Gennaro Carmine Tozza                         |
| 14/11/2025  | 1.1             | Stesura sezione 1        | Gennaro Carmine Tozza                         |
| 15/11/2025  | 1.2             | Inizio Stesura sezione 2 | Gabriele Malanga,<br>Gennaro Carmine<br>Tozza |
| 17/11/2025  | 1.3             | Fine Stesura sezione 2   | Gabriele Malanga,<br>Gennaro Carmine<br>Tozza |
| 21/11/2025  | 1.4             | Stesura sezione 3        | Gennaro Carmine Tozza                         |
| 21/11/2025  | 1.5             | Aggiunto glossario       | Gennaro Carmine Tozza                         |

# Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduzione</b>                                 | <b>4</b>  |
| 1.1      | Descrizione del Problema . . . . .                  | 4         |
| 1.2      | Design goals . . . . .                              | 4         |
| 1.2.1    | DG.1 - Criteri di Performance . . . . .             | 5         |
| 1.2.2    | DG.2 - Criteri di Affidabilità . . . . .            | 5         |
| 1.2.3    | DG.3 - Criteri di Costo . . . . .                   | 5         |
| 1.2.4    | DG.4 - Criteri di Mantenimento . . . . .            | 5         |
| 1.2.5    | DG.5 - Criteri di Utente . . . . .                  | 5         |
| 1.3      | Trade-off . . . . .                                 | 6         |
| 1.4      | Definizioni, acronimi e abbreviazioni . . . . .     | 6         |
| 1.5      | Riferimenti . . . . .                               | 6         |
| 1.6      | Panoramica . . . . .                                | 7         |
| <b>2</b> | <b>Architettura del Sistema</b>                     | <b>7</b>  |
| 2.1      | Panoramica . . . . .                                | 7         |
| 2.2      | Decomposizione in sottosistemi . . . . .            | 7         |
| 2.2.1    | Macrodecomposizione in sistemi . . . . .            | 7         |
| 2.2.2    | Microdecomposizione in sottosistemi . . . . .       | 8         |
| 2.3      | Mappatura Hardware/software . . . . .               | 19        |
| 2.4      | Gestione dei Dati Persistenti . . . . .             | 19        |
| 2.4.1    | Modello logico . . . . .                            | 20        |
| 2.4.2    | Tabelle . . . . .                                   | 20        |
| 2.5      | Controllo degli accessi . . . . .                   | 22        |
| 2.6      | Controllo software globale . . . . .                | 23        |
| 2.7      | Condizioni limite . . . . .                         | 23        |
| 2.7.1    | Start-up . . . . .                                  | 23        |
| 2.7.2    | Start-up (a seguito di un fallimento) . . . . .     | 24        |
| 2.7.3    | Terminazione . . . . .                              | 24        |
| 2.7.4    | Fallimento . . . . .                                | 24        |
| <b>3</b> | <b>Servizi dei sottosistemi</b>                     | <b>25</b> |
| 3.1      | SS_ACC - Gestione Account . . . . .                 | 25        |
| 3.2      | SS_PROD - Gestione Produzione . . . . .             | 25        |
| 3.3      | SS_TEAM - Gestione Team . . . . .                   | 26        |
| 3.4      | SS_CAST - Gestione Casting . . . . .                | 26        |
| 3.5      | SS_CAND - Gestione Candidature . . . . .            | 26        |
| 3.6      | SS_VAL - Gestione Valutazione Candidature . . . . . | 27        |
| 3.7      | SS_MON - Monitoraggio Produzione . . . . .          | 27        |
| <b>4</b> | <b>Glossario</b>                                    | <b>27</b> |

# 1 Introduzione

## 1.1 Descrizione del Problema

Audire è una piattaforma web progettata per centralizzare e digitalizzare l'intero processo di casting nel settore dell'intrattenimento (cinema, teatro, televisione). Il sistema ha lo scopo di superare le inefficienze del processo tradizionale di casting, che attualmente si basa su comunicazioni frammentate tramite email, fogli di calcolo e messaggi non strutturati.

Il sistema facilita l'incontro tra talenti (performer) e opportunità professionali, fornendo una piattaforma unificata dove:

- I **Production Manager** possono creare e gestire produzioni
- I **Casting Director** possono pubblicare annunci di casting per delle specifiche produzioni e valutare le candidature
- I **Performer** possono scoprire opportunità pertinenti e candidarsi

Il sistema copre l'intero workflow del processo di casting, dalla creazione di una produzione fino alla selezione finale dei candidati, includendo:

- Gestione completa degli utenti con tre ruoli distinti (Performer, Casting Director, Production Manager)
- Creazione e gestione di produzioni (film, serie TV, teatro, pubblicità)
- Pubblicazione e gestione di annunci di casting
- Sistema di ricerca e filtro per casting aperti
- Invio e tracciamento delle candidature
- Valutazione strutturata dei candidati con feedback
- Dashboard personalizzate per ogni tipologia di utente
- Strumenti di supervisione per i Production Manager

## 1.2 Design goals

Il sistema Audire è progettato per essere una piattaforma web moderna, intuitiva ed efficiente. Per raggiungere questi obiettivi, il sistema rispetta i seguenti criteri di design:

### 1.2.1 DG\_1 - Criteri di Performance

**DG\_1.1 Tempi di risposta:** Il sistema deve garantire tempi di risposta brevi per ogni funzionalità. Le pagine del sistema devono caricarsi in meno di 3 secondi su connessioni standard (10 Mbps). I risultati delle ricerche di casting devono essere visualizzati in meno di 2 secondi

**DG\_1.2 Upload File:** Il sistema deve supportare l'upload di file fino a 10 MB con feedback visivo sul progresso. I file caricati (foto profilo,CV) devono essere elaborati e memorizzati in modo efficiente.

**DG\_1.3 Dati Persistenti:** Il sistema utilizzerà un database relazionale (MySQL) per memorizzare tutti i dati relativi agli utenti, produzioni,casting e candidature, garantendo accesso rapido e affidabile alle informazioni.

### 1.2.2 DG\_2 - Criteri di Affidabilità

**DG\_2.1 Robustezza:** Il sistema deve gestire gracefully gli errori, mostrando messaggi informativi all'utente e loggando gli errori per analisi successive. Tutti i form devono implementare validazione in tempo reale con messaggi di errore chiari, specifici e costruttivi.

**DG\_2.2 Affidabilità:** Il sistema deve garantire il corretto svolgimento di tutte le funzionalità e produrre l'output atteso evitando errori indesiderati. Le operazioni critiche (e.g. modifica status) devono essere transazionali per garantire la consistenza dei dati.

**DG\_2.3 Sicurezza:** Il sistema deve proteggere i dati sensibili degli utenti attraverso autenticazione sicura, crittografia delle comunicazioni(HTTPS), e controllo degli accessi basato sui ruoli. Le password devono essere sottoposte a hashing con algoritmi sicuri (bcrypt o Argon2).

### 1.2.3 DG\_3 - Criteri di Costo

**DG\_3.1 Costo:** Prima dell'inizio dello sviluppo della piattaforma verranno valutati i costi di sviluppo del sistema, costi di manutenzione e costi di amministrazione.

### 1.2.4 DG\_4 - Criteri di Mantenimento

**DG\_4.1 Modificabilità:** il sistema permetterà di apportare modifiche alle funzionalità già implementate senza la necessità di modificare i sottosistemi.

### 1.2.5 DG\_5 - Criteri di Utente

**DG\_5.1 Utilità:** Il sistema deve soddisfare tutte le esigenze degli utenti sviluppando tutti i requisiti funzionali identificati in fase di analisi,fornendo valore

reale a tutti gli stakeholder (Performer, Casting Director, Production Manager).

**DG\_5.2 Usabilità:** L'interfaccia utente deve essere chiara, moderna e facile da navigare anche per utenti non esperti di tecnologia. Il sistema deve essere completamente funzionale e ottimizzato per dispositivi desktop, tablet e smartphone (responsive design).

### 1.3 Trade-off

| Trade-off                | Descrizione   |
|--------------------------|---|
| Performance vs Sicurezza | Si è scelto di privilegiare la sicurezza dei dati degli utenti rispetto a miglioramenti marginali delle performance, le quali potrebbero aumentare tempi di risposta poco superiori ai 3 secondi. |

### 1.4 Definizioni, acronimi e abbreviazioni

- **RAD:** Requirements Analysis Document - documento di analisi dei requisiti
- **SDD:** System Design Document - documento di design del sistema
- **PM:** Production Manager - responsabile di produzione
- **CD:** Casting Director - responsabile del casting
- **DBMS:** Database Management System - sistema di gestione database
- **MVC:** Model-View-Controller - pattern architetturale
- **HTTP/HTTPS:** HyperText Transfer Protocol (Secure) - protocollo di comunicazione web

### 1.5 Riferimenti

1. Requirements Analysis Document (RAD) Audire, Versione 2.3, 11 Novembre 2025, Gabriele Malanga e Gennaro Carmine Tozza
2. System Design Document Template, TUM Applied Software Engineering
3. Object-Oriented Software Engineering Using UML, Patterns, and Java, Third Edition, Bernd Bruegge & Allen H. Dutoit
4. Materiale didattico del Corso di Ingegneria del Software, Università degli Studi di Salerno, A.A. 2025/2026

## 1.6 Panoramica

Questo documento è strutturato come segue:

- **Sezione 1 - Introduzione:** Descrive gli obiettivi del sistema proposto e saranno descritte, in modo non approfondito, le varie funzionalità messe a disposizione per gli attori. Mostra, inoltre, gli obiettivi del design, in particolar modo i criteri che il sistema dovrà rispettare.
- **Sezione 2: Architettura del sistema:** Questa sezione definisce l'elaborazione della macrocomposizione in sottosistemi, la gestione dei dati persistenti, il mapping hardware/software del sistema, il controllo degli accessi, sicurezza e le condizioni limite.
- **Sezione 3 - Glossario dei Servizi dei sottosistemi:** Fornisce la descrizione per ogni servizio fornito del sottosistema e un glossario contenente una raccolta dei termini contenuti nel sistema proposto.

## 2 Architettura del Sistema

### 2.1 Panoramica

Il sistema prevederà l'utilizzo di un architettura ibrida tra Client-Server e Model-View-Controller.

- Il Lato Client si occuperà delle pagine di presentazione del sistema(View)
- Il Lato Server si occuperà della logica applicativa e di controllo (rispettivamente Model e Controller).

### 2.2 Decomposizione in sottosistemi

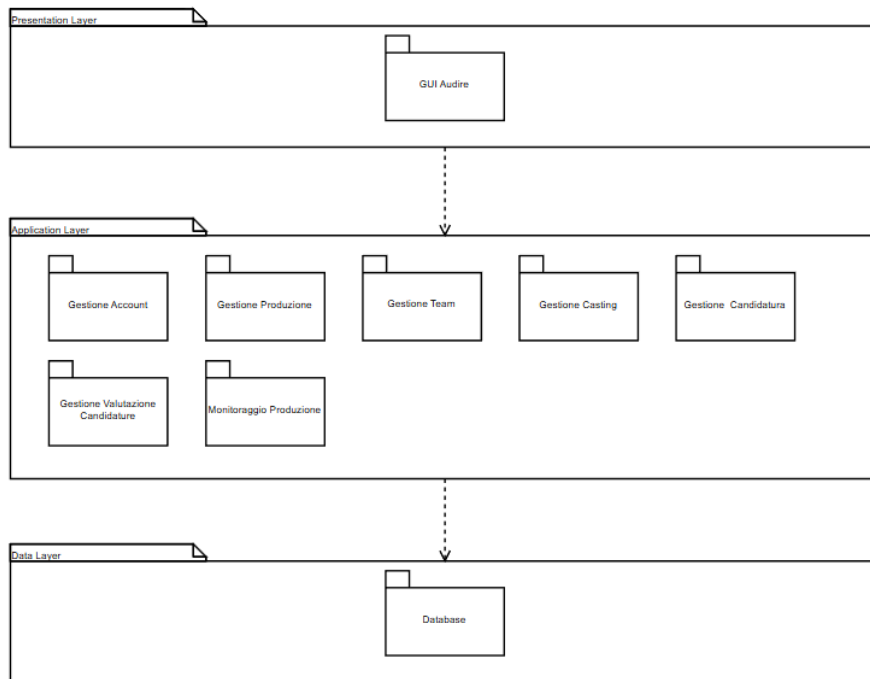
#### 2.2.1 Macrodecomposizione in sistemi

La macro decomposizione divide il sistema nei seguenti sottosistemi principali, basati sulle funzionalità identificate nel RAD:

1. Gestione Account - Gestisce registrazione, login, logout, modifica profilo per tutti i tipi di utenti
2. Gestione Produzione - Gestisce la creazione, modifica, eliminazione e visualizzazione delle produzioni (PM)
3. Gestione Team - Gestisce l'assegnazione e rimozione dei Casting Director alle produzioni (PM)
4. Gestione Casting - Gestisce la creazione, modifica, eliminazione e visualizzazione degli annunci di casting (CD)
5. Gestione Candidatura - Gestisce la ricerca, visualizzazione e invio di candidature (Performer)

6. Gestione Valutazione Candidature - Gestisce la valutazione e il feedback sulle candidature (CD)
7. Monitoraggio Produzione - Gestisce la supervisione delle produzioni e statistiche (PM)

Di seguito è mostrato il diagramma in cui vengono evidenziate le relazioni di dipendenza tra i package di ogni sottosistema e tra i diversi sottosistemi.



### 2.2.2 Microdecomposizione in sottosistemi

Per semplificare la progettazione e lo sviluppo dell'applicazione, i sottosistemi saranno decomposti secondo lo schema previsto dall'architettura software Three-Layer, ovvero:

1. Presentation Layer: si occupa di gestire l'interfaccia grafica del sottosistema e contiene tutti gli elementi che interagiscono con l'utente.
2. Application Layer: si occupa, dati gli input dagli utenti, di elaborarli e fornire dei risultati che verranno poi mostrati graficamente.
3. Data Layer: è la parte del sottosistema che gestisce l'accesso ai dati persistenti.



### 2.2.2.1 Gestione Account

Questo sottosistema si occupa della gestione completa degli account utente, in particolare fornisce le funzioni di registrazione nuovo utente, login, visualizzazione profilo, modifica profilo e logout.

#### **GestioneAccountPresentationLayer:**

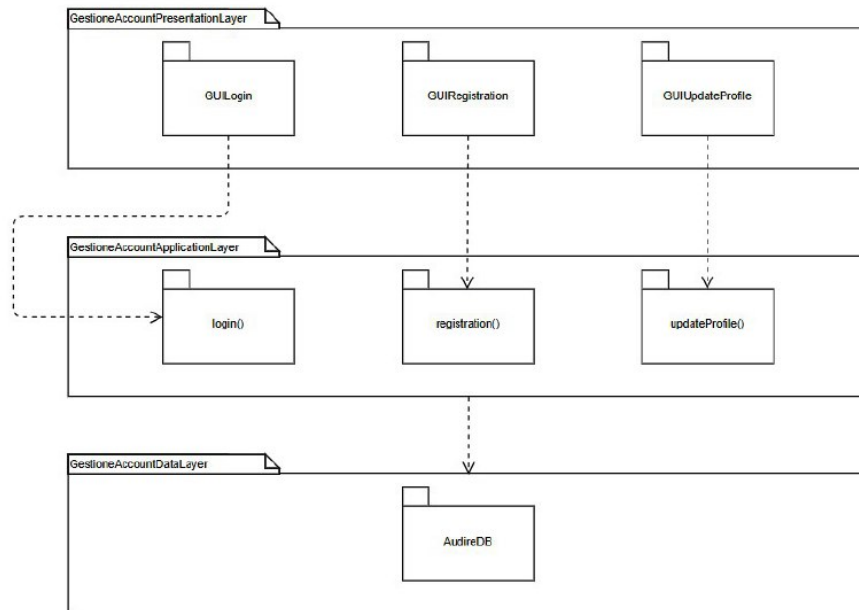
- **GUILogin:** comprende le interfacce che consentono all'utente di effettuare il login al sistema inserendo email e password.
- **GUIRegistration:** comprende le interfacce che consentono a un nuovo utente di registrarsi al sistema. Include form con campi per email, password, nome, cognome, telefono e selezione del ruolo.
- **GUIUpdateProfile:** comprende le interfacce che consentono all'utente autenticato di modificare le informazioni del proprio account (nome, cognome, telefono, e per i Performer anche foto profilo e CV).
- **GUILogout:** comprende il pulsante che consente all'utente di effettuare il logout dal sistema.

#### **GestioneAccountApplicationLayer:**

- **login():** operazione per effettuare l'accesso al sistema con credenziali valide.
- **registration():** operazione per la registrazione di un nuovo utente nel sistema.
- **updateProfile():** operazione per modificare i dati dell'account.
- **logout():** operazione per terminare la sessione e invalidare i token di autenticazione.

#### **GestioneAccountDataLayer:**

Si occupa di rendere reperibili i dati, presenti all'interno del database, relativi agli account utente (User, Performer, CastingDirector, ProductionManager).



### 2.2.2.2 Gestione Produzione

Questo sottosistema gestisce il ciclo di vita completo delle produzioni, permettendo ai Production Manager di creare, visualizzare, modificare ed eliminare produzioni.

#### GestioneProduzionePresentationLayer:

- GUIAddProduction: Interfaccia per creare una nuova produzione, include form con campi titolo e tipologia (Serie TV, Film, Teatro, Musical, Pubblicità, etc.).
- GUIGetProductions: Interfaccia che mostra l'elenco completo delle produzioni del PM, con informazioni sintetiche (titolo, tipologia, data creazione, numero casting, numero CD assegnati).
- GUIUpdateProduction: Dialog modale per modificare titolo e tipologia di una produzione esistente.
- GUIRemoveProduction: Dialog di conferma per l'eliminazione di una produzione, con avviso sulle conseguenze (eliminazione casting, candidature, assegnazioni CD).

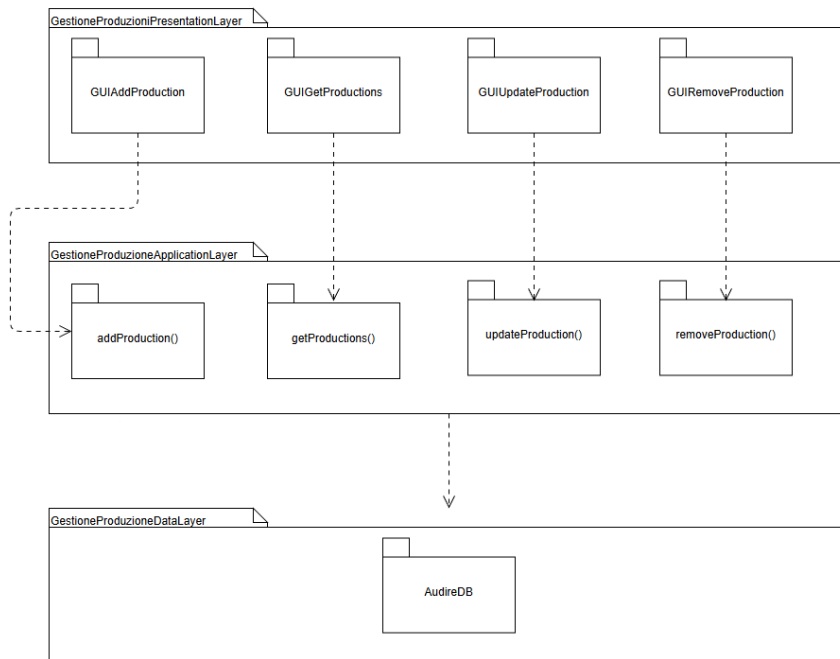
#### GestioneProduzioneApplicationLayer:

- addProduction() - operazione la creazione di una nuova produzione

- `getProductions()` - operazione per ottenere tutte le produzioni associate a un PM
- `updateProduction()` - operazione che permette la modifica dei dati di una produzione
- `removeProduction()` - operazione per l'eliminazione di una produzione

#### **GestioneProduzioneDataLayer:**

Si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alla produzione.



#### **2.2.2.3 Gestione Team**

Questo sottosistema gestisce l'assegnazione e la rimozione di Casting Director alle produzioni, permettendo ai Production Manager di costruire il team di lavoro per ogni produzione.

#### **GestioneTeamPresentationLayer:**

- `GUIAssignCD`: Form per assegnare un CD a una produzione, include campo email e pulsante di conferma.

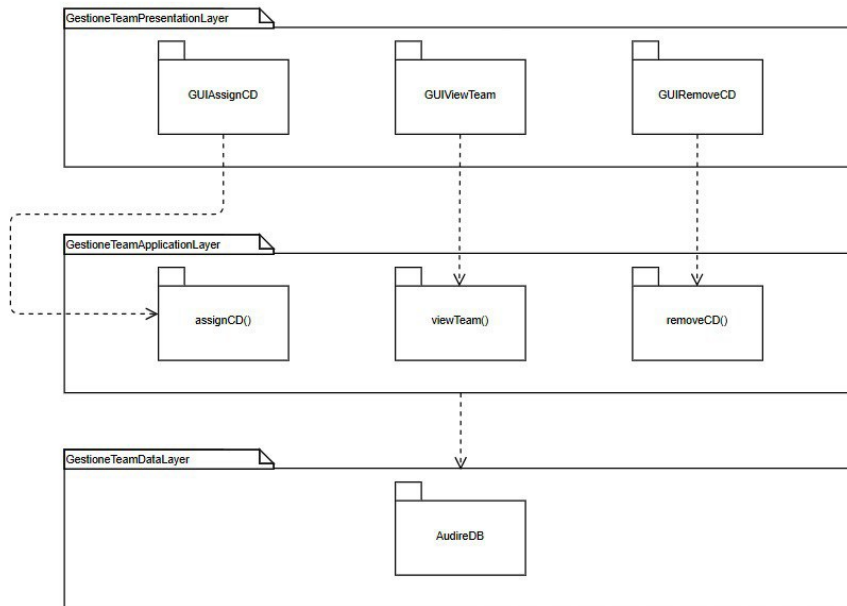
- **GUIViewTeam**: Interfaccia che mostra l'elenco completo dei CD assegnati a una produzione, con nome, cognome, email, data assegnazione.
- **GUIRemoveCD**: Dialog di conferma per rimuovere un CD da una produzione.

#### **GestioneTeamApplicationLayer:**

- **assignCD()** - operazione che permette l'assegnazione di un CD a una produzione
- **viewTeam()** - operazione che permette la visualizzazione dell'elenco dei Casting Director assegnati a una produzione
- **removeCD()** - operazioni che permette la rimozione di un Casting Director da una produzione

#### **GestioneTeamDataLayer:**

Si occupa di rendere reperibili i dati, presenti all'intero del database, relativi a un'associazione tra una produzione e un Casting Director.



#### **2.2.2.4 Gestione Casting**

Questo sottosistema gestisce il ciclo di vita completo degli annunci di casting, permettendo ai Casting Director di creare, visualizzare, modificare ed eliminare

annunci per le produzioni a cui sono assegnati.

#### **GestioneCastingPresentationLayer:**

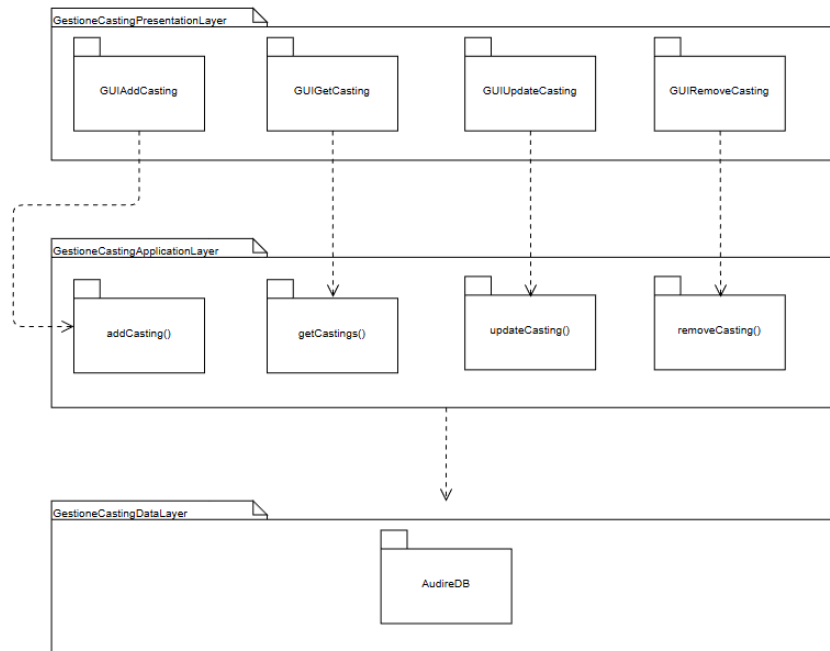
- GUIAddCasting: Form per creare un nuovo annuncio, include campi: nome ruolo, descrizione dettagliata, location, categoria richiesta, data chiusura.
- GUIGetCastings: Interfaccia che mostra l'elenco annunci di una produzione, con nome ruolo, categoria, date, numero candidature, stato.
- GUIUpdateCasting: Form per modificare un annuncio esistente (disponibile solo prima della deadline).
- GUIRemoveCasting: Dialog di conferma per eliminare un annuncio, con avviso sull'eliminazione delle candidature.

#### **GestioneCastingApplicationLayer:**

- addCasting() - operazione che permette al Casting Director di creare un nuovo annuncio per un casting
- getCastings() - operazione che permette al Casting Director di visualizzare tutte le informazioni relative ai casting da lui creati
- updateCasting() - operazione che permette di modificare le informazioni relative ad un'annuncio
- removeCasting() - operazione che permette di eliminare un annuncio

#### **GestioneCastingDataLayer:**

Si occupa di rendere reperibili i dati, all'intero del database, relativi al casting.



### 2.2.2.5 Gestione Candidature

Questo sottosistema gestisce la ricerca, visualizzazione e invio delle candidature da parte dei Performer, permettendo loro di scoprire opportunità e candidarsi ai casting aperti.

#### GestioneCandidaturePresentationLayer:

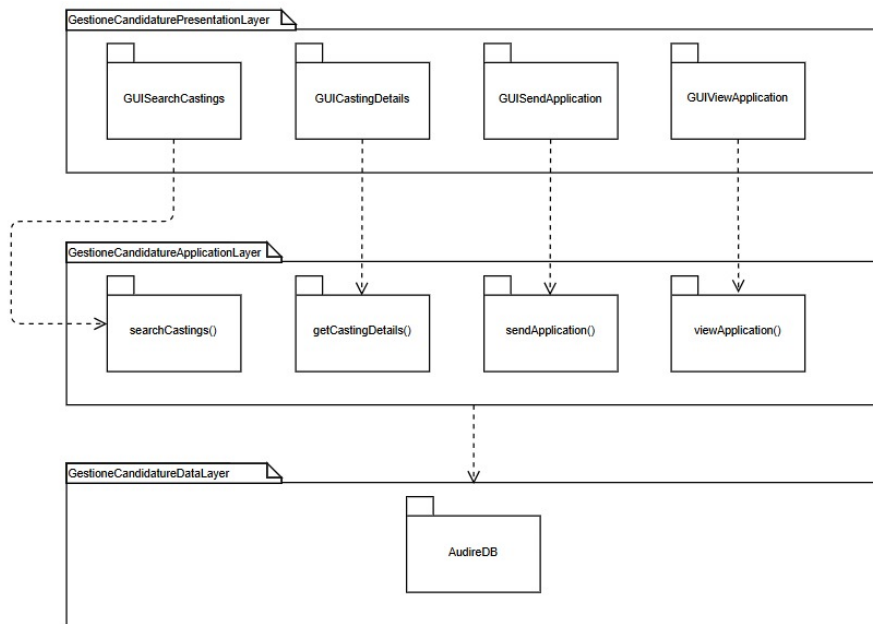
- **GUISearchCastings:** Interfaccia che mostra tutti i casting aperti con possibilità di applicare filtri per location, categoria professionale e tipo di produzione. Include anche ordinamento per data pubblicazione o deadline.
- **GUICastingDetail:** Interfaccia che mostra i dettagli completi di un casting, includendo nome ruolo, descrizione personaggio, produzione associata, location, categoria richiesta, date e pulsante "Candidati" se applicabile.
- **GUISendApplication()** - Modal di conferma per l'invio candidatura, mostra riepilogo dei materiali che verranno inviati (foto profilo, CV, info professionali).
- **GUIViewApplications:** Interfaccia che mostra l'elenco completo delle candidature del Performer con nome ruolo, produzione, data invio, status, feedback.

### GestioneCandidatureApplicationLayer:

- searchCastings() - operazione che permette la ricerca e il filtraggio dei casting aperti
- getCastingDetails() - operazione che permette di visualizzare i dettagli completi di un casting
- sendApplication() - operazione che permette l'invio di una candidatura a un casting
- viewApplications() - operazione che permette di visualizzare tutte le candidature inviate con i relativi status

### GestioneCandidatureDataLayer:

Si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alle candidature e ai casting.



#### 2.2.2.6 Gestione Valutazione Candidature

Questo sottosistema permette ai Casting Director di valutare le candidature ricevute per i loro annunci, visualizzare i profili completi dei candidati, scaricare i CV e aggiornare lo status delle candidature con feedback.

### GestioneValutazioneCandidaturePresentationLayer:

- **GUIApplicationsList:** Interfaccia che mostra l'elenco completo delle candidature ricevute per un casting, con foto profilo, nome, cognome, categoria professionale, data candidatura e status corrente.
- **GUIPerformerProfile:** Interfaccia che mostra il profilo completo del candidato con foto profilo, informazioni anagrafiche, categoria professionale, pulsante per scaricare il CV e controlli per aggiornare status e feedback.
- **GUIUpdateStatus:** Form per selezionare il nuovo status (In attesa, Shortlist, Selezionata, Rifiutata) e inserire un feedback testuale opzionale (max 1000 caratteri).
- **GUIFilterApplications:** Sezione filtri che permette di filtrare le candidature per status, data candidatura e categoria professionale.

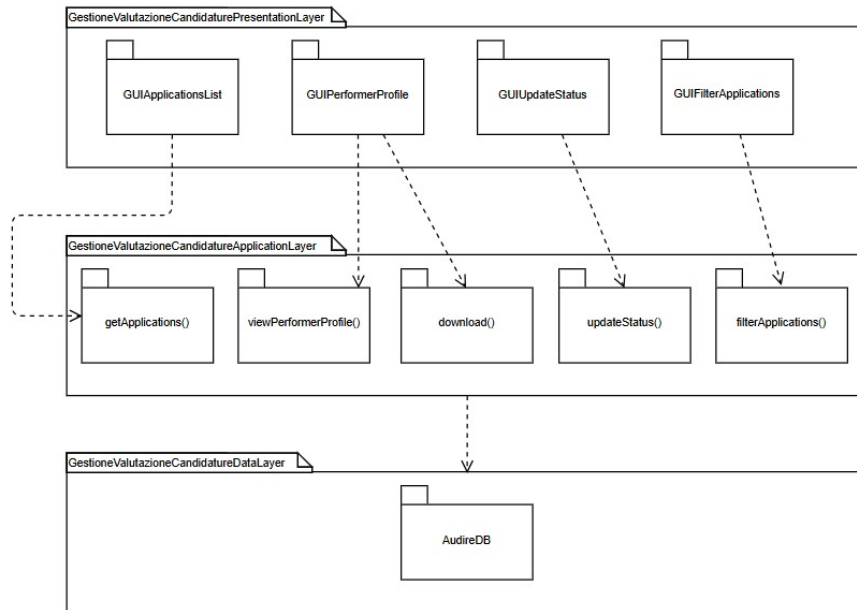
#### **GestioneValutazioneCandidatureApplicationLayer:**

- **getApplications()** - operazione che permette di visualizzare tutte le candidature di un casting
- **filterApplications()** - operazione che permette di filtrare le candidature secondo criteri specifici
- **viewPerformerProfile()** - operazione che permette di visualizzare il profilo completo di un candidato
- **updateStatus()** - operazione che permette di aggiornare lo status di una candidatura e inserire un feedback
- **downloadCV()** - operazione che permette di scaricare il curriculum vitae di un candidato

#### **GestioneValutazioneCandidatureDataLayer:**

Si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alle candidature, ai performer e ai loro materiali (CV, foto profilo).





### 2.2.2.7 Monitoraggio Produzione

Questo sottosistema permette ai Production Manager di monitorare l'avanzamento delle loro produzioni, visualizzare statistiche sui casting e sulle candidature, e avere una visione d'insieme del processo di selezione.

#### MonitoraggioProduzioneApplicationLayer:

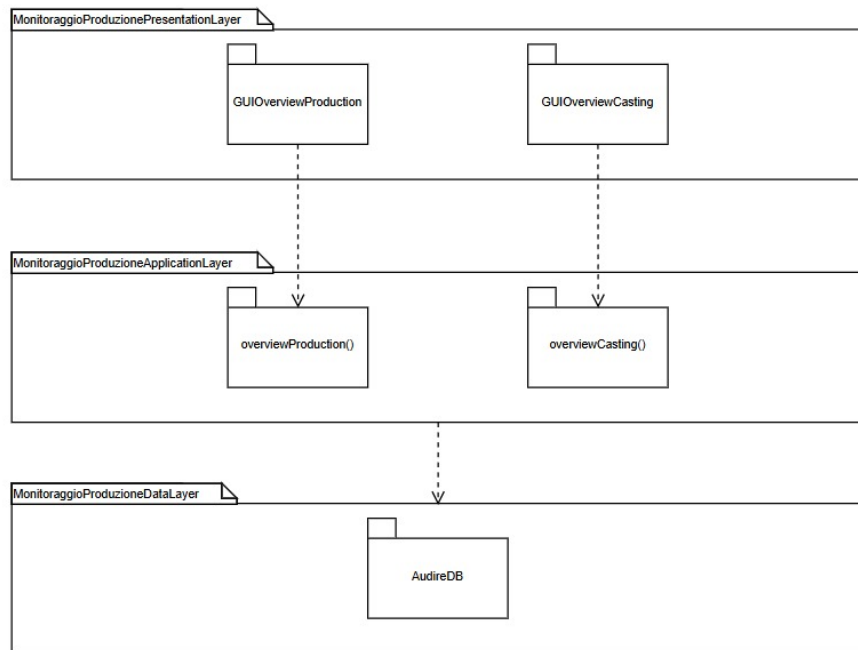
- **GUIOverviewProduction**: Interfaccia che mostra la panoramica completa di una produzione con informazioni generali, team di lavoro, statistiche casting (numero casting totali, aperti, chiusi, totale candidature).
- **GUIOverviewCasting**: Interfaccia in modalità sola lettura che mostra i dettagli di un casting specifico con statistiche dettagliate sulle candidature (totale, per status, timeline).

#### MonitoraggioProduzioneApplicationLayer:

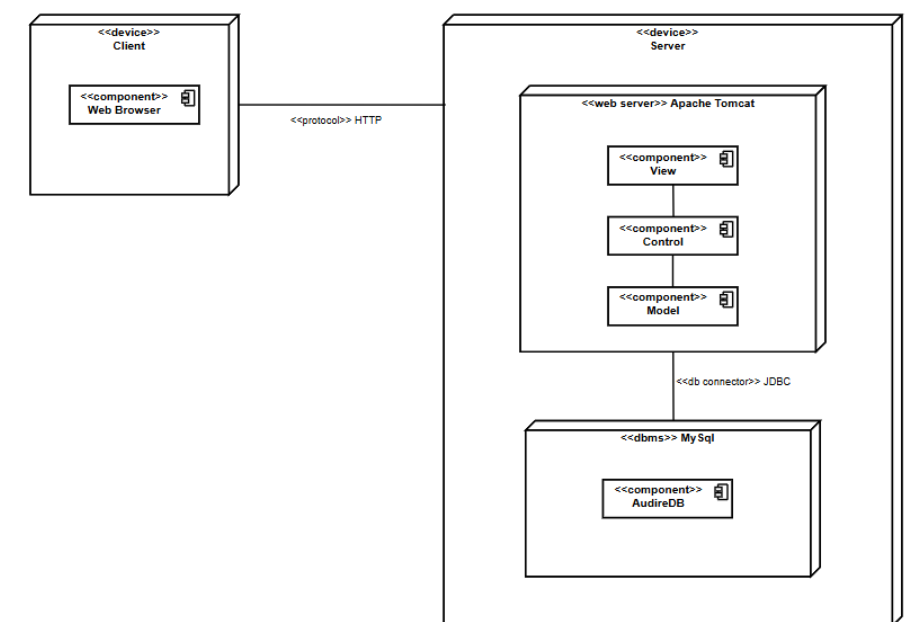
- `overviewProduction()` - operazione che permette di visualizzare la panoramica completa di una produzione
- `overviewCasting()` - operazione che permette di visualizzare le statistiche aggregate dei casting

#### MonitoraggioProduzioneDataLayer:

Si occupa di rendere reperibili i dati, presenti all'interno del database, relativi alle produzioni, ai casting e alle candidature.



## 2.3 Mappatura Hardware/software



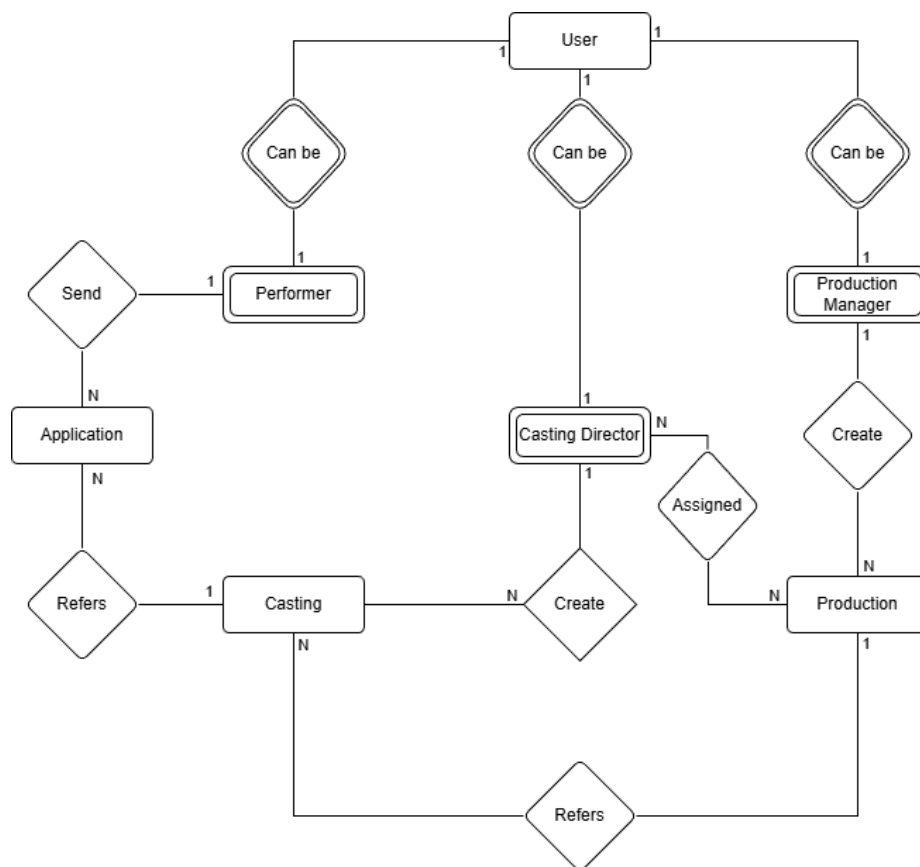
L'architettura del sistema si basa sul modello client/server, nel quale un server centralizzato eroga funzionalità a un insieme distribuito di client. Ciascun client esegue un'applicazione browser web che implementa il livello di presentazione (View), consentendo agli utenti finali di interagire con il sistema attraverso l'invio di richieste e la ricezione di risposte. Il server, d'altra parte, è deputato all'esecuzione della logica applicativa mediante i componenti Controller e Model, nonché alla gestione della persistenza dei dati.

Il protocollo di comunicazione adottato è HTTPS (HyperText Transfer Protocol Secure), il quale garantisce l'integrità, la riservatezza e l'autenticità delle comunicazioni attraverso meccanismi di crittografia. L'interazione tra le due componenti segue il pattern request-response: il client formula ed invia una richiesta HTTP/HTTPS al server; quest'ultimo processa la richiesta, esegue le operazioni richieste e genera una risposta che viene trasmessa al client originante.

## 2.4 Gestione dei Dati Persistenti

I dati persistenti del sistema Audire saranno memorizzati all'interno di un database relazionale, specificamente MySQL, come definito nei Design Goals. Il database assicura l'integrità referenziale e la persistenza delle informazioni critiche.

### 2.4.1 Modello logico



### 2.4.2 Tabelle

#### 2.4.2.1 Tabella: User

| Attributo         | Tipo  | Vincolo di chiave |
|-------------------|---|-------------------|
| userID            | INT   | Primary Key       |
| email             | VARCHAR(100)  |                   |
| passwordHash      | VARCHAR(255)  |                   |
| firstName         | VARCHAR(50)   |                   |
| lastName          | VARCHAR(50)   |                   |
| phoneNumber       | CHAR(10)  |                   |
| role              | ENUM('Performer',<br>'CastingDirector',<br>'ProductionManager') |                   |
| registrationaDate | DATE  |                   |

#### 2.4.2.2 Tabella: Performer

| Attributo    | Tipo  | Vincolo di chiave |
|--------------|---|-------------------|
| performerID  | INT   | Primary Key       |
| gender       | ENUM('M', 'F', 'Altro')   |                   |
| category     | ENUM('Attore/Attrice', 'Musicista', 'Cantante', 'Ballerino', 'Doppiatore/trice', 'Qualsiasi') |                   |
| profilePhoto | VARCHAR(255)  |                   |
| cv           | VARCHAR(255)  |                   |
| description  | TEXT  |                   |
| userID       | INT   | Foreign Key       |

#### 2.4.2.3 Tabella: Casting Director

| Attributo | Tipo | Vincolo di chiave |
|-----------|------|-------------------|
| cdID      | INT  | Primary Key       |
| userID    | INT  | Foreign Key       |

#### 2.4.2.4 Tabella: Production Manager

| Attributo | Tipo | Vincolo di chiave |
|-----------|------|-------------------|
| pmID      | INT  | Primary Key       |
| userID    | INT  | Foreign Key       |

#### 2.4.2.5 Tabella: Production

| Attributo    | Tipo   | Vincolo di chiave |
|--------------|--|-------------------|
| productionID | INT  | Primary Key       |
| title        | VARCHAR(100)   |                   |
| type         | ENUM('Serie TV', 'Film', 'Teatro', 'Musical', 'Pubblicità', 'Documentario', 'Cortometraggio', 'Web Series', 'Altro') |                   |
| creationDate | DATE   |                   |
| pmID         | INT  | Foreign Key       |

#### 2.4.2.6 Tabella: Assigned (o Team)

| Attributo    | Tipo | Vincolo di chiave     |
|--------------|------|-----------------------|
| productionID | INT  | Primary & Foreign Key |
| cdID         | INT  | Primary & Foreign Key |

#### 2.4.2.7 Tabella: Casting

| Attributo    | Tipo  | Vincolo di chiave |
|--------------|---|-------------------|
| castingID    | INT(11)   | Primary Key       |
| location     | VARCHAR(255)  |                   |
| category     | ENUM('Attore/Attrice', 'Musicista', 'Cantante', 'Ballerino', 'Doppiatore/trice', 'Qualsiasi') |                   |
| description  | TEXT  |                   |
| publishDate  | DATE  |                   |
| deadLine     | DATE  |                   |
| title        | VARCHAR(255)  |                   |
| cdID         | INT   | Foreign Key       |
| productionID | INT   | Foreign Key       |

#### 2.4.2.8 Tabella: Application

| Attributo     | Tipo   | Vincolo di chiave |
|---------------|--|-------------------|
| applicationID | INT  | Primary Key       |
| sendingDate   | DATE   |                   |
| status        | ENUM('In attesa', 'Shortlist', 'Selezionata', 'Rifiutata') |                   |
| feedback      | TEXT   |                   |
| performerID   | INT  | Foreign Key       |
| castingID     | INT  | Foreign Key       |

### 2.5 Controllo degli accessi

Il sistema Audire implementa un controllo degli accessi basato sui ruoli (RBAC - Role-Based Access Control). Ogni utente appartiene a un ruolo specifico (Performer, Casting Director, Production Manager) e può accedere solo alle funzionalità autorizzate per quel ruolo.

La seguente tabella definisce le operazioni disponibili per ogni combinazione (Attore, Oggetto):

Tabella 2: Matrice di Controllo degli Accessi

| Attori → / Oggetti ↓ | Performer  | Casting Director  | Production Manager   |
|----------------------|--|---|--|
| <b>User</b>          | registration(),<br>login(), logout(),<br>updateProfile() | registration(),<br>login(), logout(),<br>updateProfile()  | registration(), login(),<br>logout(), updateProfile()                              |
| <b>Production</b>    | <i>Nessun accesso</i>                                    | getProductions()<br>(solo lettura se assegnato)   | addProduction(),<br>getProductions(),<br>updateProduction(),<br>removeProduction() |
| <b>Team</b>          | <i>Nessun accesso</i>                                    | <i>Nessun accesso</i>   | assignCD(), removeCD(),<br>viewTeam()  |
| <b>Casting</b>       | searchCastings(),<br>getCastingDetails()                 | addCasting(),<br>getCastings(),<br>updateCasting(),<br>removeCasting()                                    | overviewCasting()<br>(solo lettura)  |
| <b>Application</b>   | sendApplication(),<br>viewApplications()                 | getApplications(),<br>filterApplications(),<br>viewPerformerProfile(),<br>updateStatus(),<br>downloadCV() | overviewProduction()<br>(statistiche aggregate)                                    |

## 2.6 Controllo software globale

Il sistema Audire è un sistema interattivo per cui ogni funzionalità viene avviata in seguito ad un comando impartito dall'utente tramite l'uso di un'interfaccia grafica. Quindi quando un utente vuole accedere ed utilizzare una funzionalità del sistema può farlo tramite l'interfaccia grafica la quale selezionerà il controllo corrispondente. L'azione scatenerà un evento il quale verrà gestito dal suo handler. L'handler a sua volta indirizzerà il controllo del flusso di eventi al sottosistema che si occupa della logica di controllo e gestore del controllo che poi si rivolge ai servizi per la logica applicativa.

Per tali motivi il sistema utilizzerà un meccanismo di controllo del flusso di tipo event-driven essendo una web-application.

## 2.7 Condizioni limite

### 2.7.1 Start-up

Per il primo avvio del sistema è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti e l'esecuzione del codice lato server. In seguito, verrà mostrata la homepage della piattaforma dalla quale è possibile accedere all'area di Login, dove sarà possibile autenticarsi tramite credenziali (email e password) come utente con accesso alle funzionalità del sistema. Una volta effettuato l'accesso, sarà possibile effettuare

tutte le operazioni consentite, che saranno diverse in base al ruolo dell'utente (Performer, Casting Director, Production Manager).

### **2.7.2 Start-up (a seguito di un fallimento)**

Il sistema può subire guasti dovuti al sovraccarico del database con successivo fallimento. Per ovviare al problema, periodicamente è previsto un salvataggio dei dati sotto forma di backup SQL. All'avvio a seguito di tale fallimento, oltre alle normali procedure previste per lo start-up, l'ultimo backup SQL memorizzato sarà eseguito per la rigenerazione del database.

### **2.7.3 Terminazione**

Al momento della chiusura dell'applicativo si ha la terminazione del sistema con un regolare Logout. Viene assicurata la consistenza dei dati, annullando eventuali operazioni che erano in esecuzione. Gli utenti connessi verranno disconnessi in modo sicuro con invalidazione delle sessioni attive.

### **2.7.4 Fallimento**

Possono verificarsi diversi casi di fallimento del sistema. Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso, è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di backup SQL per la successiva rigenerazione del DB. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione, non sono previsti metodi che ripristinino lo stato del sistema a uno stato antecedente allo spegnimento inaspettato. Un altro caso di fallimento potrebbe derivare dal software stesso che causa un crash inaspettato dovuto ad errori commessi durante la fase di implementazione; l'unico processo che potrà essere eseguito è la chiusura del sistema e il suo successivo riavvio.



### 3 Servizi dei sottosistemi

#### 3.1 SS\_ACC - Gestione Account

| Sottosistema     | Descrizione  |
|------------------|--|
| Gestione Account | Gestisce le operazioni relative alla registrazione, autenticazione e gestione del profilo utente.                        |
| Servizio         | Descrizione  |
| registration()   | Permette a un nuovo utente di registrarsi al sistema inserendo email, password, dati anagrafici e selezionando il ruolo. |
| login()          | Permette a un utente registrato di autenticarsi al sistema tramite email e password.                                     |
| logout()         | Permette a un utente autenticato di terminare la sessione corrente.  |
| updateProfile()  | Permette a un utente di modificare le informazioni del proprio profilo.  |

#### 3.2 SS\_PROD - Gestione Produzione

| Sottosistema        | Descrizione  |
|---------------------|--|
| Gestione Produzione | Gestisce il ciclo di vita delle produzioni cinematografiche, televisive e teatrali.            |
| Servizio            | Descrizione  |
| addProduction()     | Permette al Production Manager di creare una nuova produzione specificando titolo e tipologia. |
| getProductions()    | Permette al Production Manager di visualizzare l'elenco delle proprie produzioni.              |
| updateProduction()  | Permette al Production Manager di modificare i dati di una produzione esistente.               |
| removeProduction()  | Permette al Production Manager di eliminare una produzione e tutti i dati associati.           |

### 3.3 SS\_TEAM - Gestione Team

| Sottosistema  | Descrizione   |
|---------------|---|
| Gestione Team | Gestisce l'assegnazione dei Casting Director alle produzioni.                   |
| Servizio      | Descrizione   |
| assignCD()    | Permette al PM di assegnare un Casting Director a una produzione tramite email. |
| viewTeam()    | Permette al PM di visualizzare l'elenco dei CD assegnati a una produzione.      |
| removeCD()    | Permette al PM di rimuovere un CD da una produzione.                            |

### 3.4 SS\_CAST - Gestione Casting

| Sottosistema     | Descrizione  |
|------------------|--|
| Gestione Casting | Gestisce la creazione e amministrazione degli annunci di casting.          |
| Servizio         | Descrizione  |
| addCasting()     | Permette al CD di creare un nuovo annuncio di casting per una produzione.  |
| getCasting()     | Permette al CD di visualizzare tutti gli annunci di casting creati.        |
| updateCasting()  | Permette al CD di modificare un annuncio esistente (prima della deadline). |
| removeCasting()  | Permette al CD di eliminare un annuncio di casting.                        |

### 3.5 SS\_CAND - Gestione Candidature

| Sottosistema         | Descrizione  |
|----------------------|--|
| Gestione Candidature | Gestisce la ricerca dei casting e l'invio delle candidature da parte dei Performer.                    |
| Servizio             | Descrizione  |
| searchCastings()     | Permette al Performer di cercare casting aperti applicando filtri per location, categoria e tipologia. |
| getCastingDetails()  | Permette al Performer di visualizzare i dettagli completi di un casting.                               |
| sendApplication()    | Permette al Performer di inviare la propria candidatura a un casting.                                  |
| viewApplications()   | Permette al Performer di visualizzare lo storico delle proprie candidature con status e feedback.      |

### 3.6 SS\_VAL - Gestione Valutazione Candidature

| Sottosistema           | Descrizione   |
|------------------------|---|
| Gestione Valutazione   | Gestisce la valutazione delle candidature ricevute da parte dei Casting Director. |
| Servizio               | Descrizione   |
| getApplications()      | Permette al CD di visualizzare tutte le candidature ricevute per un casting.      |
| filterApplications()   | Permette al CD di filtrare le candidature per status, data o categoria.           |
| viewPerformerProfile() | Permette al CD di visualizzare il profilo completo di un candidato.               |
| updateStatus()         | Permette al CD di aggiornare lo status di una candidatura e inserire feedback.    |
| downloadCV()           | Permette al CD di scaricare il curriculum vitae di un candidato.                  |

### 3.7 SS\_MON - Monitoraggio Produzione

| Sottosistema            | Descrizione  |
|-------------------------|--|
| Monitoraggio Produzione | Gestisce la supervisione delle produzioni e la visualizzazione delle statistiche.        |
| Servizio                | Descrizione  |
| overviewProduction()    | Permette al PM di visualizzare la panoramica completa di una produzione con statistiche. |
| overviewCasting()       | Permette al PM di visualizzare le statistiche dettagliate dei casting di una produzione. |

## 4 Glossario

**Performer** - Utente del sistema che cerca opportunità di lavoro nel settore dell'intrattenimento (attori, cantanti, ballerini, etc.).

**Casting Director (CD)** - Utente responsabile della creazione degli annunci di casting e della valutazione delle candidature.

**Production Manager (PM)** - Utente responsabile della gestione delle produzioni e della supervisione del processo di casting.

**Produzione** - Progetto cinematografico, televisivo, teatrale o pubblicitario per il quale vengono organizzati i casting.

**Casting** - Annuncio pubblicato per la ricerca di performer per un ruolo specifico in una produzione.

**Candidatura (Application)** - Richiesta inviata da un Performer per partecipare a un casting.

**Status** - Stato di avanzamento di una candidatura: In attesa, Shortlist, Selezionata, Rifiutata.

**Feedback** - Commento testuale fornito dal Casting Director al Performer riguardo la sua candidatura.

**Team** - Insieme dei Casting Director assegnati a una specifica produzione.

**Dashboard** - Interfaccia principale personalizzata per ogni tipologia di utente.

**Categoria Professionale** - Specializzazione del Performer (es. Attore, Cantante, Ballerino, Modello).