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

System Design Document

Movie Desk

Versione: 3.0

Immagine che contiene testo, clipart

Descrizione generata automaticamente

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
|  |  |
| **none** | **none** |

**Partecipanti:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
|  |  |
| Caliendo Gioacchino | 05706 |
| Puglia Chiara | 06018 |
| Vitillo Ilaria | 05918 |

|  |  |
| --- | --- |
| **Scritto da:** | Caliendo Gioacchino, Puglia Chiara, Vitillo Ilaria |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 25/11/2020 | 1.0 | Prima bozza del documento | Chiara Puglia |
| 26/11/2020 | 1.1 | Prima stesura del documento con aggiunta di dettagli | Ilaria Vitillo |
| 27/11/2020 | 1.2 | Accorgimenti vari al documento | Gioacchino Caliendo |
| 28/11/2020 | 1.3 | Stesura del Purpose of the System e aggiunta dei design goals | Chiara Puglia |
| 01/12/2020 | 1.4 | Aggiunta del Current Software Architecture | Ilaria Vitillo |
| 03/12/2020 | 1.5 | Prima stesura del Proposed Software Architecture (Subsystem Decomposition e Layer Decomposition) | Chiara Puglia |
| 06/12/2020 | 1.6 | Aggiunta Hardware/Software Mapping | Gioacchino Caliendo |
| 09/12/2020 | 1.7 | Aggiunta Persistent Data Management | Ilaria Vitillo |
| 12/12/2020 | 1.8 | Aggiunta Access Control and Security | Ilaria Vitillo |
| 14/12/2020 | 1.9 | Aggiunta Global Software Control e Boundary Condition | Chiara Puglia |
| 15/12/2020 | 2.0 | Aggiunta Subsystem Services | Gioacchino Caliendo |
| 19/01/2021 | 3.0 | Revisione finale del documento con aggiunta di ultimi dettagli | Gioacchino Caliendo |

**Sommario**

[1.Introduction 4](#_Toc62119981)

[1.1 Purpose of system 4](#_Toc62119982)

[1.2 Design goals 5](#_Toc62119983)

[1.2.1 Design Trade-off 6](#_Toc62119984)

[1.3 Definitions, acronyms, and abbreviations 6](#_Toc62119985)

[1.4 References 6](#_Toc62119986)

[1.5 Overview 7](#_Toc62119987)

[2. Current Software Architecture 7](#_Toc62119988)

[3. Proposed software architecture 8](#_Toc62119989)

[3.1 Overview 8](#_Toc62119990)

[3.2 Subsystem decomposition 8](#_Toc62119991)

[3.2.1 Layer Decomposition 8](#_Toc62119992)

[3.2.2 Subsystem Decomposition 8](#_Toc62119993)

[3.3 Hardware/software mapping 10](#_Toc62119994)

[3.4 Persistent data management 11](#_Toc62119995)

[3.5 Access control and security 15](#_Toc62119996)

[3.6 Global software control 16](#_Toc62119997)

[3.7 Boundary conditions 16](#_Toc62119998)

[4. Subsystem services 17](#_Toc62119999)

# **1.Introduction**

# **1.1 Purpose of system**

Lo scopo del Sistema è quello di fornire agli utenti un’unica piattaforma in cui reperire informazioni sui contenuti cinematografici di interesse.

L’idea è quella di usare un’interfaccia semplice e user friendly così da permettere una facile lettura e interpretazione anche da parte di utenti non esperti.

Il sistema progettato è un’applicazione web a cui avranno accesso utenti di tre diverse tipologie: Utente non registrato che potrà visualizzare le informazioni del sistema, Utente Standard che potrà creare liste personalizzate di contenuti attraverso delle marcature specifiche e inviare segnalazioni in caso di inconsistenze riscontrate e Media Manager che avrà il compito di aggiungere/rimuovere contenuti e piattaforme, aggiornarne le disponibilità e verificare una segnalazione.

Il sistema può essere diviso in due categorie:

* Gestione dei contenuti multimediali: caricamento dei contenuti disponibili all’interno del sistema compresa la gestione delle informazioni ad essi collegate.
* Gestione delle piattaforme di streaming: gestione delle informazioni relative alle piattaforme di streaming per ogni contenuto multimediale contenuto nel sistema.
* Gestione delle segnalazioni: gestione delle segnalazioni inviate dagli Utenti Standard e successiva presa in incarico da parte dei Media Manager per la risoluzione di inconsistenze sulle informazioni.

Il sistema fornirà un login per le due tipologie di utente registrato (Standard e Media Manager) in base ai campi inseriti in fase di accesso.

# **1.2 Design goals**

I Design Goals sono organizzati in cinque categorie: Performance, Dependability, Cost, Maintenance e User Criteria. Illustriamo nella seguente tabella gli obiettivi di design per il sistema, con l’identificativo, il tipo di criterio, la categoria alla quale appartengono e le relative priorità (a numeri più bassi corrispondono priorità più elevate).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Priorità** | **ID** | **Descrizione** | **Categoria** | **Tipo di criterio** |
| 1 | DG\_1 | **Tempi di risposta:** il sistema deve elaborare le richieste e produrre output in meno di 2 secondi. | Performance | Criterio di performance |
| 1 | DG\_2 | **Sicurezza**: l’accesso al sistema avviene mediante l’indirizzo email dell’utente e la password da esso scelta di minimo 8 caratteri e che sarà crittografata.  Inoltre, la sicurezza è garantita in quanto ogni utente può svolgere solo le operazioni a lui consentite, differenziando ampiamente le varie tipologie di utente previste nella realizzazione del sistema stesso. | Dependability | Criterio di affidabilità |
| 1 | DG\_3 | **Costi di sviluppo:** è stimato un costo complessivo di 150 ore per la progettazione e lo sviluppo del sistema (50 ore per ogni project member). | Cost | Criterio di costo |
| 1 | DG\_4 | **Portabilità**: l’interazione con il sistema avviene tramite un browser quindi possiamo definirlo portabile poiché il sistema viene sviluppato come un’applicazione web, esso è accessibile da qualunque dispositivo, purchè abbia un browser installato e abbia una connessione ad internet. | Maintenance | Criterio di manutenzione |
| 1 | DG\_5 | **Usabilità:** il sistema sarà di facile comprensione e di utilizzo, permettendo la fruizione delle diverse funzionalità offerte in modo semplice ed immediato attraverso un’interfaccia user-friendly. | User Criteria | Criterio di usabilità |
| 1 | DG\_6 | **Affidabilità**: il sistema deve essere in grado di riconoscere situazioni anomali e mantenere modifiche ai dati persistenti al fine di garantirne la consistenza | Dependability | Criterio di affidabilità |
| 1 | DG\_7 | **Utilità:** il sistema è altamente utile poiché velocizza e quindi ottimizza la ricerca da parte di un’utente di un contenuto d’interesse | User Criteria | Criterio di usabilità |
| 2 | DG\_8 | **Robustezza**: il sistema informerà l’utente di eventuali errori nel caso di immissione di input non validi attraverso appositi messaggi | Dependability | Criterio di affidabilità |
| 2 | DG\_9 | **Leggibilità**: Il codice prodotto deve essere semplice da comprendere. | Maintenance | Criterio di manutenzione |
| 2 | DG\_10 | **Modificabilità:** le funzionalità del sistema devono essere facilmente modificabili. | Maintenance | Criterio di manutenzione |
| 3 | DG\_11 | **Estensibilità**: il sistema deve agevolare l’introduzione di nuove funzionalità. | Maintenance | Criterio di manutenzione |

# **1.2.1 Design Trade-off**

**Tempo di rilascio vs Funzionalità:** in condizione di tempi di rilascio stringenti possono essere rilasciate meno funzionalità.

**Tempo di rilascio vs Qualità:** in condizione di tempi di rilascio stringenti si intende rilasciare il software nei tempi prefissati pur se presenti bug con l’intenzione di correggerli con successivi aggiornamenti.

# **1.3 Definitions, acronyms, and abbreviations**

**User-Friendly:** “amichevole per l’utente”, si intende un qualcosa di facile utilizzo anche per chi è meno esperto in un particolare settore;

**Bug:** si riferisce ad un errore di funzionamento relativo alle funzionalità di un sistema o di un programma..

# **1.4 References**

Link: <https://wwwbruegge.in.tum.de/lehrstuhl_1/component/content/article/217-OOSE>

Libro: Object-Oriented Software Engineering using UML, Patterns, and Java

Autore: Bernd Bruegge, Allen H. Dutoit

Libro: Swebok v3.0

Autore: Pierre Bourque, Richard E. Fairley

RAD\_MovieDesk

# **1.5 Overview**

**Capitolo uno:** Introduzione agli obbiettivi del sistema, design goals, definizione acronimi e abbreviazioni.

**Capitolo due:** Descrizione delle funzionalità del sistema corrente nel caso esistano.

**Capitolo tre:** Presentazione dell’architettura del sistema proposto in cui viene gestita la decomposizione in sottosistemi, mapping hardware e software, controllo degli accessi e sicurezza, controllo del flusso globale del sistema, condizioni limite.

**Capitolo quattro:** Descrizione dei servizi dei sottosistemi.

# **2. Current Software Architecture**

Attualmente esiste un sistema che gestisce le ricerche dei contenuti da parte di un utente e fornisce loro informazioni riguardanti lo streaming.

Questo sistema è organizzato secondo l’architettura MVC (Model, View, Control). Attualmente la ricerca di determinati contenuti viene effettuata attraverso questi passaggi: L’utente si reca sulla pagina web “justwatch.com” o scarica l’app dagli store, digita sulla barra di ricerca il titolo del contenuto su cui desidera ricevere più informazioni, clicca sull’immagine di copertina del contenuto e solo a questo punto vengono visualizzate le informazioni riguardanti il contenuto selezionato.

Il sistema mostra incongruenze nelle informazioni visualizzate sono spesso piene di incongruenze oltre a mostrare un’interfaccia poco chiara in fase di ricerca, inoltre non fornisce la possibilità di ricevere notifiche riguardanti i contenuti salvati.

Il team si impegna quindi a realizzare un sistema che possa migliorare la concretezza delle informazioni visualizzate, basandosi anche sulle segnalazioni dell’utenza e che consenta all’utenza di rimanere aggiornata con notifiche personalizzate.

# **3. Proposed software architecture** **3.1 Overview**

Il Sistema da noi proposto è un’applicazione web che racchiude funzioni di ricerca e filtraggio dei contenuti multimediali d’interesse per l’utente, inoltre offre la possibilità di marcare i contenuti desiderati così da poterli recuperare rapidamente in ogni momento.

Gli utenti saranno di vario tipo: Utente non registrato, Utente Standard e Media Manager.

Il sistema offrirà funzionalità diverse a seconda della tipologia di utente.

In particolare, l’Utente generico potrà navigare liberamente nel sito, ricercare e filtrare i contenuti, marcare i contenuti in liste personalizzate, inviare segnalazioni.

Lo stile architetturale scelto è di tipo repository, perché sono adatti per applicazioni con task di elaborazione dati che cambiano di frequente, nello specifico è un sistema MVC. Quest’ultimo (Model View Controller) è un pattern architetturale molto diffuso. Il nostro sistema viene rappresentato da un’architettura di tipo multi-tier, ovvero le varie funzionalità del sito sono logicamente separate e suddivise su più strati o livelli software differenti in comunicazione fra loro.

# **3.2 Subsystem decomposition**

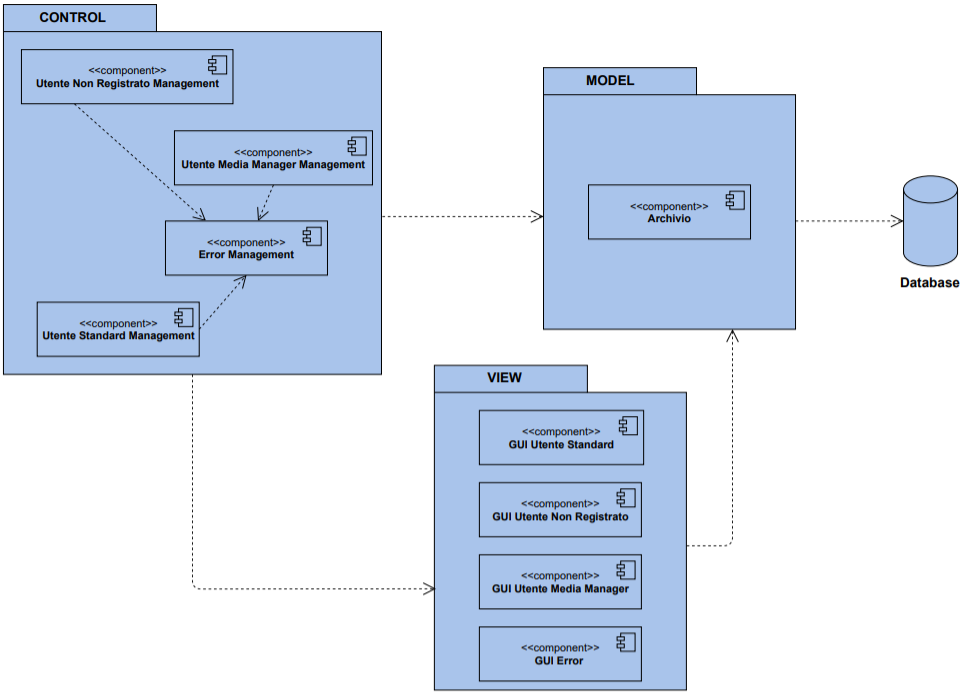
# **3.2.1 Layer Decomposition**

La decomposizione prevista per il sistema è composta da 3 layer che si occupano di gestire gli aspetti e funzionalità differenti:

* **View:** raccoglie e gestisce elementi di interfaccia grafica e gli eventi generati su di essi;
* **Controller:** si occupa della gestione della logica del sistema;
* **Model:** si occupa della gestione e dello scambio dei dati fra i vari sottosistemi.

# **3.2.2 Subsystem Decomposition**

Dopo un’attenta analisi effettuata sul sistema proposto, si è deciso di suddividerlo in tre sottosistemi sfruttando il modello architetturale MVC. Si vogliono descrivere in maniera dettagliata i componenti relativi all’utilizzo del sistema da parte degli utenti previsti al suo interno.



Il livello View prevede la gestione di quattro componenti e possiamo identificarli come:

* GUI Utente Standard;
* GUI Utente Non Registrato;
* GUI Utente Media Manager;
* GUI Error.

Il livello Control prevede la gestione di quattro componenti:

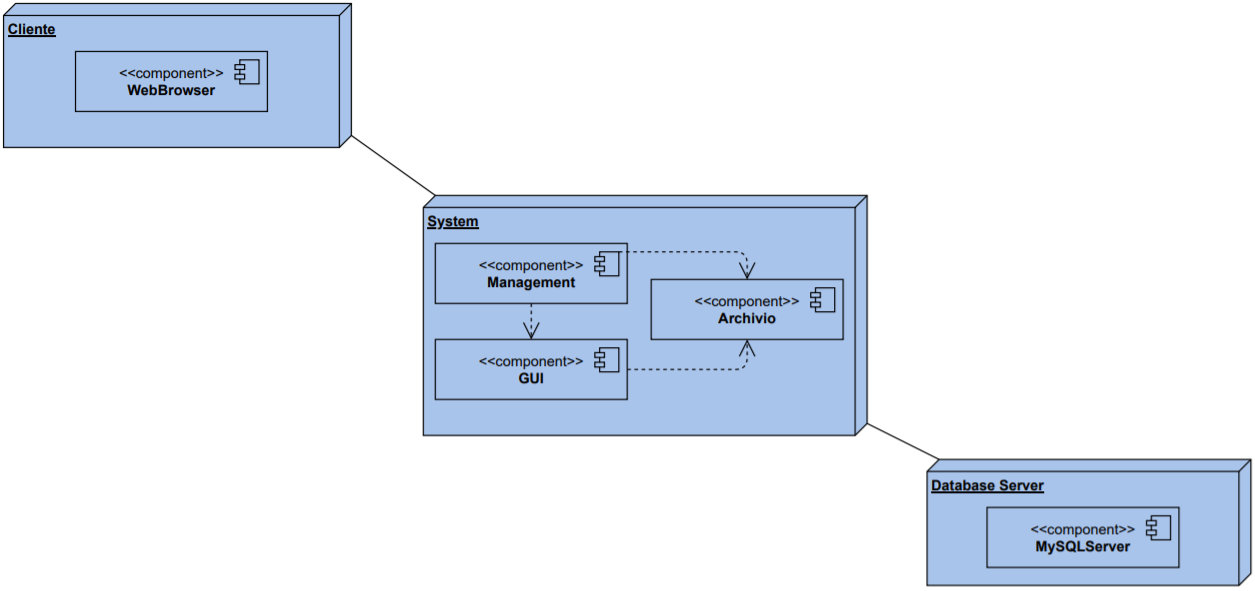
* Utente Non Registrato Management;
* Utente Media Manager Management;
* Utente Standard Management;
* Error Management.

Il livello Model prevede la gestione di un unico componente:

* Archivio.

**Deploiment Diagram**

L’utente (Cliente) richiede le funzionalità tramite l’interfaccia che il sistema mette a disposizione, a patto che si possieda un browser capace di interpretare javascript, in modo che le funzioni definite dal sistema possano essere eseguite in maniera corretta. Il tier del Cliente connette lo strato di view del System sul quale vengono eseguite le funzioni apposite al completamento degli obiettivi del Cliente.



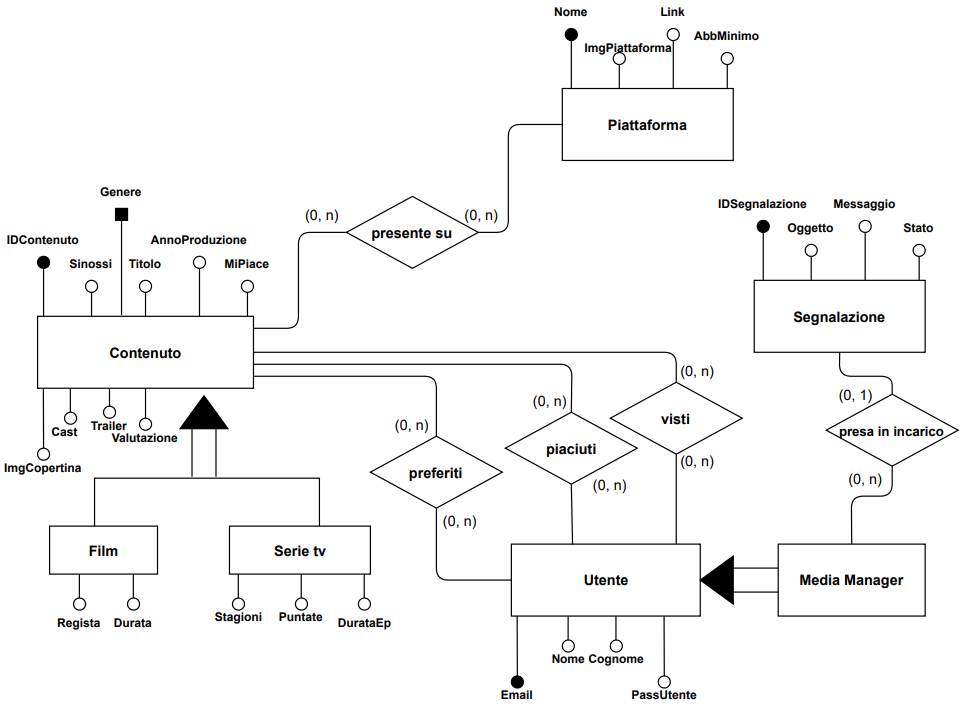
# **3.3 Hardware/software mapping**

Il sistema che si desidera sviluppare utilizzerà una struttura hardware costituita da un Server che risponderà ai servizi richiesti dai client.

Il client è una qualsiasi macchina attraverso la quale un utente potrà collegarsi, utilizzando una connessione internet per accedere al sistema, mentre la macchina server gestisce la logica e i dati persistenti contenuti nel database. Il client e il server saranno connessi tramite il protocollo HTTP, con il quale il client inoltra delle richieste al server e quest'ultimo provvederà ad utilizzare il protocollo TCP/IP per connettersi al Database e fornire i servizi richiesti dal Client. Le componenti hardware e software necessarie per il client sono una macchina (pc, tablet, smartphone…) dotata di connessione internet e di un browser installato su di essa. Per il server, invece, c'è necessità di una macchina con connessione ad Internet e con la capacità di immagazzinare una grande quantità di dati. La componente software necessaria è dunque un DBMS, per consentire la comunicazione con più client.

# **3.4 Persistent data management**

Per la memorizzazione dei dati si è scelto di utilizzare un Database relazionale con un DBMS che consente un accesso protetto, quindi utenti diversi possono accedere a diverse sezioni del database con operazioni diversificate.



**CONTENUTO**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IDContenuto** | **Sinossi** | **ImgCopertina** | **Titolo** | **AnnoProduzione** | **MiPiace** | **Cast** | **Trailer** | **Valutazione** |

**GENERE**

|  |  |
| --- | --- |
| **Contenuto\_IDContenuto** | **Tipo** |

**FILM**

|  |  |  |
| --- | --- | --- |
| **Contenuto\_IDContenuto** | **Regista** | **Durata** |

**SERIE TV**

|  |  |  |  |
| --- | --- | --- | --- |
| **Contenuto\_IDContenuto** | **Stagioni** | **Puntate** | **DurataEp** |

**SEGNALAZIONE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IDSegnalazione** | **Oggetto** | **Messaggio** | **Stato** | **Utente\_Email** |

**UTENTE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Email** | **Nome** | **Cognome** | **Password** | **MediaManager** |

**PIATTAFORMA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **ImgPiattaforma** | **Link** | **AbbMinimo** |

**PREFERITI**

|  |  |
| --- | --- |
| **Contenuto\_IDContenuto** | **Email** |

**PIACIUTI**

|  |  |
| --- | --- |
| **Contenuto\_IDContenuto** | **Email** |

**VISTI**

|  |  |
| --- | --- |
| **Contenuto\_IDContenuto** | **Email** |

**PRESENTE\_SU**

|  |  |
| --- | --- |
| **Contenuto\_IDContenuto** | **Nome** |

**CONTENUTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| IDContenuto | INT | NOT NULL | UNSIGNED, AUTO\_INCREMENT | PRIMARY KEY |
| Titolo | VARCHAR(150) | NOT NULL |  |  |
| Sinossi | VARCHAR(500) | NOT NULL |  |  |
| AnnoProduzione | VARCHAR(4) | NOT NULL |  |  |
| ImgCopertina | VARCHAR(2048) | NOT NULL |  |  |
| Trailer | VARCHAR(2048) | NOT NULL |  |  |
| MiPiace | INT | NOT NULL | UNSIGNED |  |
| Cast | VARCHAR(500) | NOT NULL |  |  |
| Valutazione | FLOAT | NOT NULL | CHECK |  |

**GENERE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Contenuto\_IDContenuto | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Tipo | VARCHAR(15) | NOT NULL | CHECK |  |

**FILM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Contenuto\_IDContenuto | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Durata | FLOAT | NOT NULL |  |  |
| Regista | VARCHAR(100) | NOT NULL |  |  |

**SERIE TV**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Contenuto\_IDContenuto | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Stagioni | VARCHAR(2) | NOT NULL |  |  |
| DurataEp | FLOAT | NOT NULL |  |  |
| Puntate | VARCHAR(4) | NOT NULL |  |  |

**PIATTAFORMA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Nome | VARCHAR(150) | NOT NULL |  | PRIMARY KEY |
| ImgPiattaforma | VARCHAR(2048) | NOT NULL |  |  |
| Link | VARCHAR(2048) | NOT NULL |  |  |
| AbbMinimo | FLOAT | NOT NULL |  |  |

**SEGNALAZIONE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| IDSegnalazione | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Oggetto | VARCHAR(25) | NOT NULL |  |  |
| Messaggio | VARCHAR(50) | NOT NULL |  |  |
| Stato | VARCHAR(20) | NOT NULL | CHECK |  |
| Utente\_Email | VARCHAR(200) | NOT NULL |  |  |

**UTENTE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Email | VARCHAR(200) | NOT NULL |  | PRIMARY KEY |
| Nome | VARCHAR(50) | NOT NULL |  |  |
| Cognome | VARCHAR(50) | NOT NULL |  |  |
| PassUtente | CHAR(32) | NOT NULL |  |  |
| Tipo | CHAR(2) | NOT NULL | CHECK |  |

**PRESENTE\_SU**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Contenuto\_IDContenuto | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Piattaforma\_Nome | VARCHAR(150) | NOT NULL |  |  |

**PREFERITI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Contenuto\_IDContenuto | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Utente\_Email | VARCHAR(200) | NOT NULL |  |  |

**PIACIUTI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Contenuto\_IDContenuto | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Utente\_Email | VARCHAR(200) | NOT NULL |  |  |

**VISTI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOME** | **TYPE** | **NULL** | **ATTRIBUTI** | **KEY** |
| Contenuto\_IDContenuto | INT | NOT NULL | UNSIGNED | PRIMARY KEY |
| Utente\_Email | VARCHAR(200) | NOT NULL |  |  |

# **3.5 Access control and security**

All’interno del sistema figurano diversi attori che hanno il permesso di eseguire diverse operazioni.

Per tener traccia del controllo degli accessi si è utilizzato una matrice degli accessi.

Il modello a cui ci siamo ispirati per realizzare la matrice degli accessi è stato il Global Access Table, il quale rappresenta ogni cella nella matrice come una tupla (actor, class, operation).

|  |  |  |
| --- | --- | --- |
| **ATTORE** | **CLASSE** | **OPERAZIONI** |
| Utente Non Registrato | CONTENUTO | * Ricerca contenuti * Filtra contenuti |
| Utente Registrato | CONTENUTO | * Marca contenuti |
| Utente Registrato | SEGNALAZIONE | * Invio segnalazioni |
| Utente Media Manager | CONTENUTO | * Aggiunge contenuto di tipo film * Aggiunge contenuto di tipo serie tv * Rimuove contenuto |
| Utente Media Manager | PIATTAFORMA | * Aggiunge piattaforma * Rimuove piattaforma |
| Utente Media Manager | CONTENUTO, PIATTAFORMA | * Aggiunge una piattaforma ad un contenuto * Rimuove una piattaforma da un contenuto |
| Utente Media Manager | SEGNALAZIONE | * Presa in incarico della segnalazione * Verifica della segnalazione |

Per gli Utenti Standard e Media Manager è richiesto il login affinché possano utilizzare le funzionalità sopra indicate.

I dati sensibili verranno criptati attraverso l’md5.

# **3.6 Global software control**

Il flusso di controllo è guidato dagli eventi.

Il sistema deve assicurarsi che molti utenti possano interagire con lo stesso, contemporaneamente, senza perdere la coerenza dei dati. Tutto questo può essere implementato dai gestori degli eventi, i quali permettono di attivare diversi eventi in tempi differenti.

L’applicazione web risponderà agli eventi in base all’ordine in cui essi arrivano. Ogni volta che l’utente interagisce con una componente, viene inviato un evento al sottosistema specifico. Un sottosistema funziona in base all’ordine degli eventi ed esso ignora il resto degli eventi che non sono rilevanti per il suo scopo.

Il meccanismo scelto per il flusso di controllo globale è di tipo Event-driven il quale risulta molto flessibile e adatto per le interfacce utenti.

# **3.7 Boundary conditions**

**Startup**

Per il primo start-up del sistema è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti. Viene visualizzata una homepage dalla quale sarà possibile accedere alla pagina relativa ai contenuti disponibili all’interno del sistema ed effettuare ricerche sugli stessi. Tramite l'interfaccia di Login, sarà possibile autenticarsi tramite opportune credenziali (indirizzo e-mail e password) per usufruire di funzionalità specifiche per l’utente che effettua l’accesso.

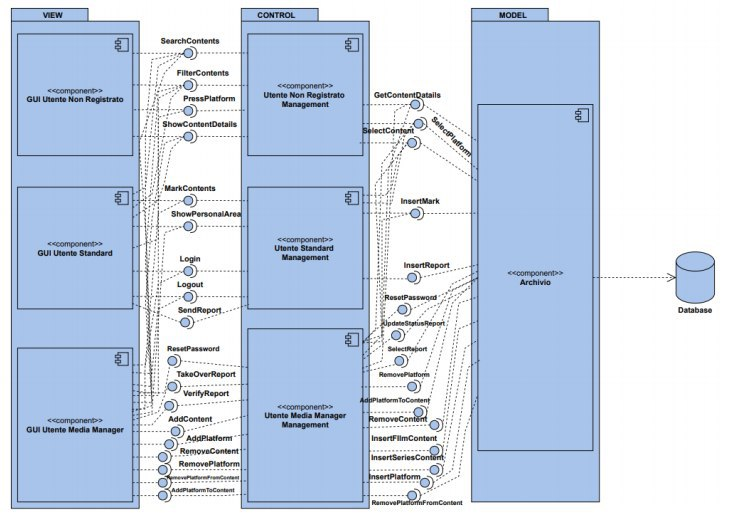
**Terminazione**

Al momento della corretta chiusura dell’applicazione web, si ha la terminazione del sistema con un regolare Log-out.

**Fallimento**

1. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione non sono previsti metodi che ripristino lo stato del Sistema precedente allo spegnimento non voluto;
2. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, contro il quale non è prevista alcuna contromisura.

# **4. Subsystem services**



L’interfaccia utente sarà il mezzo attraverso il quale gli utenti usufruiscono dei servizi dei sottosistemi.

Il sistema potrà essere utilizzato senza la necessità per l’utente di effettuare una registrazione o un login attraverso il sottosistema relativo all’utente non registrato. Naturalmente il sistema offrirà soltanto funzionalità circoscritte al ruolo che occupa l’utente. Affinché un utente possa marcare contenuti, deve dimostrare di avere accesso al sistema attraverso i servizi derivanti dal sottosistema dell’utente standard, oppure, nel caso voglia aggiungere/rimuovere contenuti e piattaforme, deve dimostrare di avere accesso al sistema attraverso i servizi derivanti dal sottosistema relativo all’utente media manager.

GUI Utente Non Registrato offre 4 servizi all’interfaccia Control:

* SearchContents;
* FilterContents;
* SignUp;
* ShowContentDetails;

GUI Utente Standard offre 10 servizi all’interfaccia Control:

* Login;
* Logout;
* MarkContents;
* SearchContents;
* PressPlatforms;
* FilterContents;
* ShowContentDetails;
* ShowPersonalArea;
* SendReport;
* ResetPassword;

GUI Utente Media Manager offre 17 servizi all’interfaccia di Visualizzazione:

* SearchContents;
* PressPlatforms;
* FilterContents;
* Login;
* Logout;
* MarkContents;
* ShowContentDetails;
* ShowPersonalArea;
* AddContent;
* AddPlatform;
* RemovePlatform;
* RemoveContent;
* RemovePlatformFromContent;
* AddPlatformToContent;
* VerifyReport;
* TakeOverReport;
* ResetPassword;

Utente Non Registrato Management offre 3 servizi all’interfaccia Model:

* GetContentDetails;
* SelectPlatform;
* SelectContent;

Utente Standard Management offre 6 servizi all’interfaccia Model:

* InsertMark;
* GetContentDetails;
* SelectPlatform;
* SelectContent;
* InsertReport;
* ResetPassword;

Utente Media Manager Management offre 16 servizi all’interfaccia Model:

* InsertMark;
* GetContentDetails;
* SelectPlatform;
* SelectContent;
* InsertFilmContent;
* InsertSeriesContent;
* InsertPlatform;
* RemovePlatform;
* RemoveContents;
* InsertPlatformToContent;
* RemovePlatformFromContent;
* SelectReport;
* UpdateStatusReport;
* VerifyReport;
* TakeOverReport;
* ResetPassword;