SDD (System Design Document)

Smart Cooking

Riferimento	
Versione	0.4
Data	08/11/2021
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Benedetta del Giudice, Daniele Colucci, Marco Luisi, Antonio Alfano, Anton Zolotko
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
08/12/2020	0.1	Inizio stesura SDD	Benedetta del Giudice, Daniele Colucci, Marco Luisi, Antonio Alfano, Anton Zolotko
08/09/2021	0.2	Modifica della sezione 3.4 (Gestione dati persistenti), 3.5(Controllo degli accessi e sicurezza) e 3.6(Controllo flusso globale del sistema) e revisione	Benedetta del Giudice
15/10/2021	0.3	Revisione	Benedetta del Giudice, Daniele Colucci, Marco Luisi, Antonio Alfano, Anton Zolotko
18/10/2021	0.4	Revisione	Benedetta del Giudice, Daniele Colucci, Marco Luisi, Antonio Alfano, Anton Zolotko

Sommario

1. Introduzione

1.1 Obiettivi del sistema

Il sistema da realizzare ha come scopo fornire agli utenti la fruizione delle videolezioni online di cucina realizzate dagli insegnanti dell'azienda cliente.
Il nostro obiettivo è di realizzare un sistema che consenta di eliminare gli attuali disagi degli utenti derivati dalla situazione d'emergenza causata dalla pandemia di Covid-19, garantendo l'erogazione costante dei contenuti dell'azienda cliente.
Per raggiungere questo obiettivo sono stati analizzati i limiti della didattica in presenza che costituisce il fulcro del sistema corrente.
La fruizione online porterebbe eventuali benefici anche in situazioni non critiche come l'attuale pandemia di Covid-19.
Il sistema proposto è una applicazione web, i quali contenuti sono gestiti dagli amministratori, che permetterà agli utenti di visualizzare le videolezioni.
Il sistema può essere diviso in due categorie:

- Gestione dei video: videolezioni caricate dagli amministratori che potranno essere visualizzate dagli utenti.
- Gestione degli account: account che gli utenti possono creare per poter accedere alla piattaforma e opzionalmente sottoscrivere un abbonamento.

Il sistema dovrà consentire il login per entrambe le figure individuate (utente, amministratore) a seconda dei dati di login immessi.
Il sistema dovrà consentire agli amministratori di effettuare il login, caricare, modificare ed eliminare le videolezioni e visualizzare la lista degli utenti registrati alla piattaforma. Il sistema dovrà consentire agli utenti di registrarsi, effettuare il login, visualizzare e ricercare le video lezioni, e sottoscrivere l'abbonamento per accedere ai contenuti non gratuiti.

1.2 Design Goals

I Design Goals sono organizzati in cinque categorie: Performance, Dependability, Cost, Maintenance, End User Criteria. I Design Goals identificati nel nostro sistema sono i seguenti:

Criteri di performance

- Tempo di risposta:

Il sistema deve garantire tempi di risposta minimi per l'esecuzione dei task al fine di evitare l'insoddisfazione degli utilizzatori.

- Memoria:

Lo spazio necessario per il funzionamento del sistema è determinato principalmente dalla memoria impiegata per l'archivio dei video.

Criteri di affidabilità

- Robustezza:

Il sistema provvederà a notificare gli utenti in caso di eventuali errori quando vengono immessi degli input non validi, attraverso l'utilizzo di appositi messaggi.

- Affidabilità:

Il sistema deve assicurare l'affidabilità delle funzionalità offerte, controllando in maniera accurata le informazioni immesse in input dagli utenti.

- Disponibilità:

Il sistema non prevederà limitazioni relative alle fasce di orario di utilizzo. Sarà pertanto disponibile a tutti gli utenti ogni qualvolta avranno intenzione di utilizzarlo; ma saranno previsti periodi di manutenzione in cui non sarà possibile l'accesso.

- Tolleranza ai guasti:

Il sistema potrebbe essere soggetto a fallimenti, durante lo svolgimento di una funzionalità, dovuti a varie cause tra cui problemi legati alla gestione del database. In questo caso viene notificato l'errore all'utente, invitandolo a riprovare successivamente.

- Sicurezza:

Il sistema garantirà sicurezza consentendo l'accesso mediante username e password e fornendo meccanismi di protezione dei dati che impediscano di usufruire di funzionalità di cui non si dispone l'autorizzazione.

Criteri di costi

- Costo di sviluppo:

È stimato un costo complessivo di 250 ore per la progettazione e lo sviluppo del sistema (50 per ogni project member).

Criteri di manutenzione

- Estensibilità:

Il sistema potrà essere esteso con l'aggiunta di nuove funzionalità dettate da vari fattori come esigenze degli

utenti e sviluppo del dominio applicativo

- Adattabilità:

Il sistema, inizialmente pensato come una piattaforma di streaming per le videolezioni di cucina, potrà essere adattato anche ad altri domini applicativi.

- Modificabilità:

Il sistema sarà di facile manutenzione in modo da modificare con relativa semplicità le funzionalità offerte in caso di eventuali problemi o per effettuare miglioramenti.

- Portabilità:

Il sistema sarà portabile in quanto viene sviluppato come un'applicazione web con la quale si interagisce tramite browser. Sarà dunque accessibile da qualsiasi dispositivo che disponga di un browser.

- Tracciabilità dei requisiti:

Sarà possibile tracciare i requisiti attraverso una matrice di tracciabilità.

Criteri di usabilità

- Usabilità:

Il sistema fornirà agli utenti un'interfaccia intuitiva che consentirà una comprensione immediata delle funzionalità e di utilizzarle in modo relativamente semplice.

- Utilità:

Il sistema si dimostra particolarmente utile per gli utenti che sono penalizzati dai limiti imposti dalla didattica in presenza.

1.2.1 Design Trade-off

Memoria vs Estensibilità: Il sistema può essere esteso attraverso l'aggiunta di nuove funzionalità, mettendo in secondo piano il consumo di memoria.
Tempo di risposta vs Affidabilità: Il sistema metterà in primo piano l'affidabilità controllando in maniera accurata i dati immessi in input ma senza trascurare totalmente i tempi di risposta.
Disponibilità vs Tolleranza ai guasti: Il sistema, in caso si verifichino errori in una funzionalità, la disattiva per un determinato periodo di tempo, restando però sempre disponibile per l'utente.
Criteri di manutenzione vs Criteri di performance: Il sistema sarà implementato preferendo la manutenibilità alla performance in modo da facilitare gli sviluppatori nel processo di aggiornamento del software a discapito delle performance del sistema.

1.3 Definizioni, acronimi e abbreviazioni

RAD: Requirements Analysis Document

SDD: System Design Document;

DB: DataBase;

DBMS: Database Management System.

SQL: Structured Query Language; linguaggio standardizzato per database basati sul modello relazionale (RDBMS) progettato per: creare e modificare schemi di database.

1.4 Riferimenti

NC02_RAD_V2.2

Slide del corso, presenti sulla piattaforma e-learning,

Libro:
-- Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition
Autori:

-- Bernd Bruegge & Allen H. Dutoit

1.5 Panoramica

Capitolo 1: Contiene l'introduzione con l'obiettivo del sistema, i design goals e un elenco di definizioni, acronimi e abbreviazioni utili alla comprensione dell'intera documentazione.

Capitolo 2: Descrive, nel caso esista, le funzionalità offerte dal sistema corrente.

Capitolo 3: Viene presentata l'architettura del sistema proposto, in cui sarà gestita la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.

Capitolo 4: Vengono presentati i servizi dei sottosistemi.

2. Architettura del Sistema corrente

Attualmente non esiste un prodotto software finalizzato alla fruizione online delle video lezioni di cucina dell'azienda cliente. Esse, infatti, vengono tenute da insegnanti specializzati in apposite sedi, previa sottoscrizione di un abbonamento periodico. Sono, quindi, effettuate in presenza pertanto per seguirle l'utente si reca nella sede indicata dove la segreteria verifica i suoi dati e lo stato dell'abbonamento, e in caso risultino regolari gli consente l'accesso. Tutto ciò risulta molto sconveniente per gli utenti che non hanno la possibilità di raggiungere fisicamente la struttura e/o di adeguarsi agli orari prestabiliti per le lezioni. Per far fronte a tali problemi, accentuati dalla situazione d'emergenza attuale causata dalla pandemia di Covid-19,

abbiamo deciso di sviluppare una piattaforma online che consentirà agli utenti la fruizione online delle lezioni, semplicemente registrandosi al sito. Questo gli concederà loro anche la possibilità di sottoscrivere l'abbonamento per visualizzare le video lezioni non gratuite.

3. Architettura del Sistema proposto

3.1 Panoramica

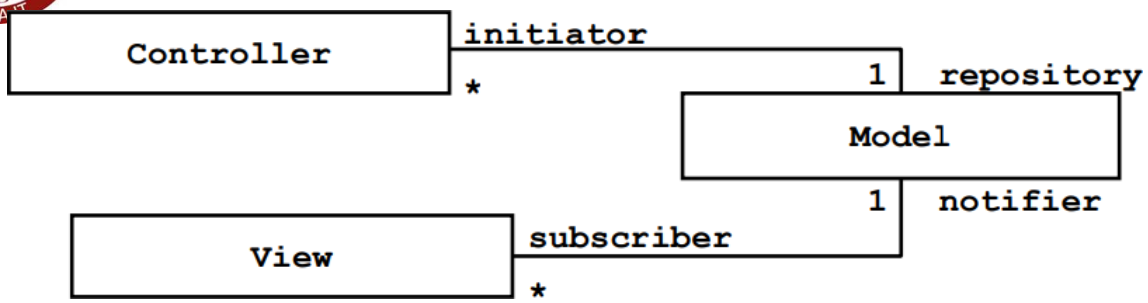
Il sistema da noi proposto è un'applicazione web che prevede due tipologie di utilizzatori: utente e amministratori. Entrambi potranno effettuare login e logout; ma i primi avranno anche la possibilità di registrarsi al sito. Il sistema prevede inoltre diverse tipologie di funzionalità, di cui alcune saranno esclusive degli amministratori mentre altre esclusive degli utenti. In particolare, questi ultimi potranno visualizzare le video lezioni gratuite, ma sottoscrivendo un abbonamento potranno accedere anche ai contenuti a pagamento. Tutte le video lezioni saranno caricate dagli amministratori che provvederanno a eliminarle e/o a modifcarne i dati in caso di necessità. 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 nello sviluppo di interfacce grafiche di sistemi software object-oriented, in grado di separare logica di presentazione dei dati, dalla logica di business. È un'architettura multi-tier: le varie funzionalità del sito sono logicamente separate e suddivise su più strati o livelli software differenti in comunicazione tra loro.

3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in Layer

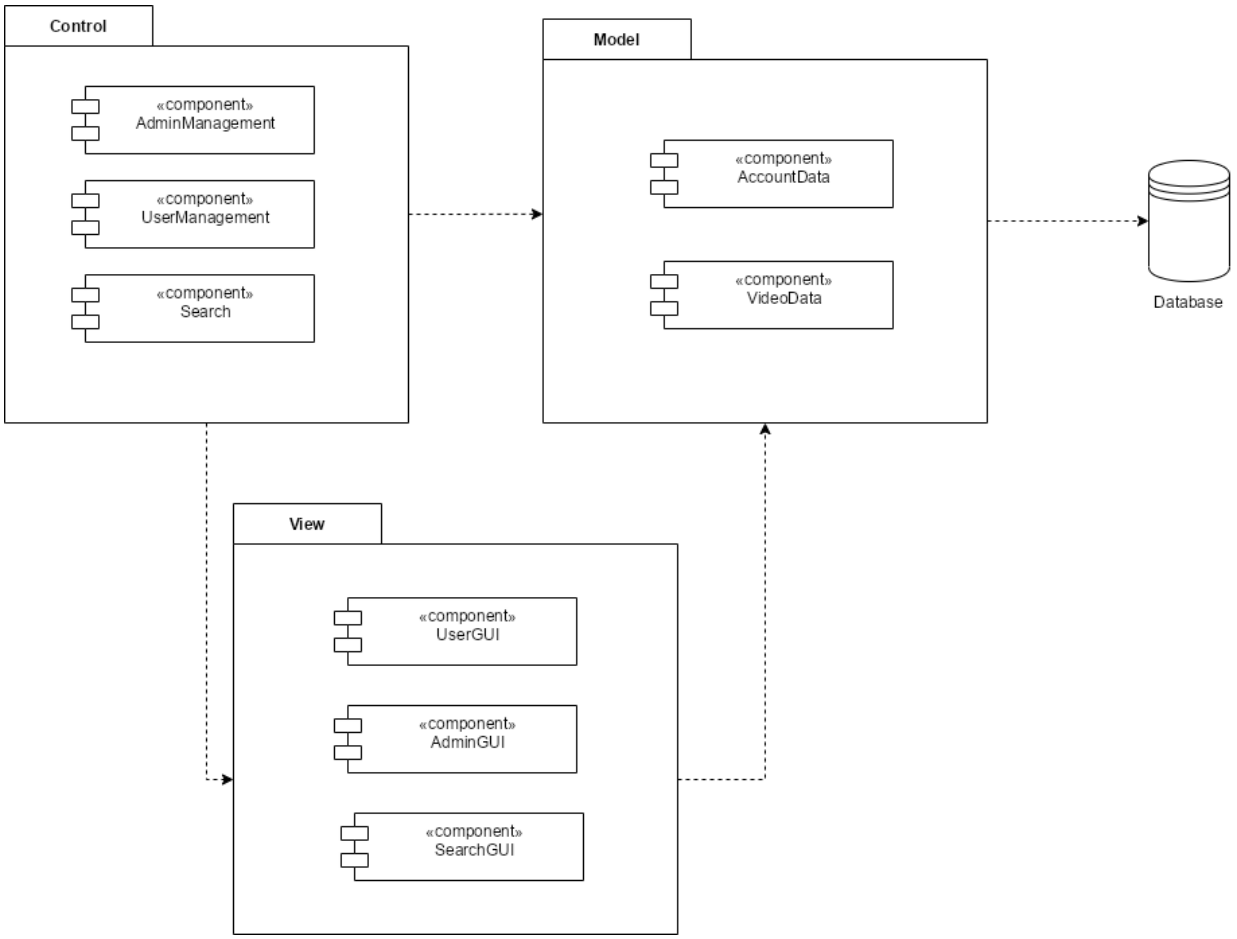
La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestirne aspetti e funzionalità differenti:

- **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 tra i sottosistemi;



3.2.2 Decomposizione in sottosistemi

Dopo un'analisi effettuata sul sistema, abbiamo deciso di suddividerlo nei seguenti sottosistemi in modo tale da utilizzare un'architettura aperta per motivi di efficienza. Si è deciso di gestire i singoli componenti con basso accoppiamento ed elevata coesione in modo tale da garantire, in caso di successive modifiche il minor numero di aggiornamenti da apportare tra tutti i sottosistemi.
Il sistema si compone di nove sottosistemi:


Control
«component»
AdminManagement
«component»
UserManagement
«component»
Search
Model
«component»
AccountData
«component»
VideoData
Database
View
«component»
UserGUI
«component»
AdminGUI
«component»
SearchGUI

Il livello View prevede la gestione di due sottosistemi e possiamo identificarli come oggetti "boundary" individuati nel RAD:

- UserGUI: si occupa di creare dinamicamente le pagine utilizzate dall'utente;
- AdminGUI: si occupa di creare dinamicamente le pagine utilizzate dall'amministratore;
- SearchGUI: si occupa di creare dinamicamente le pagine riguardanti le ricerche.

Il livello Control prevede la gestione di tre sottosistemi:

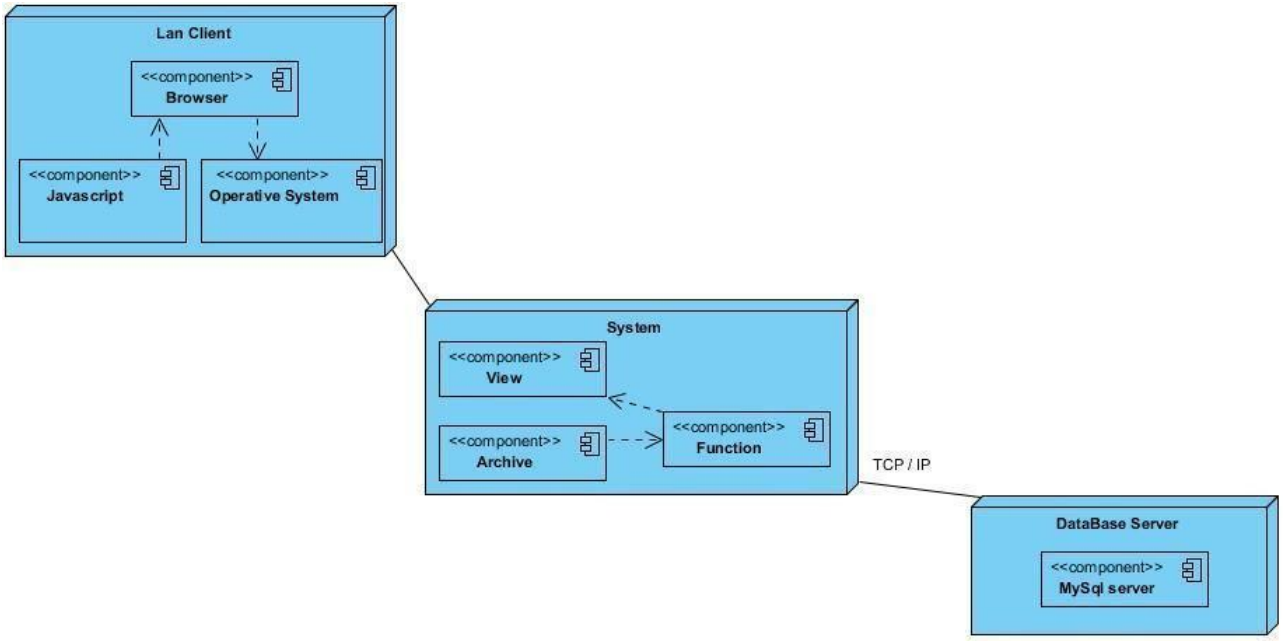
- AdminManagement: gestisce le azioni effettuate dall'amministratore, come il caricamento di video;
- UserManagement: gestisce le azioni effettuate dall'utente, come la sottoscrizione di un abbonamento;
- Search: gestisce le ricerche effettuate.

Il livello Model prevede la gestione di tre sottosistemi:

- AccountData: gestisce la comunicazione riguardante i dati degli account con il database;
- VideoData: gestisce la comunicazione riguardante i dati delle videolezioni con il database.

3.2.3 Deployment Diagram

Il deployment è diviso il 3 tier separati, ognuna caratterizzata dalle proprie funzioni. Lo strato del "Lan Client" è lo strato dedicato a chi utilizza il sito (dunque utente ed admin), il quale accede alla piattaforma tramite un qualsiasi dispositivo dotato di browser con Javascript. Il secondo strato è lo strato "System", dove vengono effettuate le operazioni di business, elaborando quindi le operazioni richieste dal Lan Client. Il terzo strato, lo strato più basso, è il "DataBase Server", che si occupa della gestione dei dati persistenti.

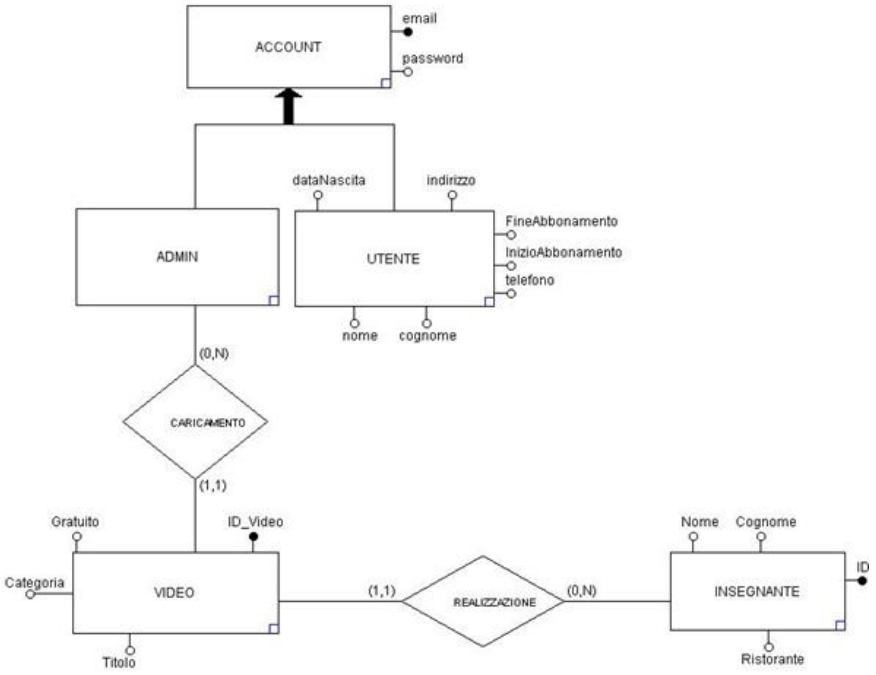


3.3 Mapping hardware/software

Il servizio richiede lo sviluppo di due parti software principali: client e server.
La parte server risponde alle richieste del cliente gestendo gli eventi e manipolando i dati persistenti. Il server è costituito da un dispositivo con connessione ad Internet, mentre le componenti software necessarie sono: un DBMS, per permettere la gestione e l'accesso ai dati persistenti in maniera efficiente e sicura, e del web server Apache Tomcat.
La parte client include il front-end, tramite il quale gli utenti e gli amministratori usufruiranno dei servizi offerti. Il client richiede un qualsiasi dispositivo dotato di browser con Javascript attivo e una connessione Internet funzionante.
Il client e il server comunicheranno tramite il protocollo HTTP.

3.4 Gestione dati persistenti

Per la gestione dei dati persistenti si è scelto di utilizzare un Database relazionale che dispone di un grande quantità di spazio per la memorizzazione delle videolezioni e permette di gestire i dati in maniera efficiente con bassi tempi di risposta. Infine utenti diversi potranno accedere a diverse sezioni del database a seconda dei loro permessi.


ACCOUNT
email
password
dataNascita
indirizzo
ADMIN
UTENTE
FineAbbonamento
InizioAbbonamento
telefono
nome
cognome
(0,N)
CARICAMENTO
(1,1)
Gratuito
ID_Video
Categoria
VIDEO
Titolo
(1,1)
REALIZZAZIONE
(0,N)
Nome
Cognome
ID
INSEGNANTE
Ristorante

ADMIN

NAME	TYPE	NULL	KEY
Email	VARCHAR (50)	NOT NULL	PRIMARY KEY
Password	VARCHAR (20)	NOT NULL	

UTENTE

NAME	TYPE	NULL	KEY
Email	VARCHAR(50)	NOT NULL	PRIMARY KEY
Password	VARCHAR(20)	NOT NULL	
Nome	VARCHAR(20)	NOT NULL	
Cognome	VARCHAR(20)	NOT NULL	
dataNascita	date	NOT NULL	
Telefono	VARCHAR(15)	NOT NULL	
InizioAbbonamento	date		
FineAbbonamento	date		

VIDEO

NAME	TYPE	NULL	KEY
ID_Video	int	NOT NULL	PRIMARY KEY
Titolo	VARCHAR(30)	NOT NULL	
Categoria	VARCHAR(20)	NOT NULL	
Gratuito	boolean	NOT NULL	
ID_Insegnante	Int	NOT NULL	FOREIGN KEY
Email_Admin	VARCHAR(50)	NOT NULL	FOREIGN KEY

INSEGNANTE

NAME	TYPE	NULL	KEY
ID_Insegnante	int	NOT NULL	PRIMARY KEY
Nome	VARCHAR(20)	NOT NULL	
Cognome	VARCHAR(20)	NOT NULL	
Ristorante	VARCHAR(30)		

3.5 Controllo degli accessi e sicurezza

In Smart Cooking ci sono diversi attori che hanno il permesso di accedere a diverse funzionalità e informazioni. Per schematizzare al meglio il controllo degli accessi si è utilizzata una matrice degli accessi, dove le righe rappresentano gli attori e le colonne le classi. Ogni entry (attore, classe) contiene le operazioni consentite da quell'attore sulle istanze di quella classe.

Sottosistema / Attore	Gestione	
	Video	Account
Utente	• Visualizzazione videolezione • Ricerca per parole chiave • Ricerca per categorie	• Registrazione • Login • Logout • Modifica dati • Sottoscrizione abbonamento
Amministratore	• Ricerca per parole chiave • Ricerca per categorie • Visualizza lista • Aggiunta • Modifica • Eliminazione	• Login • Logout

3.6 Controllo flusso globale del sistema

Il flusso del sistema Smart Cooking fornisce una funzionalità che richiede una continua interazione da parte dell'utente, ragione per cui, il controllo del flusso globale del sistema è di tipo event-driven, ovvero guidato dagli eventi.

3.7 Condizioni limite

Start-up

Il primo start-up avviene tramite l'avvio della macchina che ospita il server. In seguito si procede con l'avvio del database contenente i dati delle videolezioni e degli utenti, e, in caso di successo, l'avvio di un server Tomcat e il deployment dell'applicazione web. In questo momento, in caso di successo di ogni avvio, la piattaforma sarà accessibile come specificato nella sezione 3.5.

Terminazione

Il processo di terminazione inizia con la corretta chiusura del server Tomcat, si perde la possibilità di accesso alla piattaforma. In questo momento si potrà eseguire lo spegnimento del database. Non è possibile eseguire la disconnessione dal database prima della chiusura del server Tomcat senza causare fallimenti.

Fallimento

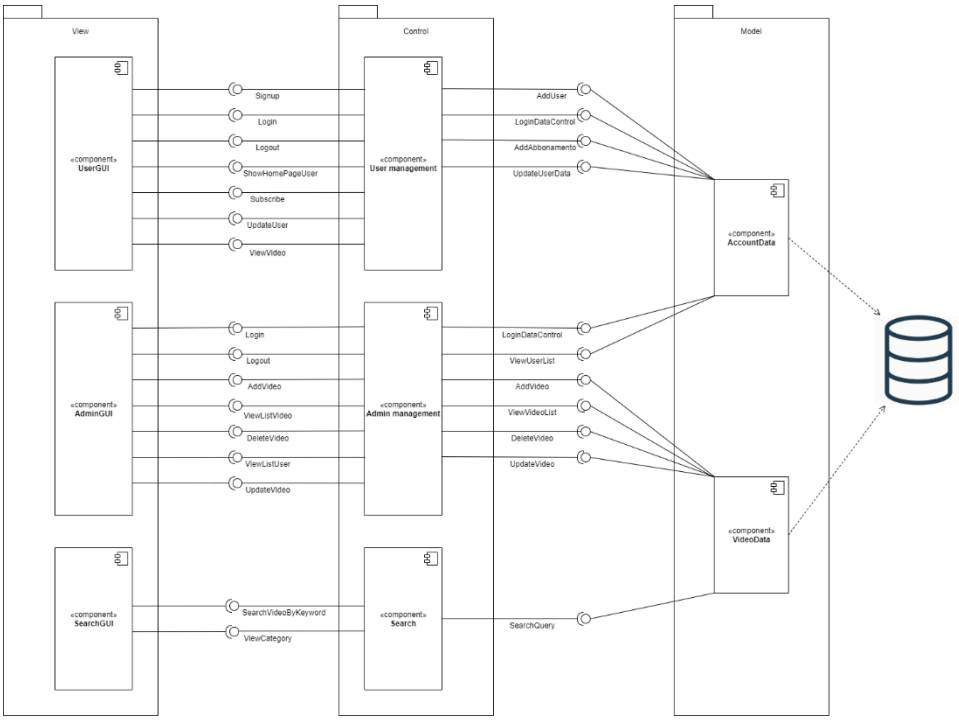
1. Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.

2. 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.

3. Nel caso di fallimento del software stesso, causato da una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione, non essendo previste politiche correttive, l'unica operazione consentita in questa particolare situazione è la chiusura del sistema e il suo successivo riavvio.

4. Nel caso di fallimento dovuto ad un errore critico nell'hardware, non è prevista alcuna contromisura.

4. Servizi dei Sottosistemi



View: Interfacce che gestiscono l'interfaccia grafica e gli eventi generati dall'interazione dell'utente con il sistema.

UserGUI offre 7 servizi all'interfaccia Control:

- Signup
- Login
- Logout
- ShowHomepageUser
- Subscribe
- UpdateUser
- ViewVideo

AdminGUI offre 7 servizi all'interfaccia Control:

- Login
- Logout

- AddVideo
- UpdateVideo

- ViewListVideo
- DeleteVideo
- ViewListUser

SearchGUI offre 2 servizi all'interfaccia Control:

- SearchVideoByKeyword
- ViewCategory

Control: Interfaccia che esegue i controlli sui dati ricevuti da "View" e invia a "Model" i dati da inserire nel database

UserManagement offre 4 servizi all'interfaccia Model:

- AddUser
- LoginDataControl
- AddAbbonamento
- UpdateUserData

AdminManagement offre 6 servizi all'interfaccia Model:

- LoginDataControl
- AddVideo
- UpdateVideo
- ViewVideoList
- ViewUserList
- DeleteVideo

Search offre 1 servizio all'interfaccia Model:

- SearchQuery