Università degli studi di Salerno

***Corso di Laurea in Informatica***



***PROGETTO DI INGEGNERIA DEL SOFTWARE – System Design Document***

***“BUY & SEE”***

**Indice**

* Introduzione

1.1. Scopo del sistema

1.2. Obiettivi di design

1.2.1 Criteri di Performance

1.2.2 Criteri di Affidabilità

1.2.3 Criteri di Costi

1.2.4 Criteri di Manutenzione

1.2.5 Criteri per l’Utente Finale

1.3 Definizioni, acronimi e abbreviazioni

1.4 Riferimenti

1.5 Panoramica

* Architettura del software corrente
* Architettura del software proposto

3.1. Panoramica

3.2.Decomposizione del sistema

3.3.Mapping Hardware/Software

3.4.Gestione dei dati persistenti

3.5.Controllo degli accessi e della sicurezza

3.6.Controllo del software globale

3.7.Condizioni Boundary

3.7.1Avvio del sistema

3.7.2Terminazione del sistema

3.7.3Fallimento del sistema

* Servizi dei sottosistemi

4.1 Gestione Autenticazione

4.2 Gestione Account

4.3 Gestione Registrazione

4.4 Gestione Prodotti

4.5 Gestione Utente

4.6 Gestione Fornitori

4.7 Gestione Store

* Glossario

**1. Introduzione**

**1.1 Scopo del sistema**

**1.2 Obiettivi di Design**

**1.2.1 Criteri di Performance**

**1.2.2 Criteri di Affidabilità**

**1.2.3 Criteri di Costo**

**1.2.4 Criteri di Manutenzione**

**1.3 Definizioni, acronimi e abbreviazioni**

**1.4 Riferimenti**

**1.5 Panoramica**

* **Architettura del software corrente**

Il nostro sistema software prende ispirazione da sistemi già esistenti che si occupano di fornire servizi simili su piattaforme online.

Il sito da noi ideato consente una gestione e un’iterazione semplice e veloce.L’utente potrà accedere ai vari servizi tramite un’interfaccia facile da comprendere e gestire. Un’interfaccia dedicata consentirà all’utente di visionare un film acquistato o gratuito in maniera comoda e pratica con il minor numero di iterazioni possibili e inserire recenzioni o valutazioni del film visionato.

Un sistema di gestione delle richieste consentirà al gestore di ricevere e valutare le proposte dell’utente per inserire nuovi film o riceve segnalazioni di problemi legati all’autenticazione o all’efficienza del sito.Un sistema di valutazione consentirà al gestore di controllare recensioni e valutazioni inserite dai vari utenti e eventualmente un sistema potrà disattivare il profilo di uno spettatore se le regole di buona condotta venissero violate.

L’utente , mediante un sistema di notifiche, verrà informato delle azioni fatte dal gestore riguardo le operazioni effettuate sul sito e le richieste inviate.

**3. Architettura del software proposto**

**3.1 Panoramica**

Il sistema da noi proposto è un applicazione web in locale, utilizzata da due tipi di utente: lo spettatore e i vari tipi di gestore(gestore catalogo, gestore account, gestore autenticazione, gestore visione contenuti). Gli spettatori per utilizzare i servizi offerti dal sito dovranno registrarsi.

La piattaforma metterà a disposizione diverse tipologie di funzionalità, a seconda del tipo di utente che effettua l'accesso.

In particolare lo spettatore potrà acquistare film e visionarli, inserire recenzioni o valutazioni e eventualemente inviare richieste che verranno inoltrate al gestore(che se ne occupa) per inserire film mancanti o per segnalare un problema.

**3.2 Decomposizione del sistema**

**3.2.1 Decomposizioni 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 sottosistem**

Abbiamo deciso di 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 livello View prevede la gestione di cinque sottosistemi:

-GUI Gestore Account

-GUI Gestore Catalogo

-GUI Gestore Autenticazione

-GUI Gestore VisioneContenuti

-GUI Spettatore

(Ogni componente rappresenta l'insieme di interfacce con cui gli utenti del sistema si interfacciano)

Il livello di Control invece contiene i seguenti sottosistemi:

-SpettatoreManger

-Gestore Catalogo Manager

-Gestore Autenticazione Manager

-Gestore VisioneContenuti Manager

-Gestore Account Manager

(Ogni componente raggruppa le funzionalità che ogni utente può effettuare e i servizi a cui può accedere)

Il livello Model prevede la gestione di un sottosistema che si interfaccia con il database:  
-Archivio

**3.2.2.1 Deployment Diagram**

L’utente 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 eseguire in maniera corretta.

Il tier Client connette lo strato di view del System sul quale vengono eseguite le funzioni apposite al completamento degli obiettivi del Lan Client

La parte Server racchiude e gestisce la persistenza dei dati. L’intera architettura non richiede ausilio di componenti hardware/software esterni.



**3.3 Mapping Hardware/Software**

Il sistema utilizzerà una struttura hardware costituita da un Server che risponderà ai servizi richiesti dai client. Il client è una qualsiasi macchina attraverso il quale un utente può 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 a fornire i servizi richiesti. Le componenti hardware e software necessarie per il client sono un computer dotato di connessione internet e di un web browser installato su di esso.

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 Gestione dei dati persistenti**

Per la memorizzazione dei dati si è scelto un Database relazionale che consente un accesso efficiente ai dati e brevi tempi di risposta.

**3.5 Controllo degli accessi e della sicurezza**

**3.6 Controllo del software globale**

**3.7 Condizioni Boundary**

**3.7.1 Avvio del sistema**

**3.7.2 Terminazione del sistema**

**3.7.3 Fallimento del sistema**

**4. Servizi dei sottosistemi**

**4.1 Gestione Autenticazione**

**4.2 Gestione Account**

**5. Glossario**