|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| System Design  Progetto  GuardaTV   |  |  | | --- | --- | | Riferimento |  | | Versione | 0.1 | | Data | 12/11/2021 | | Destinatario | Studenti di Ingegneria del Software 2021/22 | | Presentato da | Gruppo 16 | | Approvato da |  | |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 12/11/2021 | 0.1 | Design Goal | N. Cacace  S. Pastore  A. Prezioso  A. Ricchetti |
| 19/11/2021 | 0.2 | Identify subsistem e mapping hardware-software | N. Cacace  S. Pastore  A. Prezioso  A. Ricchetti |
| 26/11/2021 | 0.3 | Dati persistenti, controllo accessi, boundary condition | N. Cacace  S. Pastore  A. Prezioso  A. Ricchetti |

1. Design Goal

* **Dependability**
  + **Robustness:** Tutti i campi vengono verificati sia lato client che server in modo da sopportare input errati o non validi
  + **Security:** GuardaTV non deve permettere accesso non autorizzato ai dati degli utenti;
* **Cost** 
  + **Development Cost:** Il tempo per lo sviluppo di GuardaTV non deve superare le 50h/persona.
* **Maintenance**
  + **Estensibilità:** Possibilità di aggiungere nuove tipologie di contenuti, filtri di ricerca e di ordinamento.
  + **Portabilità\***
* **End user criteria**
  + **Usabilità:** Le funzioni di gestione delle liste devono essere intuitive e facili da utilizzare.

1. Architettura Software

Il sistema GuardaTV è un’applicazione web; nello specifico il sistema implementerà un pattern di tipo MVC, diffuso nello sviluppo di interfacce grafiche di sistemi software object-oriented in modo da separare la logica di presentazione dalla logica di business. Si tratta di un’architettura multi-tier, ovvero le funzionalità sono separate e suddivise in tre livelli (Model, View e Controller), in comunicazione tra loro.

* Decomposizione in sottosistemi

La decomposizione prevista da tre layer che si occupano di gestire aspetti e funzionalità differenti:

* + Model: memorizzazione e gestione dei dati persistenti;
  + Controller: gestione dello scambio dei dati tra i sottosistemi;
  + View: raccoglie e gestisce elementi di interfaccia grafica e gli eventi generati su di essi;

1. Accessi

Il sistema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Contenuto** | **Lista** | **Recensione** | **Utente** |
| **Utente non registrato** | Visualizza |  | Visualizza | registra |
| **Utente loggato** | Visualizza | Crea  Modifica  <elimina?> | Crea  elimina | Visualizza  Modifica |
| **Utente amministratore** | Crea  Modifica  eliminazione  Visualizza |  | Elimina |  |

1. Boundary Condition
   1. Avvio del sistema

Lo start-up del sistema prevede l’avvio del web server nel quale il sistema è installato e l’avvio del DBMS per accedere ai dati persistenti memorizzati nel database. Quando sia il web server che il DBMS sono in esecuzione, il sistema carica in memoria centrale le servlet principali attraverso le quali gli utenti possono effettuare le operazioni. Dopo l’avvio del sistema gli utenti possono interagire con esso.

* 1. Shut Down

Quando il sistema deve essere arrestato, il gestore del sistema termina l’esecutivo del web server. Quando ciò avviene tutte le risorse che il sistema utilizza (connessione al database e connessione alla rete) vengono rilasciate e nessun utente potrà più connettersi al sistema.

* 1. Fallimento