Università degli studi di Salerno

***Corso di Laurea in Informatica***

******

***Ingegneria del softwere***

***“StudyMe”***

*.*

**Studenti:**

##### **Nome Matricola**

Buono Claudia 05121 05296

Esposito Mariarosaria 05121 05692

Tripodi Maria Rachele 05121 05356

**Docente:**

Andrea De Lucia

*Anno Accademico: 2019/20*

**Sommario**

[1. Introduzione 3](#_Toc26267038)

[**1.1** **Scopo del sistema** 3](#_Toc26267039)

[**1.2** **Design goals** 3](#_Toc26267040)

[**1.3** **Definizioni, acronomi, e abbreviazioni** 4](#_Toc26267041)

[**1.4** **Riferimenti** 4](#_Toc26267042)

[2. Architettura del sistema corrente 4](#_Toc26267043)

[3. Architettura software proposta 4](#_Toc26267044)

[**3.1** **Decomposizione in sottosistemi** 4](#_Toc26267045)

[**3.2** **Mapping hardware/software** 6](#_Toc26267046)

[**3.3**  **Gestione dei dati persistenti** 6](#_Toc26267047)

[**3.3** **Controllo degli accessi e sicurezza** 9](#_Toc26267048)

[**3.4** **Controllo globale del software** 9](#_Toc26267049)

[**3.5** **Boundary Condition** 9](#_Toc26267050)

[4. Glossario dei servizi del sottosistema 10](#_Toc26267051)

**System Design Document**

# Introduzione

## **Scopo del sistema**

La crescita dell’utilizzo del World Wide Web ha consentito la nascita di molteplici piattaforme di apprendimento online utili per migliorare la qualità dell’apprendimento facilitando l’accesso alle risorse e ai servizi. Prima di essi risultava difficile seguire corsi di apprendimento, sia per una mancanza di tempo, sia per la lontananza. La soluzione al problema è fornire una piattaforma di apprendimento, StudyMe, alla quale gli utenti possono accedere, acquistare e consultare videolezioni, di vario genere. Per avere una interazione con gli acquirenti e mirare al miglioramento della piattaforma stessa, dotare il sistema di una sezione recensioni. La soluzione è proposta già da altri siti web come Udemy e Lynda.

## **Design goals**

Gli obiettivi di progettazione rappresentano le qualità desiderate nel sistema e forniscono una serie coerente di criteri che devono essere considerati quando si prendono decisioni di progettazione. Sono stati identificati i seguenti obiettivi di progettazione:

* **Mantenimento:**
* Estendibilità: Il sistema dovrà essere propenso all’inserimento di nuove funzionalità. Quindi, si adotterà una metodologia di programmazione ben strutturata, dunque, di facile comprensione;
* Modificabilità: L’accesso al sistema è controllato da un sistema di autenticazione che categorizza gli utenti non permettendo l’accesso a parti del sistema non autorizzate, oltre alla presenza di filtri che impediranno accessi non previsti;
* **Utente Finale:**
* Usabilità: L’interfaccia utente deve contenere al suo interno un menu principale con eventuali sotto-categorie che permettono all’utente di raggiungere le varie pagine del sito e un menu categorie che permette di raggiungere le relative pagine contenti i pacchetti inerenti;
* **Affidabilità:**
* Affidabilità: Ogni account deve essere non accessibile da utenti non autorizzati**;**
* Robustezza: In caso di input non valido il sistema genererà una eccezione che sarà poi mostrata all’utente sotto forma di un messaggio di errore.
* Sicurezza**:** Le password degli utenti saranno protette da chiavi crittografiche affichè nessun malintenzionato possa scoprire le password degli amministratori e/o degli acquirenti.
* **Performance:**
* Memoria: Considerando i dati da memorizzare, ci si aspetta che i dati del sistema possano occupare circa 1GB di spazio sui server.
* Tempo di risposta: Il tempo di risposta da parte del sistema a una richiesta deve essere al massimo di 15 secondi. Nel caso di una connessione molto lenta tale aspetto potrebbe essere non garantito, ovviamente i tempi di risposta saranno più rapidi quanto più è veloce la tipologia di connessione utilizzata.

## **Definizioni, acronomi, e abbreviazioni**

* StudyMe: Nome del sistema che verrà sviluppato

## **Riferimenti**

* Il contesto è ripreso dal documento di analisi dei requisiti di StudyMe.

# Architettura del sistema corrente

StudyMe è un progetto di ingegneria Greenfield, quindi lo sviluppo comincia da zero e non abbiamo alcun software attuale da sostituire.

# Architettura software proposta

## **Decomposizione in sottosistemi**

Per la realizzazione del sistema sarà utilizzata una architettura MVC. Essa si struttura in tre componenti, ognuna con un compito diverso nel sistema:

* **Model**, si occupa di gestire i dati persistenti. Contiene infatti metodi di lettura e scrittura su Database. (javaBean)
* **View**, si occupa di far visualizzare i dati all’utente e si occupa dell’iterazione tra quest’ultimo e il sistema. (JSP)
* **Controller** riceve i comandi dell’utente attraverso le view e si occupa della logica d controllo dell’applicazione interagendo con le altre due componenti. (Servet)

Il sottosistema View è stato suddiviso in quattro sottosistemi

1. Sottosistema Autenticazione: Contiene le pagine di benvenuto.

2. Sottosistema Acquirente: oltre a tutte le interfacce grafiche del sottosistema Autenticazione, potrà accedere alla funzionalità di acquistare corsi, alla funzionalità di ricerca e di inserimento commenti e\o cancellazione (solo quelli da lui inseriti). Inoltre, potrà accedere ad interfacce grafiche aggiuntive come quella per la visualizzazione delle informazioni personali e la loro modifica.

3. Sottosistema Insegnante: Tale modulo si occuperà di gestire le funzionalità di creazione di un corso e visualizzazione dei contenuti dei propri corsi. Inoltre, potrà accedere ad interfacce grafiche aggiuntive come quella per la visualizzazione delle informazioni personali e la loro modifica.

4. Sottosistema Gestore: include le interfacce accessibile al supervisore per verifica dei corsi creati dagli insegnanti del sistema.

Il sottosistema Control è stato suddiviso in vari sottosistemi:

1. Sottosistema Gestione pacchetti: comprende tutte le operazioni per la creazione, la modifica, la verifica e l’acquisto di un pacchetto

2. Sottosistema Gestione lezioni: comprende tutte le operazioni per l’inserimento e la visualizzazione di una lezione, l’aggiunta e la rimozione dei commenti;

3. Sottosistema Gestione Utente: che include tutte le operazioni di creazione di un utente, visualizzazione delle informazioni personali, eliminazione di un utente e la verifica dei dati in fase di Login.

4. Sottosistema Gestione Pagamento: include tutte le operazioni per il completamento della transazione per l’acquisto di un corso (servizio esterno)

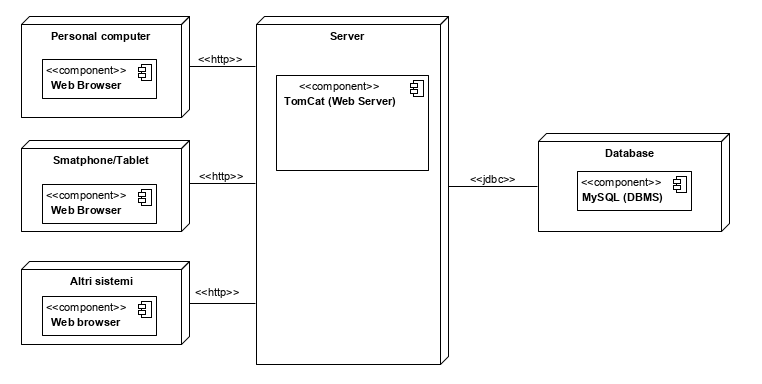
5. Sottosistema Gestione E-mail: include le operazioni per l’invio di e-mail di notifica agli utenti del sistema

## **Mapping hardware/software**

Il Sistema StudyMe utilizzerà un’architettura Client/Server.

Come server è stato scelto Apache Tomcat. Per quanto riguarda il DBMS, la scelta è ricaduta su MySQL, essendo esso open-source e largamente diffuso.

I vari client saranno infine rappresentati da i vari dispositivi attraverso cui l’utente utilizzera il sistema. Essi potranno essere un semplice pc, uno smartphone, un tablet, una smart tv o un qualsiasi dispositivo che abbia accesso ad internet mediante un web browser.



## 

## **3.3 Gestione dei dati persistenti**

Per la gestione dei dati persistenti scegliamo un database relazionale. Perché per natura i database sono potenti e supportano letture e scritture multiple.

Di seguito verrà illustrato il diagramma delle classi composto dagli entity che ci porterà alla progettazione di un database relazionale.

Immagine che contiene screenshot

Descrizione generata automaticamente

**Account**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **nomeUtente** | **varchar(20)** | **not null** | **primary key** |
| **password** | **varchar(30)** | **not null** |  |
| **email** | **varchar(50)** | **not null** |  |
| **tipo** | **varchar(20)** | **Not null** |  |

**Categoria**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **nomeCategoria** | **varchar(30)** | **not null** | **primary key** |
| **foto** | **varchar(300)** | **not null** |  |

**SottoCategoria**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **idSottocat** | **varchar(6)** | **not null** | **primary key** |
| **nomeSott** | **varchar(30)** | **not null** |  |

**Pacchetto**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **codicePacchetto** | **varchar(6)** | **not null** | **primary key** |
| **categoria** | **varchar(20)** | **not null** | **foreign key** |
| **idSott** | **varchar(6)** | **not null** | **foreign key** |
| **prezzo** | **double** | **not null** |  |
| **descrizione** | **varchar(1000)** | **not null** |  |
| **titolo** | **varchar(200)** | **not null** |  |
| **foto** | **varchar(300)** | **not null** |  |
| **nelCatalogo** | **boolean** | **not null** |  |
| **approvato** | **int** | **default= “0”** |  |

**Lezioni**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **url** | **varchar(100)** | **not null** | **primary key** |
| **titolo** | **varchar(200)** | **not null** |  |
| **durata** | **varchar(20)** | **not null** |  |
| **codiceP** | **varchar(6)** |  | **foreign key** |
| **approvato** | **int** | **default= “0”** |  |

**Ordine**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **numOrdine** | **int** | **not null,** **auto\_increment** | **primary key** |
| **nomeCliente** | **varchar(20)** | **not null** | **foreign key** |
| **dataOdierna** | **date** | **not null** |  |

**Acquisto**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **numAcquisto** | **int** | **not null, auto\_increment** | **primary key** |
| **numOrdine** | **int** | **not null** | **foreign key** |
| **codiceP** | **varchar(6)** | **not null** | **foreign key** |
| **titoloPacchetto** | **varchar(50)** | **not null** |  |
| **prezzo** | **double** | **not null** |  |

**Recensione**

|  |  |  |  |
| --- | --- | --- | --- |
| **NOME** | **TIPO** | **VINCOLI** | **KEY** |
| **idRecensione** | **int** | **not null, auto\_increment** | **primary key** |
| **userCliente** | **varchar(20)** | **not null** | **foreign key** |
| **codiceP** | **varchar(6)** | **not null** | **foreign key** |
| **commento** | **varchar(500)** | **not null** |  |
| **titolo** | **varchar(300)** | **not null** |  |

## **Controllo degli accessi e sicurezza**

Nel sistema sono presenti tre attori, essi hanno diversi ruoli e quindi possono eseguire diverse operazioni. Per schematizzare il controllo di seguito è riportata la matrice degli accessi, nella seguente tabella non ci sono i sottosistemi per la gestione del pagamento e delle e-mail. Dato che sono sistemi esterni

|  |  |  |  |
| --- | --- | --- | --- |
|  | Acquirente | Insegnante | Gestore |
| Sottosistema per la gestione degli utenti | * Login * Logout * Visualizzazione informazioni personali * Modifica profilo(e-mail, password) | * Login * Logout * Visualizzazione informazioni personali * Modifica profilo(e-mail, password) | * Login * Logout * Visualizzazione informazioni personali * Modifica profilo(e-mail, password) |
| Sottosistema per la gestione dei pacchetti | * Visionare pacchetti * Acquistare un pacchetto * Visualizzare anteprima gratuita | * Crea pacchetto * Modifica pacchetto * Elimina pacchetto | * Verifica pacchetto |
| Sottosistema per la gestione delle lezioni | * Visualizzare una lezione * Inserire un commento | * Inserire lezione * Elimina lezione * Modifica lezione | * Verifica lezione |

## **Controllo globale del software**

Il sistema ha un flusso guidato di eventi, le funzionalità richiedonoun’interazione continua da parte dell’utente; per questo motivo, il controllo del flusso globale che utilizziamo è di tipo procedure-driven.

## **Boundary Condition**

|  |  |  |
| --- | --- | --- |
| **ID:** | UC\_Avvio | |
| **Caso d’uso:** | Avvio del server | |
| **Istanze di attori partecipanti:** | Amministratore | |
| **Condizione di entrata:** | L’amministratore accede al sistema | |
| **Condizione di uscita:** | Il server è attivo e i relativi servizi sono disponibili | |
| **Flusso di eventi:** | L’amministratore avvia il server del sistema cliccando su “Avvia” | Il sistema si avvia e attiva i servizi in remoto andando a renderli disponibili per le richieste  Il sistema notifica all’utente che l’operazione di avvio del server è avvenuta con successo |
| **Eccezioni:** | Durante la fase di avvio del server, se si verifica un errore, l’amministratore riceverà una notifica di errore. | |

|  |  |  |
| --- | --- | --- |
| **ID:** | UC\_Spegnimento | |
| **Caso d’uso:** | Spegnimento del server | |
| **Istanze di attori partecipanti:** | Amministratore | |
| **Condizione di entrata:** | L’amministratore accede al sistema | |
| **Condizione di uscita:** | Il server smette di essere attivo e i relativi servizi non sono più disponibili | |
| **Flusso di eventi:** | L’amministratore spegne il sistema cliccando su “Spegni” | Il sistema effettua una scansione per verificare che non ci siano richieste in attesa, se vi sono, porta a termina tali richieste e, quando non ne ha più in sospeso, notifica all’amministratore l’inizio della procedura di spegnimento che sta eseguendo per poi terminare. |
| **Eccezioni:** | Durante la fase di spegnimento, se si verifica un qualsiasi errore, questo viene notificato all’amministratore. | |

# Glossario dei servizi del sottosistema

* **Gestione pacchetti**: Sottosistema che gestisce tutte le operazioni che riguardano i pacchetti del sistema. Le operazioni permesse sono:
  + *Crea pacchetto*: permette la creazione di un nuovo pacchetto
  + *Modifica pacchetto*: Permette la modifica di un pacchetto. Tale modifica può avvenire sia in via di creazione che dopo
  + *Elimina pacchetto*: permette la eliminazione di un pacchetto
  + *Verifica pacchetto*: Permette a un gestore di confermare o rifiutare un pacchetto appena creato
  + *Visionare pacchetti*: Permette di prendere visione dei pacchetti esistenti
  + *Acquista pacchetto*: Permette di acquistare un pacchetto
  + *Visualizzare anteprima gratuita:* Permette all’utente di visualizzare l’anteprima gratuita della lezione

Sono richiesti i servizi esterni di: e-mail, per notificare l’insegnate dell’eventuale disapprovazione del pacchetto e per notificare l’acquirente dell’eventuale pagamento; pagamento, per acquistare un pacchetto, nel sistema di occupa interamente paypal.

* **Gestione lezioni**: Sottosistema che gestisce tutte le operazioni che riguardano le lezioni del sistema. Le operazioni permesse sono:
  + *Inserire lezione:* Permette di inserire una lezione nel pacchetto
  + *Elimina lezione:* Permette di eliminareuna lezione dal pacchetto
  + *Modifica lezione:* Permette di modificare l’url e… di una lezione
  + *Verifica lezione:* Permette a un gestore di confermare o rifiutare una lezione inserita in un secondo momento nel pacchetto
  + *Visualizzare una lezione:* Permette di prendere visione delle lezioni
  + *Inserire un commento:* Permette l’inserimento di un commento
* **Gestione Utente**: Sottosistema che gestisce le informazioni e le operazioni dell’utente
  + *Login:* Permette a un utente registrato di loggarsi al sistema
  + *Logout:* Permette a un utente loggato di uscire dal sistema
  + *Visualizzazione informazioni personali*: Permette la visualizzazione delle informazioni personali a un utente
  + *Modifica profilo*: Permette a un utente la modifica del proprio profilo( username, password)
* **Gestione Pagamento**: Sottosistema che gestisce il pagamento, ovviamente solo l’acquirente potrà acquistare i pacchetti e il bonifico andrà al rispettivo insegnante
* **Gestione E-mail**: Sottosistema che gestisce l’invio delle e-mail
  + *E-mail RecuperoPassword*: Permette il recupero e quindi la modifica della password, per qualsiasi utente (Acquirente, Insegnante, Gestore)
  + *E-mail AvvertenzaRifiuto*: Permette al gestore di notificare all’Insegnante un eventuale rifiuto del pacchetto