RAD Requirement Analysis Document

YouLearn Platform Project



RAD Requirement Analysis Document

YouLearn Platform Project

|  |  |  |  |
| --- | --- | --- | --- |
| Mario Sessa, Luigi Crisci, Pasquale Ambrosio |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Cambiamenti | Autori |
| 13/11/2018 | 1.0 | Strutturazione RAD | Mario Sessa |
| 15/11/2018 | 1.0.1 | Inserimento del capitolo 1 e 2 | Mario Sessa |
| 16/11/2018 | 1.0.2 | Inserimento dei requisiti funzionali.  Inserimento dei requisiti non funzionali secondo il modello FURPS+. | Mario Sessa |

Sommario

1. [1. Introduzione 3](#_Toc433975253)

[1.1 Scopo del sistema](#_Toc433975254)

[1.2 Ambito del sistema](#_Toc433975255)

[1.3 Obiettivi e criteri di successo del sistema](#_Toc433975256)

[1.4 Definizioni, acronimi e abbreviazioni](#_Toc433975257)

[1.5 Riferimenti](#_Toc433975258)

[1.6 Panoramica](#_Toc433975259)

1. [2. Sistema corrente 3](#_Toc433975260)
2. [3. Sistema proposto 3](#_Toc433975261)

[3.1 Panoramica](#_Toc433975262)

[3.2 Requisiti funzionali](#_Toc433975263)

[3.3 Requisiti non funzionali](#_Toc433975264)

[3.4 Modello di sistema](#_Toc433975265)

1. [4. Glossario 3](#_Toc433975266)

1. Introduzione

## 1.1 Scopo del sistema

Il sistema nasce dall’idea di creare una piattaforma in grado di offrire un servizio di e-learning in cui l’utente può seguire uno o più corsi in maniera gratuita o a pagamento. Inoltre, ogni utente può creare, senza aver bisogno di specifiche autorizzazioni, uno o più corsi gratuiti o a pagamento. Nel secondo caso, il creatore dovrà pagare una quota fissa per completare la creazione del corso. I guadagni di un docente sono pari al 20% dei guadagni mensili ottenuti da un suo corso, questo perché si vuole creare una piattaforma in grado di incentivare la creazione di contenuti e migliorare l’esperienza e-learning che si vuole fornire. Il sistema offre un servizio di valutazione per ogni corso, cosi da poter valutare la qualità dei contenuti ancora prima di visualizzarli.

## 1.2 Ambito del sistema

Il sistema è riferito a quel dominio di utenti che vogliono condividere conoscenze o vogliono acquisirne di nuove. I contenuti presenti nella piattaforma sono di ogni tipo, ogni utente può seguire o tenere uno o più corsi su qualunque ambito che non violi il regolamento del sistema. In sintesi, quindi, la piattaforma è utilizzabile da qualsiasi utente che vuole condividere o approfondire una propria passione, qualsiasi essa sia.

## 1.3 Obiettivi e criteri di successo del sistema

Il sistema deve offrire un servizio in cui:

1. Preservare dati sensibili inseriti dall’utente
2. Mantenere i contenuti visibili a qualsiasi utente online
3. Eliminare risorse non idonee al regolamento
4. Poter effettuare una ricerca e trovare ciò che si stava ricercando tramite l’inserimento di poche keywords.
5. Poter fornire e-mail da sfruttare per far comunicare direttamente più utenti tra loro. Ad esempio, è utile per creare un contatto diretto tra un docente e i suoi studenti.

## 1.4 Definizioni, acronimi e abbreviazioni

## Studente: abbreviazione in “st.”, raffigura un utente che segue un corso.

* **Docente:** abbreviazione in “doc.”, raffigura un utente che gestisce un corso.
* **DA:** acronimo di “Docente Affiliato”, raffigura un utente che aiuta a gestire un corso di un docente.
* **Amministratore:** abbreviazione in “admin.”, raffigura un utente che gestisce i contenuti del sistema.
* **Corso:** Raffigura una raccolta di video-lezioni e informazioni varie di un medesimo contesto.
* **Lezione:** Video che illustra una spiegazione procedurale al fine di far imparare un determinato argomento ad uno studente del corso.
* **Segnalazione:** Invio di un messaggio per avvertire un amministratore di una violazione del regolamento.
* **Regolamento:** Insieme di norme, regole e vincoli che l’utente deve rispettare.

## 1.5 Riferimenti

È stato usato come riferimento il libro:

**Object-Oriented Software Engineering:**   
**Using UML, Patterns, and Java, 3rd Edition**   
Prentice Hall, Upper Saddle River, NJ, September 25, 2009.

Sono stati usati dei materiali di supporto visionabili al link: https://wwwbruegge.in.tum.de/lehrstuhl\_1/component/content/article/217-OOSE

## 1.6 Panoramica

Il documento è strutturato in modo da far visionare al lettore i procedimenti di analisi dei requisiti eseguita durante la progettazione della piattaforma YouLearn.

Il documento è diviso in 3 parti principali:

1. **Introduzione**:Descrive una panoramica generale del sistema, degli elementi principali e delle definizioni che si utilizzano durante la fase di progettazione.
2. **Sistema proposto:** Raffigura il sistema che si vuole progettare, i suoi requisiti, scenari, casi d’uso e comportamenti del sistema con un alto livello di astrazione.
3. **Glossario:** Raccolta delle terminologie tecniche o con significati propri del progetto.

2. Sistema corrente

Lo sviluppo di internet e la popolarità acquisita dal World Wide Web hanno permesso l’accesso ad una quantità smisurata di informazioni a chiunque disponga di una connessione internet, rendendo gran parte del sapere umano alla portata di tutti. Ciò ha portato alla nascita di enciclopedie, videocorsi e molteplici altre forme di contenuti volti alla diffusione e/o all’insegnamento di qualsivoglia disciplina, quali letteratura, scienze naturali e sociali, storia, filosofia etc.

Ciononostante, tali contenuti si presentano frammentati sul web, dispersi nella rete su pagine web differenti tra loro. Inoltre, avendo chiunque la possibilità di accedere alla rete, si pone il problema dell’affidabilità delle informazioni presenti che, spesso, non citano le fonti necessarie o sono influenzate dalle opinioni personali degli autori. Queste criticità rendono problematico l’apprendimento tramite il web, esponendo l’utente a contenuti non corretti, completi, chiari e coerenti.

Un’altra caratteristica importante è data dalla variabilità delle risorse del web. Molte informazioni sono utili per brevi periodi se non vengono aggiornate, una nuova versione di una risorsa rende inutile o parzialmente inutile le versioni precedenti. I sistemi di raccolta dati esistenti non forniscono meccanismi in grado di poter far aggiornare i loro contenuti in maniera dinamica, cosi da poter utilizzare sempre delle risorse aggiornate per problemi del presente.

3. Sistema proposto

## 3.1 Panoramica

L’idea alla base di YouLearn è quella di sviluppare un sistema di raccolta di video-lezioni dinamico divisi per argomenti. Tali raccolte sono identificate dai corsi; un utente può iscriversi gratuitamente o versando una quota specifica. I corsi sono strutturati in modo da poter inserire nuove lezioni visionabili successivamente da tutti gli studenti iscritti al corso aggiornato. Il sistema garantisce un meccanismo di ricerca di contenuti categorizzato in base alla qualità del corso. La qualità viene definita in base alle valutazioni che sono generate dai feedback degli studenti iscritti. Inoltre, il docente viene valutato tramite una media delle valutazioni dei corsi che sta tenendo cosi da orientare l’utente non solo in base ai corsi ma anche in base a chi li gestisce.

YouLearn si presenta come una piattaforma in cui sono presentati dei corsi gestiti da terzi. Una architettura di questo genere, prevede che tutte le responsabilità di un corso passano direttamente al docente (esclusi provvedimenti per la violazione del regolamento della piattaforma).

YouLearn si presenta anche come un servizio di intermediazione tra gli utenti, andando a richiamare servizi di messaggistica esterni per la ricezione o l’invio di e-mail grazie alla condivisione di indirizzi di posta elettronica personali e visibili a tutti gli utenti della piattaforma.

## 3.2 Requisiti funzionali

I requisiti sono suddivisi in base al ruolo che un utente adotta all’interno della piattaforma. Quindi un utente può assumere il ruolo di:

1. ***Studente*:** Uno *studente* deve poter:

1. Seguire un corso, che sia gratuito o a pagamento.
2. Dare un feedback ad un corso in cui è iscritto
3. Segnalare malfunzionamenti e corsi o utenti che violino il regolamento
4. Disiscriversi ad un corso
5. Inviare un messaggio ad un utente (DV)

2. ***Docente*:** Un *docente* deve poter:

1. Creare un corso, che sia gratuito o a pagamento
2. Aggiungere lezione ad un corso in maniera diretta o andando a confermare una lezione nella lista delle lezioni in attesa di conferma caricati dai docenti affiliati
3. Rispondere alle domande degli studenti (DV)
4. Creare notifiche per un corso
5. Modificare informazioni corso

3. ***Docente affiliato:***Un docente affiliato deve poter:

1. Inserire una lezione nella coda delle lezioni in attesa di approvazione

4. ***Amministratore:***Un *amministratore* deve poter:

1. Cancellare un corso

2. Bannare un utente

3. Visionare una segnalazione

avviare le procedure di cancellazione dei corsi e prendere provvedimenti sul dominio degli utenti in base alle segnalazioni che riceve.

## 3.3 Requisiti non funzionali

### Usabilità

* L’interfacciamento al sistema è semplice e comprensibile dall’utente, riesce a comprendere le sue funzionalità dopo il primo utilizzo.
* Il sistema riconosce e soddisfa le richieste di ricerca corso e utente all’inserimento di un massimo di 5 keywords.
* Il sistema mette a disposizione un manuale utente in grado di far comprendere all’utente il funzionamento del sistema in tutti i suoi aspetti.
* Il linguaggio utilizzato è semplice e intuitivo (ad eccezione del contratto di registrazioni che è descritto in maniera formale)

### Affidabilità

* Il sito è online 24 ore su 24 e 7 giorni su 7 salvo manutenzioni che verranno rese note mediante un avviso al posto della pagina di benvenuto.
* Le manutenzioni per la risoluzione di malfunzionamenti ha una durata massima di 5 ore (salvo casi gravi che possono estendere il tempo di manutenibilità fino ad un massimo di 12 ore).

### Prestazioni

* Il sistema deve poter offrire un servizio di streaming parallelo tra più utenti
* Un utente deve avere una connessione internet di almeno 1 Mbps per la fruizione corretta dei contenuti.
* Il sistema può subire un massimo di 2 secondi di ritardo quando è in uno stato di stress (Eccessivo numero di richieste di servizi del sistema da parte di un dominio di utenti molto ampio)

### Supportabilità

* La piattaforma è supportata dai principali web browser quali Firefox, Google Chrome, Internet Explorer, Safari.

### Implementazione

* Si usa l’UPPER-CASE Tool Balsamiq Mockup per l’implementazione dei Mockup in fase di analisi.
* Si usa l’UPPER-CASE Tool Visual Paradigm per la notazione UML.
* Si usa il LOWER-CASE Tool Eclipse per la fase di codifica.

### Interfaccia

* Si utilizza un API esterna per completare le transazioni su piattaforme come Paypal, Postepay e Mastercard.
* Si utilizza un API esterna per richiamare un servizio di messaggistica esterno in grado di inviare o ricevere e-mail.

### Packaging

* Il sistema non ha bisogno di un’operazione di installazione poiché è accessibile attraverso un web browser.

### Legali

* L’Utente in fase di registrazione firma un contratto per:
  + Rendere disponibile e visibile la propria e-mail di registrazione a tutti gli utenti che visitano il suo profilo personale
  + Rendere chiaro che i rimborsi di un corso non sono a responsabilità della piattaforma ma a carico legale del docente creatore.
  + Rendere chiaro che se un corso viene chiuso, i soldi spesi per l’acquisto di tale corso non sono rimborsabile.

## 3.4 Modello di sistema

### 3.4.1 Scenari

[Esempi di utilizzo, informali e completi]

### 3.4.2 Use Case

[PER OGNI REQUISITO FUNZIONALE produrre uno use case.   
INFO VALUTAZIONE: ogni componente del team deve comunicare alle PM il numero di use case prodotti e la stima del tempo impiegato. Sottolineiamo che il tempo necessario per capire come gli use case sono strutturati e le prove fatte per produrne alcuni di esempio NON deve essere contemplato nel tempo dedicato alla realizzazione degli use case per il progetto.]

|  |  |
| --- | --- |
| **Nome** |  |
| **Attore** |  |
| **Condizione di Ingresso** |  |
| **Flusso degli Eventi** |  |
| **Condizione di Uscita** |  |
| **Condizioni di Errore** |  |
| **Requisiti di Qualità** |  |

### 3.4.3 Modello ad oggetti

<CONVENZIONE NOMI:   
 entity: usare nome usato dall’utente finale e in accordo al dominio dell’applicazione  
 control: [nome\_use\_case]Control (di solito dovremmo avere un oggetto control in funzione di ogni attore coinvolto in ogni use case. Dato che l’attore è sempre il presidente di area didattica noi avremo un solo oggetto control per use case)  
 boundary: strumenti che fungono da interfaccia con l’utente per erogare la funzionalità richiesta dall’utente.

TABELLA RIASSUNTIVA DI TUTTI GLI OGGETTI:>

|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
|  | [boundary, entity, control] |  |
|  |  |  |

<PER OGNI USE CASE RIPORTARE IL DIAGRAMMA DELLE RELAZIONI TRA GLI OGGETTI ENTITY; BOUNDARY E CONTROL COINVOLTI NELLO USE CASE CORRENTE. Per gli oggetti Entity specificare gli attributi, per gli oggetti control specificare le operazioni/metodi>

### 3.4.4 Modelli dinamici

[Diagramma di sequenza o diagrammi della attività.   
INFO VALUTAZIONE: Comunicare il tempo impiegato per la produzione del diagrammi di sequenza o attività. Sottolineiamo che il tempo necessario per capire come gli use case sono strutturati e le prove fatte per produrne alcuni di esempio NON deve essere contemplato nel tempo dedicato alla realizzazione dei diagrammi per il progetto]

### 3.4.5 Interfaccia utente e mock-ups

### [CONSIGLIO: i programmi che abbiamo usato per la produzione dei mock up sono balsamiq (free per 30 giorni) e Pencil. Nel momento in cui sviluppate i mockup tenete presente che il sistema dovrebbe essere una web application.]

4. Glossario

[Inserire qui tutti i termini utilizzati nel documento in modo da disambiguarli.]