****

****

CHANGE REQUEST

**Versione 1.0**

**TOP MANAGER:**

*Prof. Andrea De Lucia*

**PROJECT MANAGER:**

*Antonio Pizza*

**TOP MANAGER:**

|  |
| --- |
| **Nome** |
| Prof. De Lucia Andrea |

**PROJECT MANAGER:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Pizza Antonio | 05121 02367 |

**PARTECIPANTI:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Benincasa Mirko | 05121 03524 |
| De Stefano Manuel | 05121 03896 |
| Monaco Salvatore | 05121 03456 |
| Pangaro Luca | 05121 03846 |
| Soldà Stefano | 05121 03576 |
| Trerè Marialuisa | 05121 03770 |

**REVISION HISTORY:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 11/10/2017 | 1.0 | Stesura del documento | Antonio Pizza |

**Scopo del documento**

In questo capitolo saranno descritti gli obiettivi del processo di estensione del dominio applicativo ed aggiunta di funzionalità, in riferimento al documento di “Identificazione e classificazione delle modifiche richieste”.

Verrà studiato l’impatto di queste modifiche sul resto del sistema, analizzando il rapporto costo/benefici e i possibili rischi derivati dalla fase di progettazione.

Infine verrà fatto uno studio di affidabilità e verranno pianificate le fasi successive di progettazione, implementazione e testing.

**Panoramica del sistema attuale**

Il sistema attuale ha lo scopo di aiutare l’organizzazione, da parte di un bibliotecario, per la gestione degli inserimenti, eliminazione, prestito e consegna dei libri. Il sistema progettato funziona solo in un ambiente locale ed è rivolto soltanto al gestore della biblioteca.

**Attori e funzionalità**

Il sistema attuale prevede il seguente attore con le relative funzionalità:

1. Bibliotecario (non registrato )
   * Aggiungi libro
   * Rimuovi libro
   * Presta libro
   * Restituisci libro
   * Ricerca libro

**Design**

Essendo un sistema relativamente piccolo ed utilizzando file system per il salvataggio delle informazioni , l’implementazione è stata portata avanti prendendo in considerazione l’architettura Three-Tier ma non rispettandola del tutto. È stata fatta solo una netta divisione tra interface (tutti gli elementi che compongono l’interfaccia grafica) e application layer (i file e le classi che contengono la logica applicativa).

Il sistema è stato sviluppato utilizzando come linguaggio di programmazione Java.

**Implementazione**

L’implementazione del sistema risulta corrispondere, in parte, al pattern MVC dove è possibile indentificare tre tipi di oggetti, quali **Viev, Model e Control**.

**Testing**

Stando alla documentazione attuale, non risulta che il software sia stato testato con particolari Framework o software. Pertanto si prevedere di implementare classi di test delle componenti da riutilizzare utilizzando il framework JUnit e Katalon.

**Analisi delle modifiche richieste**

**Problematiche affrontate**

**Issue 1:** Continuare a conservare la stessa interfaccia utente?

|  |
| --- |
| Proposal 1.1 |
| Conservare l’interfaccia utente |
| Proposal 1.2 |
| Cambiare interfaccia utente con un web services |

|  |
| --- |
| Criterion 1.1 |
| Cambiare l’interfaccia rendendola user-friendly |

|  |
| --- |
| Argument 1.1 |
| Rimpiazzare l’interfaccia utente ha costi elevati e potrebbe comportare complicazioni ma comporterebbe un maggior utilizzo da parte degli utenti. In più lo staff ho uno skill più elevato per quanto riguarda la programmazione web |

|  |
| --- |
| Resolution 2.1 |
| L’unica soluzione è quella di riscrivere l’intera interfaccia utente |

**Issue 2:** Quali funzionalità conservare?

|  |
| --- |
| Proposal 2.1 |
| Conservare tutte le funzionalità |
| Proposal 2.2 |
| Aggiungere nuove funzionalità mantenendo, inoltre, quelle esistenti |

|  |
| --- |
| Criterion 2.1 |
| Scegliere le funzionalità che permettono il corretto funzionamento del sistema |

|  |
| --- |
| Argument 2.1 |
| Essendo che si passa da un sistema locale ad un web-based le vecchie funzionalità per l’amministratore della biblioteca devono essere conservate ma anche riscritte per essere adeguate al nuovo linguaggio di programmazione. Inoltre si è deciso di aggiungerne nuove per garantire un miglior utilizzo del sistema e permettendo l’utilizzo a più utenti |

|  |
| --- |
| Resolution 2.1 |
| Analizzando le argomentazioni Argument 2.1, l’issue 2 è stato risolto applicando la Proposal 2.2 |

**Issue 3:** Conservare tutti gli attori del sistema attuale?

|  |
| --- |
| Proposal 3.1 |
| Mantenere gli attori |
| Proposal 3.2 |
| Aggiungere nuovi attori quali Utente, Amministratore e Dipendente |

|  |
| --- |
| Criterion 3.1 |
| Preservare il funzionamento del sistema |

|  |
| --- |
| Argument 3.1 |
| Mantenendo gli stessi attori i requisiti funzionali non verrebbero alterati ma il sistema interesserebbe solamente un numero ristretto di utenti |
| Argument 3.2 |
| Aggiungendo nuovi attori, il target utente sarà fortemente ampliato, in quanto consentirebbe l’utilizzo della piattaforma anche ad utenti diversi da Bibliotecari. Inoltre verrebbe a crearsi una gerarchia rendendo più semplice la suddivisione dei ruoli con le rispettive funzionalità. |

|  |
| --- |
| Resolution 3.1 |
| Analizzando le argomentazioni Argument 3.1 e 3.2, l’issue 3 è stato risolto applicando la Proposal 3.2 |

**Soluzione individuata**

La soluzione individuata consiste nell’unione di tutte le “Resolution” ottenute nel paragrafo 4.1. In dettaglio essa dovrà avere le seguenti caratteristiche:

1. Riscrivere la User Interface rendendola responsive e compatibile con i vari browser e le varie piattaforme;
2. Apportare modiche alle funzionalità attuali ed aggiungere nuove funzionalità quali la prenotazione di libri da parte di un utente;
3. Aggiungere nuovi attori per ampliare il target utente e sfruttando a pieno le nuove funzionalità aggiunte

**Soluzioni alternative**

Le soluzioni alternative sono rappresentate da tutte le altre proposal affrontate nel paragrafo precedente.

**Identificazione dell’Impact Set**

Si svolgerà quindi un’attività di re-engineering per la modifica e la riorganizzazione di un sistema per renderlo più manutenibile. In particolare si svolgerà un’attività di source code translation, conversione di un codice da un linguaggio ad un altro quindi non verranno convolti artfatti del sistema attuale in quanto sarà del tutto re-implementato.

**Studio di affidabilità**

**Identificazione, descrizione e valutazione dei costi**

|  |  |  |
| --- | --- | --- |
| Identificazione | Valutazione | Motivazioni |
| Aggiunta nuovi requisiti funzionali | FORTE | Il numero dei requisiti da aggiungere risulta molto elevato, ma essenziale per l’idea del progetto da realizzare |
| Sostituzione interfaccia utente | FORTE | Si deve realizzare una nuova interfaccia per semplificare l’uso del sistema e renderla user-friendly |
| Aggiunta di nuovi attori | FORTE | L’aggiunta di nuovi attori è essenziale, in quanto si passa da un software gestionale ad una piattaforma online |
| Implementazione con linguaggi HTML, CSS, Servlet | FORTE | Tutto ho il team ha già lavorato con queste tecnologie, quindi non viene richiesta una fase di apprendimento. |
| Organizzazione di lavoro nel team di sviluppo | MEDIA | I componenti del team di sviluppo hanno in gran parte le stesse conoscenze quindi non ci sarà bisogno di organizzare il lavoro in base a queste |
| Testing funzionale | FORTE | Il sistema attuale non risulta testato, pertanto la fase di testing dovrà essere svolta e documentata |

**Identificazione, descrizione e classificazione dei benefici**

|  |  |  |
| --- | --- | --- |
| Identificazione | Valutazione | Motivazioni |
| Accesso multipiattaforma | FORTE | L’accesso alla piattaforma è indipendente dal dispositivo hardware e dal sistema operativo. L’utilizzo di bootstrap garantisce la corretta visualizzazione su ogni dispositivo |
| Estensione del dominio applicativo | FORTE | Essendo la piattaforma rivolta un numero maggiore di utenti, quest’ultimo dovrebbe avere una crescita notevole. |

**Identificazione, descrizione e classificazione dei rischi**

|  |  |  |
| --- | --- | --- |
| Identificazione | Valutazione | Motivazioni |
| Introduzione dei fault | FORTE | L’implementazione di un nuovo sistema comporterà nuovi fault. Quindi si prevede anche una fase di testing per correggerli |
| Malfunzionamento del server | MEDIA | La possibilità che accada è molto basse ma le conseguenze potrebbero essere gravi. |
| Attacchi informatici | MEDIA | Essendo la piattaforma rivolta un numero maggiore di utenti, quest’ultimo dovrebbe avere una crescita notevole. |
| Carenza di skill del team | DEBOLE | Tutto il team conosce le tecnologie utilizzate, quindi non sono previsti rallentamenti nella fase di implementazione |

**Risultati previsti**

La realizzazione del sistema software è definita secondo la soluzione individuata nel capitolo 4, tenendo in considerazione costi, benefici e rischi riportati nel capitolo 6.