



**Impact Analysis**

**Partecipanti**:

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Valenti Andrea | 0522500403 |
| Papaleo Giuseppe | 0522500465 |
| Tammaro Ruggero | 0522500512 |

|  |  |
| --- | --- |
| **Scritto da:** | Andrea Valenti, Giuseppe Papaleo |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 20/01/2018 | 0.2 | Modificati gli impatti sul RAD ereditato | Valenti Andrea |
|  |  |  |  |
|  |  |  |  |

# 1.Introduzione

Il documento di Impact Analysis descrive i ripple effects che l’intervento di manutenzione evolutiva genera sul sistema ereditato; gli artefatti afflitti dagli effetti collaterali su menzionati verranno individuati tramite *traceability* o tramite interazione umana. Dopo questa fase, verranno specificate modifiche e aggiunte al sistema esistente, individuando *SIS* (Starting Impact Set), *CIS* (Candidate Impact Set), *AIC* (Actual Impact Set), e gli eventuali *FPIS* (False Positive Impact Set) e *DIS* (Discovered Impact Set).

**Tecnologie necessarie**

Sono state utilizzate, per lo sviluppo di questo progetto, le seguenti tecnologie:

- Linguaggi di mark-up;

- DBMS;

- JSON;

- React JS;

- JSP

- Java Servlet;

- Maven;

-JDBC

# 2.Aggiunte rispetto al sistema precedente

2.1 AutoLogin

Abbiamo aggiunto una funzionalità di autologin che il sistema ereditato non prevedeva. Tale funzionalità risulta essere molto utile per gli utenti del sistema, poiché memorizza le credenziali dell’utente in un cookie e non richiede ogni volta all’utente di inserire le credenziali. Senza questa funzionalità l’utente abituale del sistema spreca tempo a compiere un azione tediosa, che facilmente può essere automatizzata sfruttando le potenzialità delle tecnologie web scelte per effettuare la manutenzione. L’utente al primo avvio dell’applicazione può abilitare tale funzionalità, facendo check in corrispondenza della chech-box “remember me”.

2.2 Classe per la persistenza

A differenza del sistema precedente, è stata aggiunta una classe separata che viene utilizzata nei vari punti dove è necessario utilizzare la persistenza dei dati. Nel vecchio sistema dopo una analisi da parte degli sviluppatori, è stata notata la presenza di funzioni verticalizzate sui dati che sono state replicate più volte in vari punti del codice.

3.Modifica rispetto al sistema precedente

3.1.1 Ristrutturazione delle classi dei ruoli

Ristrutturazzione classe Admin e User utilizzando una classe Astratta Utente. In questo modo sono state accorpate funzioni ridondanti presenti nelle suddette classi ereditate. Inoltre sono stati eliminati dei metodi verticalizzati sulla vecchia struttura.

La realizzazione della modifica non richiederà costi eccessivi

3.1.2 Redesign UI

Rispetto al sistema ereditato, l’interfaccia utente sarà modificata tramite l’utilizzo di linguaggi di mark-up e JSP, in modo tale da migrarlo sul web e utilizzare comunque parte della logica ereditata.

3.1.3 Redsign Model

A differenza del sistema precedente, la parte del model è stata modificata migrando dai file ad un sistema DBMS.

**Questa modifca ha impatto su tutta l’architettura del sistema, in quanto:**

* Al livello “Controller”, si deve sviluppare del codice che faccia da middleware verso il codice ereditato. Necessiteremo quindi di sviluppare metodi per ottenere l’accesso ai dati in maniera generica, e nelle diverse classi del sistema che prima accedevano ai file.
* Al livello “Model”, abbiamo dovuto riscrivere il codice per interagire con la nuova fonte dati e abbiamo dovuto implementare completamente da zero il database.

4. Impact analysis, tracciabilità dei documenti

Il nostro sistema avrà un grosso impatto sui documenti ereditati dal sistema precedente. Precisiamo che non c’è stato bisogno di una re-engineering dei documenti in quanto la stesura era concorde agli standard imposti dall’IEEE, solo in alcuni casi è stato necessario ampliare delle sezioni poco chiare.

4.1 RAD

1) Requisiti funzionali:

- RF-1.3 Autologin: funzionalità molto comoda che può essere sfruttata dal Cliente per evitare di dover inserire ad ogni accesso le sue credenziali

2) Requisiti non funzionali:

-Packaging : Verrà registrato un dominio sulla rete internet sulla quale verrà installato il sistema.

-Affidabilità: Il sistema sarà affidabile e accessibile in qualsiasi momento, salvo problemi relativi alla caduta del server. Il sistema deve rispondere correttamente alle richieste impartite dall’ utente.

- Flessibilità: l'ampia gamma delle funzionalità, unite a un'interfaccia grafica intuitiva e aggiornata, consentono di soddisfare le esigenze gestionali richieste.

- Prestazioni: Le operazioni messe a disposizione dal sistema interagiscono con un database. Abbiamo bisogno di un server che funga da servlet container e che ospiti il DB. Il server è stato pensato per garantire tempi di risposta adeguati alle richieste effettuate dall’utente.

-Operatività: Il sistema offre un’interfaccia user-friendly, in modo che il suo funzionamento sia comprensibile sin dal primo utilizzo. L’ UI è basata su una pagina del browser principale che comprende un menù dal quale è possibile accedere alle altre pagine che mostreranno le altre funzionalità.

- Implementazione (modificato) : Il lato UI verrà implementato tramite React JS e JSP. React JS è un framework JavaScript-based utilizzato per facilitare le operazioni di sviluppo di UI dinamiche. Mentre la tecnologia JSP combinata con React, da un lato permette di creare pagine web accattivanti, e dall’altro lato permette di sfruttare molto delle classi ereditate poiché si interfaccia con JAVA, la tecnologia con cui era scritto Prezzi Pazzi. Il server sarà sviluppato creando un set di Servlet (JavaEE) che si interfacceranno con le classi Java esistenti e le pagine web tramite le JSP. Il DB verrà prima progettato e poi implementato, visto che nel vecchio sistema per la parte di model venivano utilizzati i file. Per una gestione efficiente del nuovo envoirorment verrà introdotto Maven.

4.2 DD

Dal sistema precedente, noi abbiamo ereditato i documenti di design (precisamente SDD e ODD). Il nostro documento di desygn, ovviamente, ha un forte impatto con questi poichè:

- Il disegno di sistema ereditato è stato totalmente modificato e proprio per questo motivo, la struttura ereditata è stata adattata ai nuovi design goal.

- Per il fatto che il nostro progetto prevede una migrazione del sistema precedente verso il web, si necessita una creazione di uno strato intermedio (middleware) che faccia da tramite tra il nostro sistema e quello precedente. E questo ha un forte impatto su i documenti precedenti suddetti.