



**RAD**

**Requirement Analysis Document**

**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 | Aggiunta modifiche casi d’uso | Andrea Valenti |
| 21/02/2018 | 0.3 | Modifiche alla descrizione UI | Andrea Valenti |
|  |  |  |  |

Sommario

[1. Introduzione 4](#_Toc504514599)

[1.1 Definizioni, acronimi e abbreviazioni 4](#_Toc504514600)

[2. Sistema corrente 4](#_Toc504514601)

[3. Sistema proposto 5](#_Toc504514602)

[4. Modifiche al RAD 6](#_Toc504514603)

[4.1 Requisiti Funzionali (RAD 3.2) 6](#_Toc504514604)

[4.2 Requisiti non-funzionali (RAD 3.3) 6](#_Toc504514605)

[4.3 Modelli di Sistema (RAD 3.4) 7](#_Toc504514606)

[Identificazioni degli Scenari(RAD 3.4.1) 7](#_Toc504514607)

[Casi d’uso(RAD 3.4.2) 8](#_Toc504514608)

# 

# 1. Introduzione

Il seguente documento si propone di descrivere le funzionalità del sistema che si andrà a realizzare; più precisamente, il suddetto documento è allegato al Requirement Analysis Document esistente, nel quale si integrano/modificano le sezioni che saranno impattate dal processo di migrazione.

## 1.1 Definizioni, acronimi e abbreviazioni

Di seguito, verranno descritti gli acronimi e le abbreviazioni utilizzati in questo documento.

|  |  |
| --- | --- |
| **Acronimo** | **Definizione** |
| DBMS | Database Management System |
| DB | Base di Dati |
| UI | User Interface |
| API | Application Programming Interface |
| JavaEE | Java Enterprise Edition |
| URL | Uniform Resource Locator |

**Tabella 1.1.1 Definizione, acronimi e avvreviazioni**

# 2. Sistema corrente

Prezzi pazzi è un green-field engeneering project, ovvero un sistema che non fa riferimento a nessun altro sistema esistente. Il suddetto sistema si occupa della gestione di catalogo di offerte, dando la possibilità di acquistare i prodotti da parte di clienti. Il sistema si basa sulla gestione di un catalogo con il quale si può interagire attraverso UI/Funzioni progettate per dare supporto alla varie funzioni che offre il sistema.

Esistono vari file che contengono :

* Dati relativi ai clienti per gestire gli ordini
* Dati relativi alle offerte per gestire quantità e prezzi;
* Dati relativi ai pagamenti effettuati dai clienti
* Dati relativi agli Admin

# 3. Sistema proposto

Il sistema che viene proposto ha l’obiettivo di creare una versione web-based di PrezziPazzi, andando a riprenderne tutte le funzionalità del cliente, migliorandole ove necessario, al fine di rendere il sistema stesso di più facile utilizzo. La migrazione verso il web di questo sistema risponde alla necessità di poter utilizzare PrezziPazzi da qualsiasi luogo e da qualsiasi terminale, previa connessione ad Internet. Trattandosi quindi di una migrazione verso il web, PrezziPazzi sarà accessibile attraverso l’indirizzo http://localhost:8080/PrezziPazzi/index.html; nella homepage, sarà possibile effettuare l’accesso al sistema attraverso l’inserimento di username e password. Nel progetto ereditato la consistenza dei dati veniva garantita dall’utilizzo di file a supporto dell’applicativo JAVA, e per adattare quindi questo strato di model al nuovo ambiente richiesto è stata effettuata una migrazione vero un DBMS. Ciò ha portato quindi alla completa sostituzione di uno strato dell’architettura, con la conseguenza di dover scrivere del codice ex-novo. Prima di tutto la parte di chiamate al DBMS sono state inserite in una classe, mentre prima non esisteva una modularità così netta nel progetto.

L’UI del sistema migrato verrà progettata tramite un framework JavaScript-based React JS, per facilitare le operazioni di design, e la tecnologia JSP. Quest’ultima tecnologia permette di utilizzare quasi tutte le entità del progetto ereditato, e si accoppia molto bene con lo strato di servlet che verrà usato per creare il back-end. Nulla della vecchia UI verrà mantenuto, e visto che il codice ereditato ha creato una netta separazione tra logica applicativa e UI, tale operazione è risultata poco costosa. Quindi la web application lavorerà sul DB fornito, interfacciandosi con la vecchia architettura di classi attraverso l’utilizzo delle API di JavaEE (servlet).

# 

# 4. Modifiche al RAD

In questa sezione, vengono presentate le modifiche e le integrazioni relative ai paragrafi del RAD originario di PrezziPazzi individuate nell’ambito del processo di migrazione.

Per ogni elemento modificato, è indicato il numero della sezione nella quale è presente all’interno del RAD esistente; dove non indicato, si considera la sezione come aggiunta al documento originario.

## 4.1 Requisiti Funzionali (RAD 3.2)

Il Sistema migrerà la parte del Cliente e non tutti gli attori individuati in accordo con la CR proposta.

Attore: Admin,Cliente

* RF1.3 - Autologin: funzionalità che permette di accedere al sistema senza dover inserire ogni volta le credenziali.

***Avvio dell’applicazione***

L’utente lancia l’applicazione inserendo nella barra degli indirizzi del proprio browser l’URL http://localhost:8080/PrezziPazzi/index.html; a questo punto, inserendo username e password, avrà accesso alla home page da cui può visualizzare il catalogo e la propria area utente.

***Uscita dall’applicazione***

L’utente effettuerà l’uscita dall’applicativo chiudendo la finestra del browser in cui PrezziPazzi era in esecuzione. Oppure eseguirà l’operazione si logout.

## 4.2 Requisiti non-funzionali (RAD 3.3)

***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 verticale dal quale è possibile accedere alle altre pagine che mostreranno le altre funzionalità.

***Packaging:***

Il sistema verrà reso fruibile sul sito internet http://localhost:8080/PrezziPazzi/index.html.

***Implementazione:***

Il lato UI verrà implementato tramite React, un framework JavaScript-based utilizzato per facilitare le operazioni di sviluppo. Il server sarà sviluppato creando un set di Servlet (JavaEE) che si interfacceranno con le classi Java esistenti. 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.

**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.

## 4.3 Modelli di Sistema (RAD 3.4)

Di seguito vengono riportate le modifiche ai diagrammi ereditati dal progetto.

## Identificazioni degli Scenari(RAD 3.4.1)

|  |  |
| --- | --- |
| **Scenario** | **Autologin** |
| **Attore** | **Ruggero: Cliente** |
| **Flusso degli eventi** | 1. **Ruggero accede al sistema inserendo le proprie credenziali, e selezionando la check box “Remember Me”;** 2. **Il sistema risponde presentando sulla pagina le azioni che Ruggero potrà compiere** 3. **Ruggero dopo aver concluso i suoi acquisti, spegne la sua macchina** 4. **Ruggero ritorna sul sitema digitando l’indirizzo nel broswer** 5. **Il sistema risponde automaticamente mostrandogli la home page, senza che Ruggero inserisca le credenziali** |

## Casi d’uso(RAD 3.4.2)

|  |  |
| --- | --- |
| **UC\_ID** | **AUT\_LOG1** |
| **Nome :** | **Autologin;** |
| **Attori :** | **Amministratore del Sistema,Cliente** |
| **Condizione d’ Entrata:** | **L’ Amministratore/Cliente vuole entrare nel sistema senza dover inserire ogni volta le credenziali** |
| **Flusso degli eventi :** | 1. **L’ Amministratore/Cliente accede al sistema con le proprie credenziali.** 2. **Prima di premere su login l’ Amministratore/Cliente, seleziona la voce “remember me”.** 3. **Dopo aver svolto i propri compiti l’ Amministratore/Cliente chiude il broswer.** 4. **L’ Amministratore/Cliente digita l’indirizzo del sistema nel broswer;** 5. **Il sistema mostra automaticamente la home page dedicata all’ Amministratore/Cliente, senza che egli digiti le proprie credenziali.** |
| **Condizioni d’ uscita :** | **L’utente accede al sistema senza dover inserire le credenziali ad ogni accesso.** |
| **Flussi alternativi :** | * **Mancata selezione della check box “remmber me”.** * **Continua al punto 1.** |