UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea in Informatica Magistrale

Ingegneria, gestione ed evoluzione del software

M2fas_management

Report

Versione 1.0

Docente

Prof.ssa Andrea De Lucia

Studente

Maria Cristina Ricciardi
Matr.: 0522500406

Anno Accademico 2022/2023

Cronologia revisioni

Data	Versione	Descrizione	Autore
24/06/2023	0.1	Prima stesura	M.C.Ricciardi
	0.2		M.C.Ricciardi

Sommario

1. Overview del sistema esistente

M2fas management è un software per la gestione di un magazzino per un negozio di abbigliamento. E' stato progettato per essere utilizzato da due attori principali: l'amministratore del sistema e un operatore addetto alla cassa. Utilizza un database centralizzato per contenere dati persistenti che tengono traccia dei prodotti, fornitori e bilancio del negozio.

La figura dell'amministratore permette di garantire la sicurezza dei dati in quanto è l'unico in grado di accedere a dati sensibili e fondamentali per la gestione del magazzino, perché permette l'accesso esclusivo al magazzino, dove gli permette di aggiungere, modificare ed eliminare i prodotti presenti nel database. Ha la responsabilità di gestire i fornitori e relativi pagamenti. Supervisiona il bilancio generale, può visionare gli andamenti finanziari tenendo conto del periodo di riferimento e visualizzando le spese sostenute, i pagamenti pendenti e le entrate giornaliere.

L'operatore invece tratta la merce acquistata, ha la facoltà di inserire i dati dei prodotti, collabora all'aggiornamento dei dati in magazzino, come la mancanza di determinati prodotti, le entrate giornaliere e quindi in sostanza fa riferimento alla gestione cassa.

Il sistema è stato implementato in Java 2 con l'uso di un database MySQL. Dai documenti forniti insieme al codice sorgente, il software funziona correttamente su sistemi con supporto alla JVM, con interfaccia grafica diversa per l'operatore e l'amministratore.

L'architettura scelta del sistema esistente è di tipo repository con tre strati indipendenti:

- Interface layer

  Contiene le componenti delle interfacce grafiche che gli attori usano per poter accedere ai dati

- Logic layer

  Sottosistema delle componenti di logica e delle entità del sistema

- Repository layer

  Sistema di gestione dei dati, per il recupero e aggiornamento dei dati nel database

1.1 Attori e funzionalità

Riepilogando il sistema prevede due attori principali con le seguenti funzionalità:

- Amministratore
  - Gestione Fornitori
  - Gestione Magazzino
  - Bilancio
- Operatore
  - Gestione cassa

2. Testing iniziale del sistema

Nella documentazione del sistema iniziale non sono presenti files relativi al testing effettuato su tale sistema; quindi, è stato necessario creare un test case ex novo.

Nel documento Piano di test, file **m2fas_test_plan_v.0.2,** viene spiegato l'approccio utilizzato per la creazione dei test e gli strumenti utilizzati. I test case sono stati sviluppati utilizzando il metodo del **Category Partition**. Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Infine, a ogni scelta verrà assegnato un valore.

Per ogni test case, viene indicato una tabella che specifica tutti i parametri necessari.

Inoltre, si è deciso di presentare in una lista le classi da testare che sono ritenute fondamentali per il corretto funzionamento dell'intero sistema, e riportiamo qui per chiarezza:

La lista delle classi scelte che sono state testate:

- ProductManagement.java
- SuppliesManagement.java
- ExitsManagement.java
- EntrancesManagement.java
- AdminAccess.java

Sono classi che implementano le diverse funzionalità, considerate essenziali del sistema m2fas management:

- Gestione prodotto

Permette all'amministratore di poter aggiornare, inserire, eliminare e visualizzare i prodotti disponibili nel catalogo del negozio.

- Gestione bilancio

Si tiene traccia delle spese in entrata e uscita

- Gestione spese

Inserimento e eliminazione delle spese relative al negozio e ai fornitori.

2.2 Report

Nel documento Test Case Specification, file **m2fas_TCS_v.0.2,** sono indicati tutti i test effettuati al sistema iniziale. Di seguito si evidenziano gli errori riscontrati dai test:

1. ProductManagement
   a. Non viene segnalato nessun messaggio di errore nonostante il valore dell'ID non è presente nel database.
   b. Non viene segnalato nessun messaggio di errore quando si inserisce un prodotto già presente nel database.
   c. Non viene segnalato nessun messaggio di errore quando si elimina un prodotto non presente nel database.
   d. Non viene segnalato nessun messaggio di errore quando si inserisce un'ID come stringa, invece di numero.
2. SuppliesManagement
   a. Errore nel database di duplicazione di chiave primaria, quando si prova a inserire un fornitore con una P.IVA già presente, non viene segnalato nessun messaggio di errore.
   b. Error. Data truncated for column 'pagato' at row 1. Errore che avviene se si aggiorna solo il campo Debito con P.IVA già presente nel database.
   c. Error. Data truncated for column 'debito at row 1. Errore che avviene se si aggiorna solo il campo Pagato con P.IVA già presente nel database.
3. ExitsManagement
   a. Error. Data truncated for column 'importo' at row 1. Non viene visualizzato Nessun messaggio di errore se viene inserito un'importo che non è un numero.
   b. Non viene segnalato nessun messaggio di errore se si inserisce un formato di data non idoneo, come mese = 33
   c. Error. Incorrect integer value: 'ok' for column 'conto' at row 1
4. EntrancesManagement
   a. Non viene segnalato nessun messaggio di errore se si inserisce un formato di data non idoneo, come mese = 33

3. Change request

3.1 Descrizione

Si richiede la migrazione del sistema esistente, inizialmente concepito per girare su terminali interni all'azienda, sul web.

3.2 Scopo

Lo scopo della change request è migrare l'applicazione esistente, implementata in JAVA, in una web app con tecnologia JSP e Servlet, riutilizzando le classi che accedono ai dati del database MySql della vecchia applicazione. I vantaggi della migrazione darà la possibilità a utenti di tutto il mondo, che non possono fisicamente visitare il negozio, di acquistare prodotti dell'azienda e avrà più libertà di scelta degli articoli. L'azienda che utilizzerà il nuovo sistema avrà maggiori profitti dagli ordini ricevuti attraverso il web con minori costi di implementazione del nuovo sistema, in quanto sarà riutilizzata tutta la logica applicativa del sistema esistente.

4. Analisi e progettazione delle change request

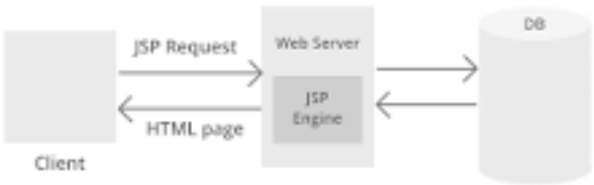
La manutenzione che si intende effettuare sul sistema è di adattare tale sistema sul web, riutilizzando i componenti core e la logica applicativa già progettati, al nuovo modulo web. Inoltre, l'intenzione è di connettere il nuovo sistema alla database attuale. La componente grafica o interfaccia grafica verrà totalmente riprogrammata in base alla tecnologia utilizzata per il web, conservando il look and feel invariato.

Le funzionalità presenti nel sistema esistenti saranno presenti nel nuovo sistema con l'aggiunta di nuovi componenti quali la sezione di Login, Registrazione utente e Shop dove l'utente visualizza i prodotti del negozio e li acquista nel carrello. L'aggiunta della componente Registrazione, darà più libertà all'utente di registrarsi al sito web cosicché visualizzare tutti i prodotti del negozio.

4.1 Architettura

L'architettura software del nuovo sistema è 3-tier, in quanto si è scelto di utilizzare come tecnologie JSP e le Servlet. L'architettura JSP ha un Client, un Web Server e un Database. Il Client è un web browser utilizzato dall'utente per visitare il sito web. Il Web Server usa JSP Engine, come un container che processa JSP.

Il Web Server utilizzato è Apache Tomcat Server che ha un motore JSP integrato. JSP engine intercetta la richiesta JSP e fornisce l'ambiente per la comprensione e l'elaborazione dei file JSP. Legge, crea e analizza JAVA Servlet, compila ed esegue codice JAVA, e restituisce la pagina HTML. Il server web ha accesso al Database.



Questo tipo di architettura ci permette di avere livelli di presentazione e logica applicativa ben separati, separazione che ritroviamo anche nel sistema esistente.

L'utilizzo di un Web Server Apache è la soluzione ideale per contenere i costi di sviluppo, in quanto è open-source e gratuita e fornisce una piattaforma per l'esecuzione di applicazioni scritte in JAVA.

4.2 Database

Si adotta lo stesso database utilizzato dal sistema esistente, quindi il nuovo sistema si interfaccerà al DBMS MySQL Server, anch'esso open-source e gratuito.

4.3 Linguaggi di programmazione

Per il lato Client si utilizzano HTML, CSS e JavaScript per l'implementazione della presentazione e grafica del sistema web, invece per quanto riguarda il lato Server, l'implementazione richiede l'utilizzo di JAVA, JSP e Servlet Java. Si è deciso di utilizzare JAVA per riutilizzare le classi del vecchio sistema, anch'esso scritto in JAVA.

Il lato client, utilizzando tali tecnologie, avrà la possibilità di essere utilizzato in qualsiasi piattaforma web e funzionerà su ogni browser web compatibile.

5. Testing post modifiche

5.1 Test comuni ad entrambi i sistemi desktop e web