

UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica



Corso di Laurea in Informatica Magistrale

***Ingegneria, gestione ed evoluzione
del software***

M2fas_management

Test Plan

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
15/03/2023	0.1	Prima stesura	M.C.Ricciardi

Sommario

1. Introduzione.....	4
2. Relazione con altri documenti	4
3. System overview	4
4. Funzionalità da testare e non	4
5. Pass/Fail criteria.....	4
6. Approccio	4
7. Specifica Test cases	5

1. Introduzione

Questo documento ha lo scopo di pianificare lo sviluppo e le attività di testing del sistema m2fas_management.

2. Relazione con altri documenti

Per verificare il corretto funzionamento del sistema, abbiamo ricavato le sue funzionalità dai documenti, quali il RAD – Requirement analysis document, dove sono specificati i requisiti e gli scenari del software, in modo da poter avere una base di partenza per la creazione dei test cases.

Facciamo riferimento ai documenti creati durante la fase di sviluppo, quali:

- RAD_m2fas
Comprende scenari e analisi dei requisiti
- SDD_m2fas
Componenti del sistema

3. System overview

Il sistema è stato sviluppato allo scopo di gestire un magazzino per qualsiasi tipo di negozio (abbigliamento, scarpe sportive, etc.). Permette all'amministratore del sistema, di organizzare e tenere traccia dei prodotti, le spese di bilancio, le entrate, i fornitori e la cassa.

4. Funzionalità da testare e non

Durante la fase di testing sono state prese in considerazione diverse funzionalità, considerate essenziali del sistema m2fas:

- 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.

5. Pass/Fail criteria

Il testing ha **successo** se l'output osservato è diverso dall'output atteso: questo significa che ha individuato una failure. In tal caso, questa verrà analizzata, e se legata a una fault, si procederà alla correzione.

I test sono iterati per verificare che la modifica effettuata non impatti su altri componenti del sistema.

Il criterio di **fallimento** invece avviene quando il test non riesce ad individuare un errore, ovvero l'output osservato sarà uguale all'oracolo.

6. Approccio

Il testing ha un approccio di tipo "black box". E' un approccio che permette di non analizzare in dettaglio il codice, ma ha il compito di testare le funzionalità in base ai dati in ingresso e in uscita.

L'approccio prevede l'esecuzione di **test di unità**, che consiste di effettuare test sui singoli metodi e classi del sistema, cercando possibili errori. Viene utilizzato il framework **JUnit** dove, ad ogni classe testata, viene creata la relativa classe di test.

La lista delle classi scelte per essere testate:

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

Per sviluppare i test cases è stato utilizzato 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.

Si adotterà una strategia Bottom-up per i **testing d'integrazione**, ovvero si prevede di integrare passo dopo passo ogni sottosistemi dal layer più basso nella scala gerarchica.

Il **testing di sistema** ci aiuterà a controllare la correttezza dell'intero sistema, in quanto cercherà errori di tutti i sottosistemi. Sarà utilizzato il framework Selenium che permetto un pieno controllo sui sistemi web-based.

I test case sono definiti nel documento Test Cases Specification – m2fas_TCS.

7. Specifica Test cases

Product Management - Inserimento Prodotto

Parametro -> ID	
Categorie	Scelte
Lunghezza LID	1: lunghezza <= 4 [errore] 2: lunghezza 5-10 [property lunghezzaLID ok] 3: lunghezza > 10 [errore]
Formato FID	1: rispetta il formato [if property lunghezzaLID ok] [property formatoFID ok] 2: non rispetta il formato [if property lunghezzaLID ok] [errore]
Valore VID	1: valore <=0 [if lunghezzaLID ok AND formatoFID ok] [errore] 2: valore 00001-9999999999 [if lunghezzaLID ok AND formatoFID ok] [property valoreVID ok] 3: valore > 9999999999 [if lunghezzaLID ok AND formatoFID ok] [errore]
Esiste EID	1: esiste se [if formatoFID ok] [property esisteEID ok] 2: non esiste se [if lunghezzaLID ok AND formatoFID ok] [errore]
Parametro -> marca	
formato: [A-Za-z0-9]	

Categorie	Scelte
Lunghezza LM	1: lunghezza < 2 [errore] 2: lunghezza >= 2 [property lunghezzaLM ok]
Formato FM	1: rispetta il formato [if property lunghezzaLM ok] [property formatoFM ok] 2: non rispetta il formato [if property lunghezzaLM ok] [errore]
Parametro -> modello formato: [A-Za-z0-9]	
Categorie	Scelte
Lunghezza LMOD	1: lunghezza < 2[errore] 2: lunghezza >= 2 [property lunghezzaLMOD ok]
Formato FMOD	1: rispetta il formato [if property lunghezzaLMOD ok] [property formatoFMOD ok] 2: non rispetta il formato [if property lunghezzaLMOD ok] [errore]
Parametro -> categoria formato: [A-Za-z0-9]	
Categorie	Scelte
Lunghezza LCAT	1: lunghezza < 2 [errore] 2: lunghezza >= 2 [property lunghezzaLCAT ok]
Formato FCAT	1: rispetta il formato [if property lunghezzaLCAT ok] [property formatoFCAT ok] 2: non rispetta il formato [if property lunghezzaLCAT ok] [errore]
Parametro -> prezzo	
Categorie	Scelte
Lunghezza LPR	1: lunghezza == 0 [errore] 2: lunghezza >= 1 [property lunghezzaLPR ok]
Formato FPR	1: rispetta il formato [if property lunghezzaLPR ok] [property formatoFPR ok] 2: non rispetta il formato [if property lunghezzaLPR ok] [errore]
Valore VPR	1: valore <=0 [if lunghezzaLPR ok AND formatoFPR ok] [errore]

	2: valore 1-9999999999 [if lunghezzaLPR ok AND formatoFPR ok] [property valoreVPR ok]
Parametro -> quantità	
Categorie	Scelte
Lunghezza LQ	1: lunghezza == 0 [errore] 2: lunghezza >= 1 [property lunghezzaLQok]
Formato FQ	1: rispetta il formato [if property lunghezzaLQ ok] [property formatoFQ ok] 2: non rispetta il formato [if property lunghezzaLQ ok] [errore]
Valore VQ	1: valore <=0 [if lunghezzaLQ ok AND formatoFQ ok] [errore] 2: valore 1-9999999999 [if lunghezzaLQ ok AND formatoFQ ok] [property valoreVQ ok]

ExitsManagement

Parametro -> day	
Categorie	Scelte
Lunghezza LD	1: lunghezza == 0 [errore] 2: lunghezza 1-2 [property lunghezzaLDok] 3: lunghezza > 2 [errore]
Formato FD	1: rispetta il formato [if property lunghezzaLDok] [property formatoFD ok] 2: non rispetta il formato [if property lunghezzaLD ok] [errore]
Valore VD	1: valore <=0 [if lunghezzaLD ok AND formatoFD ok] [errore] 2: valore 1-31 [if lunghezzaLD ok AND formatoFD ok] [property valoreVD ok] 3: valore > 31 [if lunghezzaLD ok AND formatoFD ok] [errore]
Parametro -> month	
Categorie	Scelte
Lunghezza LD	1: lunghezza == 0 [errore] 2: lunghezza 1-2 [property lunghezzaLDok]

	3: lunghezza > 2 [errore]
Formato FD	1: rispetta il formato [if property lunghezzaLDok] [property formatoFD ok] 2: non rispetta il formato [if property lunghezzaLD ok] [errore]
Valore VD	1: valore <=0 [if lunghezzaLD ok AND formatoFD ok] [errore] 2: valore 1-12 [if lunghezzaLD ok AND formatoFD ok] [property valoreVD ok] 3: valore > 12 [if lunghezzaLD ok AND formatoFD ok] [errore]