Test Plan

E-Balance

Riferimento	E-Balance_C17_TP
Versione	1.0
Data	11/12/2023
Destinatario	F. Ferrucci, F. Palomba
Presentato da	Simone Cirma, Mario De Luca, Antonio Di Giorgio, Donato Folgieri, Daniela Palma, Emanuele Vitale
Approvato da	Matteo Ercolino, Simone Silvestri

Revision History

Data	Versione	Descrizione	Autori
11/12/2023	1.0	Stesura del documento	Antonio Di Giorgio, Daniela Palma

Sommario

Revision History 2
1. Introduzione 4
2. Relazione con altri documenti 4
3. Panoramica del sistema 4
4. Features da testare/da non testare 5
5. Pass/Fail criteria 5
6. Approccio 5
6.01 Testing di Sistema 6
Functional testing 6
Performance Testing 6
Pilot Testing 6
Acceptance Testing 6
Installation Testing 6
6.02 Testing di integrazione 6
6.03 Testing di Unità 6
6.04 Ispezione del codice 6
7. Sospensione e ripristino 7
7.01 Criteri di sospensione 7
7.02 Criteri di ripristino 7
8. Materiale di testing 7
9. Test cases 7
9.01 Gestione Amministratore 7
9.01.01 Registrazione nuovo amministratore 7
9.01.02 Genera nuovo report 8
9.02 Gestione Contratto 9
9.02.01 Aggiungi nuovo contratto 9
9.02.02 Modifica un contratto 10
9.03 Gestione Accesso 11
9.03.01 Login 11
9.04 Gestione IA 11
9.04.01 Modifica Parametri IA 11
10. Testing schedule 11

1. Introduzione

E-Balance si propone di migliorare la gestione energetica dell'Università degli Studi di Salerno, creando un sistema in grado di monitorare e ottimizzare l'utilizzo delle risorse energetiche con l'aiuto dell'intelligenza artificiale.
Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing per il software di E-Balance. Il fine è quello di garantire che ogni aspetto funzioni in modo corretto.
All'interno del documento sono riportate le strategie di testing adottate, quali funzionalità saranno testate e gli strumenti scelti per la rilevazione degli errori, con lo scopo di presentare un sistema privo di malfunzionamenti. Sono state pianificate attività di testing per le seguenti gestioni:

- Gestione Amministratore
- Gestione Contratto
- Gestione Accesso
- Gestione IA

2. Relazione con altri documenti

Per la corretta individuazione dei test case, si fa riferimento ad altri documenti prodotti:

- Relazioni con il Requirements Analysis Document (RAD): I test case pianificati nel Test Plan sono elaborati in relazione ai requisiti funzionali e non funzionali presentati nel RAD.
- Relazioni con il System Design Document (SDD): I test case pianificati nel Test Plan devono rispettare la suddivisione in sottosistemi presentata nell'SDD.

3. Panoramica del sistema

Il sistema E-Balance è realizzato secondo l'architettura MVC, nel dettaglio:

- per la gestione del database saranno usati:
  - MySQL
  - JDBC
- per la parte di front-end e la generazione delle view usiamo:
  - JSP
  - HTML
  - CSS
  - jQuery
  - JavaScript
- per la parte di back-end usiamo:
  - Apache Tomcat
  - Java

4. Features da testare/da non testare

Di seguito la lista delle features di cui si effettuerà il testing per le varie gestioni:

- Gestione Amministratore
  - Aggiungi nuovo amministratore
  - Genera un nuovo report
- Gestione Contratto
  - Aggiungi un nuovo contratto
  - Modifica un contratto
- Gestione Accesso
  - Login
- Gestione IA
  - Modifica parametri IA

Le funzionalità di cui non si andrà ad effettuare le attività di testing riguardano requisiti funzionali di bassa o media priorità; sono inoltre escluse le funzionalità che non prevedono input manuale da parte dell'amministratore, ad esempio attività riguardanti esclusivamente visualizzazioni di dati.

5. Pass/Fail criteria

Il testing è finalizzato a rilevare la presenza di difetti all'interno del sistema, al fine di intervenire successivamente per eliminarli.
L'esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso della sua esecuzione, basandosi sui requisiti.
Un test ha successo (pass) se, dato un input al sistema, l'output ottenuto è diverso dall'output atteso dall'oracolo.
Un test fallisce (fail) se, dato un input al sistema, l'output ottenuto è uguale all'output atteso dall'oracolo.
Tutto il testing sarà considerato valido se tutti i seguenti vincoli saranno rispettati:

- Testare tutti i requisiti funzionali ad alta priorità;
- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;
- Raggiungere un branch coverage non inferiore al 50%.

6. Approccio

Il testing dell'intero sistema si compone di tre fasi: testing di sistema, testing di integrazione e testing di unità.
Verranno progettati nell'ordine appena definito, ma verranno eseguiti in ordine inverso.
Prima della fase di implementazione del sistema, avverrà la progettazione dei casi di test di sistema, perfezionati in seguito nella loro fase di esecuzione; durante la fase implementativa avverrà la progettazione dei casi di test di unità.
Durante lo sviluppo saranno eseguite periodiche attività di revisione sul codice prodotto.
Poiché la progettazione è organizzata seguendo un modello simile al modello a V, il testing di sistema è stato pianificato in seguito alla stesura del documento Requirements Analysis Document, mentre la pianificazione del testing di integrazione avverrà dopo la stesura del Object Design Document.

6.01 Testing di Sistema

Per il testing di sistema sarà utilizzato il tool "Selenium IDE", che permette di registrare le azioni che un utente può intraprendere sul browser, in modo da poter implementare ed eseguire i test case di sistema. Il server, per la fase di testing, verrà deployato in localhost.

Functional testing

Il functional testing ha il fine di validare i requisiti funzionali. Consiste nell'individuare i possibili faults generati dagli input degli utenti.

Performance Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione del performance testing.

Pilot Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione del pilot testing.

Acceptance Testing

L'acceptance testing verrà effettuato solo sul functional testing, ed il Project Manager simulerà la figura del cliente.

Installation Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione dell'installation testing.

6.02 Testing di integrazione

Verrà utilizzato un approccio bottom-up, metodo ritenuto più adatto per un software basato sul paradigma Object Oriented. La definizione dei test case avverrà tramite il framework JUnit, mentre verrà usato Mockito per il mocking. Verrà utilizzato Travis CI. L'automatizzazione del run dei test sarà gestita da Maven, ed infine come tool di misurazione e report coverage sarà utilizzato JaCoCo. Il test di integrazione sarà il medesimo per tutte le componenti da testare. Nello specifico, si procederà prima con il test delle classi Service, e successivamente con il test delle classi Controller. Durante questa seconda esecuzione, la chiamata al controller sarà mockata usando Mockito. Per ciò che concerne le dipendenze tra i sottosistemi, si riporta di seguito il diagramma architetturale.

6.03 Testing di Unità

Per il testing di unità la strategia prevista consiste nel testare ogni metodo delle classi del sistema. Da esse, sono escluse le interfacce e le classi entity, poiché quest'ultime presentano solo metodi getters e setters. I casi di test saranno definiti attraverso un approccio black-box e saranno documentati direttamente nel codice, attraverso l'uso del framework per il testing di classi Java JUnit. Per ogni Production Class sarà definita una Test Class che rispetterà il formato NomeProductionClassTest. Tali classi saranno scritte prima della stesura delle Production class, per eliminare effetti BIAS dovuti alla conoscenza del codice. Le stesse classi saranno poi revisionate e modificate da sviluppatori differenti.
Altre tecnologie usate in tale fase saranno:

- Mockito: per la costruzione degli stub e l'isolamento della componente testata.
- JaCoCo: per il calcolo di metriche tra le quali la Branch Coverage.
- Maven: per la build e l'esecuzione automatica dei tests.

6.04 Ispezione del codice

Per aumentare la qualità del codice ci si affiderà principalmente ai controlli del tool Checkstyle e della CI/CD con Travis. Sebbene vi sia l'intenzione di fare ispezione del codice, anche se non costante e approfondita, a causa del basso budget non si può assicurare tale pratica.

7. Sospensione e ripristino

In questa sezione verranno specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

7.01 Criteri di sospensione

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. Il testing potrà essere momentaneamente sospeso nel caso venga restituito, al momento dell'esecuzione, un errore nella definizione di uno dei test stessi.

7.02 Criteri di ripristino

Il testing verrà ripreso dopo aver risolto i fault individuati.

8. Materiale di testing

L'hardware necessario per l'attività di test è un semplice computer.

9. Test cases

L'approccio per la definizione dei test frame sarà il category partition. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza. Per definire l'output atteso si userà un oracolo umano, per via dell'assenza di specifiche formali/semi-formali.

9.01 Gestione Amministratore

9.01.01 Registrazione nuovo amministratore

Parametro: Nome	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LN]	1. Lunghezza > 255 AND Lunghezza = 0 = false[error] 2. Lunghezza <= 255 AND Lunghezza > 0 = true [PROPERTY LN_OK]
Parametro: Cognome	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LC]	1. Lunghezza > 255 AND Lunghezza = 0 = false[error] 2. Lunghezza <= 255 AND Lunghezza > 0 = true [PROPERTY LC_OK]
Parametro: Data Nascita	
Nome Categoria	**Scelte per la categoria**
Correttezza [FDN]	1. Data >= DataCorrente OR Data > (DataCorrente - 16 anni) = false [errore] 2. Data <= (DataCorrente - 16 anni) = true [PROPERTY_FDN_OK]
Parametro: Email	
FORMATO: **^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,10}$**	

Nome Categoria	Scelte per la categoria
Match [FE]	1. Formato = false [error] 2. Formato = true [PROPERTY FE_OK]
Parametro: Password	
FORMATO: ^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$	
Nome Categoria	**Scelte per la categoria**
Lunghezza[LP]	1. Lunghezza < 8 = false[error] 2. Lunghezza >= 8 = true [PROPERTY LP_OK]
Formato [FP]	1. Formato = false [IF LP_OK] [error] 2. Formato = true [IF LP_OK] [PROPERTY FP_OK]

Test Case ID	Test Frame	Esito
TC_GA_1.01	LN1	Errato: Nome non corretto
TC_GA_1.02	LN2, LC1	Errato: Cognome non corretto
TC_GA_1.03	LN2, LC2, FDN1	Errato: Data Nascita non corretta
TC_GA_1.04	LN2, LC2, FDN2	Errato: Data Nascita non corretta
TC_GA_1.05	LN2, LC2, FDN3, FE1	Errato: Email non corretta
TC_GA_1.06	LN2, LC2, FDN3, FE2, LP1	Errato: Password non corretta
TC_GA_1.07	LN2, LC2, FDN3, FE2, LP2, FP1	Errato: Password non corretto
TC_GA_1.08	LN2, LC2, FDN3, FE2, LP2, FP2	Corretto

9.01.02 Genera nuovo report

Parametro: Data Inizio	
Nome Categoria	**Scelte per la categoria**
Correttezza [FDI]	1. Data >= DataCorrente OR Data < '2021-01-01' 2. Data < DataCorrente && Data > '2021-01-01' = true [PROPERTY_FDN_OK]
Parametro: Data Fine	
Nome Categoria	**Scelte per la categoria**
Correttezza [FDF]	1. Data >= DataCorrente OR Data < '2021-01-01' 2. Data < DataCorrente && Data > '2021-01-01' = true [PROPERTY_FDN_OK]

Test Case ID	Test Frame	Esito
TC_GA_2.01	FDI1	Errato: Data Inizio non corretta
TC_GA_2.02	FDI2	Errato: Data Inizio non corretta
TC_GA_2.03	FDI3, FDF1	Errato: Data Fine non corretta
TC_GA_2.04	FDI3, FDF2	Errato: Data Fine non corretta
TC_GA_2.05	FDI3, FDF3	Corretto

9.02 Gestione Contratto

9.02.01 Aggiungi nuovo contratto

Parametro: Nome Ente	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LNE]	1. Lunghezza > 255 AND Lunghezza = 0 = false[error] 2. Lunghezza <= 255 AND Lunghezza > 0 = true [PROPERTY LNE_OK]
Parametro: Data Sottoscrizione	
Nome Categoria	**Scelte per la categoria**
Correttezza [EDS]	1. Data > DataCorrente = false [errore] 2. Data <= DataCorrente = true [PROPERTY_ EDS _OK]
Parametro: Costo Medio Unitario	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LCMU]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LCMU _OK]
Parametro: Consumo Medio Annuale	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LCMA]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LCMA _OK]
Parametro: Durata	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LD]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LD_OK]
Parametro: Prezzo Vendita	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LPV]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LD_OK]

Test Case ID	Test Frame	Esito
TC_GC_1.01	LNE1	Errore: Nome Ente non corretto
TC_GC_1.02	LNE2, EDS1	Errore: Data Sottoscrizione non corretto
TC_GC_1.03	LNE2, EDS2, LCMU1	Errore: Costo Medio Unitario non corretto
TC_GC_1.04	LNE2, EDS2, LCMU2, LCMA1	Errore: Consumo Medio Annuale non corretto
TC_GC_1.05	LNE2, EDS2, LCMU2, LCMA2, LD1	Errore: Durata non corretto
TC_GC_1.06	LNE2, EDS2, LCMU2, LCMA2, LD2, LPV1	Errore: Prezzo Vendita non corretto
TC_GC_1.07	LNE2, EDS2, LCMU2, LCMA2, LD2, LPV2	Corretto

9.02.02 Modifica un contratto

Parametro: Nome Ente	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LNE]	1. Lunghezza > 255 AND Lunghezza = 0 = false[error] 2. Lunghezza <= 255 AND Lunghezza > 0 = true [PROPERTY LNE_OK]
Parametro: Data Sottoscrizione	
Nome Categoria	**Scelte per la categoria**
Correttezza [EDS]	1. Data > DataCorrente = false [errore] 2. Data <= DataCorrente = true [PROPERTY_ EDS _OK]
Parametro: Costo Medio Unitario	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LCMU]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LCMU _OK]
Parametro: Consumo Medio Annuale	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LCMA]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LCMA _OK]
Parametro: Durata	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LD]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LD_OK]
Parametro: Prezzo Vendita	
Nome Categoria	**Scelte per la categoria**
Lunghezza [LPV]	1. Lunghezza <= 0 = false[error] 2. Lunghezza > 0 = true [PROPERTY LD_OK]

Test Case ID	Test Frame	Esito
TC_GC_2.01	LNE1	Errore: Nome Ente non corretto
TC_GC_2.02	LNE2, EDS1	Errore: Data Sottoscrizione non corretto
TC_GC_2.03	LNE2, EDS2, LCMU1	Errore: Costo Medio Unitario non corretto
TC_GC_2.04	LNE2, EDS2, LCMU2, LCMA1	Errore: Consumo Medio Annuale non corretto
TC_GC_2.05	LNE2, EDS2, LCMU2, LCMA2, LD1	Errore: Durata non corretto
TC_GC_2.06	LNE2, EDS2, LCMU2, LCMA2, LD2, LPV1	Errore: Prezzo Vendita non corretto
TC_GC_2.07	LNE2, EDS2, LCMU2, LCMA2, LD2, LPV2	Corretto

9.03 Gestione Accesso

9.03.01 Login

Parametro: Email	
Nome Categoria	**Scelte per la categoria**
Match [ME]	1. Formato = false [error] 2. Formato = true [PROPERTY ME_OK]
Parametro: Password	
Nome Categoria	**Scelte per la categoria**
Match [MP]	1. Match con password = false[error] 2. Match con password = true [PROPERTY MP_OK]

Test Case ID	Test Frame	Esito
TC_GAL_1.01	ME1	Errato: Email non corretta
TC_GAL_1.02	ME2, MP1	Errato: Password non corretta
TC_GAL_1.03	ME2, MP2	Corretto

9.04 Gestione IA

9.04.01 Modifica Parametri IA

Parametro: Preferenza Sorgente	
Nome Categoria	**Scelte per la categoria**
Selezione [EPS]	1. Selezione = false [error] 2. Selezione = true [PROPERTY EPS_OK]
Parametro: Percentuale Utilizzo Sorgente	
Nome Categoria	**Scelte per la categoria**
SommaPercentuale [EPU]	1. SommaPercentuale <= 0 = false [error] 2. SommaPercentuale > 0 = true [PROPERTY EPU_OK]

Test Case ID	Test Frame	Esito
TC_GIA_1.01	EPS1	Errore: Preferenza Sorgente non corretta
TC_GIA_1.02	EPS2, EPU1	Errore: Percentuale Utilizzo non corretta
TC_GIA_1.03	EPS2, EPU2	Corretto

10. Testing schedule

Le attività di pianificazione del testing avverranno come definito nei capitoli precedenti, cioè subito dopo la fase di design necessaria per la pianificazione.
La scrittura dei casi di test avverrà in contemporanea con lo sviluppo del codice.
L'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema. Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantirne il corretto funzionamento e produrre i report finali.
Per altre informazioni si rimanda ai documenti di management sullo schedule.