



Integration Test Plan

EasyExpo

Riferimento	
Versione	1.0
Data	09/12/2020
Destinatario	Prof.ssa Filomena Ferrucci
Presentato da	Recupito Gilberto
Approvato da	



Revision History

Data	Versione	Cambiamenti	Autore
09/12/2020	0.1	Stesura capitoli	Recupito Gilberto



1	Introduzione	4
2	Documenti correlati	4
	Relazione con il System Design Document (SDD)	4
3	Test d'Integrazione.....	4
4	Criteri di pass/fail	5
5	Criteri di sospensione e ripresa	5
5.1	Criteri di sospensione	5
5.2	Criteri di ripresa.....	6
6	Test deliverables.....	6
7	Materiale per effettuare il testing.....	6
8	Attività di training per lo staff	6
9	Responsabilità	6
10	Glossario	6

1 Introduzione

Il testing di integrazione rappresenta una delle fasi di testing più importanti, in quanto consiste nella verifica delle interazioni tra due o più componenti. L'obiettivo del testing consiste nella verifica della corretta interazione tra le componenti e il rispetto delle interfacce. Questo documento ha il compito di identificare la strategia di testing di integrazione per il sistema EasyExpo.

2 Documenti correlati

Relazione con il System Design Document (SDD)

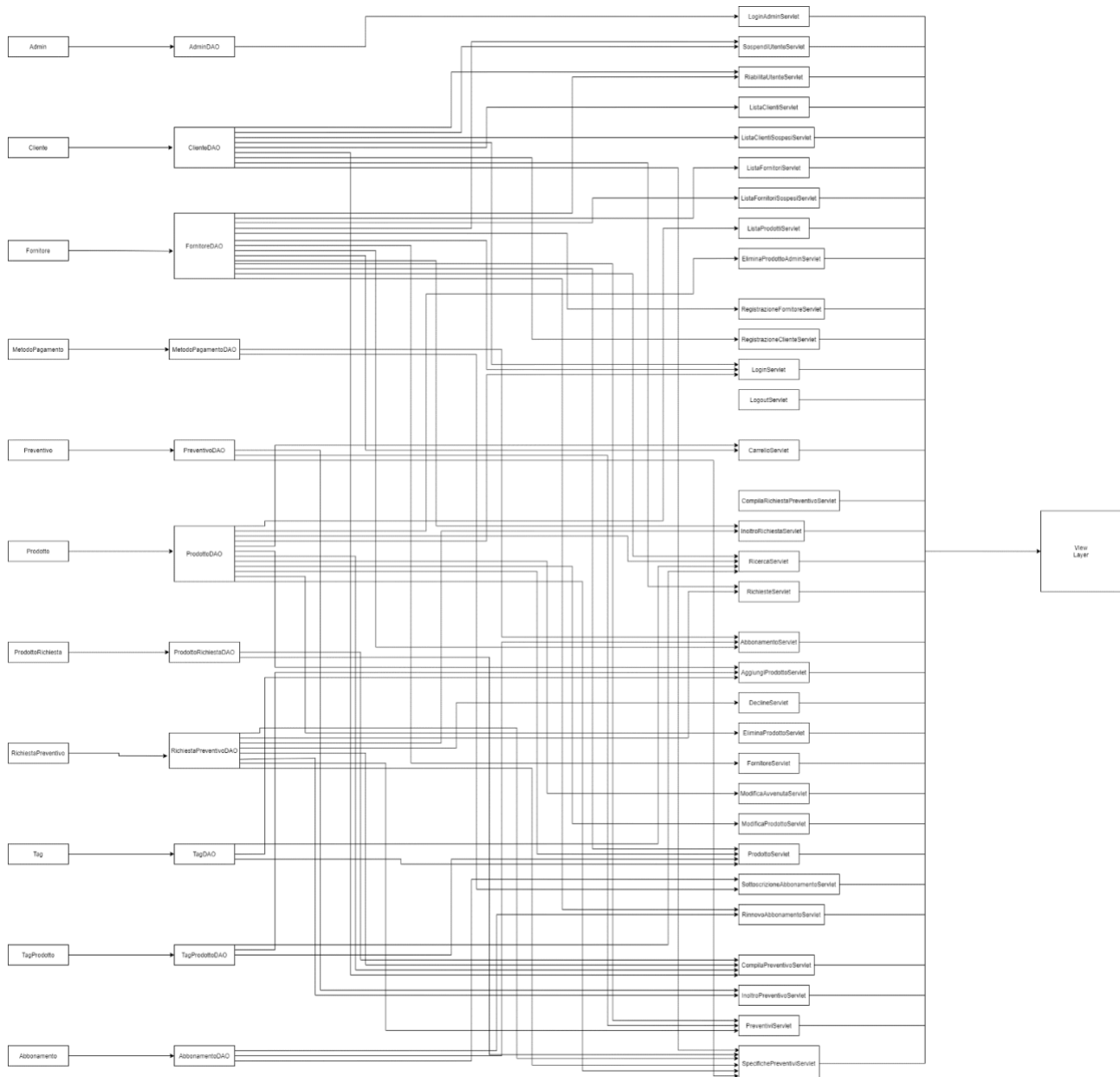
Per verificare la corretta integrazione dei sottosistemi del sistema EasyExpo i test case sono stati predisposti sulla base della suddivisione in sottosistemi presenti nel documento "C10_SDD_v1.0".

3 Test d'Integrazione

Lo scopo del testing di integrazione è quello di "mettere insieme" le componenti testate precedentemente tramite il test di unità per vedere come funzionano una volta integrate tra loro.

3.1 Approccio scelto

Per effettuare il testing di integrazione si è scelto di adoperare un approccio bottom-up. Il vantaggio fondamentale di questa tipologia di testing è quello della riusabilità del codice. Questo tipo di approccio prevede però la costruzione driver per simulare l'ambiente chiamante. È stato scelto quindi questo tipo di approccio perché sembra quello più intuitivo e semplice.



4 Criteri di pass/fail

Dopo aver individuato tutti i dati di input del sistema, quest'ultimi verranno raggruppati insieme in base alle caratteristiche in comune. Questa tecnica ci servirà per poter diminuire il numero di test da dover effettuare. Diremo che la fase di test ha successo se viene individuata effettivamente una failure all'interno del sistema, cioè l'output atteso per quel determinato input non è lo stesso previsto dall'oracolo. Successivamente la failure sarà analizzata e si passerà eventualmente alla sua correzione e verranno eseguiti nuovamente tutti i test necessari per verificare l'impatto che la modifica ha avuto sull'intero sistema. Diremo invece che il testing fallirà se l'output mostrato dal sistema coincide con quello previsto dall'oracolo.

5 Criteri di sospensione e ripresa

5.1 Criteri di sospensione

La fase di testing verrà sospesa nel momento in cui saranno raggiunti i risultati previsti in accordo con quello che è il budget a disposizione.

5.2 Criteri di ripresa

Tutte le attività di testing riprenderanno nel momento in cui verranno effettuate modifiche all'interno del sistema.

6 Test deliverables

I documenti che saranno prodotti durante questa fase sono i seguenti:

- Test Plan;
- Test Case Specification;
- Test Execution Report;
- Test Incident Report;
- Test Summary Report.

7 Materiale per effettuare il testing

Le risorse che vengono utilizzate dalle attività di testing comprendono i documenti di progetto Requirement Analysis Document, System Design Document ed Object Design Document, a partire dai quali vengono individuate le componenti da testare, rispettivamente, nel testing di sistema, nel testing di integrazione e nel testing d'unità. Per l'esecuzione di queste attività, invece, vengono utilizzati gli strumenti Selenium e JUnit su IntelliJ, insieme ad altri tool qualora sia necessario.

8 Attività di training per lo staff

Nessun team member attualmente conosce Selenium. Sarà, quindi, pianificata una sessione di training da parte del PM.

9 Responsabilità

Ogni team member sarà responsabile della fase di testing. Non è stata prevista una suddivisione tra sviluppatori e tester in quanto ogni team member deve svolgere tutte le mansioni.

10 Glossario

Form: termine utilizzato per riferirsi all'interfaccia di una applicazione che consente all'utente di inserire e inviare dati digitali.

GUI: è un tipo di interfaccia che consente all'utente l'interazione uomo - macchina in maniera visuale.

Testing: procedimento utilizzato per individuare le carenze di correttezza, completezza e affidabilità dei componenti software nel corso dello sviluppo.

Tool: strumento software utilizzato per ottenere un determinato risultato.