

Mille Miglia In Viaggio
Integration Test Plan
Versione 1.0



Data: 08/02/2019

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Integration Test Plan	Data: 08/02/2019

Coordinatore del progetto:

Nome	Matricola
De Lucia Andrea	

Partecipanti:

Nome	Matricola
Antonio Scognamiglio	0512104868
Giuliana Muto	0512104598
Matteo Fasolino	0512103038

Scritto da:	Matteo Fasolino
--------------------	-----------------

Revision History

Data	Versione	Descrizione	Autore
04/03/2019	1.0	Descrizione del documento	Matteo Fasolino
04/03/2019	1.0	Relazione con altri documenti	Matteo Fasolino
04/03/2019	1.0	Dettagli di level testing	Matteo Fasolino
04/03/2019	1.0	Pass Fail / Criteria	Matteo Fasolino
04/03/2019	1.0	Glossario	Giuliana Muto

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Integration Test Plan	Data: 08/02/2019

Sommario

1.DESCRIZIONE DEL DOCUMENTO	4
2. RELAZIONE CON ALTRI DOCUMENTI	4
3.DETTAGLI DEL LEVEL TESTING	4
3.1 Approccio di Testing Integration.....	4
3.2 Diagramma di Testing Integration.....	6
Componente da testare	7
4. PASS / FAIL CRITERIA	8
5.GLOSSARIO	8

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Integration Test Plan	Data: 08/02/2019

1.DESCRIZIONE DEL DOCUMENTO

Il testing di integrazione rappresenta una delle fasi di testing più importanti in quanto rileva bug che non sono stati determinati durante lo “unit-testing” e consente di assicurare che i vari sottosistemi sviluppati funzionino correttamente insieme. Questo documento ha il compito di identificare la strategia di testing di integrazione per il sistema MilleMigliaInViaggio. Verranno specificate le componenti da testare e il modo in cui queste verranno integrate.

2. RELAZIONE CON ALTRI DOCUMENTI

Per verificare la corretta integrazione dei sottosistemi sono stati predisposti dei test case basati sulla divisione in sottosistemi definita in fase di System Design nel documento “03. System Design Document.doc”.

3.DETTAGLI DEL LEVEL TESTING

3.1 Approccio di Testing Integration

La strategia adottata per il testing di integrazione è una strategia ibrida che comprende quella “Verticale” e la “Bottom-up”. La scelta di tali strategie è stata fatta per i seguenti motivi: la parte verticale è dovuta ai brevi tempi per raggiungere l’evasione dell’acquisto o per la messa in vendita: in tal modo, infatti, è possibile produrre candidate release da rilasciare. Per un dato caso d’uso, infatti, verrà testata la fetta di funzionalità di ogni componente collegata, ossia l’interfaccia, la logica e la gestione della persistenza. D’altra parte, con la strategia Bottom-up, verranno testati prima i livelli bassi e, successivamente, verranno combinati tutti i sottosistemi che sono chiamati da quelli già testati in precedenza, e poi testata la collezione risultante. Il tutto è ripetuto fino a quando tutti i sottosistemi non sono incorporati nel test. Per il testing verranno utilizzati i Test Stub che simulano le componenti dei livelli più alti, che non sono state ancora integrate. Tale strategia ibrida risulta essere la più semplice e naturale con cui eseguire il testing di integrazione, in quanto non ci sono dipendenze “orizzontali” all’interno dei layer. Il vantaggio con questo approccio è che i test case possono essere definiti in termini delle funzionalità del sistema

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Integration Test Plan	Data: 08/02/2019

(requisiti funzionali). L'application layer viene diviso in due parti: storage application layer e business application layer. Di seguito vengono mostrati i sottosistemi che compongono l'intero sistema:

Presentation Layer:

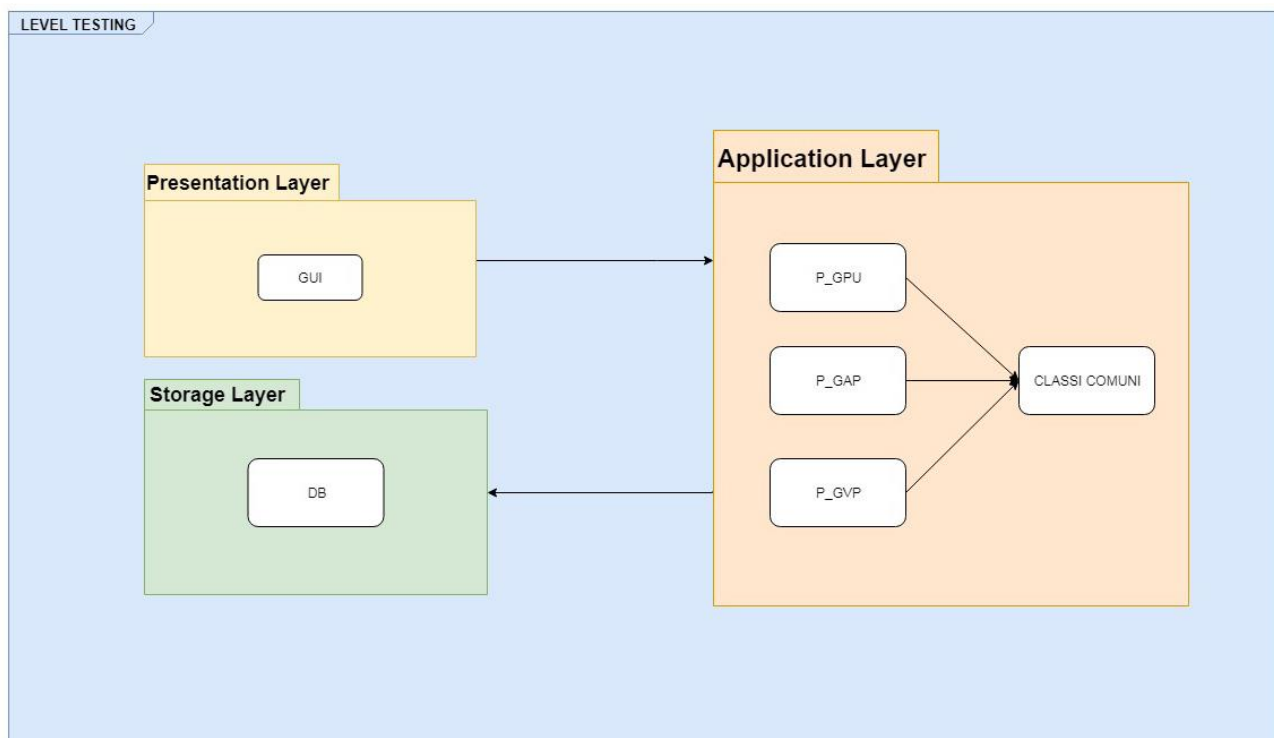
- Interfaccia Grafica (GUI)

Application Layer:

- Package Gestione Profilo Utente (P_GPU)
- Package Gestione Acquisto Pacchetti (P_GAP)
- Package Gestione Vendita Pacchetti (P_GVP)

Storage Layer:

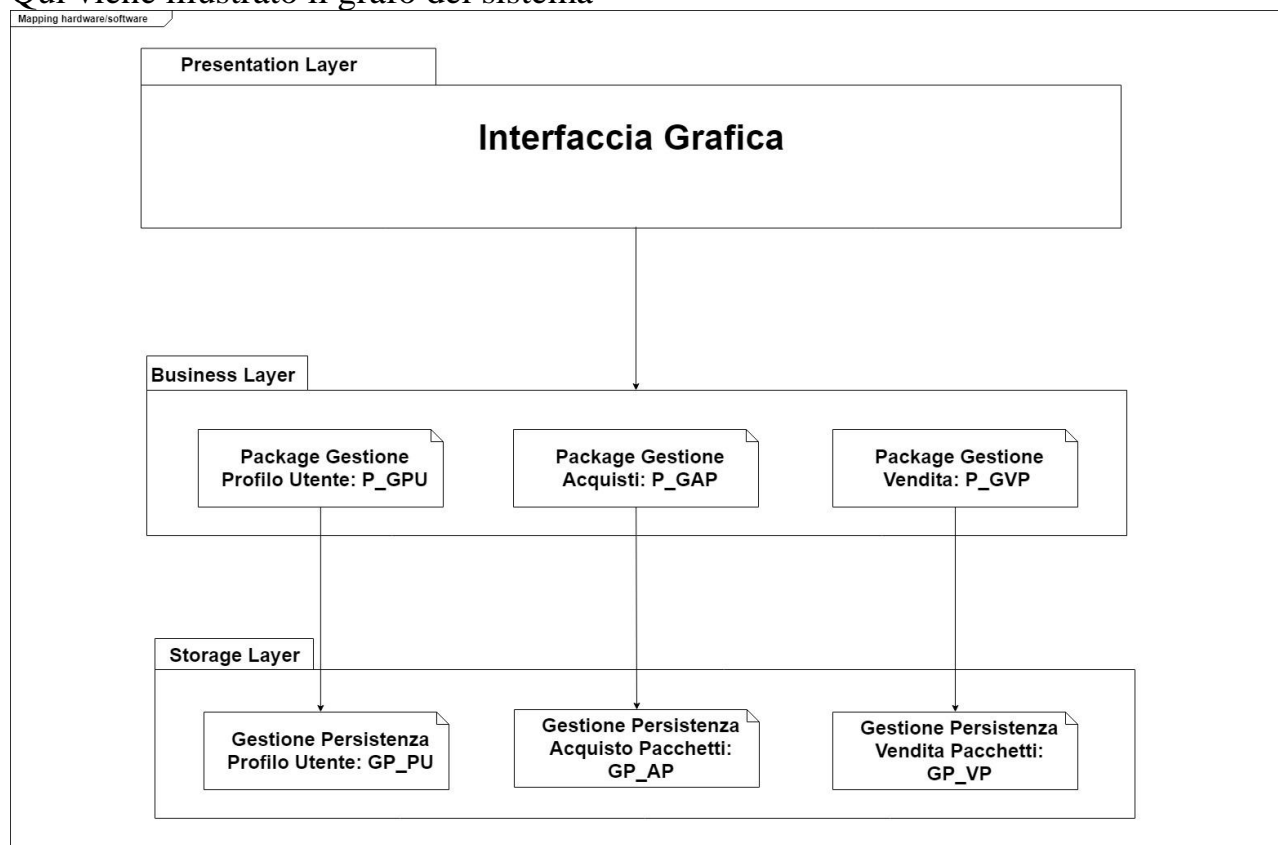
- Database (DB)



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Integration Test Plan	Data: 08/02/2019

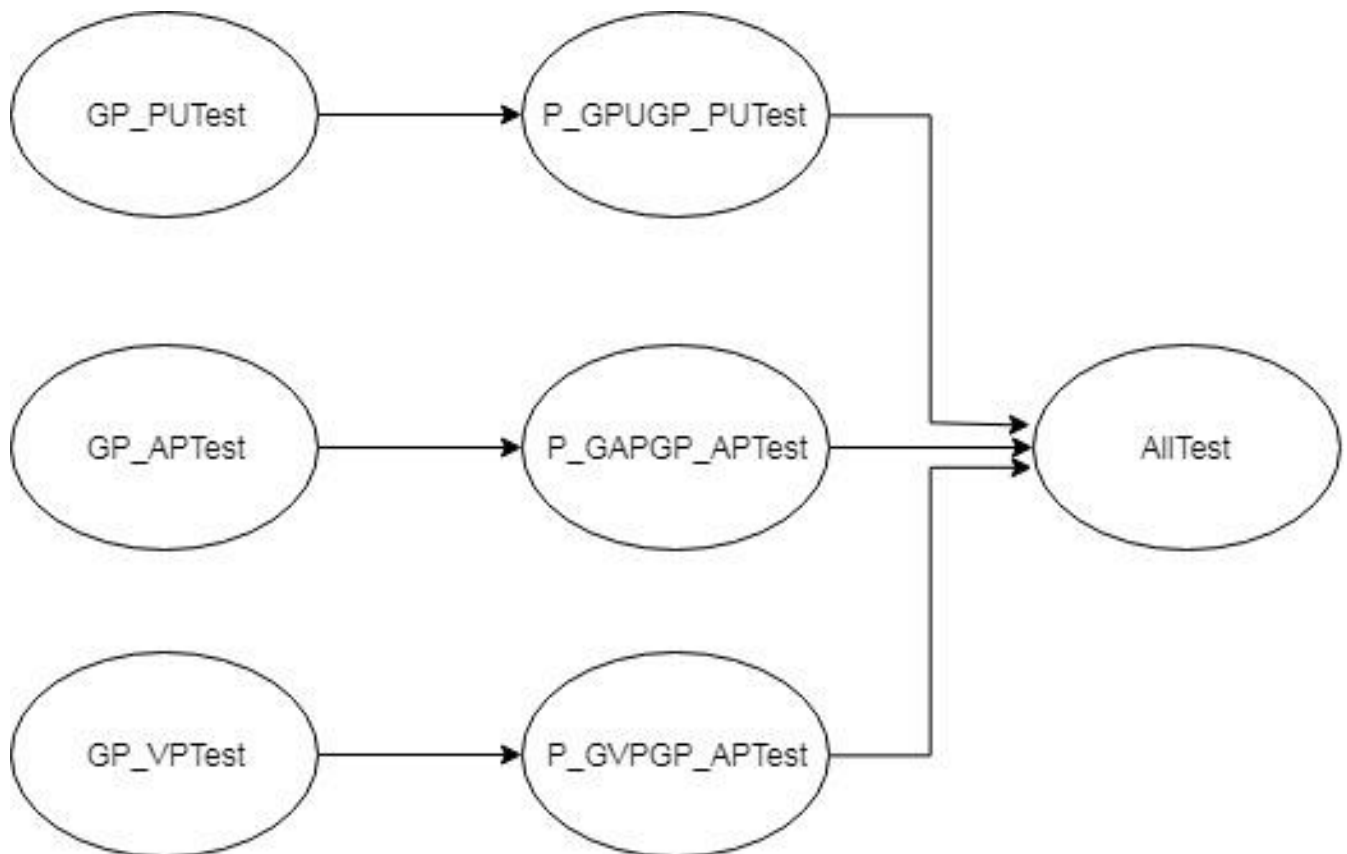
3.2 Diagramma di Testing Integration

Qui viene illustrato il grafo del sistema



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Integration Test Plan	Data: 08/02/2019

Il seguente diagramma mostra come verranno testate le componenti e quali verranno testate insieme:



Componente da testare

La scelta delle componenti da testare segue la decisione di eseguire una strategia di testing “Bottom-up”.

Le componenti da testare sono:

- Gestione Profilo Utente;
- Gestione Acquisto Pacchetti;
- Gestione Vendita Pacchetti.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: Integration Test Plan	Data: 08/02/2019

4. PASS / FAIL CRITERIA

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che la fase di testing avrà successo se individuerà una failure. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. La fase di testing sarà iterata per verificare che la modifica non abbia portato impatti su altri componenti del sistema. Al contrario, il testing fallirà se l'output osservato sarà uguale all'oracolo.

5.GLOSSARIO

Bug: errore al codice sorgente di un programma;

Testing: processo o metodo per trovare gli errori in un'applicazione o un programma software in modo che l'applicazione funzioni in base ai requisiti dell'utente finale;

Componente: un blocco di programma riutilizzabile anche in combinazione con altre componenti;

Failure: Fallimento in una o più fasi di test;

Pass: Successo in una o più fasi di test

Specifica: Nota, distinta in cui sono specificati uno per uno gli elementi utili a un determinato fine;

Successo: Esito positivo;

Test: Esperimento variamente espletato allo scopo di saggiare, mediante determinate reazioni, l'entità o la consistenza di un'attitudine o di una capacità individuale;

Verifica: Operazione di controllo per mezzo della quale si procede all'accertamento di determinati risultati.

Unit-testing: attività di testing di singole unità software.