



Laurea Magistrale in Informatica - Università di Salerno corso di  
*Ingegneria Gestione ed Evoluzione del Software* - Prof. Andrea De  
Lucia



# Test Plan Progetto EasyPlan

Riferimento	
Versione	1.0
Data	29/06/2023
Destinatario	Prof. Andrea De Lucia
Presentato da	Francesco Aurilio Oscar Granese
Approvato da	



## Sommario

Revision History.....	3
Test Plan del Progetto EasyPlan.....	4
1. Introduzione .....	4
2. Funzionalità da testare .....	4
3. Criteri pass/failed.....	4
4. Approccio.....	4
4.1. Testing di unità .....	5
4.2. Testing d'integrazione .....	5
4.3. Testing di sistema .....	5
4.4. Test di regressione .....	5
5. Criteri di sospensione e di ripresa .....	6
6. Materiale di testing.....	6
7. Test case .....	6
7.1. Utente.....	7
7.1.1. Formulazione piano di studi triennale .....	7
7.1.2. Formulazione piano di studi magistrale.....	8



## Revision History

Data	Versione	Descrizione	Autori
29/06/2023	0.1	Prima Stesura	Francesco Aurilio Oscar Granese



## Test Plan del Progetto EasyPlan

### 1. Introduzione

---

Lo scopo di questo documento è quello di pianificare l'attività di testing che verrà svolta sul software "EasyPlan" in seguito all'implementazione della CR\_02. In questo documento verranno descritte le funzionalità da testare, gli approcci, i criteri e il materiale utilizzati per l'attività di testing.

### 2. Funzionalità da testare

---

La CR\_02 è stata una change request di tipo perfeztivo che mira a separare il gruppo esami a scelta libera dai gruppi di esami opzionali. Si procederà prima al test delle classi che hanno subito le modifiche e poi verrà testato tutto il resto del sistema tramite il test di regressione per assicurarsi che le modifiche apportate non abbiano causato problemi a moduli che funzionavano correttamente.

### 3. Criteri pass/failed

---

Un caso di test ha esito positivo se l'output osservato è differente dal risultato previsto dall'oracolo; al contrario, un caso di test ha esito negativo se l'output osservato coincide con il risultato previsto dall'oracolo. Pertanto, le attività di test hanno successo nei casi in cui riescono ad individuare dei comportamenti anomali nell'esecuzione delle funzionalità del sistema.

### 4. Approccio

---

Le attività di testing da effettuare sul sistema si dividono in tre tipologie:

- Testing di unità che si occupa di testare il comportamento dei singoli componenti del sistema assicurandosi il corretto funzionamento di questi ultimi.
- Testing di integrazione che si effettua per testare l'interoperabilità delle componenti testate in precedenza dal testing di unità.



- Testing di sistema che mira a testare le funzionalità del sistema nella loro interezza verificando che il comportamento del sistema sia conforme ai requisiti funzionali e non funzionali.
- Testing di regressione che si effettua per assicurarsi che le modifiche apportate non abbiano introdotto malfunzionamenti ad altri moduli del sistema.

## 4.1. Testing di unità

---

Durante il test di unità verranno apportate se necessario le modifiche ai test di unità già presenti relativi alle classi che hanno subito modifiche durante la CR\_02. In particolare per quanto riguarda la formulazione del piano di studi il testing dovrà essere modificato in quanto ora è presente un nuovo gruppo di esami ovvero quelli a scelta libera.

## 4.2. Testing d'integrazione

---

Il testing di integrazione seguirà il testing di unità e verranno modificate e testate le componenti che effettuano il test di integrazione dei moduli interessati dalla CR\_02.

## 4.3. Testing di sistema

---

Il testing di sistema prevede il delle funzionalità che sono state modificate dalla CR\_02. In particolare la funzionalità interessata è la formulazione del piano di studi alla quale verranno quindi apportate delle modifiche.

## 4.4. Test di regressione

---

In seguito al testing delle classi che hanno subito modifiche si procederà al test di regressione dove verrà testato tutto il sistema con i test precedentemente prodotti per assicurarsi di non aver introdotto fault durante l'implementazione della CR\_02. Tale test verrà svolto tenendo conto della programmazione e delle scelte prese durante il test effettuato sul sistema a monte degli interventi di manutenzione. La documentazione per il test di regressione è presente nel documento "Test\_Plan\_Pre\_CRs\_EasyPlan.pdf" e "Test\_Plan\_Post\_CR\_01\_EasyPlan.pdf".



## 5. Criteri di sospensione e di ripresa

---

Le attività di testing pianificate dovranno protrarsi fino a quando tutti i test effettuati non presentino esito negativo. Nel momento in cui un test presenta un esito positivo evidenziando un potenziale problema, si procede pianificando e mettendo in atto un'opportuna soluzione. Successivamente, una volta eliminato il difetto rilevato, si ripeterà l'esecuzione del test precedente e di quelli strettamente correlati, per assicurarsi che la soluzione adottata abbia effettivamente risolto il problema e non ne abbia causato degli altri.

## 6. Materiale di testing

---

Per le attività di testing sono necessari i seguenti software:

- Eclipse
- Apache Tomcat
- MySQL DBMS
- Selenium
- Junit
- Mockito

## 7. Test case

---

La test suite del testing funzionale è definita seguendo un criterio di copertura che implica la definizione dei casi di test mediante category partition. In questo documento verranno illustrate solo le scelte riguardanti i moduli che hanno subito modifiche durante la CR\_02. Il restante testing essendo test di regressione farà riferimento al category partition presente nel documento "Test\_Plan\_Pre\_CRs\_EasyPlan.pdf" e "Test\_Plan\_Post\_CR\_01\_EasyPlan.pdf".



## 7.1. Utente

### 7.1.1. Formulazione piano di studi triennale

Parametro: CFU_gruppoOp Formato: Intero	
CATEGORIE	SCELTE
Somma [sto]	1. somma < somma_gruppo [errore] 2. somma >= somma_gruppo [property sommaSTOok]

Parametro: CFU_gruppoLib Formato: Intero	
CATEGORIE	SCELTE
Somma [stl]	1. somma < somma_gruppo [errore] 2. somma >= somma_gruppo [property sommaSTLok]

Codice	Combinazione	Esito
TC_1.4_1	sto1	errore
TC_1.4_2	stl1	errore
TC_1.4_3	sto2.stl1	errore
TC_1.4_4	stl2.sto1	errore
TC_1.4_5	sto2.stl2	Piano formulato correttamente



### 7.1.2. Formulazione piano di studi magistrale

Parametro: CFU_gruppoOp1 Formato: Intero	
CATEGORIE	SCELTE
Somma [som]	1. somma < somma_gruppo [errore] 2. somma >= somma_gruppo [property <b>sommaSOMok</b> ]

Parametro: CFU_gruppoOp2 Formato: Intero	
CATEGORIE	SCELTE
Somma [smo]	1. somma < somma_gruppo [errore] 2. somma >= somma_gruppo [property <b>sommaSMOok</b> ]

Parametro: CFU_gruppoLib Formato: Intero	
CATEGORIE	SCELTE
Somma [sml]	1. somma < somma_gruppo [errore] 2. somma >= somma_gruppo [property <b>sommaSMLok</b> ]

Codice	Combinazione	Esito
TC_1.5_1	som1	errore
TC_1.5_2	smo1	errore
TC_1.5_3	sml1	errore





TC_1.5_4	som2.smo1	errore
TC_1.5_5	smo2.som1	errore
TC_1.5_6	som2.sml1	errore
TC_1.5_7	smo2.sml1	errore
TC_1.5_8	sml2.som1	errore
TC_1.5_9	sml2.smo1	errore
TC_1.5_10	som2.smo2.sml1	errore
TC_1.5_11	som2.sml2.smo1	errore
TC_1.5_12	smo2.sml2.som1	errore
TC_1.5_13	som2.smo2.sml2	Piano formulato correttamente