



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	23/05/2023
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. Panoramica del sistema.....	4
3. Funzionalità da testare .....	4
4. Criteri pass/failed.....	5
5. Approccio.....	6
5.1. Testing di unità .....	6
5.2. Testing d'integrazione .....	7
5.3. Testing di sistema.....	7
6. Criteri di sospensione e di ripresa .....	7
7. Materiale di testing.....	7
8. Test case .....	8
8.1. Utente.....	8
8.1.1. Seleziona offerta formativa .....	8
8.1.2. Seleziona corso di laurea .....	9
8.1.3. Seleziona curriculum.....	9
8.1.4. Formulazione piano di studi triennale.....	10
8.1.5. Formulazione piano di studi magistrale .....	10
8.2. Gestione Amministratore.....	12
8.2.1. Login .....	12
8.2.2. Logout .....	13



## Revision History

Data	Versione	Descrizione	Autori
23/05/2023	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" che dovrà poi subire diverse modifiche. In questo documento verranno descritte le funzionalità da testare, gli approcci, i criteri e il materiale utilizzati per l'attività di testing.

### 2. Panoramica del sistema

---

Il sistema EasyPlan fornisce tutte le sue funzionalità attraverso un sito web ed è strutturato secondo il pattern architetturale MVC:

- **Model:** si occupa della memorizzazione di dati persistenti e del loro recupero dal database attraverso l'uso di query.
- **View:** composto da tutte le interfacce grafiche utili per interfacciare l'utente con il sistema.
- **Control:** composta dagli oggetti che racchiudono la logica responsabile del corretto funzionamento del sistema.

Sono stati individuati i seguenti sottosistemi:

- Gestione Amministratore
- Gestione curricula
- Gestione esami
- Gestione gruppo esami
- Gestione offerta formativa
- Utente

### 3. Funzionalità da testare

---

Per snellire l'attività di testing in modo da consentire al team di sviluppo di concentrarsi sulle modifiche da apportare al software si è scelto di testare esclusivamente le parti del sistema che rientrano nell'ambito delle future modifiche. Si è scelto pertanto di testare le seguenti componenti:



UTENTE
<ul style="list-style-type: none"><li>• Seleziona offerta formativa</li><li>• Seleziona corso di laurea</li><li>• Seleziona curriculum</li><li>• Formulazione piano di studi</li></ul>

GESTIONE AMMINISTRATORE
<ul style="list-style-type: none"><li>• Login</li><li>• Logout</li></ul>

## 4. 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. In presenza di casi di test



aventi successo sarà individuato il fault e si procederà alla sua correzione. Successivamente, sarà reiterata la fase di testing per verificare che le modifiche apportate non abbiano avuto un impatto dannoso sulle altre componenti del sistema.

## 5. 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.

### 5.1. Testing di unità

---

In questa fase vengono testate le singole unità per verificare che funzionino correttamente in isolamento. Durante il test di unità verranno creati dei casi di test per ogni unità al fine di verificare che producano i risultati desiderati. L'obiettivo principale di quest'attività di testing è di individuare eventuali difetti o errori nelle unità del software prima di integrarle con altre parti del sistema.

Per effettuare il testing di unità si è scelto di utilizzare l'approccio black-box in cui si esamina il sistema senza avere conoscenza interna della sua struttura o implementazione. Pertanto, la costruzione dei test si utilizzano i requisiti ignorando come sia stato realizzato il sistema al suo interno. L'approccio black-box è particolarmente utile per verificare se il sistema soddisfa i requisiti funzionali specificati e a rilevare eventuali malfunzionamenti funzionali.

Nel sistema EasyPlan si testeranno le singole classi BeanDAO che si occupano di effettuare query al database. Tali classi utilizzano delle corrispondenti classi Bean che, tuttavia, non verranno testate in quanto sono composte esclusivamente dalle proprie variabili di istanza e dai relativi metodi getter e setter associati. Successivamente durante l'attività di testing di unità verranno testate le singole servlet relative alle funzionalità da testare (par. 3).



## 5.2. Testing d'integrazione

Il testing di integrazione seguirà il testing di unità ed avrà l'obiettivo di individuare eventuali errori o problemi che potrebbero sorgere quando le unità vengono integrate e lavorano insieme. Per il test d'integrazione del sistema di EasyPlan si è deciso di utilizzare un approccio bottom-up che si concentra sull'integrazione delle componenti a partire da quelle che si trovano nel livello più basso della gerarchia del sistema.

## 5.3. Testing di sistema

Il testing di sistema è una fase del processo di testing del software che si concentra sulla validazione e sulla verifica del sistema software nel suo insieme, in conformità con i requisiti specificati.

Le attività di testing di sistema verranno effettuate tramite un approccio black-box e verrà svolta con l'aiuto del tool Selenium il quale fornisce un insieme di librerie Java utili a codificare i vari passaggi necessari a testare il sistema. Nel caso in cui le funzionalità da testare prevedessero degli input da parte dell'utente, è stata utilizzata la strategia del category partition.

## 6. 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.

## 7. Materiale di testing

Per le attività di testing sono necessari i seguenti software:

- Eclipse
- Apache Tomcat



- MySQL DBMS
- Selenium
- Junit
- Mockito

## 8. 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. Pertanto, per ciascuna funzionalità da testare saranno individuati i parametri di input e per ciascun parametro saranno individuate le categorie di cui si compone. Infine per ogni categoria saranno individuate tutte le possibili scelte. Tramite questo approccio verranno individuate le relazioni di incompatibilità, per le quali non saranno realizzati test case. La specifica di ognuno di questi test case si trova nel documento *Test\_Case\_Specification Pre\_CRs\_EasyPlan*.

### 8.1. Utente

#### 8.1.1. Selezione offerta formativa

Parametro: Pulsante	
CATEGORIE	SCELTE
Cliccato [clo]	1. Cliccato 2018/2019 [property cliccatoCLOok]

Codice	Combinazione	Esito
TC_1.1_1	clo1	Il sistema visualizza la pagina seleziona corso di laurea





### 8.1.2. Seleziona corso di laurea

Parametro: Pulsante	
CATEGORIE	SCELTE
Cliccato [cli]	1. Cliccato Triennale [property <b>cliccatoCLLtri</b> ] 2. Cliccato Magistrale [property <b>cliccatoCLCmag</b> ]

Codice	Combinazione	Esito
TC_1.2_1	cli1	Il sistema visualizza la pagina contenete i curricula della laurea triennale
TC_1.2_2	cli2	Il sistema visualizza la pagina contenete i curricula della laurea magistrale

### 8.1.3. Seleziona curriculum

Parametro: Pulsante	
CATEGORIE	SCELTE
Cliccato [clc]	1. Cliccato Standard [property <b>cliccatoCLCstd</b> ] 2. Cliccato SITS [property <b>cliccatoCLCsits</b> ] 3. Cliccato IOT [property <b>cliccatoCLCiot</b> ] 4. Cliccato SIC [property <b>cliccatoCLCsic</b> ] 5. Cliccato CC [property <b>cliccatoCLCcc</b> ] 6. Cliccato DS [property <b>cliccatoCLCds</b> ]

Codice	Combinazione	Esito
TC_1.3_1	clc1	Il sistema visualizza la pagina per la formulazione di un piano di studi basato sul curriculum standard.



TC_1.3_2	clc2	Il sistema visualizza la pagina per la formulazione di un piano di studi basato sul curriculum Software Engineering and IT Management.
TC_1.3_3	clc3	Il sistema visualizza la pagina per la formulazione di un piano di studi basato sul curriculum Internet of Things.
TC_1.3_4	clc4	Il sistema visualizza la pagina per la formulazione di un piano di studi basato sul curriculum Sicurezza informatica.
TC_1.3_5	clc5	Il sistema visualizza la pagina per la formulazione di un piano di studi basato sul curriculum Cloud Computing.
TC_1.3_6	clc6	Il sistema visualizza la pagina per la formulazione di un piano di studi basato sul curriculum Data Science & Machine Learning.

#### 8.1.4. 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]

Codice	Combinazione	Esito
TC_1.4_1	sto1	errore
TC_1.4_2	sto2	Piano formulato correttamente

#### 8.1.5. 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> ]

Codice	Combinazione	Esito
TC_1.5_1	som1	errore
TC_1.5_2	smo1	errore
TC_1.5_3	som2.smo1	errore
TC_1.5_4	smo2.som1	errore
TC_1.5_5	som2.smo2	Piano formulato correttamente



## 8.2. Gestione Amministratore

### 8.2.1. Login

Parametro: Username Formato: [0-9a-zA-Z\S]+	
CATEGORIE	SCELTE
Lunghezza [lnu]	1. Lunghezza==0 (errore) 2. Lunghezza>=1[property <b>lunghezzaLNUok</b> ]
Formato [fam]	1. Rispetta il formato [if <b>lunghezzaLNUok</b> ] [property <b>formatoFAMok</b> ] 2. Non rispetta il formato [if <b>lunghezzaLNUok</b> ] [errore]
Esiste [eua]	1. Esiste nel database [if <b>formatoFAMok</b> ] [property <b>esisteEUAok</b> ] 2. Non esiste nel database [if <b>formatoFAMok</b> ] [error]

Parametro: Password Formato: [A-Za-z0-9]	
CATEGORIE	SCELTE
Lunghezza [lnp]	1. Lunghezza<2 (errore) 2. Lunghezza>16(errore) 3. Lunghezza >=3 && Lunghezza <=16[property <b>lunghezzaLNPOk</b> ]
Formato [fpm]	1. Rispetta il formato [if <b>lunghezzaLNPOk</b> ] [property <b>formatoFPMok</b> ] 2. Non rispetta il formato [if <b>lunghezzaLNPOk</b> ] [errore]
Corrisponde [cp]	1. Corrisponde all'username [if <b>formatoFPMok</b> and <b>esisteEUAok</b> ] [property <b>corrispondeCPOk</b> ] 2. Non corrisponde all'username [if <b>formatoFPMok</b> and <b>esisteEUAok</b> ] [error]

Codice	Combinazione	Esito
--------	--------------	-------



TC_2.1_1	Inu1	errore
TC_2.1_2	Inu2.fam2	errore
TC_2.1_3	Inu2.fam1.eua2	errore
TC_2.1_4	Inp1	errore
TC_2.1_5	Inp2	errore
TC_2.1_6	Inp3.fpm2	errore
TC_2.1_7	Inp3.fpm1.cp2	errore
TC_2.1_8	Inu2.fam1.eua1.lnp 1	errore
TC_2.1_9	Inu2.fam1.eua1.lnp 2	errore
TC_2.1_10	Inu2.fam1.eua1.lnp 3.fpm2	errore
TC_2.1_11	Inu2.fam1.eua1.lnp 3.fpm1.cp2	errore
TC_2.1_12	Inu2.fam1.eua1.lnp 3.fpm1.cp1	Accesso effettuato con successo

### 8.2.2. Logout

Parametro: Pulsante	
CATEGORIE	SCELTE
Cliccato [cllg]	1. Cliccato [property <b>cliccatoCLLGok</b> ]



Codice	Combinazione	Esito
TC_2.2_1	cllg	Logout