****

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **T**est **P**lan  **D**ocument  Easy Pass   |  |  | | --- | --- | | Riferimento |  | | Versione | 1.0 | | Data | 4/12/2021 | | Destinatario | Prof. C. Gravino | | Presentato da | Montefusco Alberto  Mulino Martina  Rinaldi Viviana  Spina Gennaro | | Approvato da |  | |

Sommario

[Sommario 2](#_Toc530825397)

[RevisionHistory 3](#_Toc530825396)

1. [Introduzione 5](#_Toc530825398)

1.1 Descrizione del Documento 5

1. Relazioni con altri Documenti 5
2. Panoramica del Sistema 5
3. Funzionalità da testare e no 5

4.1 Matrice di tracciabilità del Testing 5

1. Criteri di successo/fallimento 5
2. Approccio 5

6.1 Testing di unità 5

6.2 Testing Funzionale 5

1. [Strumenti per il testing (Hardware/Software) 5](#_Toc530825398)
2. [Test Case 5](#_Toc530825398)
3. Specifica dei Test Cases 5

9.1 Registrazione di un Docente 5

9.2 Selezione di un numero di studenti da validare 5

9.3 Invio di un Green Pass 5

9.4 Ricerca dei report 5

9.5 Selezione del formato dei report 5

1. Pianificazione del testing e Assegnazione dei ruoli 5

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 5/12/2021 | 0.1 | Prima stesura: aggiunta capitoli 1 – 2 – 3 – 4 | Alberto Montefusco |
| 6/12/2021 | 0.2 | Aggiunta capitoli da 5 – 7 – 8 | Alberto Montefusco  Viviana Rinaldi |
|  | 0.3 | Aggiunta capitoli 6 | Martina Mulino |
|  | 1.0 | Aggiunta capitoli 6 |  |

1. Introduzione
   1. Descrizione del Documento

Il Test Plan del progetto Easy Pass presenta la pianificazione e la specifica di quelli che sono i Test Case, ovvero l’insieme di input e di risultati attesi che servono a testare una componente del Sistema. Nello specifico utilizzeremo la tecnica di testing “Category Partition”. Nel caso in cui delle attività di testing evidenziassero degli errori che possano causare comportamenti diversi da quelli attesi e che possano compromettere il buon utilizzo del Sistema da parte degli utenti, quest’ultimo può essere sottoposto ad un processo di correzione degli errori individuati, per garantire agli utenti finali un prodotto software che rispecchi tutte le specifiche finora stabilite nelle precedenti fasi di sviluppo.

Nello specifico, in questo Documento tratteremo il testing Black Box, cioè andremo a testare se una determinata componente si comporta come richiesto dal Documento di specifica e di analisi dei requisiti.

1. Relazioni con altri Documenti

Il Test Plan presenta diversi punti di correlazione con i documenti stilati durante le fasi precedenti dello sviluppo di Easy Pass. In particolare, il presente Documento fa riferimento ai casi d’uso descritti nel Requirement Analysis Document (RAD).

1. Panoramica del Sistema

Il Sistema Easy Pass fornisce tutte le sue funzionalità attraverso una Web Application. Per assicurarsi che ciascuna funzione si comporti come previsto, bisogna quindi assicurarsi che vengano testate tutte le funzionalità offerte dal Sistema. Nel caso in cui queste funzionalità si comportino nella maniera prevista, si può ritenere che il Sistema Easy Pass soddisfi gli obiettivi che si erano prefissati.

1. Funzionalità da testare e no

Le attività di testing previste per il Sistema Easy Pass prevedono di testare il corretto funzionamento della maggior parte delle funzionalità del sistema. Queste sono:

1. Registrazione di un Docente;
2. Selezione di un numero di studenti da validare;
3. Ricerca dei report tramite filtri;
4. Selezione di un formato dei report;
5. Invio di un Green Pass.
   1. Matrice di tracciabilità del Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisiti** | **Casi di Test** | | | | |
| **TC\_1** | **TC\_2** | **TC\_3** | **TC\_4** | **TC\_5** |
| **RF[1]** | **🗸** |  |  |  |  |
| **RF[2]** |  |  |  |  |  |
| **RF[3]** |  | **🗸** |  |  |  |
| **RF[4]** |  |  | **🗸** |  |  |
| **RF[5]** |  |  |  |  |  |
| **RF[6]** |  |  |  |  |  |
| **RF[7]** |  |  |  |  |  |
| **RF[8]** |  |  |  |  |  |
| **RF[9]** |  |  |  |  |  |
| **RF[10]** |  |  |  |  |  |
| **RF[11]** |  |  |  |  |  |
| **RF[12]** |  |  |  | **🗸** |  |
| **RF[13]** |  |  |  |  | **🗸** |

1. Criteri di successo/fallimento

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. Nel caso in cui uno o più casi di testi riscuotono successo, è possibile attuare un’opportuna procedura di correzione del difetto riscontrato e, successivamente, ricorrere ad un test di regressione per testare nuovamente la funzionalità modificata ed accertarsi che il problema sia stato risolto.

1. Approccio

L’attività di Testing del Sistema si articola nell’individuazione dei Test Case, nella scrittura dei Test Frames e nella descrizione di ogni Test Case fornendo la specifica per ognuno, che include l’oracolo.

* Testing di Sistema (che dovrebbe essere il testing funzionale?): si testano i requisiti funzionali individuati nel RAD attraverso il Black-Box testing e sfruttando la tecnica della Category partition.

1. Strumenti per il testing (Hardware/Software)
2. Test Case

Per sviluppare i Test Cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell’identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore. I Test Cases verranno definiti nel documento di Test Cases Specification (TCS).

1. Specifica dei Test Cases
   1. Registrazione di un Docente

**TC\_1 Registrazione Docente**

|  |  |
| --- | --- |
| **Parametro**: Nome  **Formato**: ^[a-zA-Z]+$ | |
| **Categorie** | **Possibili Scenari** |
| **Campo vuoto cvn** | 1. Il campo non è stato compilato [**errore**] 2. Il campo è stato compilato   [property **campoVuotoCVNok**] |
| **Lunghezza ln** | 1. Lunghezza <= 0 [if **campoVuotoCVNok**] [**errore**] 2. Lunghezza > 30 [if **campoVuotoCVNok**] [**errore**] 3. 0 < Lunghezza < 30 [if **campoVuotoCVNok**]   [property **lunghezzaLNok**] |
| **Formato fn** | 1. Non rispetta il formato   [if **campoVuotoCVNok** and l**unghezzaLNok**] [**errore**]   1. Rispetta il formato [if **campoVuotoCVNok** and **lunghezzaLNok**]   [property **formatoFNok**] |

|  |  |
| --- | --- |
| **Parametro:** Cognome  **Formato:** ^[a-zA-Z]+$ | |
| **Categorie** | **Scelte** |
| **Campo vuoto cvc** | 1. Il campo non è stato compilato [**errore**] 2. Il campo è stato compilato [property **campoVuotoCVCok**] |
| **Lunghezza lc** | 1. Lunghezza <= 0 [if **campoVuotoCVCok**] [**errore**] 2. Lunghezza > 30 [if **campoVuotoCVCok**] [**errore**] 3. 0 < Lunghezza < 30 [if **campoVuotoCVCok**]   [property **lunghezzaLCok**] |
| **Formato fc** | 1. Non rispetta il formato   [if **campoVuotoCVCok** and **lunghezzaLCok**] [**errore**]   1. Rispetta il formato [if **campoVuotoCVCok** and **lunghezzaLCok**] [property **formatoFCok**] |

|  |  |
| --- | --- |
| **Parametro:** Dipartimento (Dropdown menù)  **Formato:** - | |
| **Categorie** | **Scelte** |
| **Scelta sd** | 1. Scelta non effettuata [**errore**] 2. Scelta effettuata [property **sceltaSDok**] |

|  |  |
| --- | --- |
| **Parametro:** E-mail  **Formato:** ^[a-zA-Z0-9.!#$%&'\*+/=?^\_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)\*$ | |
| **Categorie** | **Scelte** |
| **Campo vuoto cve** | 1. Il campo non è stato compilato [**errore**] 2. Il campo è stato compilato [property **campoVuotoCVEok**] |
| **Formato fe** | 1. Non rispetta il formato [if **campoVuotoCVEok**] [**errore**] 2. Rispetta il formato [if **campoVuotoCVEok**] [property **formatoFEok**] |
| **Esiste ee** | 1. Esiste nel database [if **campoVuotoCVEok** and **formatoFEok**] [**errore**] 2. Non esiste nel database [if **campoVuotoCVEok** and **formatoFEok**] [property **esisteEEok**] |

|  |  |
| --- | --- |
| **Parametro:** Password  **Formato:** ^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$ | |
| **Categorie** | **Scelte** |
| **Campo vuoto cvp** | 1. Il campo non è stato compilato [**errore**] 2. Il campo è stato [property **campoVuotoCVPok**] |
| **Lunghezza lp** | 1. Lunghezza < 8 [if **campoVuotoCVPok**] [**errore**] 2. Lunghezza >= 8 [if **campoVuotoCVPok**]   [property  **lunghezzaLPok**] |
| **Formato fp** | 1. Non rispetta il formato [if **campoVuotoCVPok** and **lunghezzaLPok**] [**errore**] 2. Rispetta il formato [if **campoVuotoCVPok** and **lunghezzaLPok**] [property **formatoFPok**] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_1\_01 | cvn1 | FAIL |
| TC\_1\_02 | cvn2.ln1 | FAIL |
| TC\_1\_03 | cvn2.ln2 | FAIL |
| TC\_1\_04 | cvn2.ln3.fn1 | FAIL |
| TC\_1\_05 | cvn2.ln3.fn2.cvc1 | FAIL |
| TC\_1\_06 | cvn2.ln3.fn2.cvc2.lc1 | FAIL |
| TC\_1\_07 | cvn2.ln3.fn2.cvc2.lc2 | FAIL |
| TC\_1\_08 | cvn2.ln3.fn2.cvc2.lc3.fc1 | FAIL |
| TC\_1\_09 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd1 | FAIL |
| TC\_1\_10 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd2.cve1 | FAIL |
| TC\_1\_11 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd2.cve2.fe1 | FAIL |
| TC\_1\_12 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd2.cve2.fe2.ee1 | FAIL |
| TC\_1\_13 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd2.cve2.fe2.ee2.cvp1 | FAIL |
| TC\_1\_14 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd2.cve2.fe2.ee2.cvp2.lp1 | FAIL |
| TC\_1\_15 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd2.cve2.fe2.ee2.cvp2.lp2.fp1 | FAIL |
| TC\_1\_16 | cvn2.ln3.fn2.cvc2.lc3.fc2.sd2.cve2.fe2.ee2.cvp2.lp2.fp2 | PASS |

* 1. Selezione di un numero di studenti da validare

**TC\_2** **Inserimento numero studenti**

|  |  |
| --- | --- |
| **Parametro:** Numero Studenti  **Formato:** ^[1-9]\d\*$ | |
| **Categorie** | **Scelte** |
| **Campo vuoto cvns** | 1. Il campo non è stato compilato [**errore**] 2. Il campo è stato [property **campoVuotoCVNSok**] |
| **Formato fns** | 1. Non rispetta il formato [if **campoVuotoCVNSok**] [**errore**] 2. Rispetta il formato [if **campoVuotoCVNSok**] [property **formatoFNSok**] |
| **Valore vns** | 1. Valore <= 0 [if **campoVuotoCVNSok** and **formatoFNSok**] [**errore**] 2. Valore > 0 [if **campoVuotoCVNSok** and **formatoFNSok**] [property **valoreVNSok**] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_2\_01 | cvn1 | FAIL |
| TC\_2\_02 | cvn2.ln1 | FAIL |
| TC\_2\_03 | cvn2.ln2 | FAIL |
| TC\_2\_04 | cvn2.ln3.fn1 | PASS |

* 1. Invio di un Green Pass

**TC\_3 Invio del Green Pass**

|  |  |
| --- | --- |
| **Parametro:** Green Pass  **Formato:** immagine(png, jpeg), pdf | |
| **Categorie** | **Scelte** |
| **Formato fgp** | 1. campo vuoto [**errore**] 2. input non corrispondente al formato [**errore**] 3. input corrispondente al formato [property **formatoFGPok**] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_3\_01 | fgp1 | FAIL |
| TC\_3\_02 | fgp2 | FAIL |
| TC\_3\_03 | fgp3 | PASS |

* 1. Ricerca dei report

**TC\_4 Ricerca dei Report**

|  |  |
| --- | --- |
| **Parametro:** Nome completo del Docente  **Formato:** ^[a-zA-Z ,.'-]+$/i | |
| **Categorie** | **Scelte** |
| **Formato fndoc** | 1. non esiste nel DB [**errore**] 2. input non corrispondente al formato [**errore**] 3. input corrispondente al formato [property **formatoFNDOCok**] |

|  |  |
| --- | --- |
| **Parametro:** PrimaData  **Formato:** Data | |
| **Categorie** | **Scelte** |
| **Formato ffdat** | 1. input non corrispondente al formato [**errore**] 2. input corrispondente al formato [property **formatoFFDATok**] |

|  |  |
| --- | --- |
| **Parametro:** SecondaData  **Formato:** Data | |
| **Categorie** | **Scelte** |
| **Campo vuoto cvsdat** | 1. campo vuoto [if **formatoFFDATok**] [**errore**] 2. campo non vuoto [if **formatoFFDATok**]   [property **campoVuotoCVSDATok**] |
| **Formato fsdat** | 1. input non corrispondente al formato [if **campoVuotoCVSDATok**] [**errore**] 2. input corrispondente al formato [if **campoVuotoCVSDATok**] [property **formatoFSDATok**] |
| **Confronto csdat** | 1. PrimaData > SecondaData   [if **campoVuotoCVSDATok** and **formatoFSDATok**] [**errore**]   1. PrimaData <= SecondaData [if **formatoFSDATok** and **campoVuotoCVSDATok**] [property **confrontoCSDATok**] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_4\_01 | fndoc1 | FAIL |
| TC\_4\_02 | fndoc2 | FAIL |
| TC\_4\_03 | fndoc3.ffdat1 | FAIL |
| TC\_4\_04 | fndoc3.ffdat2.cvsdat1 | FAIL |
| TC\_4\_05 | fndoc3.ffdat2.cvsdat2.fsdat1 | FAIL |
| TC\_4\_06 | fndoc3.ffdat2.cvsdat2.fsdat2.csdat1 | FAIL |
| TC\_4\_07 | fndoc3.ffdat2.cvsdat2.fsdat2.csdat2 | PASS |