****

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **T**est **Summary**  **Report**  Easy Pass   |  |  | | --- | --- | | Riferimento |  | | Versione | 1.0 | | Data | 15/1/2022 | | 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](#_Toc530825398) 3

1.1 Riferimenti 4

1. Testing di Unità e di Integrazione 4
2. Testing di Sistema 4

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 15/01/2022 | 0.1 | Completamento | Viviana Rinaldi |
| 16/01/2022 | 1.0 | Revisione | Alberto Montefusco  Martina Mulino  Viviana Rinaldi  Gennaro Spina |

1. Introduzione

Easy Pass è una webApp che si propone di semplificare e rendere più sicure le procedure di controllo dei Green Pass da parte dei docenti. Grazie alla stesura di più documenti è possibile presentare una completa visione delle attività di testing che sono state svolte. Il loro scopo è garantire agli utilizzatori il corretto funzionamento di ogni funzionalità del Sistema.

Sono state svolte attività di testing per le funzionalità riguardanti:

* Docente
* Direttore di Dipartimento
* Studente

Nel documento in questione saranno presentate le coverage dei test e i risultati (successo o fallimento) dei testing di sistema effettuati.

* 1. Riferimenti

Il documento presente fa riferimento al TPD (per l’individuazione delle funzionalità da testare) e al TPD (in quanto presenta l’esito delle esecuzioni dei casi di test individuati nel documento appena citato).

Inoltre, il Test Summary Report contiene il sunto dei risultati sull’esecuzione specificati nel Test Incident Report tramite link esterno.

1. Testing di Unità e di Integrazione

Il testing unitario è stato utilizzato per testare le singole componenti del Sistema mentre quello di Integrazione è stato usato per integrare le unità testate in modo isolato per garantirne la corretta collaborazione.

Per controllare l’avanzamento della coverage dei test è stato utilizzato il tool JaCoCo. A fine testing, questi sono stati i risultati:

|  |  |
| --- | --- |
| **Branche Coverage** | **Line Coverage** |
|  |  |

Per un report completo si lascia il link al report di JaCoCo: <https://stefanolambiase.github.io/biblionet/>

1. Testing di Sistema

Per il testing di Sistema, utilizzato per testare le funzionalità del Sistema definite dai requisiti funzionali, sono state definite varie test suites tramite il tool Selenium IDE. Il suo utilizzo si basa sulla registrazione delle azioni che un utente può effettuare nell’interazione con il Sistema, così da poter eseguire i casi di test.

Di seguito vengono riportati i risultati delle esecuzioni dei test.

|  |  |  |
| --- | --- | --- |
| **Esecuzione** | **#Fallimenti** | **#Successi** |
|  |  |  |