

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Integration  Test Plan  Easy Traineeship   |  |  | | --- | --- | | Riferimento |  | | Versione | 1.0 | | Data | 12/01/2020 | | Destinatario | Top Management | | Presentato da | Alessandro Serritiello, Elio Testa, Felice Montella, Giuseppe Barisano, Luigi Barbato,  Pellegrino Aurilia, Salvatore Amendola, Simona Grilletto | | Approvato da | Vincenzo Belmonte, Vincenzo De Rosa | |

## 

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autori |
| 11/12/2019 | 0.1 | Inizio stesura documento | Alessandro Serritiello,  Luigi Barbato,  Simona Grilletto |
| 13/12/2019 | 0.2 | Analisi delle dipendenze e stesura dei diagrammi | Alessandro Serritiello,  Luigi Barbato,  Simona Grilletto |
| 12/01/2020 | 1.0 | Revisione del documento | Alessandro Serritiello |

Sommario

[Revision History 2](#_Toc27151358)

[1. Introduzione 4](#_Toc27151359)

[2. Riferimenti 4](#_Toc27151360)

[3. Test di Integrazione 4](#_Toc27151361)

[3.1 Approccio di Integration Testing 4](#_Toc27151362)

[3.2 Componenti da Testare 4](#_Toc27151363)

[4. Criteri di successo 8](#_Toc27151364)

## Introduzione

Il Testing di Integrazione è una delle fasi di testing più importante. Essa vuole sopperire ai mancati *“fault”* difficilmente individuabili nello *“Unit Testing”*, integrando differenti componenti insieme.

Questo documento ha il compito di identificare la strategia di testing di integrazione per il Sistema *ET*.

## Riferimenti

Per verificare la corretta Integrazione dei sottosistemi del Sistema *ET*, sono stati predisposti dei Test Case basati sulla divisione in sottosistemi proposta in fase di System Design, nel documento

*“ET\_SDD”*.

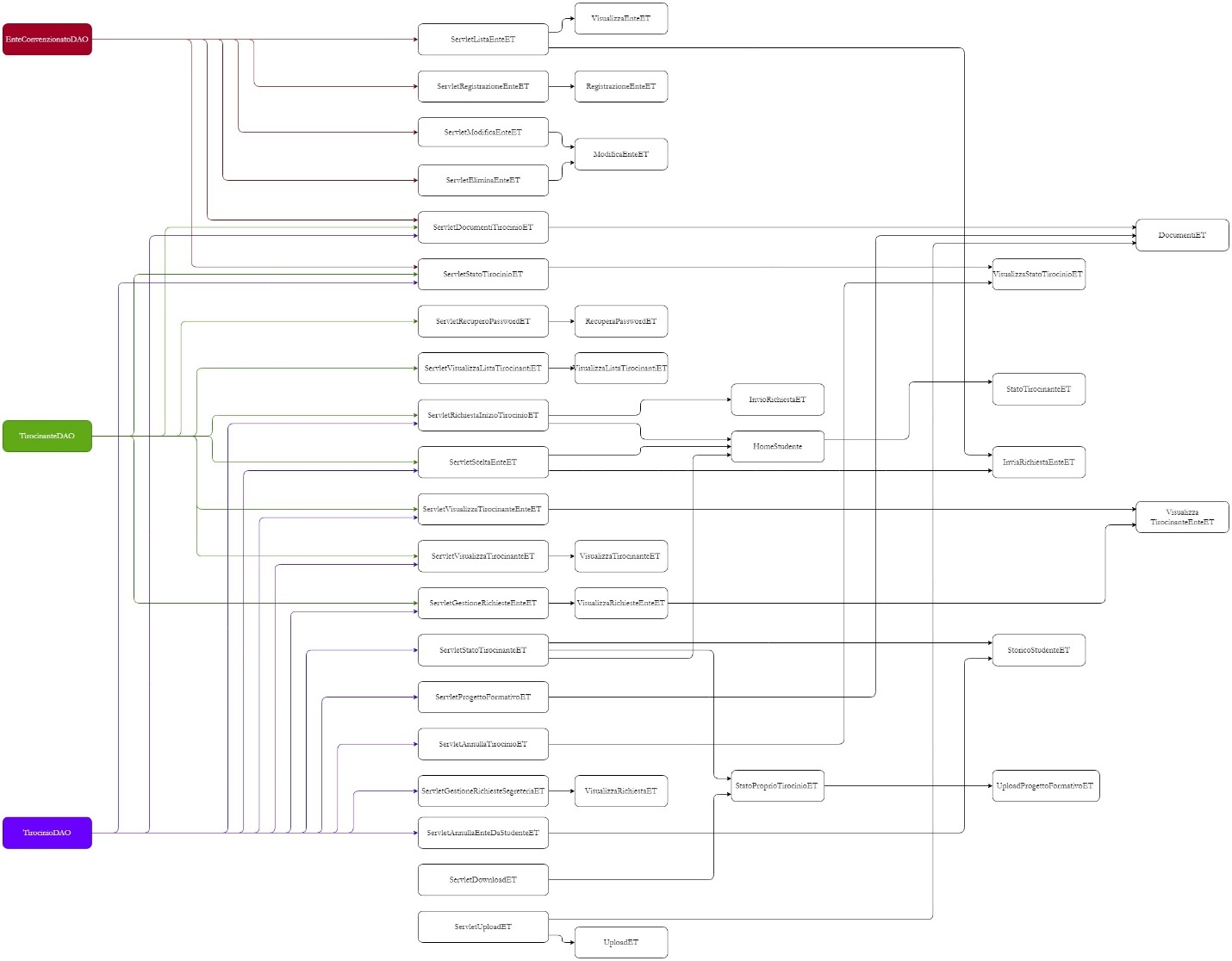
## Test di Integrazione

### Approccio di Integration Testing

La strategia che verrà adottata per il Testing di Integrazione è quella *“Bottom-Up”.*L’utilizzo di questa strategia consiste nel testare dapprima i sottosistemi nel layer più basso (in questo caso, quello dei DAO) e successivamente i sottosistemi dipendenti da quel layer, e quindi ripetere questa operazione fino a quando tutti i sottosistemi non sono stati testati, facendo utilizzo di Test Driver appositi.

### Componenti da Testare

|  |  |
| --- | --- |
| Package | Nome |
| DAO | EnteConvenzionatoDAO.java |
| DAO | TirocinanteDAO.java |
| DAO | TirocinioDAO.java |
| Servlet | ServletListaEnteET.java |
| Servlet | ServletRegistrazioneEnteET.java |
| Servlet | ServletModificaEnteET.java |
| Servlet | ServletEliminaEnteET.java |
| Servlet | ServletDocumentiTirocinioET.java |
| Servlet | ServletStatoTirocinioET.java |
| Servlet | ServletRecuperoPasswordET.java |
| Servlet | ServletVisualizzaListaTirocinantiET.java |
| Servlet | ServletRichiestaInizioTirocinioET.java |
| Servlet | ServletSceltaEnteET.java |
| Servlet | ServletVisualizzaTirocinanteEnteET.java |
| Servlet | ServletVisualizzaTirocinanteET.java |
| Servlet | ServletGestioneRichiesteEnteET.java |
| Servlet | ServletStatoTirocinanteET.java ServletVisualizzaTirocinioET.java |
| Servlet | ServletProgettoFormativoET.java |
| Servlet | ServletAnnullaTirocinioET.java |
| Servlet | ServletGestioneRichiesteSegreteriaET.java |
| Servlet | ServletAnnullaEnteDaStudenteET.java |
| Servlet | ServletDownloadET.java |
| Servlet | ServletUploadET.java |
| View | VisualizzaEnteET.jsp |
| View | RegistrazioneEnteET.jsp |
| View | ModificaEnteET.jsp |
| View | RecuperaPasswordET.jsp |
| View | VisualizzaListaTirocinantiET.jsp |
| View | VisualizzaTirocinanteET.jsp |
| View | VisualizzaRichiesteEnteET.jsp |
| View | VisualizzaRichiestaET.jsp |
| View | UploadET.jsp |
| View | InviaRichiestaET.jsp |
| View | HomeStudente.jsp |
| View | StatoProprioTirocinioET.jsp |
| View | VisualizzaStatoTirocinioET.jsp |
| View | StatoTirocinanteET.jsp |
| View | InviaRichiestaEnteET.jsp |
| View | StoricoStudenteET.jsp |
| View | UploadProgettoFormativoET.jsp |
| View | DocumentiET.jsp |
| View | VisualizzaTirocinanteEnteET.jsp |



## Criteri di successo

Il Testing ha successo quando il comportamento osservato dal Sistema implementato è diverso dal comportamento atteso, specificandolo attraverso il modello del Sistema.

Avremo, quindi, un SUCCESSO se il test individuerà una failure. In tal caso questa verrà analizzata e si procederà alla sua eventuale correzione. Successivamente alla correzione, la fase di testing verrà ripetuta per verificare che tale modifica non abbia impattato su altre componenti del Sistema. Se il test non riesce ad individuare un errore, parliamo di FALLIMENTO.