|  |  |
| --- | --- |
| Riferimento |  |
| Versione | 0.1 |
| Data | 14/01/2020 |
| Destinatario | Prof.ssa F. Ferrucci |
| Presentato da | Marco Delle Cave, Francesco Pagano,  Manuel Pisciotta, Alessia Olivieri |
| Approvato da |  |



Test Summary Report

TutoratoSmart

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Cambiamenti** | **Autori** |
| 14/01/2020 | 0.1 | Aggiunta screenshots casi di test DAO | Delle Cave Marco,  Pagano Francesco |

Sommario

1. [1. Introduzione 4](#_Toc29997619)

[1.1 Identificativo del documento 4](#_Toc29997620)

[1.2 Scopo 4](#_Toc29997621)

[1.3 Riferimenti 4](#_Toc29997622)

1. [2. Risultati di JUnit per il Model 4](#_Toc29997623)

[2.1. Features testate 4](#_Toc29997624)

[2.2. Panoramica dei risultati del test delle classi 5](#_Toc29997625)

[2.2.1. Classe ActivityTutorDAO 5](#_Toc29997626)

[2.2.2. Classe AppointmentDAO 5](#_Toc29997627)

[2.2.3. Classe ContainedInDAO 5](#_Toc29997628)

[2.2.4. Classe ManagesDAO 6](#_Toc29997629)

[2.2.5. Classe RegisterDAO 6](#_Toc29997630)

[2.2.6. Classe RequestDAO 6](#_Toc29997631)

[2.2.7. Classe StudentDAO 7](#_Toc29997632)

[2.2.8. Classe TutorDAO 7](#_Toc29997633)

[2.2.9. Classe UserDAO 7](#_Toc29997634)

[2.2.10. Classe ValidatesDAO 7](#_Toc29997635)

[2.2.11. Classe ActivityBean 8](#_Toc29997636)

[2.2.12. Classe AppointmentBean 8](#_Toc29997637)

[2.2.13. Classe ContainedIdBean 9](#_Toc29997638)

[2.2.14. Classe ManagesBean 9](#_Toc29997639)

[2.2.15. Classe RegisterBean 10](#_Toc29997640)

[2.2.16. Classe RequestBean 10](#_Toc29997641)

[2.2.17. Classe StudentBean 10](#_Toc29997642)

[2.2.18. Classe TutorBean 11](#_Toc29997643)

[2.2.19. Classe UserBean 11](#_Toc29997644)

[2.2.20. Classe ValidatesBean 11](#_Toc29997645)

1. [3. Risultati Mockito per il Controller 12](#_Toc29997646)

[3.1. Features testate 12](#_Toc29997647)

[3.2. Panoramica dei risultati del test delle classi 12](#_Toc29997648)

[3.2.1. Classe ActivityServlet (Aggiunta) 12](#_Toc29997649)

[3.2.2. Classe ActivityServlet (Modifica) 12](#_Toc29997650)

[3.2.3. Classe AppointmentServlet (Aggiunta) 13](#_Toc29997651)

[3.2.4. Classe AppointmentServlet (Modifica) 13](#_Toc29997652)

[3.2.5. Classe RequestServlet(Aggiunta) 13](#_Toc29997653)

[3.2.6. Classe RequestServlet(Modifica) 14](#_Toc29997654)

[3.2.7. Classe RequestServlet(Manage) 14](#_Toc29997655)

[3.2.8. Classe LoginServlet 15](#_Toc29997656)

[3.2.9. Classe RegistrationServlet 15](#_Toc29997657)

[3.2.10. Classe RegistrationServlet (Tutor) 16](#_Toc29997658)

# 1. Introduzione

Questo documento ha lo scopo di riportare i risultati dell’esecuzione dei test case di unità delle varie componenti del software web TutoratoSmart. In particolare, sono state testate le classi del package project.Model e project.Controller con l’utilizzo di JUnit..

## Identificativo del documento

Questo documento è caratterizzato da alcune proprietà che lo definiscono univocamente; tali proprietà sono: data, autore ed approvazione dei project manager. Il documento contiene il nome di chi si è occupato della stesura dello stesso.

## Scopo

Lo scopo di questo documento è quello di fornire una presentazione dei casi di test di unità per il software TutoratoSmart. I vari membri del team si sono impegnati nel verificare che le singole unità (classi e metodi) abbiano il comportamento atteso.

## Riferimenti

* TS\_RAD\_V\_1.1.docx
* TS\_SDD\_V\_0.6.docx
* TS\_ODD\_V\_0.4.docx
* TS\_Test\_case\_Specification\_V\_0.3.docx

# 2. Risultati di JUnit per il Model

Di seguito sono riportati i risultati dei test di unità per le classi del Package project.Model

## 2.1. Features testate

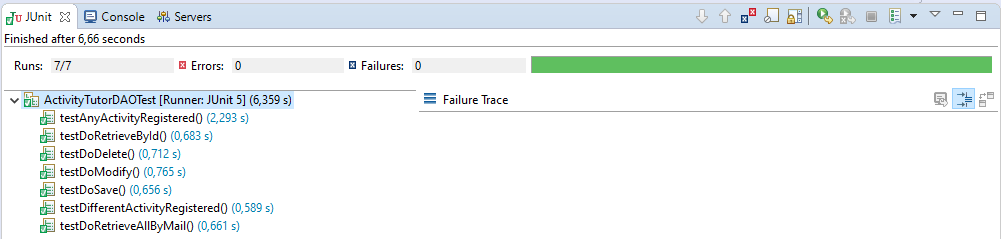
Il testing di unità si propone di effettuare il test delle classi presenti all’interno del Package project.Model. In particolare, saranno eseguiti:

* ActivityTutorDAO
* AppointmentDAO
* ContainedInDAO
* ManagesDAO
* RegisterDAO
* RequestDAO
* StudentDAO
* TutorDAO
* UserDAO
* ValidatesDAO
* ActivityTutorBean
* AppointmentBean
* ContainedInBean
* ManagesBean
* RegisterBean
* RequestBean
* StudentBean
* TutorBean
* UserBean
* ValidatesBean

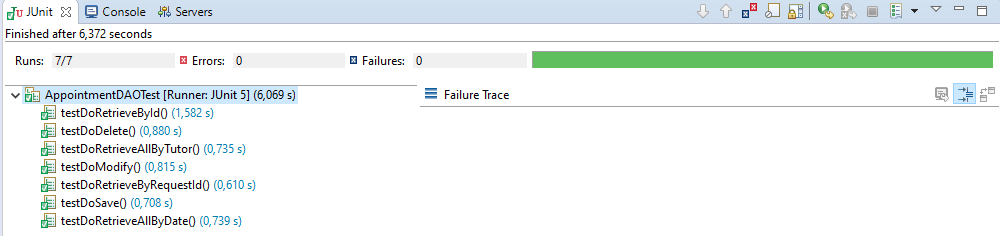
## 2.2. Panoramica dei risultati del test delle classi

In nessuna classe sono stati riscontrati errori e/o failure. Di seguito sono riportati I risultati dell’esecuzione di ogni classe.

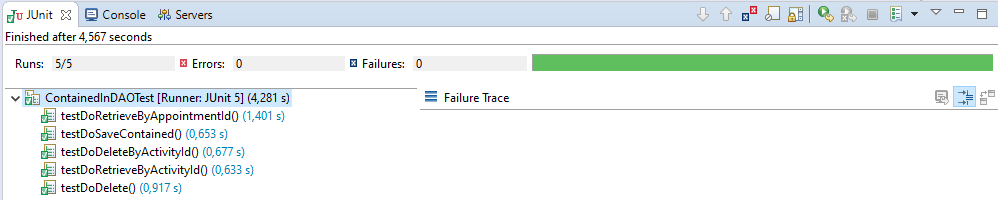
### 2.2.1. Classe ActivityTutorDAO

****

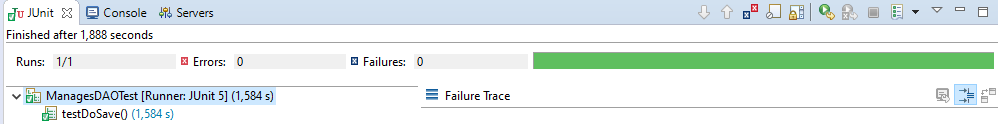
### 2.2.2. Classe AppointmentDAO

****

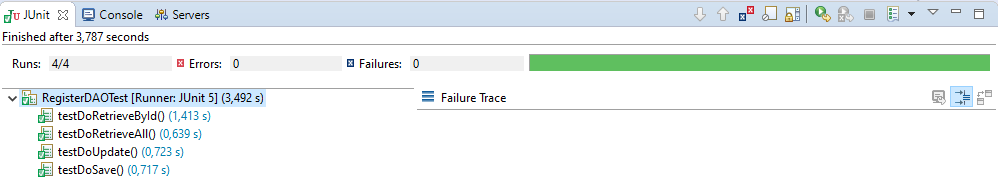
### 2.2.3. Classe ContainedInDAO



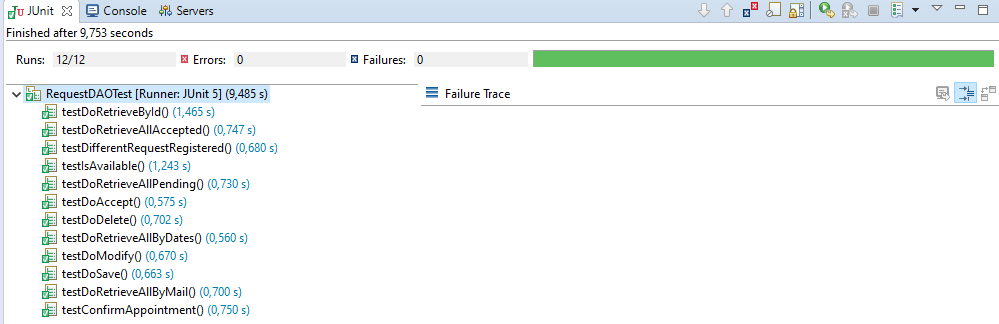
### 2.2.4. Classe ManagesDAO



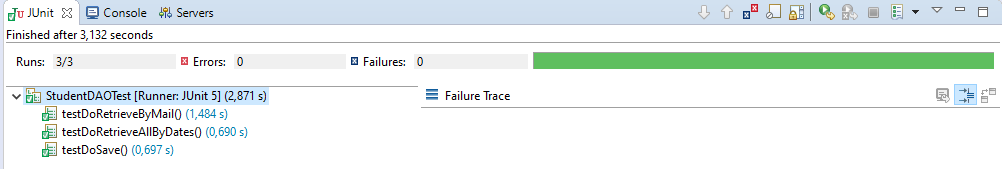
### 2.2.5. Classe RegisterDAO



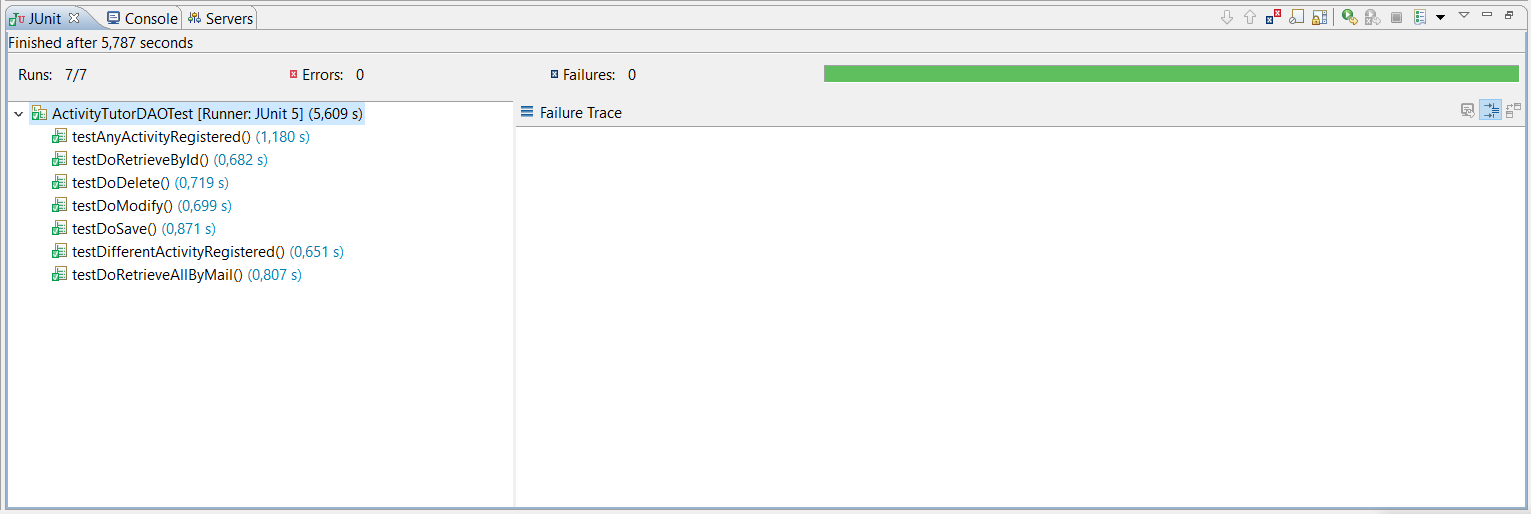
### 2.2.6. Classe RequestDAO



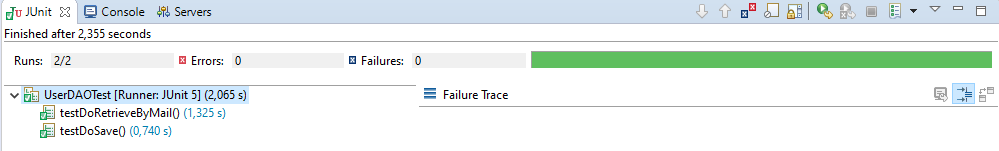
### 2.2.7. Classe StudentDAO



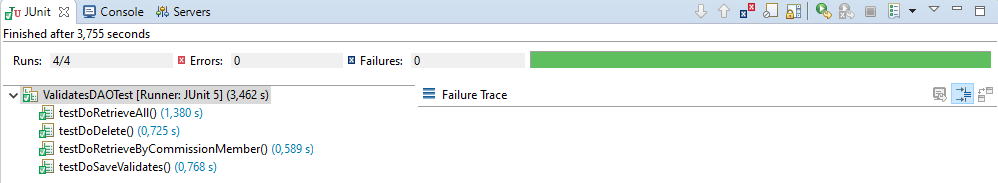
### 2.2.8. Classe TutorDAO

****

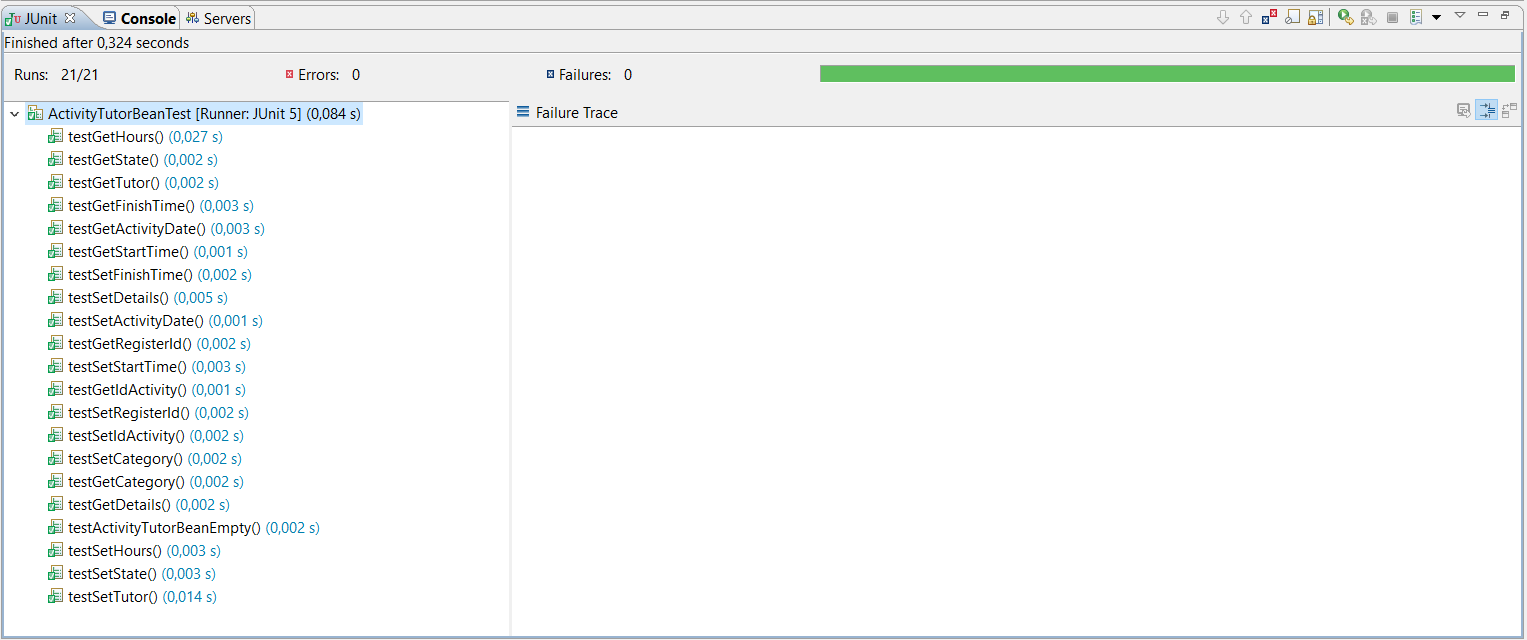
### 2.2.9. Classe UserDAO

****

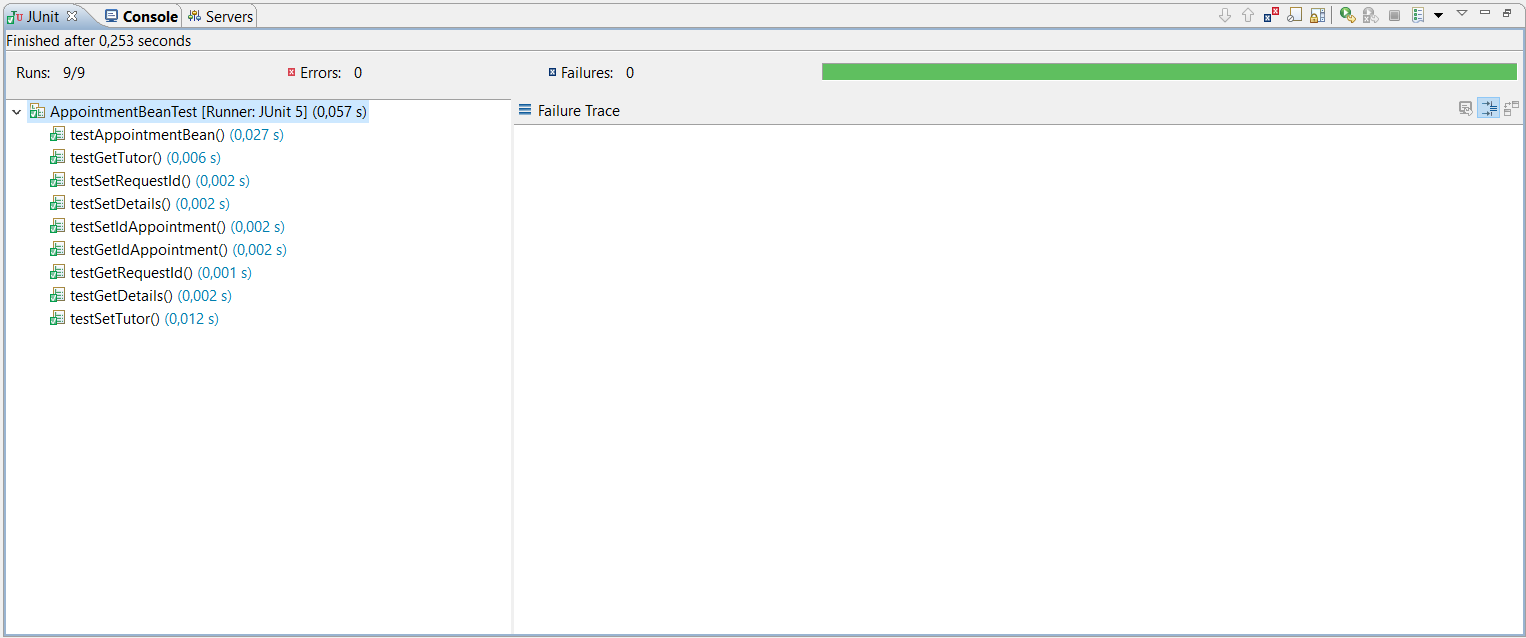
### 2.2.10. Classe ValidatesDAO



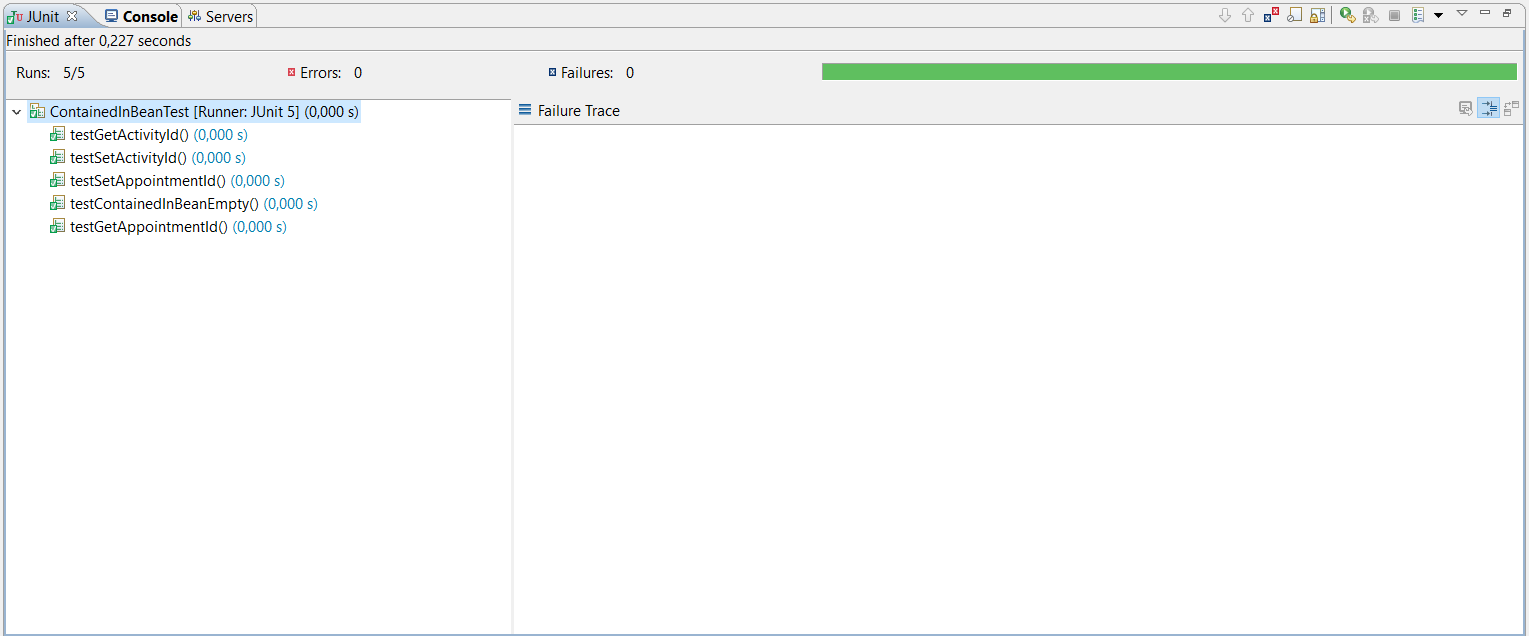
### 2.2.11. Classe ActivityBean



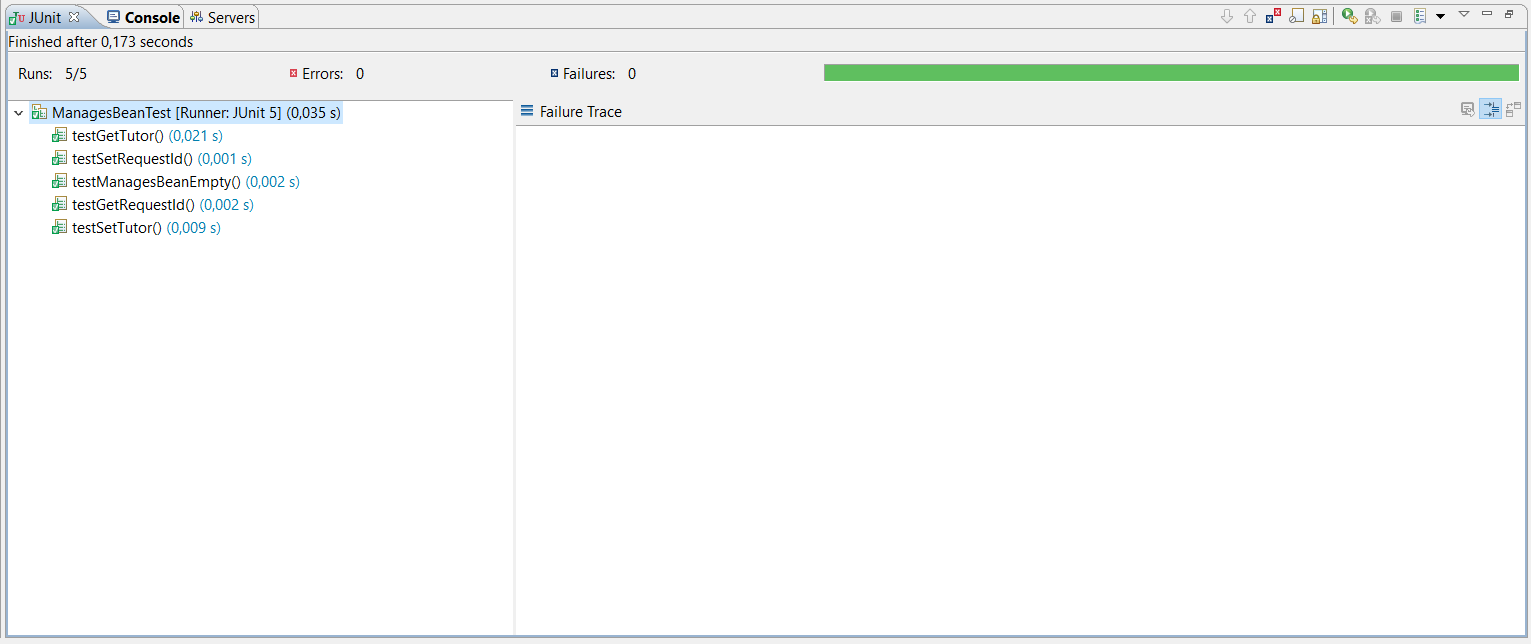
### 2.2.12. Classe AppointmentBean



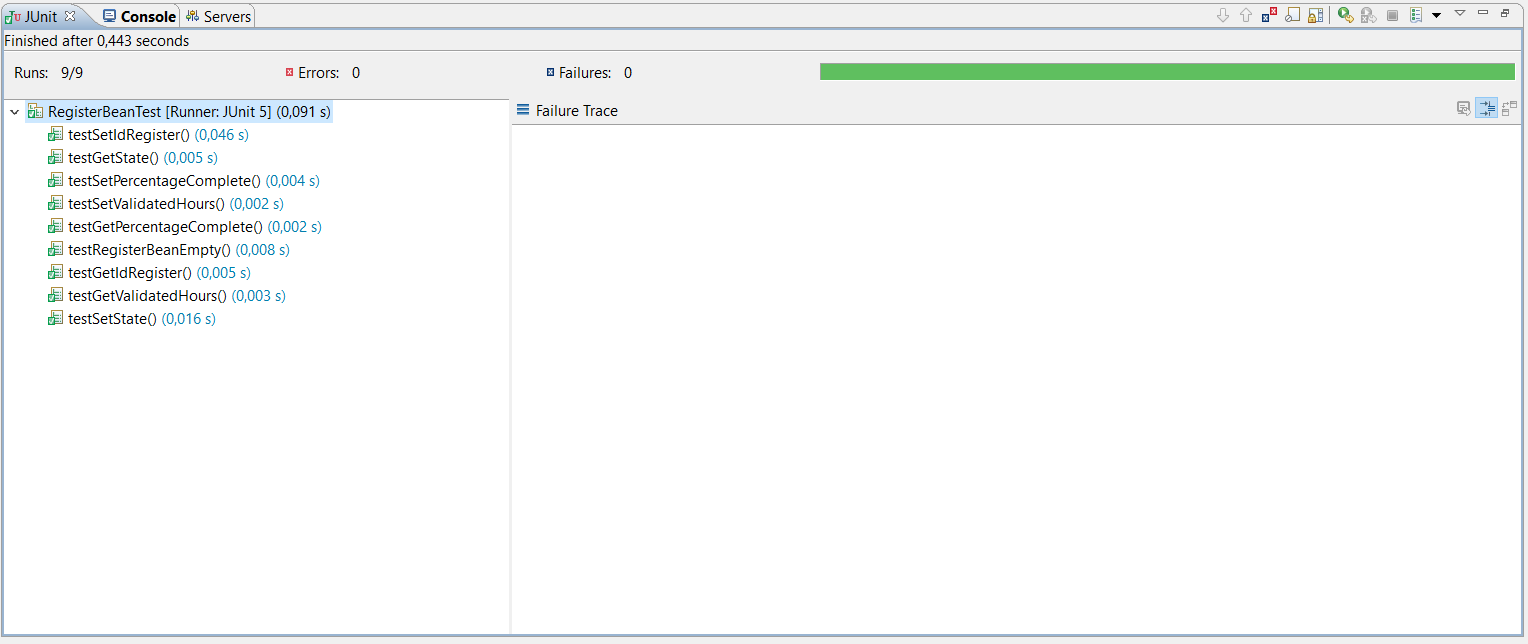
### 2.2.13. Classe ContainedIdBean



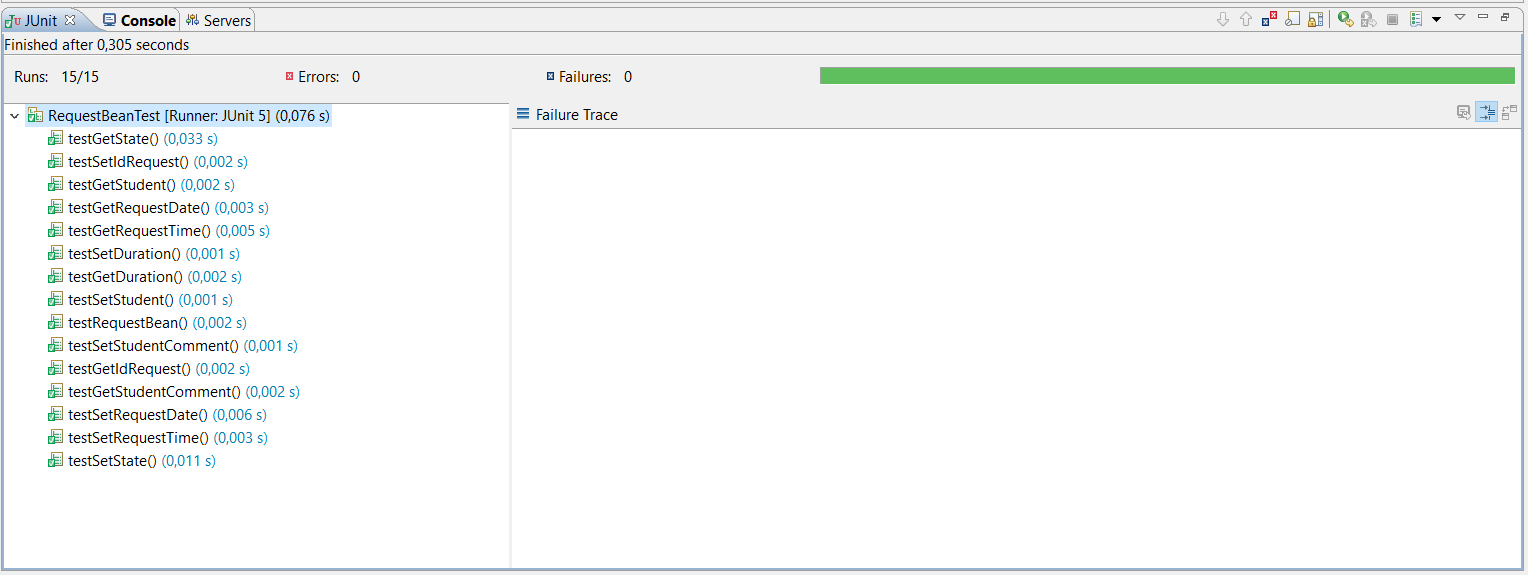
### 2.2.14. Classe ManagesBean



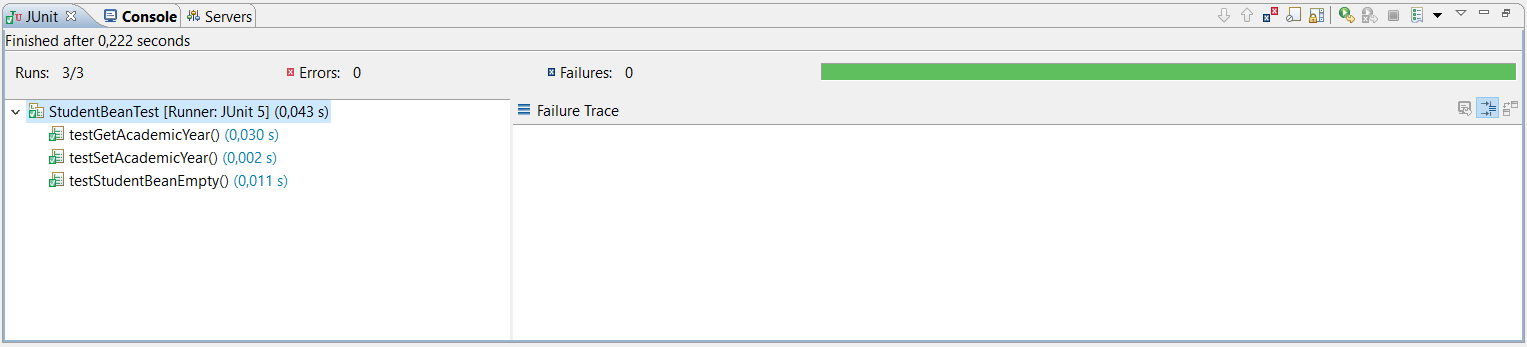
### 2.2.15. Classe RegisterBean



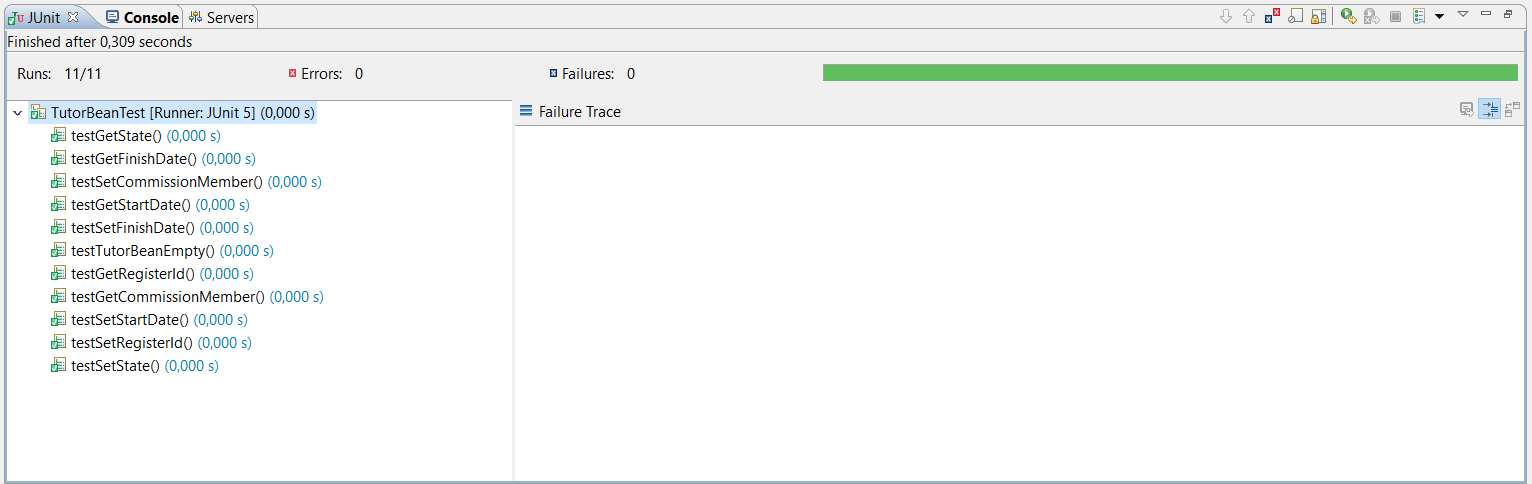
### 2.2.16. Classe RequestBean



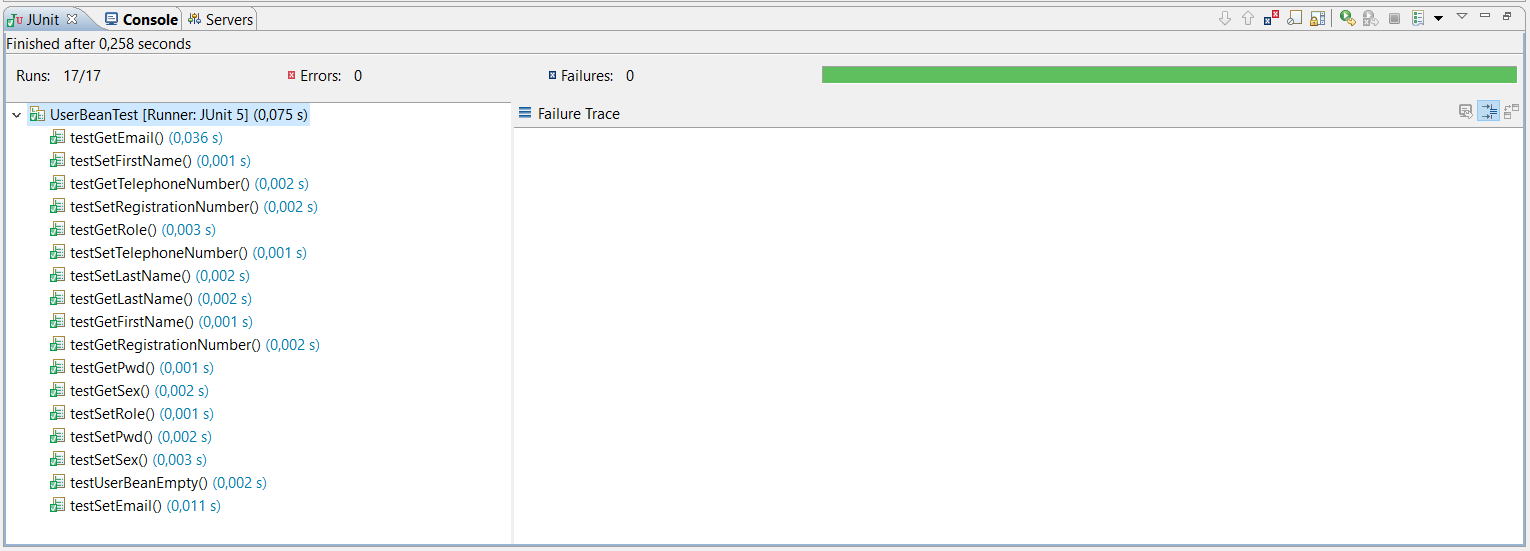
### 2.2.17. Classe StudentBean



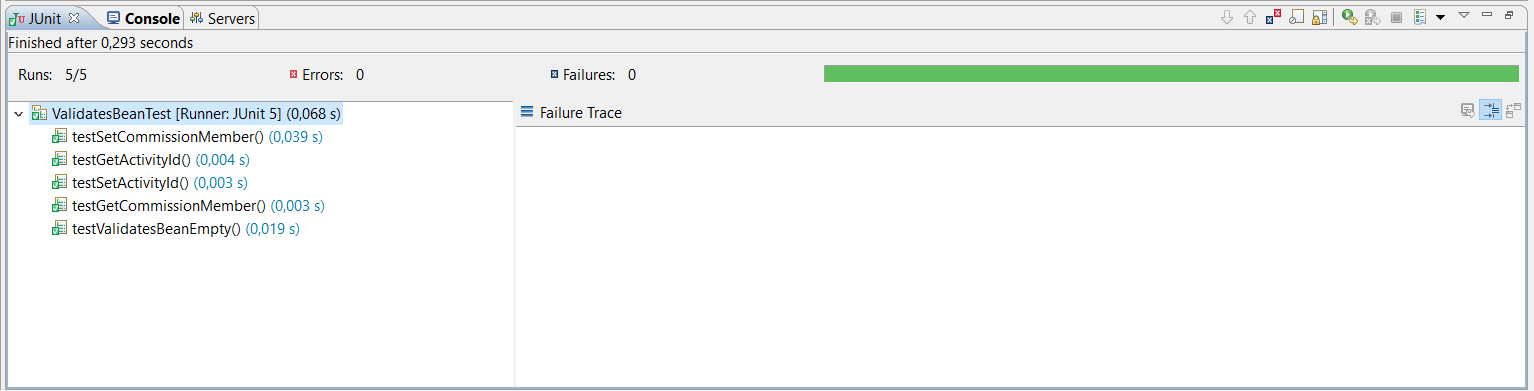
### 2.2.18. Classe TutorBean



### 2.2.19. Classe UserBean



### 2.2.20. Classe ValidatesBean

****

# 3. Risultati Mockito per il Controller

Di seguito sono riportati i risultati dei test di unità per le classi del Package project.Control

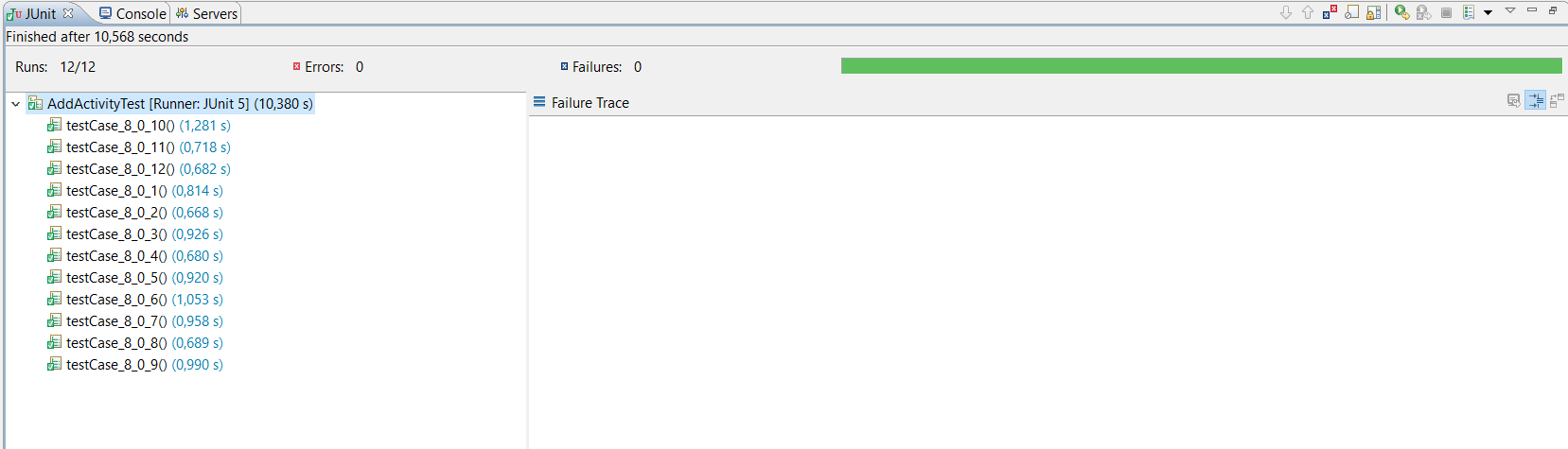
## 3.1. Features testate

Il testing di unità si propone di effettuare il test delle classi presenti all’interno del Package project.Control. In particolare, saranno eseguiti:

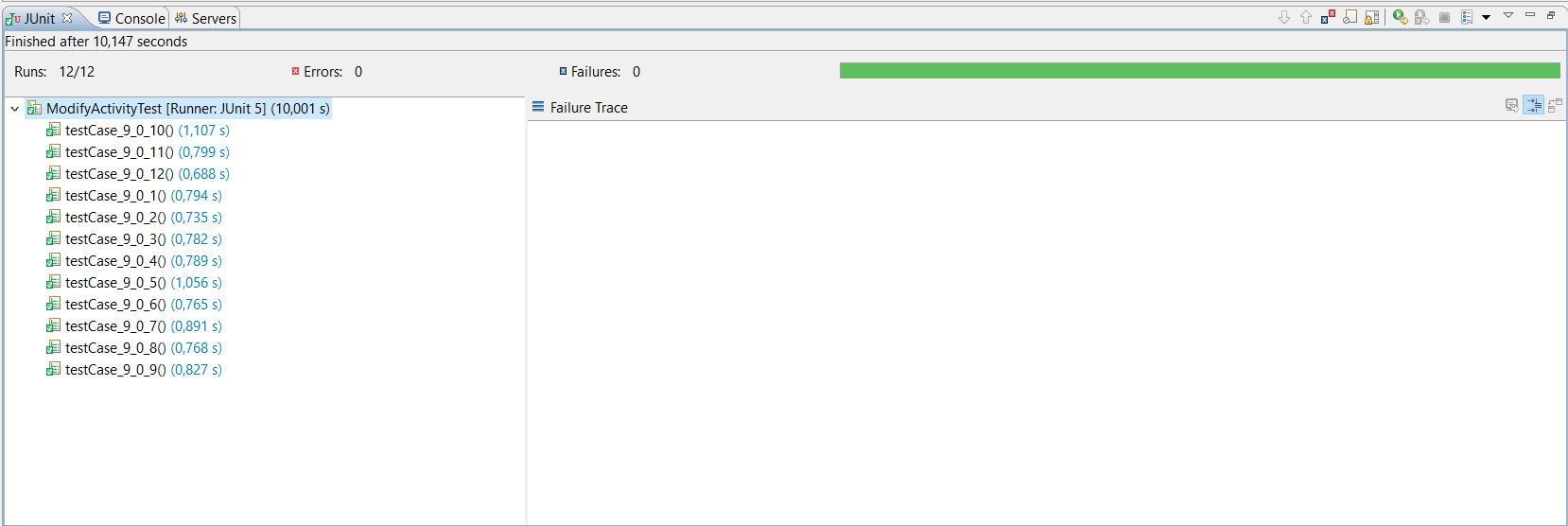
* ActivityServlet
* AppointmentServlet
* LoginServlet
* RegistrationServlet
* RequestServlet

## 3.2. Panoramica dei risultati del test delle classi

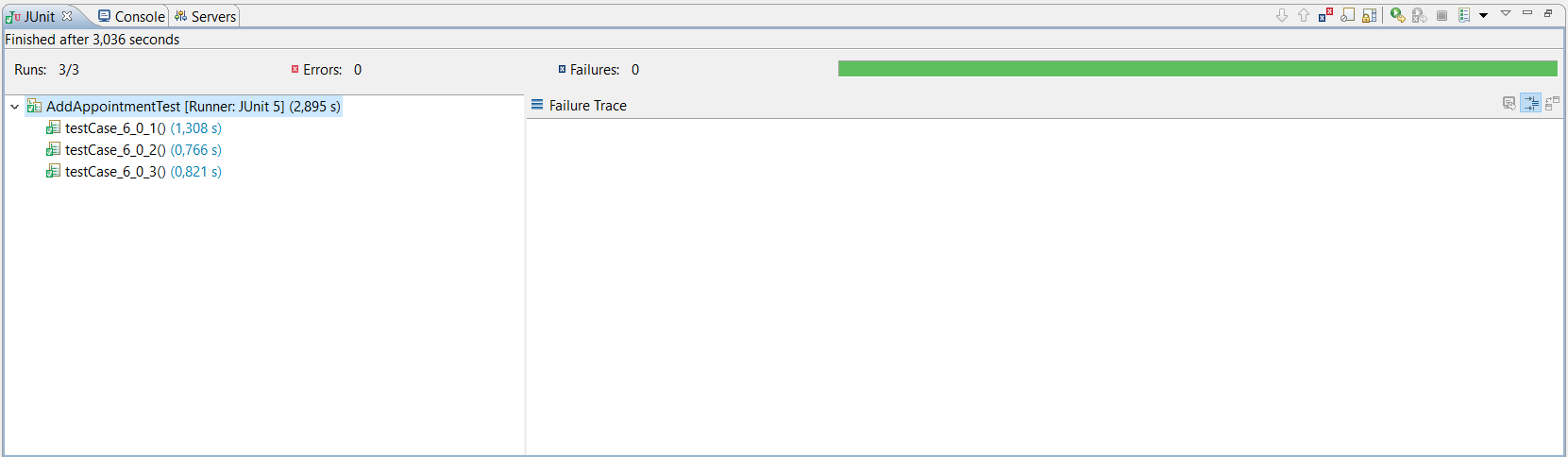
### Classe ActivityServlet (Aggiunta)



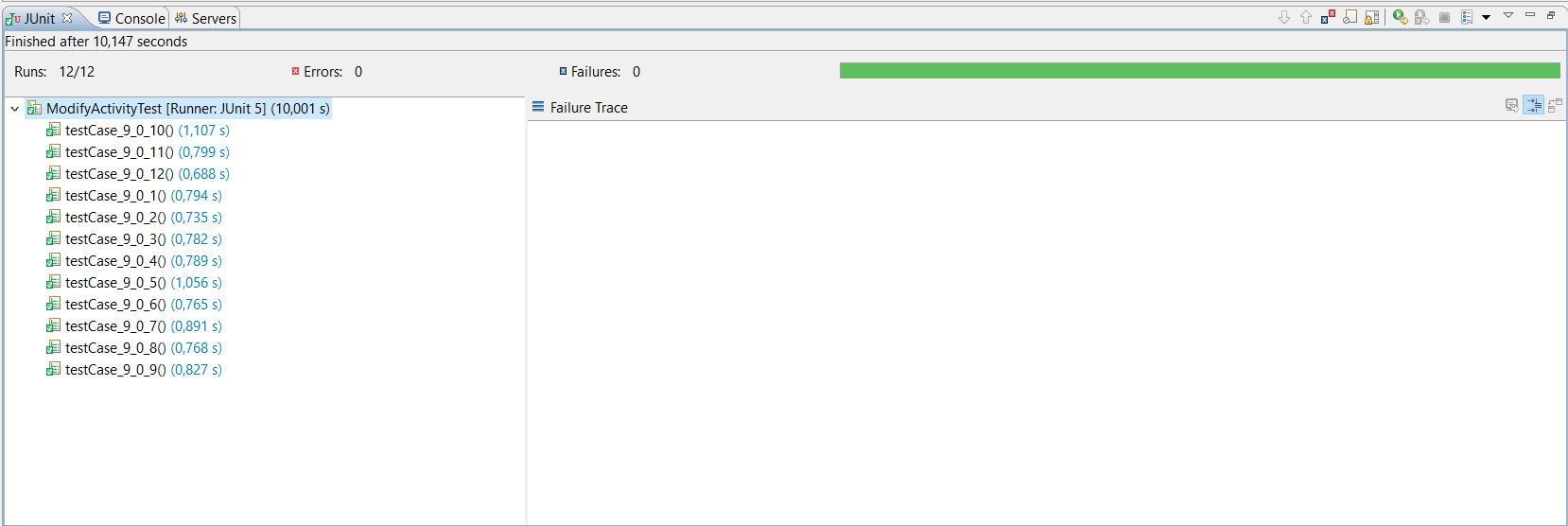
### Classe ActivityServlet (Modifica)



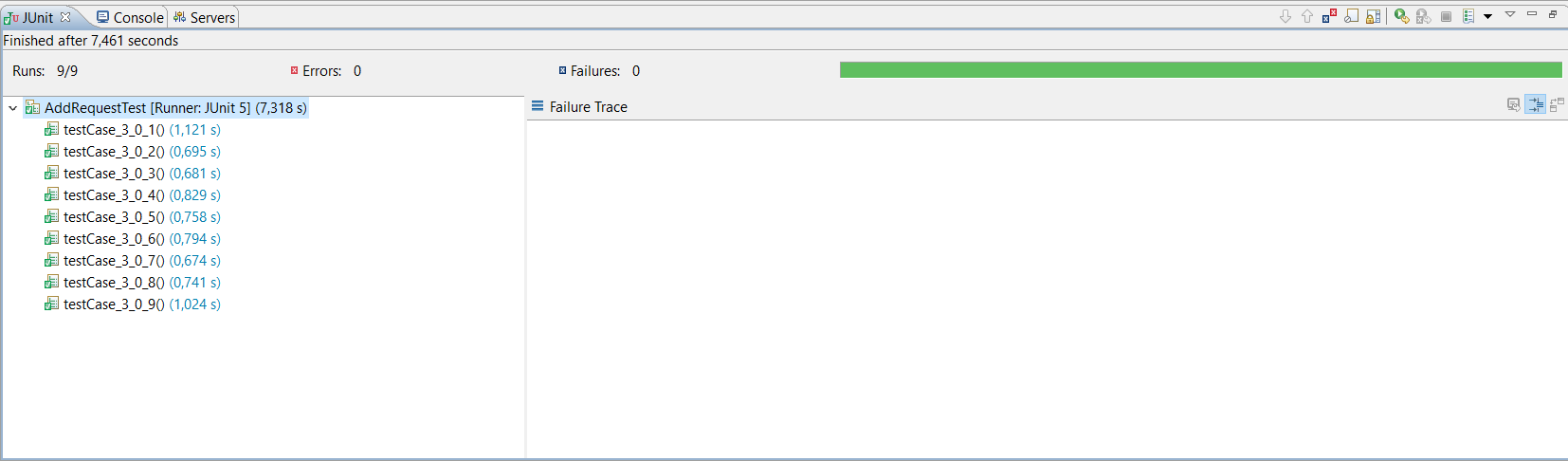
### Classe AppointmentServlet (Aggiunta)

****

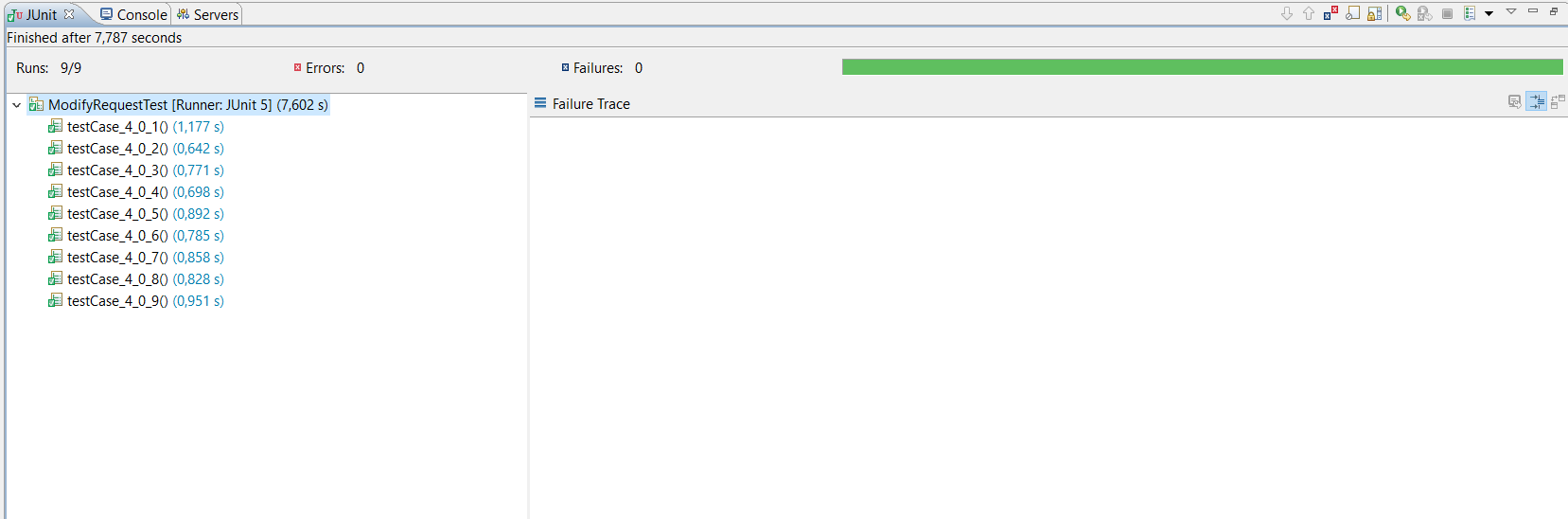
### Classe AppointmentServlet (Modifica)



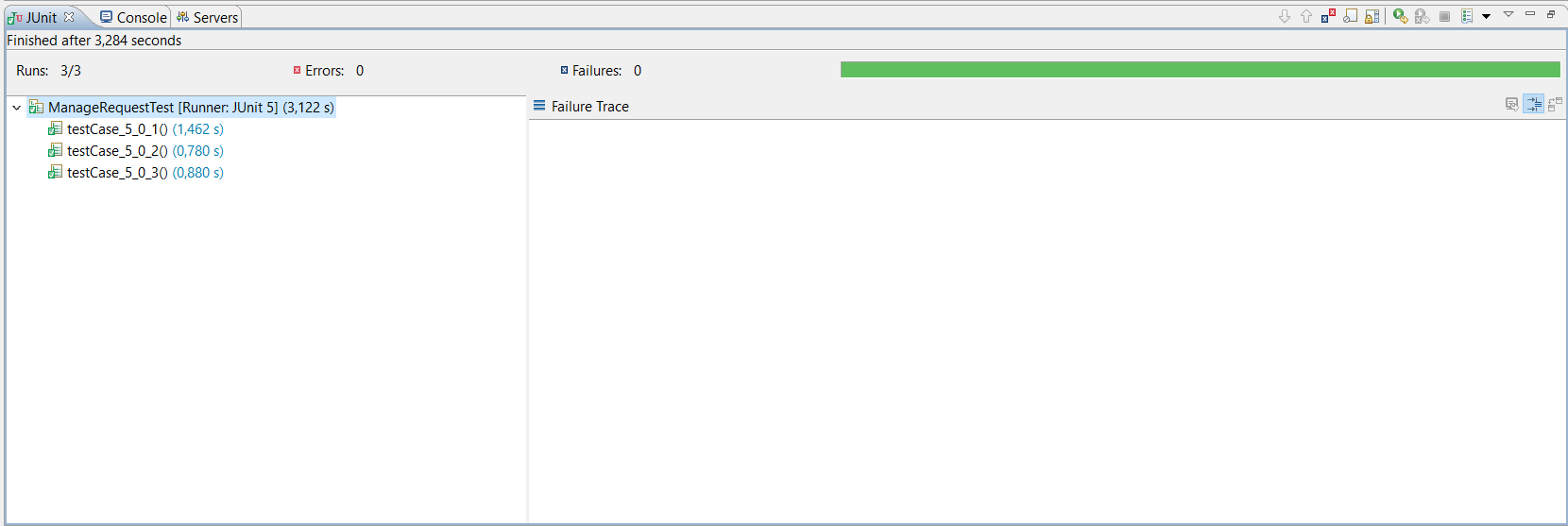
### Classe RequestServlet(Aggiunta)



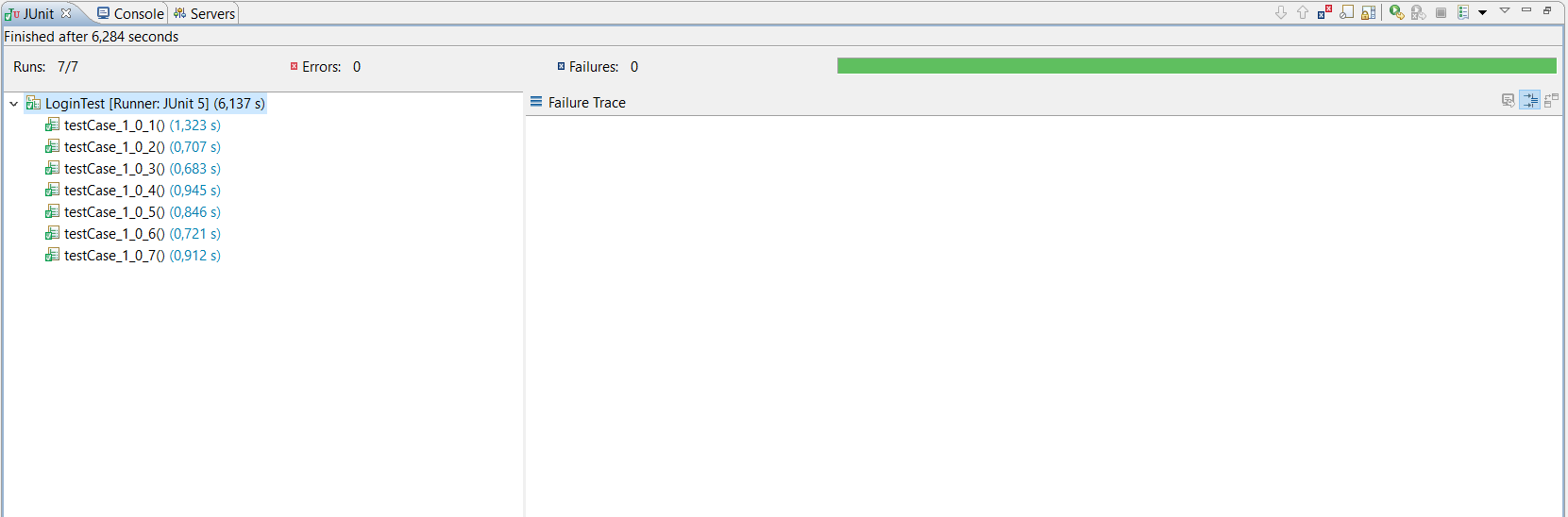
### Classe RequestServlet(Modifica)



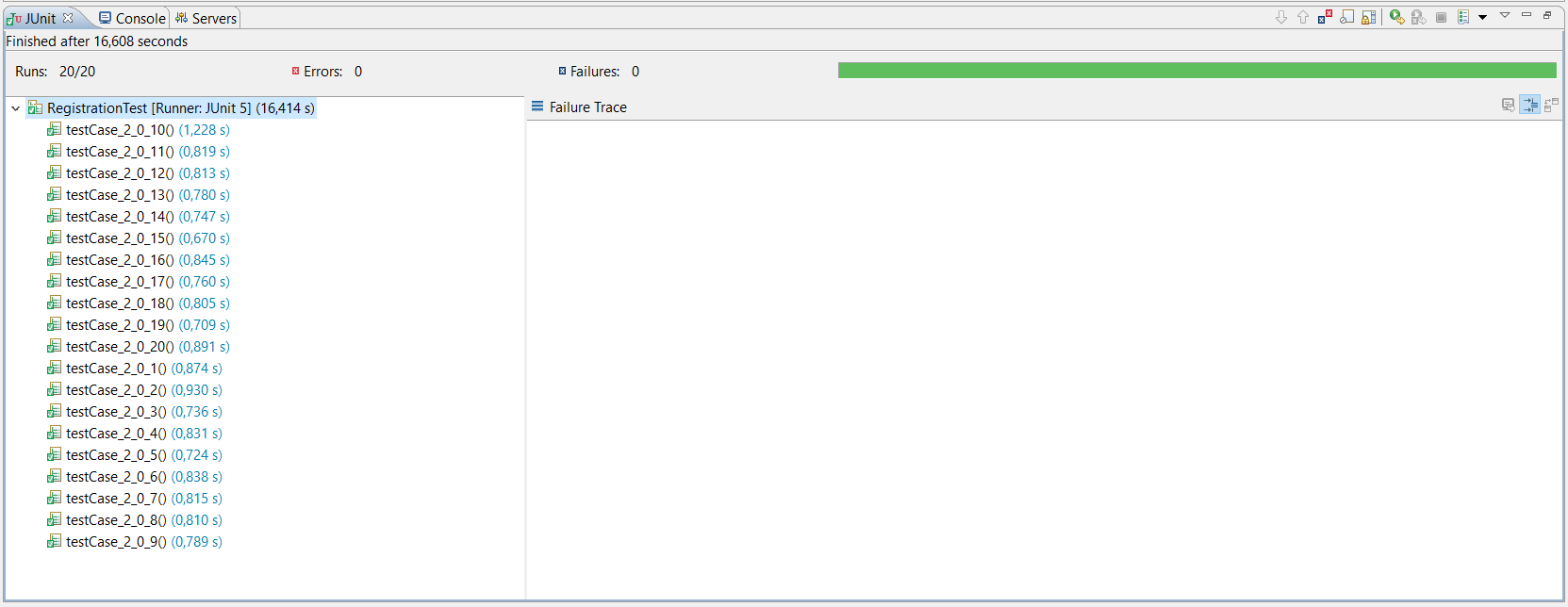
### Classe RequestServlet(Manage)



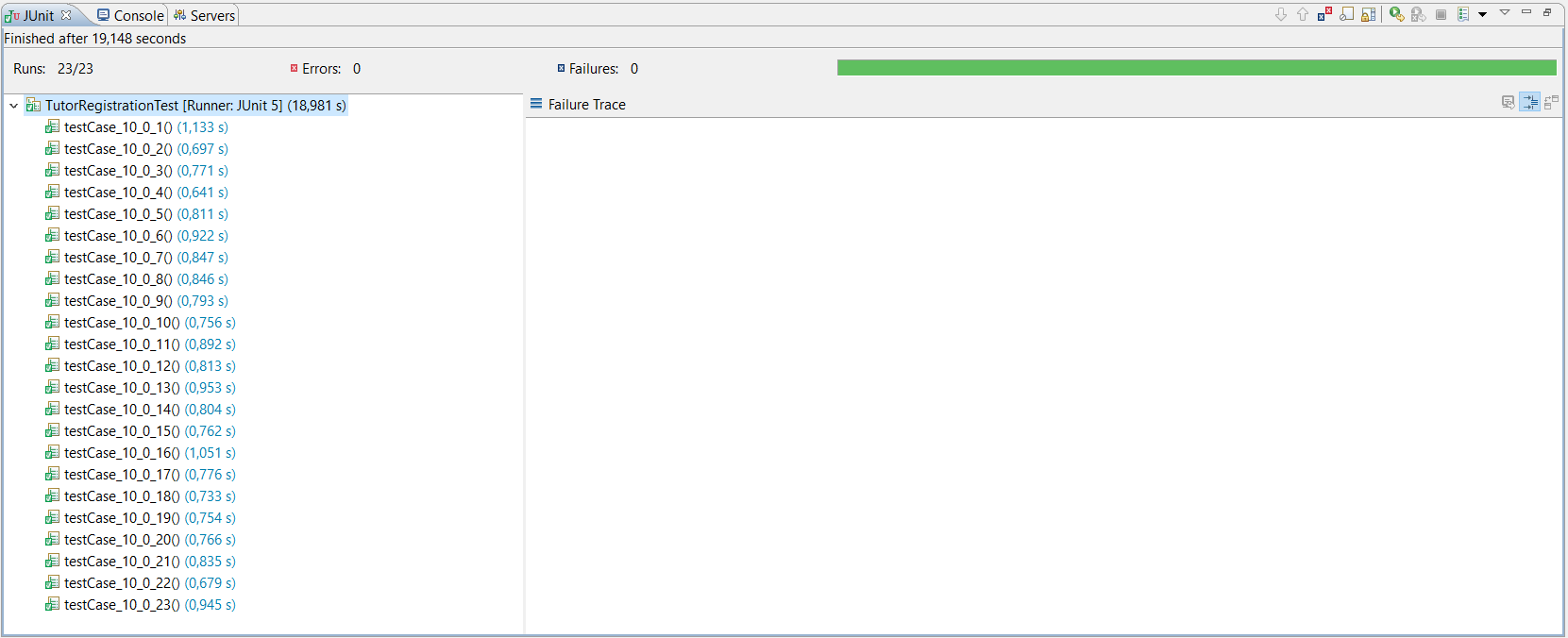
### Classe LoginServlet

****

### Classe RegistrationServlet



### Classe RegistrationServlet (Tutor)



## COVERAGE:

# 4. Riepilogo del testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Numero di componenti testate** | **Numero di errori trovati** | **Numero di errori corretti** | **Numero di componenti non testate** |
| **Model** | 20 | 0 | 0 | 2 |
| **Control** | 5 | 0 | 0 | 8 |

# 5. Glossario

● RAD (Requirement Analysis Document): Documento di Raccolta e analisi dei Requisiti; contiene l’elenco dei requisiti funzionali e non funzionali individuati in fase di individuazione degli stessi e la loro analisi sotto forma di scenari e casi d'uso. I mock-up mostrano una possibile implementazione dell’interfaccia del sistema.

● SDD (System Design Document): Documento all’interno del quale viene riportata la progettazione del sistema come risultato di una prima fase di modellazione: contiene una suddivisione ad alto livello del sistema nei sottosistemi che lo comporranno.

● ODD (Object Design Document): Documento che riporta e analizza gli oggetti che compongono il sistema analizzando le componenti a più basso livello, riportandole così come saranno implementate.

● TCD (Test Case Document): Documento che riporta i casi di test pianificati.