



Laurea Triennale in Informatica - Università di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci



TutoratoSmart

Unit Test Report

TutoratoSmart

Riferimento	
Versione	0.1
Data	01/02/2020
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Marco Delle Cave, Francesco Pagano, Manuel Pisciotta, Alessia Olivieri
Approvato da	



Revision History

Data	Versione	Cambiamenti	Autori
01/02/2020	0.1	Prima stesura	Pisciotta Manuel



Sommario

1. Introduzione.....	4
2. Riferimenti.....	4
3. Dettagli	4
3.1 Black-box Testing.....	4
3.1.1 TC_1: Autenticazione Utente	4
3.1.2 TC_2: Registrazione Studente	4
3.1.3 TC_3: Compilazione Richiesta	5
3.1.4 TC_4: Modifica Prenotazione.....	5
3.1.5 TC_5: Gestione Richiesta.....	5
3.1.6 TC_6: Conferma Appuntamento	5
3.1.7 TC_7: Modifica Appuntamento	5
3.1.8 TC_8: Aggiunta Attività Lavorativa.....	5
3.1.9 TC_9: Modifica Attività Lavorativa	6
3.1.10 TC_10: Registrazione Tutor	6
3.2 White-box Testing	6
3.2.1 Request_Management.....	6
3.2.1.1 AppointmentServlet	6
3.2.1.2 CalendarServlet	6
3.2.1.3 RequestServlet.....	6
3.2.1.4 ShowAppointmentServlet	6
3.2.1.5 ShowRequestServlet	6
3.2.1.6 Coverage.....	7
3.2.2 Tutoring_Activity_Management	7
3.2.2.1 ActivityServlet	7
3.2.2.2 ShowActivityServlet	7
3.2.2.3 ShowRegisterServlet.....	7
3.2.2.4 Coverage.....	7
3.2.3 Tutoring_Request.....	7
3.2.3.1 CalendarServlet	7
3.2.3.2 RequestServlet.....	7
3.2.3.3 ShowRequestServlet	7



3.2.3.4 Coverage.....	8
3.2.4 Tutoring_Supervision	8
3.2.4.1 ActivityServlet	8
3.2.4.2 RegistrationServlet.....	8
3.2.4.3 ShowActivityServlet	8
3.2.4.4 ShowRegisterServlet.....	8
3.2.4.5 StudentsServlet.....	8
3.2.4.6 TutorsServlet	8
3.2.4.7 Coverage.....	8
3.2.5 User.....	8
3.2.5.1 Login.....	8
3.2.5.2 Registration.....	9
3.2.5.3 Coverage.....	9
4. Conclusioni e raccomandazioni	9



1. Introduzione

Questo documento ha lo scopo di riportare i risultati dell'esecuzione dei test case di unità e di integrazione delle varie componenti del software web "TutoratoSmart"; il package Control rappresenta il secondo layer da testare nel processo di testing d'integrazione. Le classi sono state testate con l'utilizzo di JUnit e Mockito.

In particolare, sono state adottate due strategie di testing, black-box e white-box.

- **Black-Box Testing:**

Come linea guida abbiamo seguito gli use case identificati in precedenza nel RAD. Abbiamo effettuato un test per ogni test-case individuato nel documento Test Plan e specificato nel documento Test Case Specification.

- **White-Box Testing:**

Abbiamo utilizzato la tecnica del branch testing per assicurarci che il codice prodotto funzionasse correttamente e abbiamo scelto come percentuale di branch coverage delle classi di almeno il 75%.

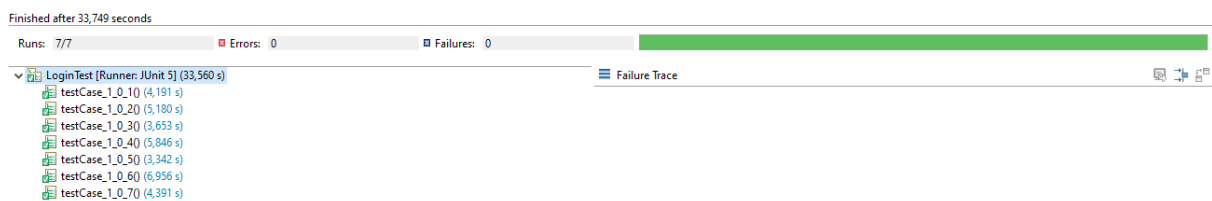
2. Riferimenti

- TS_TSR_V_0.1
- TS_TP_V_0.2
- TS_TCS_V_0.3
- TS_ITD_V_0.2

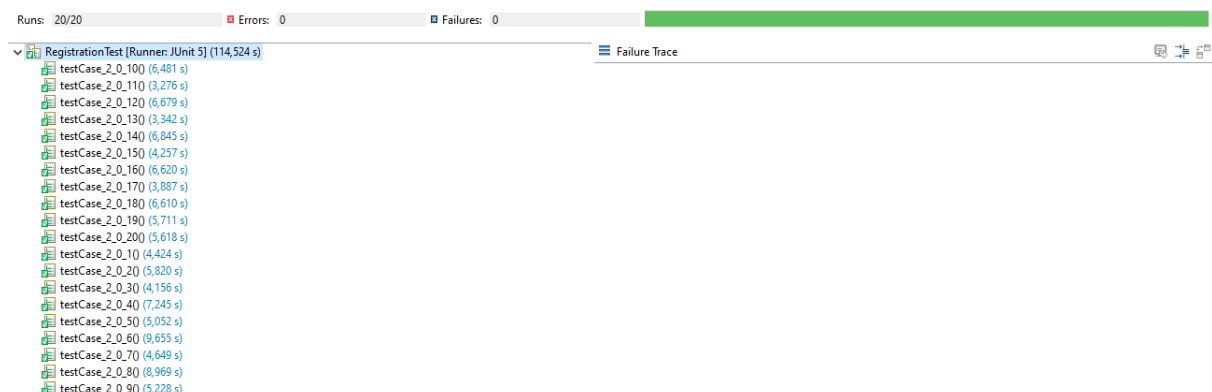
3. Dettagli

3.1 Black-box Testing

3.1.1 TC_1: Autenticazione Utente

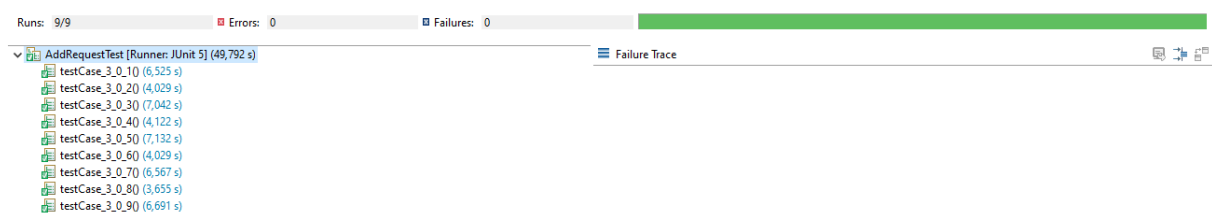


3.1.2 TC_2: Registrazione Studente

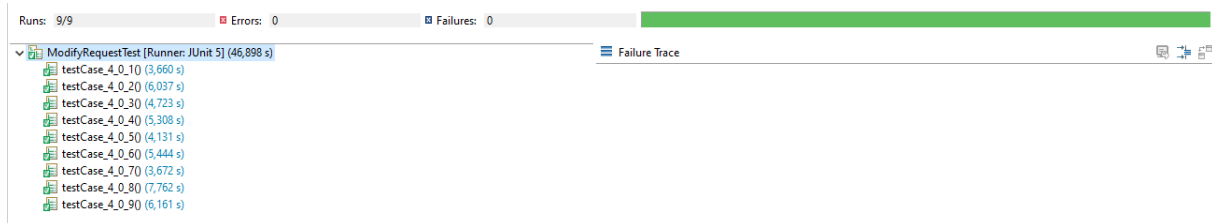




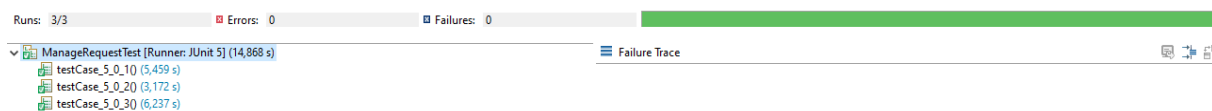
3.1.3 TC_3: Compilazione Richiesta



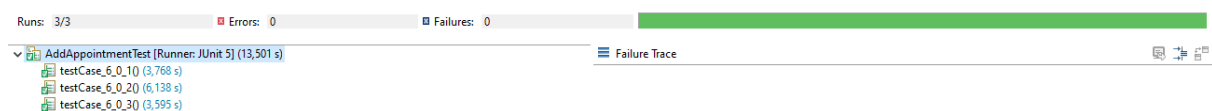
3.1.4 TC_4: Modifica Prenotazione



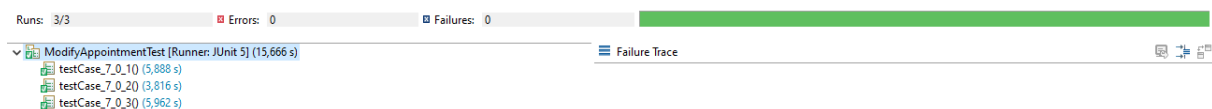
3.1.5 TC_5: Gestione Richiesta



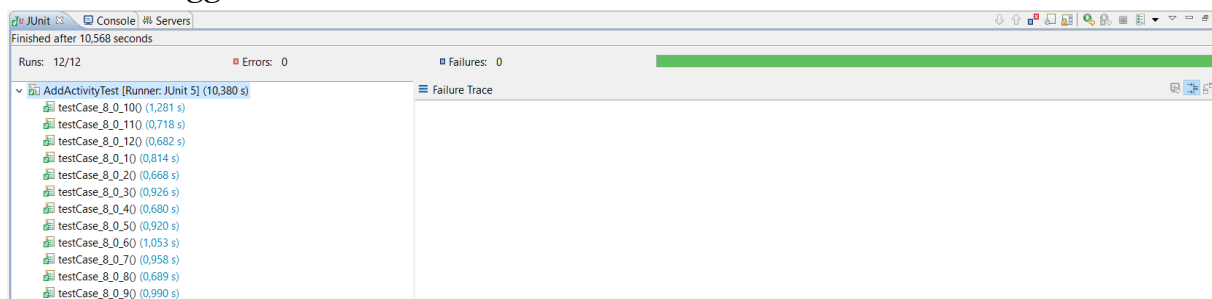
3.1.6 TC_6: Conferma Appuntamento



3.1.7 TC_7: Modifica Appuntamento

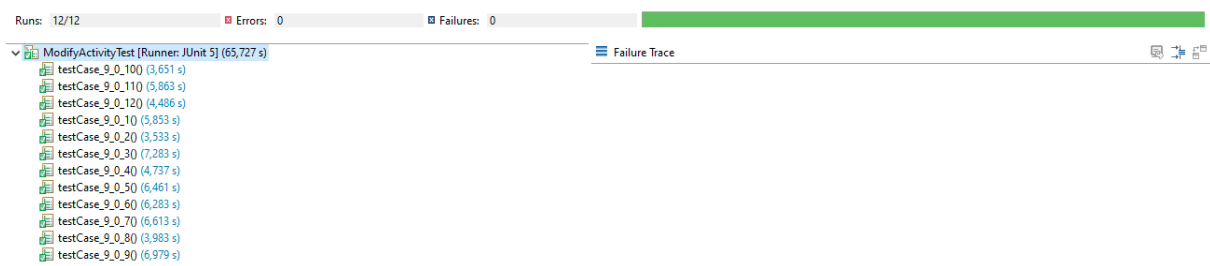


3.1.8 TC_8: Aggiunta Attività Lavorativa

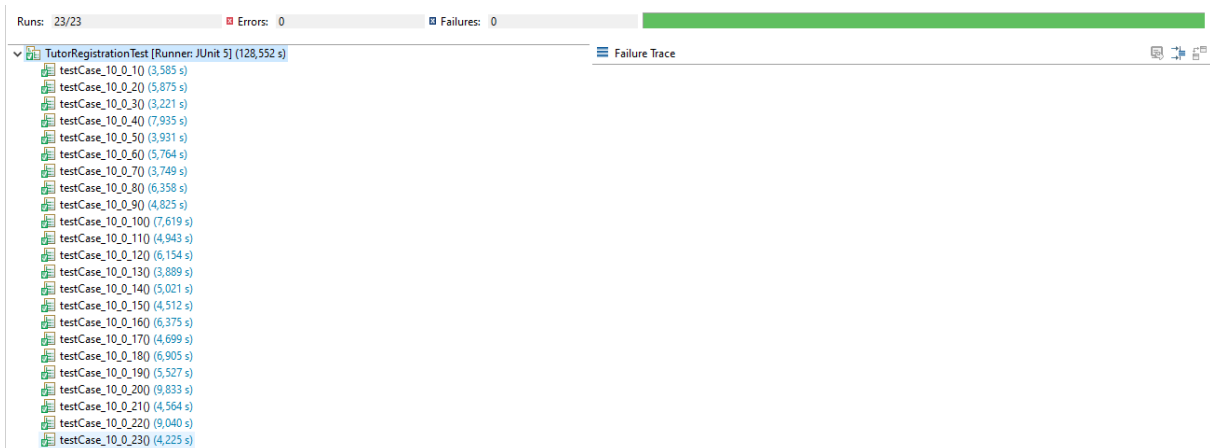




3.1.9 TC_9: Modifica Attività Lavorativa



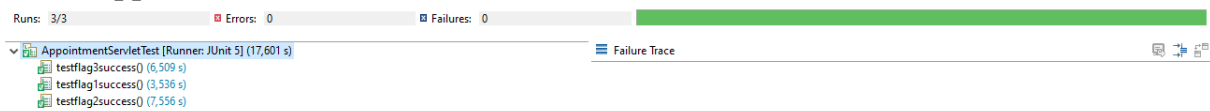
3.1.10 TC_10: Registrazione Tutor



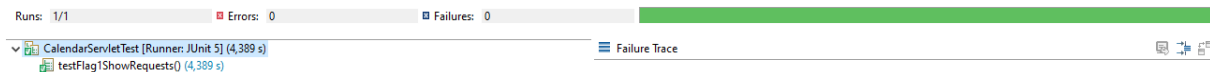
3.2 White-box Testing

3.2.1 Request_Management

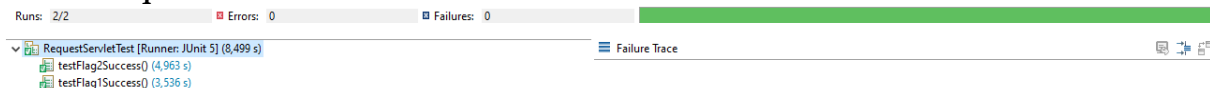
3.2.1.1 AppointmentServlet



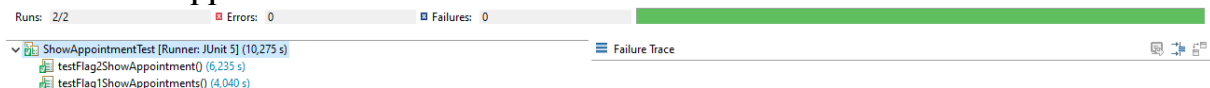
3.2.1.2 CalendarServlet



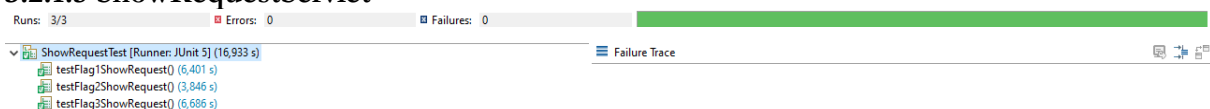
3.2.1.3 RequestServlet



3.2.1.4 ShowAppointmentServlet



3.2.1.5 ShowRequestServlet





3.2.1.6 Coverage

✓	project.Control.Request_Management	80,5 %	762	185	947
>	RequestServlet.java	76,2 %	163	51	214
>	ShowRequestServlet.java	77,5 %	134	39	173
>	AppointmentServlet.java	77,8 %	207	59	266
>	ShowAppointmentServlet.java	84,9 %	90	16	106
>	CalendarServlet.java	89,4 %	168	20	188

3.2.2 Tutoring_Activity_Management

3.2.2.1 ActivityServlet

Runs:	6/6	Errors:	0	Failures:	0
✓	ActivityServletTest [Runner: JUnit 5] (31,544 s)				
	testAjax() (8,187 s)				
	testFlag2Delete() (4,118 s)				
	testAjaxCheckModify() (6,019 s)				
	testFlag3ModifySuccess() (3,934 s)				
	testFlag1Success() (5,996 s)				
	testAjaxCheck() (3,290 s)				

3.2.2.2 ShowActivityServlet

Runs:	1/1	Errors:	0	Failures:	0
✓	ShowActivityTest [Runner: JUnit 5] (6,239 s)				
	testShowActivity() (6,239 s)				

3.2.2.3 ShowRegisterServlet

Runs:	1/1	Errors:	0	Failures:	0
✓	ShowRegisterTest [Runner: JUnit 5] (4,474 s)				
	testShowRegister() (4,474 s)				

3.2.2.4 Coverage

✓	project.Control.Tutoring_Activity_Management	82,2 %	739	160	899
>	ActivityServlet.java	80,9 %	642	152	794
>	ShowActivityServlet.java	90,0 %	36	4	40
>	ShowRegisterServlet.java	93,8 %	61	4	65

3.2.3 Tutoring_Request

3.2.3.1 CalendarServlet

Runs:	1/1	Errors:	0	Failures:	0
✓	CalendarServletTest [Runner: JUnit 5] (6,465 s)				
	testFlag1ShowRequests() (6,465 s)				

3.2.3.2 RequestServlet

Runs:	11/11	Errors:	0	Failures:	0
✓	RequestServletTest [Runner: JUnit 5] (61,353 s)				
	testFlagAjaxModifyAvailable() (4,223 s)				
	testFlag2Success() (6,825 s)				
	testErrorDayNotAvailable() (4,145 s)				
	testFlagAjaxAvailable() (5,713 s)				
	testErrorDayNotSelected() (3,421 s)				
	testErrorCommentLength() (6,946 s)				
	testFlag3Success() (4,474 s)				
	testErrorTimeNotSelected() (6,444 s)				
	testFlag1Success() (5,989 s)				
	testErrorTimeNotValid() (8,419 s)				
	testFlagAjaxNotAvailable() (4,752 s)				

3.2.3.3 ShowRequestServlet

Runs:	2/2	Errors:	0	Failures:	0
✓	ShowRequestTest [Runner: JUnit 5] (12,488 s)				
	testFlag1ShowRequests() (7,361 s)				
	testFlag2ShowRequest() (5,127 s)				



3.2.3.4 Coverage

▼	project.Control.Tutoring_Request	82,1 %	606	132	738
>	RequestServlet.java	78,6 %	411	112	523
>	ShowRequestServlet.java	85,1 %	86	15	101
>	CalendarServlet.java	95,6 %	109	5	114

3.2.4 Tutoring_Supervision

3.2.4.1 ActivityServlet

Runs:	2/2	Errors:	0	Failures:	0
▼	ActivityServletTest [Runner: JUnit 5] (11,973 s)				
	testFlag3ValidateSuccess() (4,248 s)				
	testFlag3DeleteSuccess() (7,725 s)				

3.2.4.2 RegistrationServlet

Runs:	5/5	Errors:	0	Failures:	0
▼	RegistrationServletTest [Runner: JUnit 5] (23,404 s)				
	testErrorLenghtFirstName() (4,246 s)				
	testRegistrationSuccess() (5,236 s)				
	testErrorFormatFirstName() (3,724 s)				
	testErrorFormatLastName() (5,697 s)				
	testErrorLenghtLastName() (4,501 s)				

3.2.4.3 ShowActivityServlet

Runs:	1/1	Errors:	0	Failures:	0
▼	ShowActivityTest [Runner: JUnit 5] (6,588 s)				
	testShowActivity() (6,588 s)				

3.2.4.4 ShowRegisterServlet

Runs:	1/1	Errors:	0	Failures:	0
▼	ShowRegisterTest [Runner: JUnit 5] (3,682 s)				
	testShowRegister() (3,682 s)				

3.2.4.5 StudentsServlet

Runs:	1/1	Errors:	0	Failures:	0
▼	StudentsServletTest [Runner: JUnit 5] (5,712 s)				
	testSearchStudents() (5,712 s)				

3.2.4.6 TutorsServlet

Runs:	1/1	Errors:	0	Failures:	0
▼	TutorsServletTest [Runner: JUnit 5] (3,302 s)				
	testSearchStudents() (3,302 s)				

3.2.4.7 Coverage

▼	project.Control.Tutoring_Supervision	81,8 %	585	130	715
>	RegistrationServlet.java	75,9 %	233	74	307
>	ActivityServlet.java	82,6 %	133	28	161
>	TutorsServlet.java	84,8 %	56	10	66
>	StudentsServlet.java	86,5 %	64	10	74
>	ShowActivityServlet.java	89,7 %	35	4	39
>	ShowRegisterServlet.java	94,1 %	64	4	68

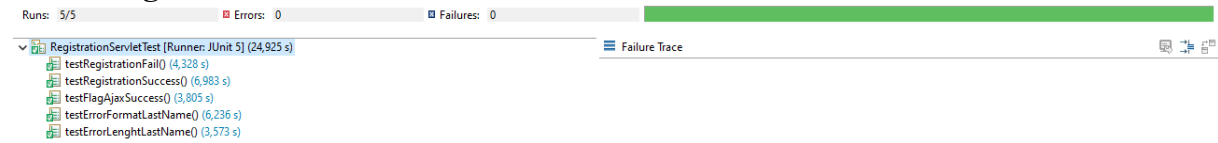
3.2.5 User

3.2.5.1 Login

Runs:	5/5	Errors:	0	Failures:	0
▼	LoginTest [Runner: JUnit 5] (27,227 s)				
	testErrorLenghtEmail() (6,300 s)				
	testLoginSuccess() (4,109 s)				
	testErrorFormatEmail() (5,753 s)				
	testErrorFormatPassword() (4,300 s)				
	testErrorLenghtPassword() (6,765 s)				



3.2.5.2 Registration



3.2.5.3 Coverage

▼	project.Control.User	76,0 %	383	121	504
>	LogoutServlet.java	0,0 %	0	28	28
>	RegistrationServlet.java	76,3 %	251	78	329
>	LoginServlet.java	89,8 %	132	15	147

4. Conclusioni e raccomandazioni

I risultati ottenuti mostrano che non ci sono errori. Si può passare quindi alla fase di testing successiva, il testing di sistema.