



# TEST SUMMARY REPORT

Nefrapp

---

Riferimento	Nefrapp_RAD_Vers.1.8, Nefrapp_TCS_Vers.0.7, Nefrapp_TP_Vers.1.1, Nefrapp_TER.Vers.1.0
Versione	1.0
Data	14/01/2020
Destinatario	Professoressa F.Ferrucci
Proposto da	Eugenio Corbisiero, Sara Corrente, Silvio Di Martino, Antonio Donnarumma, Luca Esposito, Matteo Falco, Domenico Musone, Davide Benedetto Strianese.
Approvato da	Antonio Fasulo, Francesco Garofalo



### Project Manager

Nome	Cognome	Matricola
Francesco	Garofalo	0522500615
Antonio	Fasulo	0522500627

### Partecipanti

Nome	Cognome	Matricola
Eugenio	Corbisiero	0512105449
Sara	Corrente	0512105695
Silvio	Di Martino	0512105629
Antonio	Donnarumma	0512105083
Luca	Esposito	0512105123
Matteo	Falco	0512109201
Domenico	Musone	0512105689
Davide Benedetto	Strianese	0512105257



### Revision History

Data	Versione	Descrizione	Autori
12/01/2020	0.5	Stesura iniziale documento	Matteo Falco
13/01/2020	0.9	Test di usabilità	Sara Corrente
14/01/2020	1.0	Performance Test	Antonio Donnarumma



## Sommario

<b>1. Introduzione</b>	<b>5</b>
<b>2. Testing di Unità</b>	<b>6</b>
2.1 Classi testate	6
2.2 Riassunto dei risultati del testing di unità	7
2.3 Coverage del testing di unità	8
<b>3. Testing di Sistema</b>	<b>9</b>
3.1 Test Case eseguiti	9
3.2 Riassunto dei risultati del testing di sistema	10
<b>4. Testing di Integrazione</b>	<b>11</b>
<b>5. Testing di Usabilità</b>	<b>12</b>
<b>6. Testing della Performance</b>	<b>13</b>
6.1 Stress test	13



## 1. Introduzione

Questo documento ha lo scopo di fornire un riassunto dei risultati del testing eseguito sul progetto. Precisamente, saranno riportati i risultati di:

- Testing di unità eseguito con JUnit
- Testing di sistema eseguito con Katalon
- Testing di integrazione eseguito con Mockito
- Testing di usabilità
- Test di performance



## 2. Testing di Unità

Di seguito sono riportati i risultati del testing di Unità, effettuato con JUnit sulle classi Java di Model, Control e Bean.

### 2.1 Classi testate

Il testing di unità si è concentrato sulle seguenti classi:

- Package Bean:
  - Amministratore
  - AnnuncioCompleto
  - AnnuncioProxy
  - Medico
  - MessaggioCompleto
  - MessaggioProxy
  - Paziente
  - PianoTerapeuticoi
  - SchedaParametri
  - Utente
- Package Model:
  - AmministratoreModel
  - AnnuncioModel
  - MedicoModel
  - MessaggioModel
  - PazienteModel
  - PianoTerapeuticoModel
  - SchedaParametriModel
  - UtenteModel



- Package Control:
  - GestioneAccesso
  - GestioneAmministratore
  - GestioneAnnunci
  - GestioneComunicazione
  - GestioneMessaggi
  - GestioneParametri
  - GestionePaziente
  - GestionePianoTerapeutico
  - GestioneRegistrazione

## 2.2 Riassunto dei risultati del testing di unità

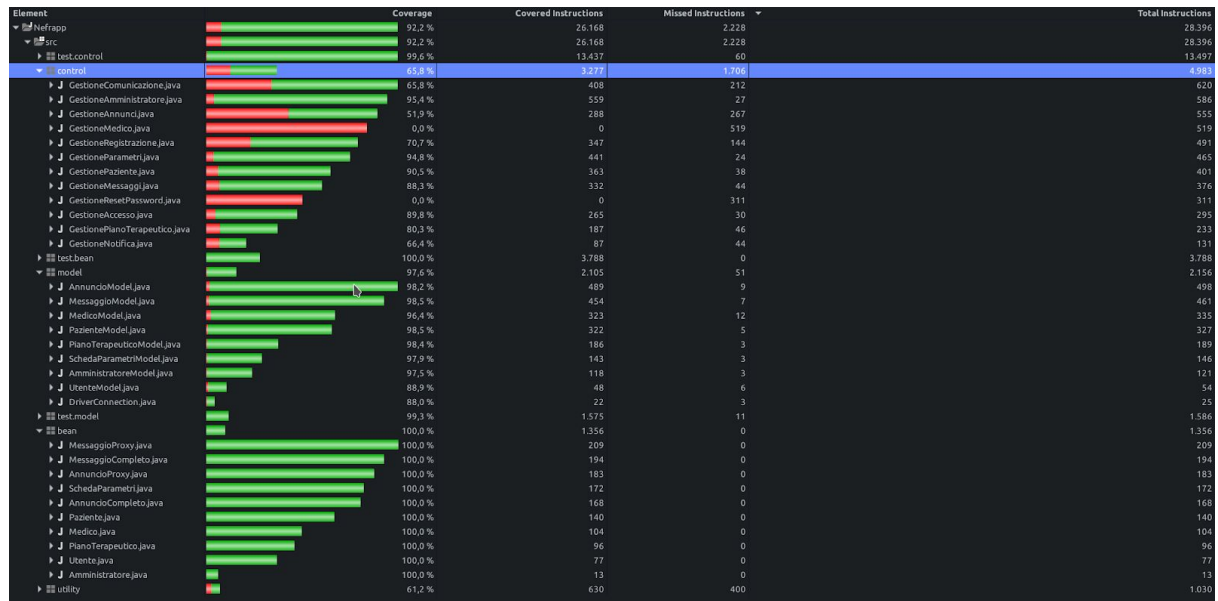
	Componenti Testate	Errori Trovati	Errori Corretti	Componenti non testate
<b>Bean</b>	10	0	0	0
<b>Model</b>	8	0	0	0
<b>Control</b>	10	4	4	2

Nelle classi di Bean e Model non sono stati riscontrati errori, e sono state testate tutte le componenti dei package.

Nelle classi Control, sono stati riscontrati 4 errori, e sono stati tutti documentati e corretti, come descritto nel Test Unit Report. Non sono state testate 2 componenti, poiché era già stato raggiunto un coverage ritenuto soddisfacente.



## 2.3 Coverage del testing di unità



Il testing di unità ha raggiunto un coverage complessivo del 92,2%.





## 3. Testing di Sistema

Di seguito sono riportati i risultati del testing di sistema, effettuato con Katalon Studio. Il testing di sistema viene effettuato seguendo i test case pianificati nel Test Case Specification Document.

### 3.1 Test Case eseguiti

Sono stati eseguiti i seguenti Test Case:

- Gestione Medico
  - TC\_GM\_1
  - TC\_GM\_2
  - TC\_GM\_3
  - TC\_GM\_4
  - TC\_GM\_5
  - TC\_GM\_6
  - TC\_GM\_7
  - TC\_GM\_8
  - TC\_GM\_9
  - TC\_GM\_10
  - TC\_GM\_11
  - TC\_GM\_12
- Gestione Paziente
  - TC\_GP\_1
  - TC\_GP\_2
  - TC\_GP\_3
  - TC\_GP\_5
  - TC\_GP\_6
  - TC\_GP\_7
  - TC\_GP\_8
  - TC\_GP\_9
  - TC\_GP\_10



- Gestione Amministratore
  - TC\_GA\_1
  - TC\_GA\_2
  - TC\_GA\_3
  - TC\_GA\_4
  - TC\_GA\_5
  - TC\_GA\_6
  - TC\_GA\_7
  - TC\_GA\_8

Non sono stati eseguiti i seguenti test case:

- Gestione Paziente
  - TC\_GP\_4

### 3.2 Riassunto dei risultati del testing di sistema

	TC eseguiti	Errori Trovati	Errori Corretti	TC non eseguiti
<b>Gestione Medico</b>	12	5	4	0
<b>Gestione Paziente</b>	9	0	0	1
<b>Gestione Amministratore</b>	8	1	0	0

Nei test case eseguiti per la Gestione Medico sono stati riscontrati 5 errori: 4 errori sono stati corretti, mentre la correzione di 1 errore, rilevato in TC\_GM\_11, è stata riservata alle fasi future di sviluppo.

Nei test case eseguiti per la Gestione Paziente non sono stati riscontrati errori.

Nei test case eseguiti per la Gestione Amministratore è stato riscontrato 1 errore in TC\_GA\_7, la cui correzione è stata riservata alle fasi future di sviluppo.

L'analisi completa dei risultati del testing è riportata nel Test Execution Report.



## 4. Testing di Integrazione

Questa fase ha lo scopo di integrare tutte le componenti di una funzionalità al fine di testarle nel complesso utilizzando una strategia Bottom-Up, che mira a ridurre le dipendenze fra funzionalità differenti e a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.

Il software utilizzato è Mockito.

L'approccio Bottom-up è stato applicato testando singolarmente le componenti a basso livello nel testing di unità; successivamente sono state testate le servlet e, infine, sono state testate le JSP con il testing di sistema tramite Katalon.



## 5. Testing di Usabilità

Per testare l'usabilità è stato reso fruibile il sistema al cliente e ad alcuni utenti in modo tale da ricoprire tutti gli attori individuati nella fase di analisi.

Agli utenti è stato poi sottoposto un form da compilare dando la possibilità di esprimere la loro esperienza con l'applicazione.

I risultati del test hanno evidenziato che il sistema, nonostante rispecchi appieno le aspettative del cliente, potrebbe risultare leggermente ostico da utilizzare per coloro che non sono avvezzi all'utilizzo della tecnologia.

Particolari difficoltà sono state riscontrate dagli utenti nelle seguenti funzionalità:

Attore	Funzionalità
Medico	Nessuna
Paziente	Invio messaggio e allegati al medico
Amministratore	Registrazione medico, Modifica account medico

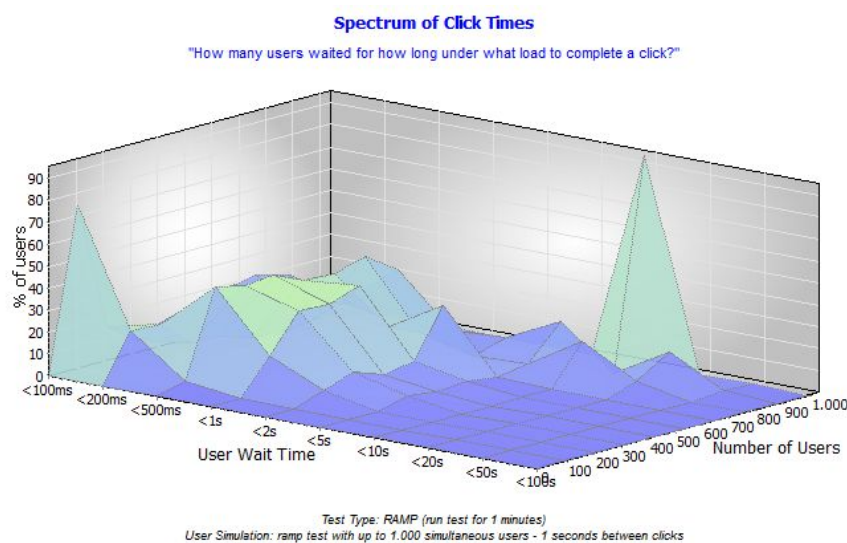
In sintesi, il sistema risulta fruibile per la maggior parte dei Requisiti Funzionali individuati eccetto quelli riportati nella tabella.

## 6. Testing della Performance

Attraverso il software Webserver Stress Tool sono stati effettuati degli scenari standard di stress tool.

### 6.1 Stress test

Il test effettuato ha simulato l'accesso con 400 utenti simulati alla piattaforma con tipologia RAMP (tipologia di test che vede l'aumentare del carico di richieste da un solo utente utente al numero specificato di utenti (400)).



I risultati ottenuti dal singolo test mostrano come il tempo di caricamento della pagina peggiora (come previsto) all'aumentare del numero di utenti. La piattaforma non è riuscita a soddisfare solo lo 0,08% delle richieste.