

Università degli Studi di Salerno

Corso di Ingegneria del Software

HairBeautyNow

TIR: Test Incident Report

Versione 1.0



Data: 11/02/2025

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Aniello Cirillo	0512117775
Christian Pio Lanziero	0512117931

Scritto da:	
-------------	--

Revision History

Data	Versione	Descrizione	Autore
14/12/2024	0.1	Creazione del documento	Tutto il team
18/12/2024	0.2	Inserimento dell'errore	Cirillo Aniello
11/02/2024	1.0	Revisione generale	Tutto il team

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

Indice

1. INTRODUZIONE.....	3
2. AMBIENTE DI ESECUZIONE.....	3
3. TEST INCIDENT REPORT.....	3
4. GLOSSARIO.....	7

1. INTRODUZIONE

Questo documento descrive una problematica riscontrata durante la fase di testing relativa all'operazione di prenotazione e tutto ciò che deriva da essa(seleziona salone, selezione professionista, seleziona orario, completa pagamento). In particolare per l'operazione di completa pagamento inizialmente ha prodotto un esito negativo.

2. AMBIENTE DI ESECUZIONE

Per il testing relativo all'operazione di Aggiunta Proiezione, sono stati utilizzati i seguenti strumenti:

- **JUnit**: per eseguire test di unità sulle funzionalità principali dell'operazione.
- **Mockito**: per simulare e verificare le interazioni tra i componenti del sistema, garantendo l'accuratezza dei flussi di integrazione.
- **Selenium IDE**: Per eseguire i test di sistema, simulando l'interazione dell'utente con l'interfaccia grafica.

Questi strumenti hanno permesso di isolare le unità di codice, validare l'interazione tra i componenti del sistema e verificare il comportamento complessivo dell'applicazione.

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

3. TEST INCIDENT REPORT

3.1 Completa Prenotazione TI01

FAILURE: Durante l'esecuzione dei test di completamento prenotazione è stato identificato un problema critico legato all'aggiunta di una prenotazione nel database. Nello specifico ci siamo resi conto che il sistema passava dati di tipo errato(Tra LocalDateTime e Date) tra il service ed il dao e di conseguenza questo generava errore sul metodo addPrenotazione.

Test Incident ID	TI01	Test Case ID	TC02.6
Input			
Categoria	Valore della scelta		
Data prenotazione	2025-02-15		
Orario Prenotazione	10:00-10:30		
Professionista ID	1		
Servizio	Taglio di capelli		
Prezzo	30.0		
Utente	Username		
Metodo di pagamento	Visa		
Numero di carta	1234567890123456		
CVV	123		

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

Scadenza di carta	12/25
Indirizzo	Via Roma 1
Oracolo	
Prenotazione aggiunta con successo	

Output attuale:

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

testDoPost()

Test doPost - Dati coinvolti:

Data prenotazione: 2025-02-18

Orario prenotazione: 10:00-10:30

Professionista ID: 1

Servizio: Taglio di capelli

Prezzo: 30.0

Utente: username

Metodo di pagamento: visa

Numero carta: 1234567890123456

CVV: 123

Scadenza carta: 12/25

Indirizzo: Via Roma 1

Verifica che il metodo di pagamento sia stato aggiunto...

Verifica che la prenotazione sia stata aggiunta...

```
prenotazioneService.addPrenotazione(
```

```
);
```

```
-> at integration.GestionePrenotazioni.CompletaPrenotazioneServletIntegrationTest.testDoPost(CompletaPrenotazioneServletIntegrationTest.java:134)
```

Actually, there were zero interactions with this mock.

Wanted but not invoked:

```
prenotazioneService.addPrenotazione(
```

```
);
```

```
-> at integration.GestionePrenotazioni.CompletaPrenotazioneServletIntegrationTest.testDoPost(CompletaPrenotazioneServletIntegrationTest.java:134)
```

Actually, there were zero interactions with this mock.

```
at integration.GestionePrenotazioni.CompletaPrenotazioneServletIntegrationTest.testDoPost(CompletaPrenotazioneServletIntegrationTest.java:134)
```

```
at java.base/java.lang.reflect.Method.invoke(Method.java:580)
```

```
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
```

```
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
```

Successful: Il problema è stato risolto mediante una modificando la servlet e passando direttamente un oggetto di tipo LocalDateTime in modo da non generare più alcuna eccezione. In questo modo il sistema riesce a creare la prenotazione in maniera corretta.

Test Incident ID	TI01	Test Case ID	TC02.6
------------------	------	--------------	--------

HairBeautyNow	Ingegneria del Software	Pagina 6 di 9
---------------	-------------------------	---------------

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

Input	
Categoria	Valore della scelta
Data prenotazione	2025-02-15
Orario Prenotazione	10:00-10:30
Professionista ID	1
Servizio	Taglio di capelli
Prezzo	30.0
Utente	Username
Metodo di pagamento	Visa
Numero di carta	1234567890123456
CVV	123
Scadenza di carta	12/25
Indirizzo	Via Roma 1
Oracolo	
Prenotazione aggiunta con successo	

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

Output attuale:

testDoPost()

Test doPost - Dati coinvolti:

Data prenotazione: 2025-02-18

Orario prenotazione: 10:00-10:30

Professionista ID: 1

Servizio: Taglio di capelli

Prezzo: 30.0

Utente: username

Metodo di pagamento: visa

Numero carta: 1234567890123456

CVV: 123

Scadenza carta: 12/25

Indirizzo: Via Roma 1

Pagamento con VISA effettuato per l'importo di: 30.0

Verifica che il metodo di pagamento sia stato aggiunto...

Verifica che la prenotazione sia stata aggiunta...

Prenotazione inserita: Prenotazione{id=0, servizioName='Taglio di capelli', professionalId=1, data=2025-02-18T10:00, username='username', prezzo=30.0}

Prenotazione inserita: Prenotazione{id=0, servizioName='Taglio di capelli', professionalId=1, data=2025-02-18T10:00, username='username', prezzo=30.0}

Verifica che la fascia oraria sia stata aggiornata...

Verifica che la risposta sia stata reindirizzata a index.jsp...

4. GLOSSARIO

Utente Generale	Rappresenta un utente non autenticato nel sistema, sono presenti quindi restrizioni nell'utilizzo di quest'ultimo.
Utente Acquirente	Rappresenta un utente autenticato nel sistema, che quindi puo' sfruttare le funzionalità riguardanti le prenotazioni e lo storico degli ordini.
Utente Gestore Sede	Rappresenta un utente registrato con il compito di gestire una singola sede dell'intera catena, esso può gestire le prenotazioni e i professionisti in salone.

Progetto: HairBeautyNow	Versione: 1.0
Documento: Test Incident Report	Data: 11/02/2025

Utente Gestore Catena	Rappresenta un utente registrato, che gestisce l'intera catena HairBeautyNow, ha il compito di amministrare le sedi, gli Utenti Gestori Sede e di creare e attivare Promozioni in tutta la catena.
Prenotazione	Rappresenta l'acquisto di un servizio con relative informazioni riguardo la sede, il professionista scelto e il servizio.
Franchise	Un sistema di collaborazione tra un produttore e un distributore, ove il primo cede al secondo la facoltà di distribuire il servizio con alcune definizioni contrattuali.