



Test Results Document - HeavyRoute

Versione 1.0

Informazioni Generali

Corso di Laurea:	Informatica
Università:	Università degli Studi di Salerno (UNISA)
Docente:	Chiar.mo Prof. DE LUCIA Andrea
Data:	13/01/2026
Anno Accademico:	2025/2026

Membri Del Gruppo

MANFREDINI Umberto	Matricola 0512119797
MANZO Ugo	Matricola 0512119071 (<i>Coordinatore</i>)
ROMANO Pino Fiorello	Matricola 0512120259

Revision History

Data	Versione	Descrizione	Autore
13/01/2026	1.0	Creazione Documento Test Results	Pino Fiorello Romano

Indice

1	Introduzione	3
2	Storico dei Tentativi di Esecuzione	3
3	Analisi dei Fallimenti e Correzioni	3
3.1	TC-CORE-03: Approvazione Richiesta	4
3.2	TC-CORE-04: Assegnazione Risorse	4
3.3	TC-CORE-07: Validazione Rotta (Approvazione)	4
3.4	TC-CORE-08: Validazione Rotta (Rifiuto)	5
4	Esito Finale dei Test	6
5	Log di Build Finale (Maven)	7
6	Conclusioni	7

1. Introduzione

Il presente documento riporta gli esiti dell'esecuzione dei test pianificati nel *Test Plan Document* (v1.1) e specificati nel dettaglio nel *Test Cases Document* (v1.1).

L'obiettivo è certificare la qualità del sottosistema **Core Business** di Heavy Route, verificando la corretta implementazione dei requisiti funzionali (creazione richieste, pianificazione viaggi, assegnazione risorse) e dei requisiti di sicurezza (RBAC).

Durante la fase di validazione, sono stati riscontrati e risolti diversi difetti software grazie all'analisi dei log di fallimento dei test automatici. Il documento traccia quindi non solo l'esito finale, ma anche lo sforzo di correzione (bug fixing) effettuato.

2. Storico dei Tentativi di Esecuzione

Durante la fase di sviluppo e integrazione continua (CI), alcuni casi di test critici hanno richiesto iterazioni multiple prima di raggiungere il successo. Di seguito si riporta il numero di esecuzioni necessarie per stabilizzare la build.

ID Test	Tentativi Totali	Statistiche Esecuzione
TC-CORE-03	4	3 Falliti, 1 Successo
TC-CORE-04	3	2 Falliti, 1 Successo
TC-CORE-07	3	2 Falliti, 1 Successo
TC-CORE-08	3	2 Falliti, 1 Successo

3. Analisi dei Fallimenti e Correzioni

In questa sezione vengono dettagliati i motivi dei fallimenti iniziali per i test case critici e le azioni correttive intraprese sul codice sorgente o sulla logica di test.

3.1. TC-CORE-03: Approvazione Richiesta

Errore Riscontrato

Errore: `AssertionError: Expected WAITING_VALIDATION but was IN_PLANNING`

Causa: Disallineamento tra il Test (che si aspettava il vecchio stato di validazione) e l'implementazione del Service (che impostava direttamente lo stato di pianificazione). Inoltre, un errore nell'uso dei Matcher di Mockito causava fallimenti nella verifica del metodo `save()`.

Soluzione Applicata

Fix: È stato aggiornato il Test per riflettere il nuovo flusso di stati (`IN_PLANNING`). È stato introdotto l'uso di `ArgumentCaptor` per intercettare correttamente l'oggetto salvato e validarne i campi.

3.2. TC-CORE-04: Assegnazione Risorse

Errore Riscontrato

Errore: `AssertionError: Expected CONFIRMED but was WAITING_VALIDATION`

Causa: Il test non considerava che, secondo la nuova logica di business, ogni modifica alle risorse (autista/veicolo) richiede una ri-validazione da parte del Traffic Coordinator, riportando lo stato indietro a `WAITING_VALIDATION`.

Soluzione Applicata

Fix: Aggiornamento dell'oracolo del test per accettare lo stato `WAITING_VALIDATION` come esito corretto dell'assegnazione risorse.

3.3. TC-CORE-07: Validazione Rotta (Approvazione)

Errore Riscontrato

Errore: `AssertionError: Expected IN_PLANNING but was CONFIRMED`

Causa: Errore nella definizione del flusso di stati nel test case. L'approvazione della rotta deve rendere il viaggio immediatamente confermato, non riportarlo in pianificazione.

Soluzione Applicata

Fix: Correzione dell'aspettativa nel test case unitario e di integrazione.

3.4. TC-CORE-08: Validazione Rotta (Rifiuto)

Errore Ricontrato

Errore: Wanted but not invoked: `tripRepository.delete(...)`

Causa: Mancata implementazione (Missing Feature). Il metodo di validazione nel Service gestiva il rifiuto solo tramite log, senza eseguire la cancellazione del viaggio richiesta dai requisiti.

Soluzione Applicata

Fix: Implementazione della logica mancante nel `TripServiceImpl`: aggiunta chiamata a `notificationService.send()` e `tripRepository.delete()` nel ramo `else` della validazione.

4. Esito Finale dei Test

A seguito delle correzioni descritte nella Sezione 3, l'intera suite di test è stata rieseguita con successo. Di seguito la matrice finale degli esiti.

ID Test	Descrizione Scenario	Categoria	Esito Finale
TC-SYS-01	Integrità del Sistema	System	PASS
TC-AUTH-01	Sicurezza - Registrazione	Security	PASS
TC-CORE-01	Creazione Richiesta Valida	Integration	PASS
TC-CORE-02.A	Rifiuto Peso Negativo	Validation	PASS
TC-CORE-02.B	Rifiuto Data Passata	Validation	PASS
TC-CORE-03	Approvazione Viaggio	Business	PASS
TC-CORE-04	Assegnazione Risorse (Happy Path)	Business	PASS
TC-CORE-05	Errore Capacità Insufficiente	Business	PASS
TC-CORE-06	Blocco Ruolo Non Autorizzato	Security	PASS
TC-CORE-07	Approvazione Rotta (Coordinator)	Business	PASS
TC-CORE-08	Rifiuto Rotta (Coordinator)	Business	PASS

5. Log di Build Finale (Maven)

Di seguito l'output della console che conferma il successo della build finale:

```
[INFO] -----  
[INFO] T E S T S  
[INFO] -----  
[INFO] Running TransportRequestControllerTest  
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.45 s  
[INFO] Running TC-ST-02-SuiteLogic  
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 1.12 s  
[INFO] Running HeavyrouteApplicationTests  
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 2.30 s  
[INFO] Running TC-INTEGRATION-Suite  
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 1.05 s  
[INFO]  
[INFO] Results:  
[INFO]  
[INFO] Tests run: 16, Failures: 0, Errors: 0, Skipped: 0  
[INFO]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----
```

6. Conclusioni

La fase di testing ha evidenziato disallineamenti iniziali tra le specifiche dei test e l'evoluzione della logica di business. Grazie all'analisi puntuale dei log di errore e al refactoring del codice (sia di produzione che di test), è stato possibile raggiungere una copertura funzionale completa e corretta.

Il sistema è ora ritenuto stabile per le funzionalità testate.