



Laurea Triennale in Informatica

Università di Salerno

Corso di Ingegneria del Software

Prof. Andrea De Lucia



Progetto EasyDrive Test Case Specification

Versione 0.1

Data: 21/12/2025

Coordinatore del progetto:

| Nome | Matricola |
|-----------------------|------------|
| Mario Mascheri | 0512120157 |
| Claudio Brizio | 0512119716 |

Partecipanti:

| Nome | Matricola |
|--------------------------|------------|
| Brizio Claudio | 0512119716 |
| Cannella Vincenzo | 0512119065 |
| Coscia Matteo | 0512121210 |
| Mascheri Mario | 0512120157 |

Scritto da: Vincenzo Cannella

Cronologia delle Revisioni

| Data | Versione | Descrizione | Autore |
|-------------|-----------------|--|-------------------|
| 21/12/2025 | 0.1 | Prima stesura e bozza del Test Case Specification | Vincenzo Cannella |
| 04/02/2026 | 1.0 | Stesura ufficiale e definitiva del Test Case Specification | Matteo Coscia |

Indice

| | |
|---|----------|
| 1 Test case specification identifier | 2 |
| 1.1 Convenzione di Denominazione | 3 |
| 2 Test items | 3 |
| 2.1 Elenco dei componenti oggetto di test | 3 |
| 3 Specifiche degli input (Input Specifications) | 4 |
| 4 Bisogni ambientali (Environmental needs) | 4 |
| 4.1 Requisiti Software | 4 |
| 4.2 Requisiti Software | 4 |
| 4.3 Test Cases | 5 |
| 5 Dettaglio Test Cases (Input e Oracolo) | 5 |
| 6 Dettaglio dei Test Case | 5 |
| 6.1 Test Cases: Registrazione Utente (UC 3) | 5 |
| 6.2 Test Cases: Login (Autenticazione) | 6 |
| 6.3 Test Cases: Prenotazione Noleggio | 6 |
| 6.4 Test Cases: Pagamento e Vendita | 7 |
| 6.5 Test Cases: Ricerca Veicoli (Filtri Catalogo) | 7 |

1 Test case specification identifier

L'identificatore è un nome univoco assegnato a ogni caso di test con lo scopo di distinguerlo dagli altri all'interno della suite di test. Per il progetto **EasyDrive**, si adotta una convenzione di denominazione strutturata che riflette la scomposizione del sistema in package e sottosistemi definita nel documento di progettazione a oggetti (ODD).

1.1 Convenzione di Denominazione

Per facilitare il riferimento e la tracciabilità dei casi di test da parte degli sviluppatori e dei tester, gli identificatori dei test case seguiranno la convenzione definita nei *Test Frame* del documento di *Test Plan*.

2 Test items

Questa sezione identifica i componenti software, gli oggetti o i sottosistemi che sono oggetto dell'attività di verifica. Nel contesto del progetto **EasyDrive**, gli elementi del test corrispondono ai package e alle classi descritte nell'Object Design Document (ODD), che costituiscono la struttura fisica del sistema.

Ogni elemento del test viene verificato rispetto alle sue interfacce pubbliche (metodi, attributi ed eccezioni) e alle responsabilità logiche definite nel documento di progettazione.

2.1 Elenco dei componenti oggetto di test

Di seguito sono elencati i principali sottosistemi di EasyDrive che verranno sottoposti a test:

- `it.unisa.easydrive.catalog`: Include le classi responsabili della logica di presentazione e dei meccanismi di ricerca dei veicoli per l'utente.
- `it.unisa.easydrive.account`: Comprende i componenti per l'identificazione (login), la registrazione e la gestione del profilo utente.
- `it.unisa.easydrive.booking`: Rappresenta il core della business logic per il controllo delle disponibilità e la gestione operativa dei noleggi.
- `it.unisa.easydrive.sales`: Oggetto di test per quanto riguarda il flusso d'acquisto, la generazione degli ordini e l'integrità dello storico transazioni.
- `it.unisa.easydrive.payment`: Interfaccia critica che gestisce la validazione e l'esecuzione dei pagamenti tramite l'interazione con il database.
- `it.unisa.easydrive.core`: Package trasversale contenente le classi di base, la connessione al database e le entità condivise (es. Veicolo, Utente) fondamentali per il funzionamento di tutti gli altri moduli.

3 Specifiche degli input (Input Specifications)

In questa sezione e in questo documento vengono definiti i dati necessari per l'esecuzione dei test, includendo gli input forniti dall'utente per l'esecuzione . Gli input devono essere selezionati non solo per verificare il corretto funzionamento (casi positivi), ma anche per gesire e gli scenari critici e i casi limite.

Per ogni caso di test, verranno inoltre descritti i risultati attesi per ogni caso di test, che fungono da “oracolo”. Il successo di un test è determinato dal confronto tra l'output effettivamente ottenuto dal sistema e quello specificato in questa sede. Un test è considerato riuscito se causa un fallimento previsto (nel caso di test negativi) o se produce il comportamento corretto specificato nei requisiti.

4 Bisogni ambientali (Environmental needs)

Questa sezione identifica i requisiti hardware, software e di rete indispensabili per l'esecuzione dei test.

4.1 Requisiti Software

Data la natura del progetto **EasyDrive**, basato su linguaggio Java e architetture orientate agli oggetti, l'ambiente deve includere i seguenti strumenti:

4.2 Requisiti Software

Data la natura del progetto **EasyDrive**, basato su linguaggio PHP e architetture orientate agli oggetti, l'ambiente deve includere i seguenti strumenti:

- **PHP:** Versione aggiornata per l'esecuzione dei componenti software definiti nei vari namespace.
- **Web Server:** Apache o Nginx configurato per eseguire applicazioni PHP.
- **Database Management System (DBMS):** Un'istanza di database (es. MySQL o PostgreSQL) necessaria per testare il modulo **Core** che gestisce le logiche principali dell'applicazione, e il modulo **Payment**, che si occupa della validazione ed esecuzione delle transazioni.

- **Framework di Testing:** Utilizzo di **PHPUnit** per l'esecuzione dei casi di test e per garantire l'automazione della *test suite*.
 - **Test Stub:** Implementazioni parziali di un componente da cui dipende il componente sotto esame.
 - **Test Driver:** Implementazioni parziali di un componente che dipende dal componente sotto esame.

L'uso combinato di driver e stub permette di isolare i singoli moduli dal resto del sistema per verificarne il comportamento in modo atomico.

4.3 Test Cases

Di seguito sono elencati tutti i test cases dei test specificati nel documento di Test Plan

5 Dettaglio Test Cases (Input e Oracolo)

6 Dettaglio dei Test Case

In questa sezione vengono riportati i casi di test concreti, strutturati per funzionalità. Ogni tabella associa i dati di input specifici al relativo oracolo (risultato atteso).

6.1 Test Cases: Registrazione Utente (UC 3)

| Input | Oracolo (Risultato Atteso) |
|---|---|
| TC_REG_01 (Successo) <i>Username:</i> “luca_esp”, <i>Email:</i> “luca@test.it”, <i>Password:</i> “Luca2026!”, <i>Data_Nascita:</i> “15/05/1995”, <i>CAP:</i> “80100”, <i>Città:</i> “Napoli”. | Registrazione completata. I dati sono salvati nel DB e l'utente viene reindirizzato alla Login. |
| TC_REG_02 (Email Errata) <i>Email:</i> “luca.test.it” (mancanza @). | Messaggio di errore: “Formato email non valido”. |
| TC_REG_03 (Email Duplicata) <i>Email:</i> “luca@test.it” (già in DB). | Messaggio di errore: “Email già associata a un account”. |

| Input | Oracolo (Risultato Atteso) |
|---|---|
| TC_REG_04 (Password Debole) <i>Password:</i> “abc” (troppo corta). | Messaggio di errore: “La password deve contenere almeno 8 caratteri e un numero”. |
| TC_REG_05 (Minorenne) <i>Data_Nascita:</i> “10/10/2015” (Età < 18). | Messaggio di errore: “Devi essere maggiorenne per iscriverti”. |
| TC_REG_06 (CAP Errato) <i>CAP:</i> “ABCDE” (non numerico). | Messaggio di errore: “Il CAP deve essere di 5 cifre”. |

6.2 Test Cases: Login (Autenticazione)

| Input | Oracolo (Risultato Atteso) |
|--|--|
| TC_LGN_01 (Successo) <i>Username:</i> “luca_esp”, <i>Password:</i> “Luca2026!”. | Accesso eseguito e redirect alla Dashboard. |
| TC_LGN_02 (Credenziali Errate) <i>Username:</i> “luca_esp”, <i>Password:</i> “Sbagliata123”. | Messaggio di errore: “username o password non corretta”. |
| TC_LGN_03 (Account Inesistente) <i>Username:</i> “utente_ignoto”, <i>Password:</i> “—”. | Messaggio di errore: “username o password non corretta”. |

6.3 Test Cases: Prenotazione Noleggio

| Input | Oracolo (Risultato Atteso) |
|---|---|
| TC_NOL_01 (Successo) <i>Inizio:</i> 10/05/2026, <i>Fine:</i> 15/05/2026. <i>Veicolo:</i> Libero. | Prenotazione salvata e veicolo bloccato a sistema. |
| TC_NOL_02 (Occupato) <i>Inizio:</i> 10/05/2026, <i>Fine:</i> 15/05/2026. <i>Veicolo:</i> Già prenotato. | Messaggio di errore: “Veicolo non disponibile nelle date scelte”. |

| | |
|--|---|
| TC_NOL_03 (Date Errate) <i>Inizio:</i> 20/05/2026, <i>Fine:</i> 10/05/2026 (<i>Inizio > Fine</i>). | Messaggio di errore: “La data di fine deve essere successiva all’inizio”. |
|--|---|

6.4 Test Cases: Pagamento e Vendita

| Input | Oracolo (Risultato Atteso) |
|---|--|
| TC_PAG_01 (Successo) <i>Numeri_Carta:</i> 16 cifre, <i>Scadenza:</i> 12/28, <i>CVV:</i> 123. | Pagamento autorizzato e creazione ordine d’acquisto. |
| TC_PAG_02 (Carta Scaduta) <i>Scadenza:</i> 01/24 (passata). | Messaggio di errore: “La carta inserita è scaduta”. |
| TC_PAG_03 (Formato Errato) <i>CVV:</i> “12” (corto) o <i>Numeri_Carta:</i> “ABCD”. | Messaggio di errore: “Dati della carta non validi”. |

6.5 Test Cases: Ricerca Veicoli (Filtri Catalogo)

| Input | Oracolo (Risultato Atteso) |
|--|--|
| TC_CAT_01 (Successo) <i>Cat:</i> “SUV”, <i>Mod:</i> “Stelvio”, <i>Prezzo_Min:</i> 100€, <i>Prezzo_Max:</i> 500€. | Visualizzazione dei modelli Stelvio nel range di prezzo. |
| TC_CAT_02 (Range Errato) <i>Prezzo_Min:</i> 500€, <i>Prezzo_Max:</i> 100€ (<i>Min > Max</i>). | Messaggio di errore: “Il prezzo minimo non può superare il massimo”. |
| TC_CAT_03 (Incoerenza) <i>Cat:</i> “SUV”, <i>Mod:</i> “Panda” (non compatibili). | Messaggio: “Modello non disponibile per la categoria scelta”. |