



# **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:**

<b>Nome</b>	Matricola
<b>Mario Mascheri</b>	0512120157
<b>Claudio Brizio</b>	0512119716

**Partecipanti:**

<b>Nome</b>	Matricola
<b>Brizio Claudio</b>	0512119716
<b>Cannella Vincenzo</b>	0512119065
<b>Coscia Matteo</b>	0512121210
<b>Mascheri Mario</b>	0512120157

**Scritto da:** Vincenzo Cannella

**Cronologia delle Revisioni**

<b>Data</b>	<b>Versione</b>	<b>Descrizione</b>	<b>Autore</b>
21/12/2025	0.1	Prima stesura e bozza del Test Case Specification	Vincenzo Cannella

# Indice

<b>1 Test case specification identifier</b>	<b>2</b>
1.1 Convenzione di Denominazione . . . . .	2
1.2 Esempi Applicativi per EasyDrive . . . . .	3
<b>2 Test items</b>	<b>3</b>
2.1 Elenco dei componenti oggetto di test . . . . .	3
<b>3 Specifiche degli input (Input Specifications)</b>	<b>4</b>
<b>4 Bisogni ambientali (Environmental needs)</b>	<b>4</b>
4.1 Requisiti Software . . . . .	5
4.2 Requisiti Software . . . . .	5
4.3 Test Cases . . . . .	5
<b>5 Dettaglio Test Cases (Input e Oracolo)</b>	<b>5</b>

## 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 da parte degli sviluppatori e dei tester, l'identificatore seguirà il formato:

#### TC ID Progressivo

Dove:

- TC: Acronimo per *Test Case*.

- **ID\_Progressivo:** Numero incrementale a due cifre per gestire varianti dello stesso test.

## 1.2 Esempi Applicativi per EasyDrive

In base ai sottosistemi individuati nell'ODD, ecco alcuni esempi di identificatori:

Identificatore	Descrizione del Test	Package di Riferimento
TC_Catalog_RicercaVeicoli_01	Verifica la ricerca corretta dei veicoli disponibili.	it.unisa.easydrive.catalog
TC 1.1 e TC 1.2	Test della procedura di login e registrazione	it.unisa.easydrive.user
TC_Booking_GestioneNoleggio_01	Verifica della logica di business per la prenotazione.	it.unisa.easydrive.booking
TC_Payment_ValidateEsecuzione_03	Test dell'interfaccia database per l'esecuzione pagamenti.	it.unisa.easydrive.payment

Tabella 1: Esempi di identificatori per i casi di test di EasyDrive.

## 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.user**: 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.staff**: Include le funzionalità CRUD (Create, Read, Update, Delete) avanzate per la manutenzione del catalogo da parte degli amministratori.
- **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)

### Test Cases: Registrazione Utente

<b>ID Test</b>	<b>Dati di Input e Stato</b>	<b>Oracolo (Output Atteso)</b>
<b>TC_REG_01</b>	<p><b>Input:</b>            Email: mario.rossi@test.it            Pwd: Pass1234            Conf: Pass1234</p> <p><b>Stato:</b> Email nuova (non presente nel DB).</p>	Utente creato correttamente nel Database. Redirect alla pagina di Login.
<b>TC_REG_02</b>	<p><b>Input:</b>            Email: mario.noat            Pwd: Pass1234            Conf: Pass1234</p> <p><b>Stato:</b> -</p>	Visualizzazione errore: "Formato email non valido".
<b>TC_REG_03</b>	<p><b>Input:</b>            Email: esiste@test.it            Pwd: Pass1234            Conf: Pass1234</p> <p><b>Stato:</b> Email già presente nel DB.</p>	Visualizzazione errore: "Email già in uso". Nessun account creato.
<b>TC_REG_04</b>	<p><b>Input:</b>            Email: new@test.it            Pwd: Short            Conf: Short</p> <p><b>Stato:</b> Email nuova.</p>	Visualizzazione errore: "Password troppo corta (min 8 caratteri)".
<b>TC_REG_05</b>	<p><b>Input:</b>            Email: new@test.it            Pwd: Pass1234            Conf: Pass9999</p> <p><b>Stato:</b> Email nuova.</p>	Visualizzazione errore: "Le password non coincidono".

## Test Cases: Login

<b>ID Test</b>	<b>Dati di Input e Stato</b>	<b>Oracolo (Output Atteso)</b>
<b>TC_LOG_01</b>	<p><b>Input:</b> Email: admin@easy.it Pwd: admin123</p> <p><b>Stato:</b> Utente esistente nel DB.</p>	Accesso consentito. Redirect alla Dashboard/Home.
<b>TC_LOG_02</b>	<p><b>Input:</b> Email: admin@easy.it Pwd: errata</p> <p><b>Stato:</b> Utente esistente nel DB.</p>	Accesso negato. Visualizzazione errore: "Credenziali non valide".
<b>TC_LOG_03</b>	<p><b>Input:</b> Email: no@easy.it Pwd: admin123</p> <p><b>Stato:</b> Utente NON presente nel DB.</p>	Accesso negato. Visualizzazione errore: "Utente non trovato".

## Test Cases: Ricerca Veicoli

<b>ID Test</b>	<b>Dati di Input e Stato</b>	<b>Oracolo (Output Atteso)</b>
<b>TC_SEARCH_01</b>	<p><b>Input:</b> Cat: SUV, Marca: BMW Min: 50€, Max: 150€</p> <p><b>Stato:</b> Nel DB presenti BMW X5 (100€).</p>	Viene visualizzata la lista contenente i veicoli trovati (es. BMW X5).

<b>ID Test</b>	<b>Dati di Input e Stato</b>	<b>Oracolo (Output Atteso)</b>
<b>TC_SEARCH_02</b>	<p><b>Input:</b> Cat: CityCar, Marca: Ferrari Min: 20€, Max: 50€</p> <p><b>Stato:</b> Nel DB NON ci sono Ferrari CityCar.</p>	Messaggio: "Nessun veicolo trovato". Viene mostrata una lista vuota.
<b>TC_SEARCH_03</b>	<p><b>Input:</b> Cat: Qualsiasi, Marca: Fiat Min: 100€, Max: 50€</p> <p><b>Stato:</b> -</p>	Errore bloccante: "Il prezzo minimo non può essere superiore al massimo".
<b>TC_SEARCH_04</b>	<p><b>Input:</b> Cat: Inesistente_XYZ Marca: Fiat Min: 0€, Max: 100€</p> <p><b>Stato:</b> Categoria non valida.</p>	Visualizzazione errore: "Categoria selezionata non valida".
<b>TC_SEARCH_05</b>	<p><b>Input:</b> Cat: Berlina, Marca: Audi Min: -10€, Max: 100€</p> <p><b>Stato:</b> -</p>	Visualizzazione errore: "Il prezzo non può essere negativo".

## Test Cases: Prenotazione Noleggio

<b>ID Test</b>	<b>Dati di Input e Stato</b>	<b>Oracolo (Output Atteso)</b>
<b>TC_RENT_01</b>	<p><b>Input:</b>            Inizio: 01/06/26            Fine: 05/06/26</p> <p><b>Stato:</b> Veicolo libero in quelle date.</p>	Ordine confermato. Lo stato del veicolo passa a "Prenotato".
<b>TC_RENT_02</b>	<p><b>Input:</b>            Inizio: 01/06/26            Fine: 05/06/26</p> <p><b>Stato:</b> Veicolo già occupato.</p>	Visualizzazione errore: "Veicolo non disponibile per le date selezionate".
<b>TC_RENT_03</b>	<p><b>Input:</b>            Inizio: 10/06/26            Fine: 05/06/26</p> <p><b>Stato:</b> -</p>	Visualizzazione errore: "Data fine antecedente data inizio".
<b>TC_RENT_04</b>	<p><b>Input:</b>            Inizio: 01/01/20            Fine: 05/01/20</p> <p><b>Stato:</b> -</p>	Visualizzazione errore: "Impossibile prenotare nel passato".

## Test Cases: Pagamento

<b>ID Test</b>	<b>Dati di Input e Stato</b>	<b>Oracolo (Output Atteso)</b>
<b>TC_PAY_01</b>	<p><b>Input:</b>            Carta: 4242... (Visa)            Scad: 12/30            Saldo: OK</p> <p><b>Stato:</b> Gateway online.</p>	Pagamento approvato con successo. Transazione registrata.
<b>TC_PAY_02</b>	<p><b>Input:</b>            Carta: 4242... (Visa)            Scad: 12/30            Saldo: KO</p> <p><b>Stato:</b> Fondi insufficienti.</p>	Errore dal Gateway: "Transazione negata per fondi insufficienti".
<b>TC_PAY_03</b>	<p><b>Input:</b>            Carta: 1234... (Errata)            Scad: 12/30            Saldo: -</p> <p><b>Stato:</b> -</p>	Visualizzazione errore: "Numero carta non valido".