



# TP

## Test Plan

### Foundly

<b>Riferimento</b>	NC05_TP_ver.1.0
<b>Versione</b>	1.0
<b>Data</b>	18/12/2025
<b>Destinatario</b>	Prof. Carmine Gravino
<b>Presentato da</b>	NC05 Team



## Revision History

---

Data	Versione	Descrizione	Autori
5/12/2025	0.1	Definizione della Strategia	SL, SDP, NN
10/12/2025	0.2	Test di Unità e Integrazione	SL, SDP, NN
12/12/2025	0.3	Definizione degli approcci da seguire	SL, SDP, NN
17/12/2025	0.4	Stesura dei test case	SL, SDP, NN
19/12/2025	0.5	Revisione e correzione refusi	SL, SDP, NN
21/12/2025	1.0	Revisione finale	SL, SDP, NN



## Team members

---

Nome	Ruolo	Acronimo	Informazioni di contatto
Salvador Davide Passarelli	Project Manager	SDP	s.passarelli2@studenti.unisa.it
Salvatore Lepore	Project Manager	SL	s.lepore11@studenti.unisa.it
Natale Nappi	Project Manager	NN	n.nappi8@studenti.unisa.it



## Sommario

<b>Revision History</b>	<b>2</b>
<b>Team members</b>	<b>3</b>
Sommario	4
<b>1. Introduzione</b>	<b>5</b>
<b>2. Relazione con altri documenti</b>	<b>6</b>
<b>3. Panoramica del sistema</b>	<b>7</b>
<b>4. Features da testare/da non testare</b>	<b>8</b>
<b>5. Pass/Fail criteria</b>	<b>9</b>
<b>6. Approccio</b>	<b>10</b>
6.1 Testing di Sistema	10
6.2 Testing di Integrazione	10
6.3 Testing di Unità	10
6.4 Ispezione del Codice	11
<b>7. Criteri di sospensione e ripresa</b>	<b>12</b>
<b>8. Materiale di testing</b>	<b>13</b>
<b>9. Test cases</b>	<b>14</b>
9.1.1 Registrazione utenti (Finder-Owner)	14
9.1.2 Registrazione Drop-Point.	17
9.1.3 Login	19
9.1.4 Modifica dati profilo	21
9.2 Gestione Segnalazioni	22
9.2.1 Inserimento nuova segnalazione	23
9.2.2 Ricerca Oggetti/Animali	26
9.2.3 Visualizzazione dettagli	28
9.2 Gestione restituzioni	31
9.3.1 Gestione restituzione (Secure claim)	31

# 1. Introduzione

---

**Foundly** si propone di semplificare il processo di recupero degli oggetti smarriti, al fine di offrire un servizio utile alla comunità mettendo in contatto chi ha perso un oggetto (**Owner**) con chi lo ha ritrovato (**Finder**) o con attività commerciali partner, per permettere degli scambi in modo sicuro (**Drop-Point**).

Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing per la piattaforma **Foundly**. Il fine è quello di garantire che ogni aspetto funzioni in modo corretto.

Considerando la dimensione del team di sviluppo e i vincoli temporali, la strategia di testing seguirà un approccio *Risk-Based*, concentrandosi sulle funzionalità principali.

Di seguito la lista delle features di cui si effettuerà il testing, suddivise per sottosistema:

## Gestione Utente

- **Registrazione Utente e Drop Point:** Verifica della creazione account per i diversi attori (Cittadino Owner/Finder e Attività Commerciale Drop-Point), con validazione dei dati obbligatori.
- **Autenticazione (Login):** Verifica dell'accesso sicuro e gestione degli errori di credenziali.
- **Recupero Password:** Verifica della procedura di reset delle credenziali .

## Gestione Segnalazioni

- **Inserimento Segnalazione:** Verifica del corretto salvataggio di un oggetto smarrito nel sistema.
- **Ricerca e Filtri:** Verifica che la ricerca restituisca i risultati pertinenti applicando vari filtri
- **Dettaglio Segnalazione:** Verifica della corretta visualizzazione della scheda oggetto o animale smarrito.

## Gestione Restituzione (Secure Claim & Drop-Point)

- **Secure Claim:** Verifica della logica di validazione del questionario di sicurezza (domanda/risposta) per inviare e accertare le domande.

**Funzionalità escluse dal testing:** Le seguenti funzionalità non saranno oggetto di test formale in questa fase:

- **Funzionalità Amministrative:** Pannelli di moderazione amministrativa, gestione eliminazione utenti, Drop-Point e cancellazione segnalazioni.
- **Scoreboard:** Visualizzazione delle classifiche utenti.
- **Gestione avanzata Profilo:** Caricamento immagini, storico notifiche e modifica dati non essenziali.
- **Mappe interattive:** Integrazione con API esterne di mappe (es. Google Maps).



## 2. Relazione con altri documenti

---

Per la corretta individuazione dei test case e la stesura del presente piano, si fa esplicito riferimento agli altri documenti prodotti dal team NC05.

**Relazioni con il Requirements Analysis Document (RAD)** I test case pianificati nel Test Plan sono elaborati partendo dai requisiti funzionali e agli scenari d'uso presentati nel RAD, garantendo la copertura delle funzionalità analizzate in precedenza.

**Relazioni con il System Design Document (SDD)** La strategia di testing e i test case pianificati rispettano l'architettura del sistema e la suddivisione in sottosistemi logici presentata nell'SDD.



### 3. Panoramica del sistema

---

Il sistema proposto adotta un'architettura basata sul pattern architetturale **MVC (Model-View-Controller)**. Per lo sviluppo dell'interfaccia utente (Front-end) e la generazione delle view, sono stati impiegati **JSP**, **CSS3** e **JavaScript**.

Per la persistenza dei dati e il collegamento al database, il sistema si appoggia all'infrastruttura cloud fornita da **Aiven**, che garantisce la gestione e l'accessibilità dei dati in fase di produzione ed esecuzione.

## 4. Features da testare/da non testare

---

Di seguito la lista delle features di cui si effettuerà il testing, suddivise per le rispettive gestioni:

### Gestione Utente

- Registrazione Utente (Owner/Finder)
- Login
- Modifica dati profilo

### Gestione Segnalazioni

- Inserimento nuova segnalazione (Oggetto trovato)
- Ricerca oggetti smarriti (Filtri per categoria/luogo)
- Visualizzazione dettagli segnalazione

### Gestione Restituzione (Secure Claim & Scambio)

- Avvio procedura di reclamo (Risposta alle domande di verifica - Secure Claim)

Le funzionalità di cui **non** si andrà ad effettuare le attività di testing riguardano requisiti funzionali di bassa o media priorità, oppure quelle funzionalità che non prevedono un input manuale complesso da parte dell'utente.

In particolare, non saranno oggetto di test in questa fase:

- Visualizzazione passiva della **Scoreboard**
- Funzionalità di moderazione lato **Admin** (es. approvazione nuovi Drop-Point), in quanto trattate separatamente o considerate di priorità inferiore rispetto al flusso principale Owner-Finder.
- Non verrà verificata la correttezza interna di librerie esterne e framework (es. rendering delle mappe, driver di connessione al Database, framework CSS), dandone per assunta l'affidabilità e la stabilità. Il test si limiterà a verificare la corretta integrazione di tali componenti nel sistema.



## 5. Pass/Fail criteria

---

Le attività di testing sono mirate ad identificare la presenza di fault all'interno del sistema, per effettuare un successivo intervento di eliminazione e garantire la qualità del prodotto finale.

L'esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso della sua esecuzione, basandosi sui requisiti specificati nel RAD.

- **PASS (Successo):** Un test case è considerato superato se, dato un input al sistema, l'output ottenuto (comportamento attuale) è **uguale** all'output atteso dall'oracolo.
- **FAIL (Fallimento):** Un test case è considerato fallito se, dato un input al sistema, l'output ottenuto è **diverso** dall'output atteso dall'oracolo, evidenziando quindi un malfunzionamento.

L'intero processo di testing sarà considerato valido e concluso se tutti i seguenti vincoli saranno rispettati:

- Esecuzione del 100% dei test case definiti per i requisiti funzionali ad alta priorità.
- Nessun bug critico (che impedisce le funzionalità core come Login, Segnalazione, Restituzione) rimasto aperto).
- Effettuazione di test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti.
- Raggiungimento di un **branch coverage non inferiore al 75%**

## 6. Approccio

---

L'attività di testing per **Foundly** è stata strutturata in tre livelli principali: **Testing di Sistema**, **Testing di Integrazione** e **Testing di Unità**. Seguendo la logica del Modello a V, la progettazione dei test è avvenuta in parallelo alle fasi di analisi e design, mentre l'esecuzione effettiva avverrà in ordine inverso, partendo dai singoli componenti fino all'intero sistema.

### 6.1 Testing di Sistema

Questa fase serve a verificare che il comportamento globale dell'applicazione corrisponda a quanto definito nel RAD. Il server verrà deployato in ambiente locale per l'esecuzione.

- **Tools:** Utilizzeremo **Selenium IDE** per registrare e riprodurre le interazioni dell'utente (click, inserimento dati) direttamente sul browser, verificando il corretto funzionamento del Front-end integrato con il Back-end.
- **Functional Testing:** Ci concentreremo sulla validazione dei requisiti funzionali, cercando di individuare eventuali fault generati da input non validi o flussi inattesi.
- **Acceptance Testing:** Verrà eseguito un test di accettazione basato sui requisiti funzionali principali. In questa fase, il Project Manager simulerà il comportamento del cliente finale per validare il rilascio.

A causa dei vincoli di budget e tempo, non garantiamo l'esecuzione formale di **Performance Testing**, **Pilot Testing** e **Installation Testing**.

### 6.2 Testing di Integrazione

Per l'integrazione adotteremo un approccio **Bottom-up**, ideale per l'architettura a oggetti del nostro sistema. L'obiettivo è verificare che i vari moduli comunichino correttamente tra loro.

- **Strategia:** I test partiranno dal livello più basso (Service Layer) per poi risalire ai Controller. Durante il test dei Controller, le chiamate ai servizi sottostanti verranno isolate tramite mocking.
- **Tools:**
  - **JUnit:** Framework principale per la definizione dei casi di test.
  - **Mockito:** Per simulare le dipendenze esterne e isolare i componenti.
  - **Maven:** Per l'automazione della build e l'esecuzione dei test.
  - **JaCoCo:** Per monitorare la copertura del codice.
  - **GitHub Actions:** Valuteremo l'uso di questo tool per la Continuous Integration (CI), in modo da lanciare i test automaticamente ad ogni push.

### 6.3 Testing di Unità

In questa fase scenderemo nel dettaglio del codice, testando i singoli metodi delle classi.

- **Strategia:** Utilizzeremo un approccio **Black-box**. Ogni classe di produzione avrà una corrispondente classe di test
- **Cosa testiamo:** Tutta la logica di business.
- **Cosa NON testiamo:** Classi Entity, Interfacce e metodi banali che non contengono logica complessa.



## 6.4 Ispezione del Codice

Puntiamo a mantenere un codice pulito e standardizzato. Per farlo, ci affideremo ai report generati da **Checkstyle** e ai controlli automatici della pipeline Maven. Anche se non potremo effettuare sessioni formali di ispezione del codice approfondita per limiti di tempo, questi strumenti ci aiuteranno a mantenere alta la qualità.



## 7. Criteri di sospensione e ripresa

---

In questa sezione vengono specificate le condizioni che determinano l'interruzione immediata delle attività di testing e le modalità per la loro successiva ripresa.

Di norma, l'esecuzione dei test prosegue fino al completamento di tutti i casi pianificati, anche qualora vengano rilevati malfunzionamenti nel software. Tuttavia, le attività verranno sospese immediatamente nei seguenti casi critici:

- Viene rilevato un errore nella **definizione stessa del Test Case** (es. logica del test errata o oracolo non valido).
- L'ambiente di test diventa instabile o non disponibile, impedendo l'esecuzione attendibile delle prove.

Le attività di testing riprenderanno non appena la causa bloccante sarà stata risolta, ripartendo dal punto in cui si era verificata l'interruzione.



## 8. Materiale di testing

---

L'ambiente di test sarà costituito da un computer necessariamente connesso alla rete. Tale configurazione serve per la verifica delle funzionalità principali in un ambiente sicuro e chiuso pre-rilascio.

## 9. Test cases

La strategia di testing adotterà il metodo Category Partition per scomporre le funzionalità in unità testabili. Gli input verranno selezionati tramite l'analisi delle Classi di Equivalenza per ridurre il numero di casi. In assenza di specifiche formali, la validazione dell'output sarà affidata a un Oracolo Umano, che si baserà su delle euristiche di usabilità e la coerenza interna del sistema.

### 9.1 Gestione Utenti

#### 9.1.1 Registrazione utenti (Finder-Owner)

Parametro: Nome	
FORMATO: $^[A-zÀ-ù ' -]\{2,30\}$$ (Standard alfabetico)	
Nome categoria	Scelte per la categoria
Formato [FNO]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FNO_OK]</li></ol>

Parametro: Cognome	
FORMATO: $^[A-zÀ-ù ' -]\{2,30\}$$ (Standard alfabetico)	
Nome categoria	Scelte per la categoria
Formato [FCO]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FCO_OK]</li></ol>

Parametro: Username	
Nome categoria	Scelte per la categoria
Unicità [UN]	<ol style="list-style-type: none"><li>1. Username già presente = true [error]</li><li>2. Username già presente = false [PROPERTY UN_OK]</li></ol>



Parametro: Email	
FORMATO: <code>^[A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}\$</code>	
Nome categoria	Scelte per la categoria
Formato [FE]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FE_OK]</li></ol>

Parametro: Telefono	
FORMATO: <code>^\d{10}\$</code> (10 cifre numeriche)	
Nome categoria	Scelte per la categoria
Formato [FNT]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FNT_OK]</li></ol>

Parametro: Password	
FORMATO: <code>^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&amp;._#-])[A-Za-z\d@\$!%*?&amp;._#-]{8,}\$</code>	
Nome categoria	Scelte per la categoria
Lunghezza [LP] Complessità [CP]	<ol style="list-style-type: none"><li>1. Lunghezza &lt; 8 = false [error]</li><li>2. Lunghezza &gt;= 8 = true [PROPERTY LP_OK]</li></ol> <ol style="list-style-type: none"><li>1. Manca Maiuscola/Minuscola/Numero/Speciale = true [error]</li><li>2. Contiene tutti i caratteri richiesti = true [PROPERTY CP_OK]</li></ol>



Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_1.1_1	FNO1	<b>Fail:</b> Nome vuoto o con caratteri speciali non ammessi
TC_1.1_2	FNO2, FCO1	<b>Fail:</b> Cognome vuoto o con caratteri speciali non ammessi.
TC_1.1_3	FNO2, FCO2, UN1	<b>Fail:</b> Username già in uso da un altro utente.
TC_1.1_4	FNO2, FCO2, UN2, FE1	<b>Fail:</b> Formato email non valido (manca @ o .dominio).
TC_1.1_5	FNO2, FCO2, UN2, FE2, FNT1	<b>Fail:</b> Numero di telefono con lettere o lunghezza errata.
TC_1.1_6	FNO2, FCO2, UN2, FE2, FNT2, LP1	<b>Fail:</b> Password troppo corta (< 8 caratteri).
TC_1.1_7	FNO2, FCO2, UN2, FE2, FNT2, LP2, CP1	<b>Fail:</b> Password non complessa (es. manca carattere speciale).
TC_1.1_8	FNO2, FCO2, UN2, FE2, FNT2, LP2, CP2	<b>Pass:</b> Registrazione effettuata con successo. Redirect al Login.





### 9.1.2 Registrazione Drop-Point.

Parametro: Nome Attività	
FORMATO: $^[A-z0-9\À-ù \ ' -]\{2,50\}$ (Alfanumerico)	
Nome categoria	Scelte per la categoria
Formato [FNA]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNA_OK]

Parametro: Email Aziendale	
FORMATO: $^[A-z0-9._\%+-]+@[A-z0-9.-]+\.[A-z]\{2,10\}$	
Nome categoria	Scelte per la categoria
Formato [FE]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]

Parametro: Provincia	
FORMATO: $^[A-Z]\{2\}$ (Esattamente 2 lettere)	
Nome categoria	Scelte per la categoria
Lunghezza/Formato [PR]	1. Lunghezza != 2 o Numeri = false [error] 2. Lunghezza = 2 e Lettere = true [PROPERTY PR_OK]



Parametro: Posizione Mappa	
FORMATO: Input hidden latitudine/longitudine popolati da JS	
Nome categoria	Scelte per la categoria
Selezione[MAP]	1. Nessun marker sulla mappa = false [error]  2. Marker posizionato (Lat/Long presenti) = true [PROPERTY MAP_OK]

Parametro: Password	
FORMATO: ^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&._#-])[A-Za-z\d@\$!%*?&._#-]{8,}\$	
Nome categoria	Scelte per la categoria
Complessità[CP]	1. Non rispetta la policy = false [error]  2. Rispetta la policy = true [PROPERTY CP_OK]

Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_1.2_1	FNA1	<b>Fail:</b> Nome attività vuoto o con caratteri non validi.
TC_1.2_2	FNA2, FE1	<b>Fail:</b> Formato email aziendale non valido.
TC_1.2_3	FNA2, FE2, PR1	<b>Fail:</b> Provincia non valida (es. "Milano" invece di "MI" o numeri).



TC_1.2_4	FNA2, FE2, PR2, MAP1	<b>Fail:</b> L'utente ha compilato l'indirizzo ma non ha selezionato/confermato il punto sulla mappa (Coordinate null).
TC_1.2_5	FNA2, FE2, PR2, MAP2, CP1	<b>Fail:</b> Password non complessa (manca numero o carattere speciale).
TC_1.2_6	FNA2, FE2, PR2, MAP2, CP2	<b>Pass:</b> Registrazione Drop-Point completata. Redirect al Login.

### 9.1.3 Login

Parametro: Email	
FORMATO: <code>^[A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}\$</code>	
Nome categoria	Scelte per la categoria
Formato[FE] Esistenza Account [EA]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK] 1. Email non presente nel DB = true [error] 2. Email presente nel DB = true [PROPERTY EA_OK]



Parametro: Password	
VINCOLO: Campo obbligatorio, verifica corrispondenza hash	
Nome categoria	Scelte per la categoria
Corrispondenza [CP]	1. Password errata per l'utente = true [error]  2. Password corretta = true [PROPERTY CP_OK]

Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_1.3_1	FE1	<b>Fail:</b> Formato email non valido (es. manca @). Il browser o il sistema bloccano l'invio.
TC_1.3_2	FE2, EA1	<b>Fail:</b> Email valida formalmente, ma account inesistente. Il sistema mostra "Credenziali errate".
TC_1.3_3	FE2, EA2, CP1	<b>Fail:</b> Account esistente, ma password errata. Il sistema mostra "Credenziali errate".
TC_1.3_4	FE2, EA2, CP2	<b>Pass:</b> Login effettuato con successo. Reindirizzamento alla Index (Home).



#### 9.1.4 Modifica dati profilo

Parametro: Username	
VINCOLO: Univocità nel DB	
Nome categoria	Scelte per la categoria
Modifica[MOD]	<ol style="list-style-type: none"><li>1. Username invariato = true [PROPERTY MOD_SKIP]</li><li>2. Username modificato e univoco = true [PROPERTY MOD_OK]</li><li>3. Username modificato ma già esistente = false [error]</li></ol>

Parametro: Immagine profilo	
VINCOLO: File immagine (jpg, png) max 10MB	
Nome categoria	Scelte per la categoria
Azione[ACT]	<ol style="list-style-type: none"><li>1. Nessuna modifica = true [PROPERTY ACT_NONE]</li><li>2. Caricamento nuovo file valido = true [PROPERTY ACT_UPLOAD]</li><li>3. Caricamento file non valido (es. .exe o &gt;10MB) = false [error]</li><li>4. Rimozione avatar esistente = true [PROPERTY ACT_REMOVE]</li></ol>

Nota: Per brevità, i parametri Nome e Cognome ereditano le categorie [FNO] e [FCO] già definite nella sezione 9.1.1, poiché i controlli di validazione sono identici.



Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_1.4_1	FNO1	<b>Fail:</b> Modifica Nome con caratteri non validi (es. numeri). Salvataggio bloccato.
TC_1.4_2	MOD3	<b>Fail:</b> Modifica Username con uno già in uso da un altro utente. Il sistema mostra errore.
TC_1.4_3	ACT3	<b>Fail:</b> Tentativo di upload di un file non immagine (es. documento .pdf) come avatar.
TC_1.4_4	FNO2, FCO2, MOD2, ACT_NONE	<b>Pass:</b> Aggiornamento solo dei dati testuali (Nome/Cognome/Username). Avatar invariato.
TC_1.4_5	FNO2, FCO2, MOD1, ACT_UPLOAD	<b>Pass:</b> Aggiornamento con caricamento di una nuova foto profilo valida.
TC_1.4_6	FNO2, FCO2, MOD1, ACT_REMOVE	<b>Pass:</b> Rimozione della foto profilo corrente (ripristino avatar default).



## 9.2 Gestione Segnalazioni

### 9.2.1 Inserimento nuova segnalazione

Parametro: Tipo e modalità	
VINCOLO: Determinare la visibilità dei campi	
Nome categoria	Scelte per la categoria
Flusso[FLOW]	<ol style="list-style-type: none"><li>1. Tipo "Animale" = true [PROPERTY FLOW_ANIMAL]</li><li>2. Tipo "Oggetto" + Consegna "Diretta" = true [PROPERTY FLOW_DIRECT]</li><li>3. Tipo "Oggetto" + Consegna "DropPoint" = true [PROPERTY FLOW_DP]</li></ol>

Parametro: Selezione Drop-Point	
VINCOLO: Obbligatorio solo se Modalità = DropPoint	
Nome categoria	Scelte per la categoria
Selezione [SEL_DP]	<ol style="list-style-type: none"><li>1. Nessun DropPoint selezionato = false [error]</li></ol>

Parametro: Provincia	
FORMATO: ^[A-Z]{2}\$ (Esattamente 2 lettere)	
Nome categoria	Scelte per la categoria
Lunghezza/Formato [PR]	<ol style="list-style-type: none"><li>1. Lunghezza != 2 o Numeri = false [error]</li><li>2. Lunghezza = 2 e Lettere = true [PROPERTY PR_OK]</li></ol>



Parametro: Data ritrovamento	
VINCOLO: 01/01/2000 <= Data <= Oggi	
Nome categoria	Scelte per la categoria
Validità [VD]	<ol style="list-style-type: none"><li>1. Data &gt; Data Odierna (Futuro) = false [error]</li><li>2. Data &lt; 01/01/2000 (Troppo vecchia) = false [error]</li><li>3. 2000 &lt;= Data &lt;= Oggi = true [PROPERTY DT_OK]</li></ol>

Parametro: Immagine (opzionale)	
VINCOLO: File immagine, max size.	
Nome categoria	Scelte per la categoria
Formato[IMG]	<ol style="list-style-type: none"><li>1. File non immagine (es. .pdf, .exe) = false [error]</li><li>2. File immagine valido o Nessun file = true [PROPERTY IMG_OK]</li></ol>





Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_2.1_1	PR1	<b>Fail:</b> Provincia non valida (es. "Roma" invece di "RM"). Il form non viene inviato.
TC_2.1_2	PR2, DT1	<b>Fail:</b> Data di ritrovamento nel futuro. Logica di validazione blocca l'inserimento.
TC_2.1_3	PR2, DT2	<b>Fail:</b> Data antecedente all'anno 2000 (es. 1999). Il sistema rifiuta segnalazioni troppo vecchie.
TC_2.1_4	PR2, DT3, IMG1	<b>Fail:</b> Tentativo di caricare un documento PDF al posto della foto.
TC_2.1_5	PR2, DT3, IMG2, FLOW_DP, SEL_DP1	<b>Fail:</b> L'utente sceglie "Consegna al DropPoint" ma <b>non seleziona</b> il negozio dal menu a tendina.
TC_2.1_6	PR2, DT3, IMG2, FLOW_ANIMAL	<b>Pass:</b> Inserimento segnalazione Animale (Specie/Razza compilati, DropPoint ignorato).
TC_2.1_7	PR2, DT3, IMG2, FLOW_DIRECT	<b>Pass:</b> Inserimento Oggetto con scambio a mano (DropPoint non richiesto).
TC_2.1_8	PR2, DT3, IMG2, FLOW_DP	<b>Pass:</b> Inserimento Oggetto con consegna al DropPoint correttamente selezionato.



### 9.2.2 Ricerca Oggetti/Animali

Parametro: Parola chiave	
VINCOLO: Input testuale libero	
Nome categoria	Scelte per la categoria
Input[KW]	<ol style="list-style-type: none"><li>1. Stringa vuota = true [PROPERTY KW_EMPTY]</li><li>2. Stringa alfanumerica presente = true [PROPERTY KW_PRES]</li></ol>

Parametro: Filtro tipo	
VINCOLO: Select Menu (Tutti, Oggetto, Animale)	
Nome categoria	Scelte per la categoria
Selezione[TYP]	<ol style="list-style-type: none"><li>1. "Tutti i tipi" (Valore vuoto) = true [PROPERTY TYP_ALL]</li><li>2. Tipo Specifico (Oggetto/Animale) = true [PROPERTY TYP_SPEC]</li></ol>

Parametro: Filtro Categoria	
VINCOLO: Select Menu (Elettronica, Chiavi, ecc.)	
Nome categoria	Scelte per la categoria
Selezione[CAT]	<ol style="list-style-type: none"><li>1. "Tutte le categorie" (Valore vuoto) = true [PROPERTY CAT_ALL]</li><li>2. Categoria Specifica = true [PROPERTY CAT_SPEC]</li></ol>



Parametro: Stato database	
VINCOLO: Determina l'output della ricerca	
Nome categoria	Scelte per la categoria
Match[DB]	<ol style="list-style-type: none"><li>1. Nessun elemento corrisponde ai criteri = true [PROPERTY DB_ZERO]</li><li>2. Uno o più elementi corrispondono = true [PROPERTY DB_MANY]</li></ol>

Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_2.2_1	KW_EMPTY, TYP_ALL, CAT_ALL, DB_MANY	<b>Pass:</b> Visualizzazione di tutte le segnalazioni recenti (stato di default della Home).
TC_2.2_2	KW_PRES, TYP_ALL, CAT_ALL, DB_MANY	<b>Pass:</b> Ricerca per sola parola chiave (es. "Chiavi"). Vengono mostrati i risultati pertinenti.
TC_2.2_3	KW_PRES, TYP_ALL, CAT_ALL, DB_ZERO	<b>Pass:</b> Ricerca infruttuosa (es. "Xyzk123"). Il sistema mostra il messaggio "0 risultati - Nessuna segnalazione trovata".
TC_2.2_4	KW_EMPTY, TYP_SPEC, CAT_ALL, DB_MANY	<b>Pass:</b> Filtraggio per Tipo (es. Solo "Animale"). Vengono mostrati solo animali, ignorando le categorie.
TC_2.2_5	KW_PRES, TYP_SPEC, CAT_SPEC, DB_MANY	<b>Pass:</b> Ricerca combinata completa (es. "Iphone" + "Oggetto" + "Elettronica"). Risultati molto specifici.



TC_2.2_6	KW_EMPTY, TYP_SPEC, CAT_SPEC, DB_ZERO	<b>Pass:</b> Filtro inconsistente (es. Tipo="Animale" + Categoria="Elettronica"). Il sistema restituisce 0 risultati (coerenza logica).
----------	---	---

### 9.2.3 Visualizzazione dettagli

Parametro: ID segnalazione	
VINCOLO: Parametro URL (es. ?id=5)	
Nome categoria	Scelte per la categoria
Esistenza[ID]	1. ID nullo o non presente nel DB = false [error]  2. ID presente nel DB = true [PROPERTY ID_OK]

Parametro: Ruolo utente	
VINCOLO: Confronto tra Utente Loggato e Utente Creatore	
Nome categoria	Scelte per la categoria
Ruolo[ROLE]	1. Utente non loggato = false [error]  2. Utente Finder (Creatore della segnalazione) = true [PROPERTY ROLE_FINDER]  3. Utente Richiedente (Diversa dal creatore) = true [PROPERTY ROLE_CLAIMANT]



Parametro: Stato del reclamo	
VINCOLO: Determina la sidebar	
Nome categoria	Scelte per la categoria
Stato[CLM]	<ol style="list-style-type: none"><li>1. Nessun reclamo effettuato = true [PROPERTY CLM_NONE]</li><li>2. Reclamo in attesa (Pending) = true [PROPERTY CLM_PENDING]</li><li>3. Reclamo Accettato (Vittoria) = true [PROPERTY CLM_WIN]</li><li>4. Reclamo Rifiutato = true [PROPERTY CLM_LOST]</li></ol>

Parametro: Modalità di consegna	
VINCOLO: Determina le info di contatto o ritiro	
Nome categoria	Scelte per la categoria
Tipo[DEL]	<ol style="list-style-type: none"><li>1. Scambio Diretto = true [PROPERTY DEL_DIRECT]</li><li>2. Drop-Point = true [PROPERTY DEL_DP]</li></ol>

Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_2.3_1	ID1	<b>Fail:</b> ID segnalazione inesistente. Redirect alla Home o pagina "Segnalazione non trovata".
TC_2.3_2	ID2, ROLE1	<b>Fail:</b> Utente non loggato tenta di accedere al dettaglio. Redirect al Login.



TC_2.3_3	ID2, ROLE_FINDER	<b>Pass:</b> Visualizzazione lato Finder. Compare la sidebar "Gestione Reclami" con la lista delle richieste ricevute e il tasto "Elimina".
TC_2.3_4	ID2, ROLE_CLAIMANT, CLM_NONE	<b>Pass:</b> Visualizzazione lato Richiedente. Compare il form "Verifica Proprietà" (Secure Claim) per inviare le risposte.
TC_2.3_5	ID2, ROLE_CLAIMANT, CLM_PENDING	<b>Pass:</b> Visualizzazione stato attesa. Sidebar mostra "In Valutazione" (Form bloccato).
TC_2.3_6	ID2, ROLE_CLAIMANT, CLM_WIN, DEL_DIRECT	<b>Pass:</b> Reclamo Accettato (Diretta). Sidebar mostra i contatti email/telefono del Finder.
TC_2.3_7	ID2, ROLE_CLAIMANT, CLM_WIN, DEL_DP	<b>Pass:</b> Reclamo Accettato (Drop-Point). Sidebar mostra il <b>Codice di Ritiro</b> univoco e l'indirizzo del negozio.
TC_2.3_8	ID2, ROLE_CLAIMANT, CLM_LOST	<b>Pass:</b> Reclamo Rifiutato. Sidebar mostra badge rosso "Richiesta Rifiutata".



## 9.2 Gestione restituzioni

### 9.3.1 Gestione restituzione (Secure claim)

Parametro: Compilazione Risposte (lato Owner)	
VINCOLO: Campi obbligatori nel form	
Nome categoria	Scelte per la categoria
Input[ANS]	<ol style="list-style-type: none"><li>1. Uno o più campi vuoti = false [error]</li><li>2. Campi compilati = true [PROPERTY ANS_OK]</li></ol>

Parametro: Decisione risposte (lato Finder)	
VINCOLO: Azione manuale del Finder	
Nome categoria	Scelte per la categoria
Valutazione[JUDGE]	<ol style="list-style-type: none"><li>1. Il Finder Rifiuta le risposte = true [PROPERTY JUDGE_REJECT]</li><li>2. Il Finder Accetta le risposte = true [PROPERTY JUDGE_ACCEPT]</li></ol>



Test Case ID	Test Frame (Combinazione)	Esito Atteso (Oracolo)
TC_3.1_1	ANS1	<b>Fail:</b> L'Owner prova a inviare il form lasciando una risposta vuota. Il sistema blocca l'invio (Client-side validation).
TC_3.1_2	ANS2, JUDGE_REJECT	<b>Pass (Stato Rifiutato):</b> L'Owner invia risposte valide. Il Finder le legge, non le ritiene corrette e clicca "Rifiuta". Il sistema notifica l'Owner del rifiuto.
TC_3.1_3	ANS2, JUDGE_ACCEPT	<b>Pass (Stato Accettato):</b> L'Owner invia risposte valide. Il Finder le legge, le ritiene corrette e clicca "Accetta". Il sistema cambia lo stato in "ACCETTATO" e sblocca i dati di contatto/codice di ritiro.