

Università degli Studi di Salerno

Corso di Ingegneria del Software

FantaUnisa

Test Case Specifications

Versione 1.2



Data: 21/01/2026

Progetto: FantaUnisa	Versione: 1.2
Documento: Test Case Specifications	Data: 21/01/2026

Partecipanti:

Nome	Matricola
Clavino Antonio	0512119692
Corona Francesco	0512119827
Sabetta Francesco	0512118990
Tiberini Monica	0512120226

Scritto da:	Tutti i partecipanti.
-------------	-----------------------

Revision History

Data	Versione	Descrizione	Autore
21/12/2025	0.1	Prima stesura Test Case Specifications	Clavino Antonio
22/12/2025	0.2	Modifiche Test Case Specifications	Clavino Antonio
22/12/2025	0.3	Aggiunte Test Case Specifications	Clavino Antonio
22/12/2025	0.4	Modifiche Use Case	Clavino Antonio
22/12/2025	1.0	Aggiunta Use Case Community	Clavino Antonio
20/01/2026	1.1	Modifica Use Case	Clavino Antonio
21/01/2026	1.2	Modifiche generali	Clavino Antonio

Indice

1.	Test Case specification identifier	3
2.	Scopo del Documento	3
3.	Test Case Specifications	4
3.1	UC1 - Registrazione Utente	4
3.2	UC2 - Login Utente	5
3.3	UC3 - Gestione Squadra	6
3.4	UC4 - Scelta Modulo	7
3.5	UC5 - Upload File Excel	7
3.6	UC6 - Eliminazione Account	8
3.7	UC7 - Logout	9
3.8	UC8 - Visualizzazione Statistiche Giocatori	9
3.9	UC9 - Modifica Dati Personali	10
3.10	UC10 - Gestione Post	11
3.11	UC11 - Inserimento Commento	12
3.12	UC12 - Gestione Reazioni	12
3.13	UC13 - Schieramento Formazione Titolare	12

1. Test Case specification identifier

Identifichiamo i Test Case secondo il seguente schema :

Nome : Test_[nomefunzionalita]

Codice : TC_[numerosottosistema]

Input : parametri della classe di equivalenza testata

Oracolo : comportamento atteso dal sistema

Glossario :

[nomefunzionalita] : nome della funzionalità del sistema da testare.

[numerosottosistema] : numero del sottosistema in cui si trova la funzionalità

2. Scopo del Documento

Il presente documento descrive le specifiche dei casi di test per il sistema **FantaUnisa**. L'obiettivo è verificare la conformità del sistema rispetto ai requisiti funzionali descritti nel *Requirements Analysis Document* e le

specifiche architetture del *System Design Document* .

L'approccio utilizzato per la progettazione dei casi di test è il **Category Partition** per ogni Caso d'Uso :

1. **Analisi delle Specifiche:** Identificazione delle unità funzionali da testare e dei relativi parametri di input e oggetti dell'ambiente.
2. **Identificazione delle Categorie:** Scomposizione dei parametri in categorie significative che influenzano il comportamento del sistema.
3. **Definizione delle Scelte:** Individuazione delle possibili scelte per ogni categoria, distinguendo tra valori validi e valori di errore ([Error]).
4. **Definizione dei Vincoli:** Applicazione di proprietà ([property]) e vincoli logici ([if]) per eliminare combinazioni non valide o ridondanti.
5. **Generazione dei Test Frame:** Creazione delle combinazioni di scelte che costituiscono gli scenari di test astratti.
6. **Creazione dei Test Case:** Istanziamento dei Test Frame con dati concreti.

3. Test Case Specifications

3.1 UC1 - Registrazione Utente

PARAMETRI: e-mail, password, conferma password, nome, cognome

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Formato E-mail
- **Categoria 2:** Unicità E-mail
- **Categoria 3:** Lunghezza Password
- **Categoria 4:** Complessità Password

SCELTE

- **Scelte per Categoria 1 (Formato E-mail)**
 - FMT1: E-mail formato valido (es. x@y.z) [property fmt_valid]
 - FMT2: E-mail formato NON valido (es. senza @) [Error]
- **Scelte per Categoria 2 (Unicità E-mail)**
 - UNQ1: E-mail NON presente nel DB (Nuovo utente) [if fmt_valid] [property email_new]
 - UNQ2: E-mail GIÀ presente nel DB [Error]
- **Scelte per Categoria 3 (Lunghezza Password)**
 - LEN1: Lunghezza >= 8 caratteri [property len_ok]
 - LEN2: Lunghezza < 8 caratteri [Error]
- **Scelte per Categoria 4 (Complessità Password)**
 - CMP1: Contiene almeno un numero e un simbolo [if len_ok]
 - CMP2: Non contiene numeri O non contiene simboli [Error]

TEST FRAME

1. **FMT1, UNQ1, LEN1, CMP1** – (Registrazione Valida)

- **Oracolo:** Creazione account, invio e-mail conferma, visualizzazione messaggio successo.
- 2. **FMT2, UNQ1, LEN1, CMP1** – (E-mail Malformata)
 - **Oracolo:** Messaggio errore validazione campo e-mail.
- 3. **FMT1, UNQ2, LEN1, CMP1** – (Utente Già Esistente)
 - **Oracolo:** Messaggio errore "E-mail già in uso".
- 4. **FMT1, UNQ1, LEN2, CMP1** – (Password Corta)
 - **Oracolo:** Messaggio errore "La password deve contenere almeno 8 caratteri...".
- 5. **FMT1, UNQ1, LEN1, CMP2** – (Password Debole)
 - **Oracolo:** Messaggio errore "La password deve contenere un numero e un simbolo".

TEST CASE

- **TC1 (Reg OK):** e-mail: new@test.it, pass: Secure1!pass DB: e-mail assente. -> **Pass**
- **TC2 (Email Dup):** e-mail: m.rossi@unisa.it, pass: Secure1!pass. DB: email presente. -> **Fail (E-mail in uso)**
- **TC3 (Pass Corta):** e-mail: new@test.it, pass: Short1!. DB: email assente. -> **Fail (Pass < 8)**
- **TC4 (Pass NoSym):** e-mail: new@test.it, pass: Password123. DB: e-mail assente. -> **Fail (Formato pass)**
- **TC5 (Frame 5):** Utente: "Marco", Email: marco@test.it, Pass: passwordlunga (no num/simb). -> **Errore Complessità Pass**

3.2 UC2 - Login Utente

PARAMETRI: e-mail, password

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Presenza dell'e-mail nel database
- **Categoria 2:** Correttezza della password associata all'e-mail
- **Categoria 3:** Stato dell'account (Attivato/Non attivato)

SCELTE

- **Scelte per Categoria 1 (E-mail)**
 - EM1: E-mail presente nel database [property e-mail_found]
 - EM2: E-mail NON presente nel database [Error]
- **Scelte per Categoria 2 (Password)**
 - PW1: Password corretta [if e-mail_found] [property pwd_correct]
 - PW2: Password NON corretta [Error]
- **Scelte per Categoria 3 (Stato Account)**
 - ST1: Account attivo [if e-mail_found and pwd_correct]
 - ST2: Account NON attivo (in attesa di conferma) [Error]

TEST FRAME

1. **EM1, PW1, ST1** – (Login Valido)
 - **Oracolo:** Utente autenticato, reindirizzamento alla Dashboard Personale.
2. **EM2, PW1, ST1** – (E-mail inesistente)
 - **Oracolo:** Messaggio di errore "E-mail o password non corretti".
3. **EM1, PW2, ST1** – (Password errata)
 - **Oracolo:** Messaggio di errore "E-mail o password non corretti".
4. **EM1, PW1, ST2** – (Account non attivo)

- **Oracolo:** Messaggio di errore "Account non attivato. Controlla la tua e-mail" e link per reinvio.

TEST CASE

- **TC1 (Login OK):** e-mail: m.rossi@studenti.unisa.it, password: Password1! – DB: Utente presente, attivo. -> **Pass**
- **TC2 (Utente ignoto):** e-mail: sconosciuto@gmail.com, password: Password1! – DB: Utente assente. -> **Fail (Msg Errore)**
- **TC3 (Password errata):** e-mail: m.rossi@studenti.unisa.it, password: Errata123! – DB: Utente presente, password diversa. -> **Fail (Msg Errore)**
- **TC4 (Non attivo):** e-mail: m.rossi@studenti.unisa.it, password: Password1! – DB: Utente presente, flag is_active=false. -> **Fail (Msg Attivazione)**

3.3 UC3 - Gestione Squadra

PARAMETRI: ListaGiocatori

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Numero totale di giocatori selezionati
- **Categoria 2:** Distribuzione ruoli
- **Categoria 3:** Duplicati nella lista

SCELTE

- **Scelte per Categoria 1 (Totale)**
 - TOT1: Totale = 25 giocatori [property total_ok]
 - TOT2: Totale != 25 giocatori [Error]
- **Scelte per Categoria 2 (Ruoli)**
 - ROL1: 3 Por, 8 Dif, 8 Cen, 6 Att [if total_ok] [property roles_ok]
 - ROL2: Distribuzione ruoli errata [Error]
- **Scelte per Categoria 3 (Duplicati)**
 - DUP1: Nessun giocatore duplicato [if roles_ok]
 - DUP2: Almeno un giocatore presente due volte [Error]

TEST FRAME

TOT1, ROL1, DUP1 – (Rosa Validata)

- **Oracolo:** Salvataggio rosa nel DB, messaggio "Squadra salvata con successo".

TOT2, ROL1, DUP1 – (Rosa Incompleta)

- **Oracolo:** Messaggio errore "Rosa incompleta - devi selezionare 25 giocatori".
- *(Nota: ROL1 e DUP1 sono irrilevanti qui, ma tecnicamente se TOT!=25, ROL1 è falso per definizione, ma Category Partition usa l'errore per dominare lo scenario).*

TOT1, ROL2, DUP1 – (Ruoli Errati)

- **Oracolo:** Messaggio errore "Numero massimo per ruolo raggiunto" o configurazione non valida.

2. **TOT1, ROL1, DUP2** – (Duplicati)

- **Oracolo:** Messaggio errore "Giocatore già inserito".

TEST CASE

- **TC1 (Rosa OK):** Input: 3 P, 8 D, 8 C, 6 A -> **Pass (Salvataggio)**
- **TC2 (Pochi Giocatori):** Input: 2 P, 8 D, 8 C, 6 A -> **Fail (Rosa incompleta)**

- **TC3 (Troppi Difensori):** Input: 3 P, 9 D, 7 C, 6 A -> **Fail (Ruoli errati)**
- **TC4 (Duplicato):** Input: 3 P, 8 D, 8 C, 6 A (ma "Lautaro" inserito 2 volte). -> **Fail (Giocatore duplicato)**

3.4 UC4 - Scelta Modulo

PARAMETRI: modulo_selezionato

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Validità del modulo scelto
- **Categoria 2:** Compatibilità con la rosa
- **Categoria 3:** Stato della rosa

SCELTE

- **Scelte per Categoria 1 (Validità Modulo)**
 - MV1: Modulo supportato [property modulo_valido]
 - MV2: Modulo non supportato/inesistente [Error]
- **Scelte per Categoria 2 (Compatibilità Rosa)**
 - CR1: Rosa compatibile con il modulo [if modulo_valido] [property rosa_compatibile]
 - CR2: Rosa NON compatibile [Error]
- **Scelte per Categoria 3 (Stato Rosa)**
 - SR1: Rosa completa [property rosa_presente]
 - SR2: Rosa incompleta o assente [Error]

VINCOLI

- CR1 e CR2 sono valutabili solo se rosa_presente è vero.

TEST FRAME

- **MV1, SR1, CR1** -- ORACOLO: Modulo salvato con successo, visualizzazione conferma.
- **MV1, SR2** -- ORACOLO: Messaggio errore "Devi prima completare la rosa".
- **MV1, SR1, CR2** -- ORACOLO: Messaggio errore "Modulo non compatibile con la tua rosa".
- **MV2, SR1** -- ORACOLO: Messaggio errore "Modulo non valido".

TEST CASE

- **TC1 (Frame 1):** Modulo "3-4-3", Rosa Completa. -> **Successo**
- **TC2 (Frame 2):** Modulo "4-4-2", Utente senza rosa creata. -> **Errore Compatibilità**
- **TC3 (Frame 3):** Input "5-5-5". -> **Errore Modulo**

3.5 UC5 - Upload File Excel

PARAMETRI: file_input

Oggetti dell'ambiente: Database, File System

CATEGORIE

- **Categoria 1:** Formato/Estensione del file
- **Categoria 2:** Validità struttura interna
- **Categoria 3:** Permessi Utente

- **Scelte per Categoria 1 (Estensione)**
 - EXT1: Estensione .xlsx o .xls corretta [property estensione_ok]
 - EXT2: Estensione errata [Error]
- **Scelte per Categoria 2 (Contenuto)**
 - DAT1: Dati coerenti e completi [if estensione_ok]
 - DAT2: Dati incoerenti o corrotti [Error]
- **Scelte per Categoria 3 (Ruolo)**
 - RL1: Utente è Amministratore/Gestore Dati [property admin]
 - RL2: Utente non è Amministratore [Error]

TEST FRAME

- **EXT1, DAT1, RL1** -- ORACOLO: Importazione completata, messaggio "Aggiornamento riuscito", statistiche aggiornate nel DB.
- **EXT2, RL1** -- ORACOLO: Messaggio errore "Formato file non riconosciuto o corrotto".
- **EXT1, DAT2, RL1** -- ORACOLO: Messaggio errore "Errore durante l'importazione: dati incoerenti".
- **EXT1, DAT1, RL2** -- ORACOLO: Accesso negato alla funzionalità.

TEST CASE

- **TC1:** File "voti.xlsx" valido, Utente Admin -> **Successo**
- **TC2:** File "vacanze.jpg", Utente Admin -> **Errore Formato**
- **TC3 (Frame 3):** File "voti.xlsx" con colonne mancanti, Admin. -> **Errore Dati**
- **TC4 (Frame 4):** File valido, Utente Standard. -> **Accesso Negato**

3.6 UC6 - Eliminazione Account

PARAMETRI: conferma_eliminazione (input utente)

Oggetti dell'ambiente: Database, Sessione

CATEGORIE

- **Categoria 1:** Stato Autenticazione (Utente loggato)
- **Categoria 2:** Scelta Utente (Conferma popup)

SCELTE

- **Scelte per Categoria 1 (Stato)**
 - LG1: Utente loggato [property is_logged]
 - LG2: Utente non loggato [Error]
- **Scelte per Categoria 2 (Conferma)**
 - CNF1: Utente clicca "Conferma" [if is_logged]

TEST FRAME

- **LG1, CNF1** – ORACOLO: Account eliminato dal DB, Logout forzato.
- **LG2 (Sessione Scaduta)** – ORACOLO: Reindirizzamento alla pagina di login.
- **ADMIN_ROLE** – ORACOLO: Utente Admin elimina un altro account con successo .

TEST CASE

- **TC1 (Frame 1):** Utente loggato conferma eliminazione. -> **Eliminato**
- **TC2 (Frame 2):** Utente sessione scaduta clicca elimina. -> **Redirect Login**
- **TC3 (Admin Delete):** Utente Admin elimina un altro utente ("target"). -> **Eliminato (Redirect Admin)**

- **TC4 (Unauthorized):** Utente standard prova a eliminare un altro utente. -> **Errore (Accesso Negato)**

3.7 UC7 - Logout

PARAMETRI: click_logout

Oggetti dell'ambiente: Sessione

CATEGORIE

- **Categoria 1:** Stato utente

SCELTE

- **Scelte per Categoria 1**
 - LGT1: Utente loggato [property logged_in]
 - LGT2: Utente già disconnesso [Error]

TEST FRAME

- **LGT1** – ORACOLO: Sessione invalidata, reindirizzamento Home pubblica.
- **LGT2** – ORACOLO: Reindirizzamento Home pubblica .

TEST CASE

- **TC1:** Utente Loggato clicca Logout -> **Sessione Chiusa**
- **TC2 (Frame 2):** Utente non loggato chiama URL logout. -> **Redirect Home**

3.8 UC8 - Visualizzazione Statistiche Giocatori

PARAMETRI: nome_giocatore

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Validità Input
- **Categoria 2:** Presenza Giocatore nel DB

SCELTE

- **Scelte per Categoria 1 (Input)**
 - IN1: Stringa valida (non vuota) [property input_valid]
 - IN2: Stringa vuota o caratteri speciali non ammessi [Error]
- **Scelte per Categoria 2 (Presenza)**
 - PRS1: Giocatore presente [if input_valid] [property player_found]
 - PRS2: Giocatore non presente [if input_valid] [Error]

TEST FRAME

- **IN1, PRS1** – ORACOLO: Visualizzazione scheda giocatore .
- **IN1, PRS2** – ORACOLO: Messaggio errore "Nessun giocatore trovato".
- **IN2, PRS1** – ORACOLO: Messaggio avviso "Inserire un nome valido".

TEST CASE

- **TC1 (Frame 1):** Cerca "Lukaku" (presente). -> **Dati Mostrati**
- **TC2 (Frame 2):** Cerca "Pelè". -> **Non Trovato**
- **TC3 (Frame 3):** Cerca "". -> **Errore Input**

3.9 UC9 - Modifica Dati Personali

Descrizione: L'utente modifica i propri dati anagrafici o aggiorna le credenziali di accesso.

PARAMETRI: nome, cognome, nuova_password, conferma_password, token

Oggetti dell'ambiente: Database, Sessione

CATEGORIE

- **Categoria 1:** Modifica Dati Anagrafici (Nome/Cognome)
- **Categoria 2:** Modifica Password
- **Categoria 3:** Corrispondenza Conferma Password

SCELTE

- **Scelte per Categoria 1 (Anagrafica)**
 - **AN1:** Nome o Cognome validi e modificati [property info_changed]
 - **AN2:** Campi vuoti o invariati [property info_skip]
- **Scelte per Categoria 2 (Password)**
 - **PW1:** Password valida (≥ 8 char, simboli, numeri) [property pwd_ok]
 - **PW2:** Password non valida o Token scaduto [Error]
 - **PW3:** Campo vuoto (nessuna modifica) [property pwd_skip]
- **Scelte per Categoria 3 (Conferma)**
 - **CP1:** Coincide con nuova password [if pwd_ok]
 - **CP2:** Non coincide [Error]

TEST FRAME

1. **AN1, PW3** – ORACOLO: Profilo aggiornato (solo nome/cognome modificati).
2. **AN2, PW1, CP1** – ORACOLO: Password aggiornata con successo (richiede re-login se via token).
3. **AN1, PW1, CP1** – ORACOLO: Profilo completo aggiornato.
4. **AN2, PW1, CP2** – ORACOLO: Errore "Le password non coincidono".
5. **AN2, PW2** – ORACOLO: Errore validazione password o token non valido.

TEST CASE

- **TC1 (Modifica Info):** Input Nome: "Mario", Cognome: "Rossi", Pass: [Vuota]. -> **Info Aggiornate**
- **TC2 (Cambio Password):** Input Nome: [Invariato], Pass: "SecurePass1!", Conf: "SecurePass1!". -> **Password Aggiornata**
- **TC3 (Errore Conferma):** Input Pass: "Pass1!", Conf: "Pass2!". -> **Errore Mismatch**
- **TC4 (Token/Pass Errata):** Input Token: "scaduto" OPPURE Pass: "corta". -> **Errore Validazione**

3.10 UC10 - Gestione Post

Descrizione: Copre la creazione (pubblicazione) e l'eliminazione dei post.

PARAMETRI: testo, allegato (formazione), operazione (crea/elimina), utente_richiedente

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Validità del Contenuto (per creazione)
 - *Nota:* Secondo l'ODD, un post deve avere testo O un allegato non vuoto.
- **Categoria 2:** Tipo Operazione
- **Categoria 3:** Permessi di Eliminazione
 - *Nota:* Secondo l'ODD, un utente può eliminare solo i propri post.

SCELTE

- **Scelte per Categoria 1 (Contenuto)**
 - CNT1: Solo testo presente [property valid_content]
 - CNT2: Solo allegato presente [property valid_content]
 - CNT3: Testo e allegato presenti [property valid_content]
 - CNT4: Testo vuoto e allegato nullo [Error]

TEST FRAME

1. **CNT1, OP1** – ORACOLO: Post testuale pubblicato con successo nel feed.
2. **CNT2, OP1** – ORACOLO: Post con formazione pubblicato con successo.
3. **CNT3, OP1** – ORACOLO: Post completo pubblicato con successo.
4. **CNT4, OP1** – ORACOLO: Errore "Il post non può essere vuoto".
5. **CNT5, OP1** – ORACOLO: Post creato automaticamente dopo schieramento formazione.

TEST CASE

- TC1 (Frame 1): Testo "Ciao", Allegato Null. -> **Pubblicato**
- TC2 (Frame 2): Testo Null, Allegato Formazione. -> **Pubblicato**
- TC3 (Frame 3): Testo "Ecco la rosa", Allegato Formazione. -> **Pubblicato**
- TC4 (Frame 4): Testo Null, Allegato Null. -> **Errore Vuoto**
- TC5 (Frame 5): Dati ricevuti da "Schiera Formazione" (Attributi). -> **Pubblicato**

3.11 UC11 - Inserimento Commento

PARAMETRI: testo_commento, id_post_target

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Validità Testo Commento
- **Categoria 2:** Stato del Post Target

SCELTE

- **Scelte per Categoria 1 (Testo)**
 - TXT1: Testo valido [property txt_ok]
 - TXT2: Testo vuoto o spazi bianchi [Error]
- **Scelte per Categoria 2 (Post Target)**
 - PST1: Post esistente nel DB [if txt_ok]
 - PST2: Post non esistente [Error]

TEST FRAME

1. **TXT1, PST1** – ORACOLO: Commento aggiunto, contatore commenti incrementato.
2. **TXT2, PST1** – ORACOLO: Errore "Il commento non può essere vuoto".
3. **TXT1, PST2** – ORACOLO: Errore "Impossibile commentare: il post non esiste più".

TEST CASE

- **TC12.1 (Commento OK):** Testo "Bel centrocamp!", Post ID 10 -> **Successo**
- **TC12.2 (Commento Vuoto):** Utente prova a inviare un commento con testo vuoto o solo spazi. -> **Errore**
- **TC12.3 (Post Non Trovato):** Utente prova a commentare il post ID #999 che è stato appena cancellato dall'autore. -> **Errore**

3.12 UC12 - Gestione Reazioni

PARAMETRI: id_post_target, tipo_reazione

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1:** Stato Reazione Precedente (Logica Toggle)
 - *Nota:* Secondo l'ODD, se la reazione esiste viene rimossa, se non esiste viene aggiunta.

SCELTE

- **Scelte per Categoria 1 (Stato)**
 - ST1: L'utente NON ha ancora reagito al post [property add_react]
 - ST2: L'utente HA già reagito al post [property remove_react]

TEST FRAME

1. **ST1** – ORACOLO: Reazione aggiunta, icona illuminata/contatore +1.
2. **ST2** – ORACOLO: Reazione rimossa, icona spenta/contatore -1.

TEST CASE

- **TC13.1 (Aggiunta):** Utente clicca "Like" su post mai votato -> **Like Aggiunto**
- **TC13.2 (Rimozione):** Utente clicca "Like" su post già votato -> **Like Rimosso**

3.13 UC13 - Schieramento Formazione Titolare

PARAMETRI: lista_titolari, lista_panchina, modulo_corrente

Oggetti dell'ambiente: Database

CATEGORIE

- **Categoria 1: Numero Titolari**
 - *Nota:* Una formazione valida deve avere esattamente 11 titolari.
- **Categoria 2: Coerenza con il Modulo**
 - *Nota:* La distribuzione dei ruoli deve rispettare il modulo attivo
- **Categoria 3: Appartenenza Giocatori**
 - *Nota:* I giocatori schierati devono appartenere alla rosa dell'utente (vincolo di sicurezza).
- **Categoria 4: Duplicati**
 - *Nota:* Un giocatore non può essere contemporaneamente titolare e in panchina.

SCELTE

- **Scelte per Categoria 1 (Numero Titolari)**
 - NUM1: Esattamente 11 titolari [property num_ok]
 - NUM2: Diverso da 11 titolari [Error]
- **Scelte per Categoria 2 (Coerenza Modulo)**
 - MOD1: Ruoli coerenti con il modulo [if num_ok] [property mod_ok]
 - MOD2: Ruoli NON coerenti [Error]
- **Scelte per Categoria 3 (Appartenenza)**
 - OWN1: Tutti i giocatori appartengono alla rosa utente [if mod_ok] [property own_ok]
 - OWN2: Uno o più giocatori NON appartengono alla rosa [Error]
- **Scelte per Categoria 4 (Duplicati)**
 - DUP1: Nessun duplicato tra titolari e panchina [if own_ok]
 - DUP2: Giocatore presente sia tra titolari che riserve [Error]

TEST FRAME

1. **NUM1, MOD1, OWN1, DUP1** – ORACOLO: Formazione salvata con successo.
2. **NUM2, MOD1, OWN1, DUP1** – ORACOLO: Errore "Devi schierare esattamente 11 giocatori".
3. **NUM1, MOD2, OWN1, DUP1** – ORACOLO: Errore "La formazione non rispetta il modulo selezionato".
4. **NUM1, MOD1, OWN2, DUP1** – ORACOLO: Errore "Uno o più giocatori non fanno parte della tua rosa".
5. **NUM1, MOD1, OWN1, DUP2** – ORACOLO: Errore "Giocatore schierato più volte".

TEST CASE

- **TC14.1 (Inserimento Valido):** Modulo 3-4-3, 11 Titolari corretti (1P, 3D, 4C, 3A) della propria rosa, Nessun duplicato -> **Successo**
- **TC14.2 (Errore Modulo):** Modulo 4-4-2, Titolari inseriti con schema 3-5-2 -> **Errore Coerenza**
- **TC14.3 (Incoerenza Modulo):** Utente ha selezionato il modulo 3-5-2 ma prova a schierare 4 Difensori. -> **Errore**
- **TC14.4 (Giocatore Non Posseduto):** Utente tenta di schierare il giocatore "Lukaku" (che non possiede) manipolando l'ID della richiesta. -> **Errore**
- **TC14.5 (Giocatore Duplicato):** Utente inserisce "Vlahovic" sia come titolare che come primo panchinaro. -> **Errore**