

Università degli Studi di Salerno

Corso di Ingegneria del Software

FakeBuster Test Report Document Versione 1.0



Data: 12/01/2026

Progetto: FakeBuster	Versione: 1.0
System Design Document	Data: 12/01/2026

Coordinatore del progetto:

Nome	Matricola
Bruno Santo	0512116161
Emiliano Di Giuseppe	0512119155

Partecipanti:

Nome	Matricola
Bruno Santo	0512116161
Emiliano Di Giuseppe	0512119155

Scritto da:	Bruno Santo & Emiliano Di Giuseppe
-------------	------------------------------------

Revision History

Data	Versione	Descrizione	Autore
12/01/2026	1.0	Creazione del Test Report	Bruno Santo & Emiliano Di Giuseppe

Indice

1.	Introduction	3
1.1.	Purpose.....	4
1.2.	Scope.....	4
1.3.	System Overview	4
1.4.	References	4
2.	Test Summary	5
2.1.	Detailed Test Result	5
2.1.1.	Gestione Utente (TC_UTE).....	5
2.1.2.	Gestione Pubblicazioni (TC_PUB).....	7
2.1.3.	Gestione Appelli (TC_APP).....	7
2.1.4.	Gestione Segnalazioni (TC_SEG)	8
2.2.	Defect Analysis (Analisi Difetti).....	8
3.	Evaluation & Approvals	8
3.1.	Evaluation	8
3.2.	Limitations & Constraints	9

1. Introduction

1.1. Purpose

Il presente documento, Test Summary Report, ha lo scopo di riepilogare e valutare i risultati delle attività di testing eseguite sul sistema FakeBuster Social. Questo report certifica la qualità del software rispetto ai requisiti definiti nel Requirement Analysis Document (RAD) e verifica la conformità tecnica rispetto al System Design Document (SDD) e all'Object Design Document (ODD). Esso analizza l'esito dei casi di test specificati nel documento Test Case Specification (TCS), determinando se il software è pronto per il rilascio o se necessita di ulteriori correzioni.

1.2. Scope

Le attività di testing hanno coperto le seguenti aree funzionali, come pianificato nel *Test Plan*:

- **Gestione Utenza:** Verifica delle procedure di Registrazione (TC_UTE_1) e Login (TC_UTE_2) per utenti e Fact-Checker.
- **Gestione Pubblicazioni:** Verifica dell'inserimento post e dell'integrazione con il servizio di Intelligenza Artificiale per il blocco/pubblicazione automatica (TC_PUB_1).
- **Gestione Appelli:** Verifica del processo di contestazione per i post bloccati (TC_APP_1).
- **Gestione Segnalazioni:** Verifica dell'invio di report per post pubblicati (TC_SEG_1).

Sono stati eseguiti test di unità, test di integrazione (con mock dell'IA) e test di sistema tramite Selenium. Come indicato nel piano, non è stata oggetto di test l'accuratezza intrinseca del modello IA, ma solo la risposta del sistema agli score restituiti.

1.3. System Overview

FakeBuster Social è una piattaforma web basata su architettura **MVC** con backend **Python/Flask** e database **MySQL**. Il core del sistema è l'interazione tra i moduli di gestione pubblicazione e un **AIService** esterno, che classifica i contenuti come attendibili o falsi. Il sistema prevede ruoli distinti per utenti standard (User) e moderatori (Fact-checker).

1.4. References

Questo documento fa riferimento alle seguenti specifiche e pianificazioni approvate:

- **Test Plan (TP_Fake-Buster):** Versione 2.0 del 12/01/2026.
- **Test Case Specification (TCS_Fake-Buster):** Versione 2.0 del 12/01/2026.

2. Test Summary

L'attività di testing è stata condotta seguendo la pianificazione temporale definita nel *Test Plan*. Sono stati eseguiti **tutti i Test Case**, coprendo le funzionalità critiche di Gestione Utente, Pubblicazioni, Appelli e Segnalazioni.

- **Copertura Funzionale:** 100% dei requisiti pianificati.

2.1. Detailed Test Result

Di seguito sono riportati i risultati dettagliati per ogni area funzionale, confrontando l'output ottenuto con l'oracolo definito nelle *Output Specifications* del TCS.

2.1.1. Gestione Utente (TC_UTE)

Questa suite ha verificato la registrazione e il login, con particolare attenzione alla validazione degli input (email, password) e alla gestione dei ruoli (User vs Fact-Checker).

ID Test Case	Descrizione Scenario	Input Principale	Esito Atteso (Oracolo)	Esito Ottenuto	Status
TC_UTE_1.1	Registrazione con Username vuoto	(vuoto)	Errore: Username obbligatorio	Errore visualizzato	PASS
TC_UTE_1.2	Registrazione con Email invalida	email_errata	Errore: Formato email non valido	Errore visualizzato	PASS
TC_UTE_1.3	Registrazione con Email duplicata	esistente@test.it	Errore: Email già registrata	Errore visualizzato	PASS
TC_UTE_1.4	Registrazione con Input Valido	NuovoUser	Redirect al Feed / Record DB creato	Utente creato e loggato	PASS
TC_UTE_2.1	Login con Username vuoto	(vuoto)	Alert: Compila questo campo	Alert visualizzato	PASS
TC_UTE_2.2	Login Utente inesistente	NonEsiste	Errore: Credenziali errate	Errore 401	PASS
TC_UTE_2.3	Login Password errata	PassErrata	Errore: Credenziali errate	Errore 401	PASS
TC_UTE_2.4	Login Successo (User)	TestUser	Redirect Feed	Redirect OK	PASS
TC_UTE_2.5	Login Successo (Fact Checker)	FactChecker	Redirect Dashboard	Redirect OK	PASS

2.1.2. Gestione Pubblicazioni (TC_PUB)

Verifica dell'integrazione con il modulo AI (Mock) e del corretto blocco/pubblicazione dei post in base allo score.

ID Test Case	Descrizione Scenario	Input (AI Score)	Esito Atteso (Oracolo)	Esito Ottenuto	Status
TC_PUB_1.1	Inserimento Post vuoto	N/A	Errore: Testo obbligatorio	Blocco UI	PASS
TC_PUB_1.2	Post con bassa attendibilità (Fake)	0.20	Stato: 'Bloccato'	DB: Bloccato	PASS
TC_PUB_1.3	Post con alta attendibilità (Real)	0.95	Stato: 'Pubblicato'	DB: Pubblicato	PASS

2.1.3. Gestione Appelli (TC_APP)

Verifica della logica di business per la creazione di ricorsi su post bloccati.

ID Test Case	Descrizione Scenario	Pre-Condizione	Esito Atteso (Oracolo)	Esito Ottenuto	Status
TC_APP_1.1	Creazione Appello valida	Post Bloccato	Successo	DB: Appello aperto	PASS

2.1.4. Gestione Segnalazioni (TC_SEG)

Verifica dell'invio segnalazioni da parte degli utenti verso contenuti ritenuti inappropriati.

ID Test Case	Descrizione Scenario	Pre-Condizione	Esito Atteso (Oracolo)	Esito Ottenuto	Status
TC_SEG_1.1	Segnalazione post bloccato	Post Bloccato	Errore: Post non disponibile	Azione negata	PASS
TC_SEG_1.2	Segnalazione duplicata	Già segnalato	Errore: Hai già segnalato	Azione negata	PASS
TC_SEG_1.3	Invio Segnalazione valido	Post Pubblicato	Successo / Record creato	DB: Record creato	PASS

2.2. Defect Analysis (Analisi Difetti)

Durante la fase di *Integration Testing* sono state riscontrate anomalie minori relative alla configurazione del Mock AI, risolte prima dell'avvio del *System Testing*.

- **Incidenti Bloccanti:** Nessuno riscontrato in fase finale.
- **Stabilità Ambiente:** Il database MySQL e il server Flask hanno mantenuto stabilità coerente con i criteri di sospensione/ripresa definiti nel Test Plan.

3. Evaluation & Approvals

L'analisi dei risultati ottenuti dalla fase di *System Testing* conferma che il sistema **FakeBuster Social** ha raggiunto un livello di stabilità e funzionalità adeguato agli obiettivi prefissati.

3.1. Evaluation

Sulla base dei criteri di successo definiti nel *Test Plan*:

- **Completezza Funzionale:** Tutte le funzionalità critiche (Registrazione, Login, Pubblicazione con IA, Appelli, Segnalazioni) sono state verificate e risultano conformi alle specifiche del RAD.
- **Qualità del Codice:** I test di unità hanno dimostrato una corretta gestione delle eccezioni e

dei casi limite (es. input vuoti o formati non validi).

- **Integrazione:** L'interazione tra il Backend Flask, il Database MySQL e il modulo Mock AI avviene senza errori di comunicazione o latenze bloccanti.

3.2. Limitations & Constraints

Come specificato nel perimetro del test (*Features not to be tested*), si evidenzia che:

- La valutazione non copre l'accuratezza semantica del modello di Intelligenza Artificiale reale, ma solo la corretta reazione del sistema agli score numerici simulati.
- I test di carico (Stress Testing) non sono stati inclusi in questa iterazione; pertanto, le performance con un alto numero di utenti concorrenti (>1000) non sono garantite.