

**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

**FakeBuster**  
**Problem Statement**  
**Versione 1.0**



Data: 13/10/2025

Progetto: FakeBuster	Versione: 1.0
Documento: Problem Statement	Data: 13/10/2025

### 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
13/10/2025	1.0	Creazione e modifica sulla base delle indicazioni del prof in aula	Bruno Santo & Emiliano Di Giuseppe

## Indice

1. Problem Domain .....	4
2. Scenarios .....	5
3. Requirement .....	5
3.1 Functional Requirements.....	5
3.2 Non Functional Requirements.....	6
4. Target environment.....	6
5. Deliverables & Deadlines .....	6
6. Client Acceptance Criteria .....	6

## 1. Problem Domain

La disinformazione online continua a rappresentare una minaccia significativa per l'informazione pubblica e la fiducia nelle fonti digitali.

Molti social network consentono la diffusione incontrollata di notizie senza un controllo di veridicità, amplificando il rischio di fake news e manipolazione dell'opinione pubblica.

FakeBuster Social nasce come piattaforma sperimentale che unisce intelligenza artificiale e partecipazione umana per creare un ecosistema informativo più affidabile.

L'obiettivo principale è ridurre la diffusione di notizie false alla radice, bloccandone la pubblicazione già in fase di inserimento.

Il sistema valuta automaticamente le notizie proposte dagli utenti e decide se pubblicarle o meno in base a una stima di attendibilità.

Parallelamente, un fact-checker può analizzare i log delle decisioni dell'IA, verificando la correttezza delle valutazioni e contribuendo al miglioramento continuo del modello.

Il progetto si propone di:

- Creare un social network controllato dove la qualità delle informazioni è prioritaria.
- Sviluppare un modello di AI in grado di stimare in tempo reale la veridicità dei contenuti testuali.
- Garantire trasparenza e tracciabilità grazie ai log delle decisioni dell'IA, accessibili ai verificatori.

In questo modo, FakeBuster Social si distingue non come un semplice rilevatore di fake news, ma come una piattaforma proattiva che previene la diffusione di contenuti falsi, mantenendo al contempo un equilibrio tra automazione e supervisione umana.

## 2. Scenarios

- Scenario 1 — Pubblicazione veloce (Utente: notizia approvata automaticamente)

Attore: Utente registrato

L'utente inserisce titolo + testo + eventuale link / immagine e clicca "Pubblica notizia" l'IA effettua un controllo e approva e il post viene pubblicato

Precondizioni: Utente autenticato; campo obbligatori compilati; limiti di lunghezza rispettati.

- Scenario 2 — Pubblicazione bloccata dall'IA

Attori: Utente registrato / fact-checker

L'utente inserisce titolo + testo + eventuale link / immagine e clicca "Pubblica notizia" l'IA effettua un controllo e blocca la pubblicazione, l'utente può cliccare appeal e manda un ricorso al fact-checker che si autentica con account fact-checker e ha visione di tutti gli appeal, se il fact-checker approva il post il viene pubblicato

Precondizioni: Utente autenticato; campo obbligatori compilati; limiti di lunghezza rispettati, fact-checker autenticato

- Scenario 3 — Pubblicazione non bloccata dall'IA ma utente che legge segnala

Attori: Utente registrato / fact-checker

L'utente inserisce titolo + testo + eventuale link / immagine e clicca "Pubblica notizia" l'IA effettua un controllo e approva e il post viene pubblicato, un utente lo legge e segnala, la segnalazione arriva ricorso al fact-checker che si autentica con account fact-checker e ha visione di tutti gli appeal, se il fact-checker approva il post il viene pubblicato

Precondizioni: Utente autenticato; campo obbligatori compilati; limiti di lunghezza rispettati, fact-checker autenticato

## 3. Requirement

### 1 Functional Requirements

- Registrazione utenti (Utente, Fact-checker, ) e login con permessi separati, utente può pubblicare e leggere e Fact-checker può solo revisionare
- Pubblicazione annunci
- Controllo automatico dell'IA
- Appeal e segnalazione
- dashboard delle segnalazione o degli appeal per i Fact-checker

### **3.2 Non Functional**

- Performance
- Usabilità & Accessibilità
- Scalabilità
- Affidabilità
- compatibilità e portabilità

### **4. Target environment**

Poiché il progetto è in fase iniziale, le tecnologie esatte non sono ancora definitive. L'ambiente di sviluppo previsto include IDE come Visual Studio Code o PyCharm, controllo versione su GitHub e modellazione UML. Lo stack tecnologico ipotizzato è: backend in Python (Flask o FastAPI) o in alternativa Node.js; frontend basato su HTML/CSS (con JavaScript minimo o framework leggero se necessario); database relazionale MySQL per metadati e stato delle notizie. Per funzionalità AI/NLP si prevede l'uso di librerie Python consolidate come scikit-learn, spaCy e transformers per preprocessing, embedding e modelli; il servizio di inferenza sarà esposto via API dal backend.

### **5. Deliverables & Deadlines**

1. Problem Statement: 14 ottobre 2025
2. Requisiti e casi d'uso: 28 ottobre 2025
3. Requirements Analysis Document: 11 novembre 2025
4. System Design Document: 25 novembre 2025
5. Specifica delle interfacce dei moduli del sottosistema da implementare e parte dell' Object Design Document  
Design Document: 16 dicembre 2025
6. Piano di test di sistema e specifica dei casi di test per il sottosistema da implementare: 16 dicembre 2025
7. Object Design Document 20 dicembre
8. Implementazione e Documenti di Esecuzione del Test 10 gennaio

### **6. Client Acceptance Criteria**

Il sistema deve essere in grado di analizzare con accuratezza le notizie, deve permettere a gli utenti di fare un appeal o segnalazione per eventuali errori dell'IA. Deve permettere la revisione umana da parte dell' fact-checker

