

FakeBuster
Requirements Analysis Document
Versione 1.0



Data: 25/10/2025

Progetto: FakeBuster	Versione: 1.0
Documento: Requirements Analysis Document	Data: 25/10/2025

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
25/10/2025	1.0	Creazione e modifica rad prima parte	Bruno Santo & Emiliano Di Giuseppe

Indice

1.	Introduction	4
1.1.	Purpose of the system.....	4
1.2.	Scope of the system.....	4
1.3.	Objectives and success criteria of the project	4
1.4.	Definitions, acronyms, and abbreviations	4
1.5.	References	4
2.	Current system.....	5
3.	Proposed system	5
3.1.	Overview	5
3.2.	Functional requirements.....	5
3.3.	Nonfunctional requirements.....	6
3.3.1.	Usability	6
3.3.2.	Reliability.....	6
3.3.3.	Performance	6
3.3.4.	Supportability.....	6
3.3.5.	Implementation	6
3.3.6.	Interface	6
3.3.7.	Packaging.....	7
3.3.8.	Legal.....	7
3.4.	System models	7
3.4.1.	Scenarios	7
3.4.2.	Use case model.....	9
3.4.3.	Object model	12
3.4.4.	Dynamic model	12
3.4.5.	User interface-navigational paths and screen mock-ups.....	13

1. Introduction

1.1.Purpose of the system

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.

1.2.Scope of the system

Il progetto riguarda lo sviluppo di un social network controllato, in cui ogni contenuto testuale viene valutato tramite modelli di AI di verifica automatica.

Il sistema consente:

- la pubblicazione di notizie solo dopo la validazione;
- la tracciabilità di ogni decisione automatica tramite dei verificatori umani;
- l'analisi delle decisioni dell'IA per migliorare il modello.

1.3.Objectives and success criteria of the project

Il progetto FakeBuster Social mira a:

- Creare un social network dove la qualità dell'informazione è prioritaria.
- Sviluppare un modello di AI capace di stimare in tempo reale la veridicità dei testi.

Criteri di successo:

- Almeno l'75% di accuratezza nella classificazione di notizie veritiere o false.
- Tempo di risposta dell'IA inferiore a 5 secondi.
- Interfaccia accessibile e facile da usare.
- Garantire revisione umana

1.4. Definitions, acronyms, and abbreviations

- IA / AI (Intelligenza Artificiale) – Tecnologia che consente al sistema di analizzare e valutare automaticamente le notizie.
- Fake news – Notizia falsa o manipolata, diffusa con intento ingannevole.
- Fact-checker – Utente esperto incaricato di verificare la correttezza delle decisioni automatiche del sistema in caso di appello o segnalazione del post
- User - Utente autenticato che può sia pubblicare che leggere post

1.5.References

- Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition - Bernd Bruegge & Allen H. Dutoit
- Slide del corso di ingegneria del software di Andre De Lucia

2. Current system

Attualmente, la diffusione di notizie sui social network avviene senza un controllo sistematico della veridicità dei contenuti.

Le piattaforme più diffuse (come X, Facebook o Instagram) si basano su meccanismi di segnalazione manuale o su algoritmi di moderazione generici, che intervengono solo dopo la pubblicazione del contenuto.

Questo approccio presenta diverse problematiche:

- Le fake news possono diffondersi rapidamente prima che vengano rimosse.
- Gli algoritmi di moderazione spesso non distinguono il contesto o l'intento del messaggio.
- La fiducia degli utenti nelle fonti digitali è in costante calo.
- Non esiste una tracciabilità trasparente delle decisioni automatizzate.

Attualmente, quindi, il controllo dell'informazione è reattivo e non preventivo, lasciando spazio a manipolazioni e disinformazione.

3. Proposed system

3.1. Overview

FakeBuster Social propone un nuovo paradigma di social network, in cui la pubblicazione di una notizia è subordinata a una valutazione automatica di attendibilità effettuata da un modello di intelligenza artificiale.

Il sistema integra inoltre una supervisione umana (fact-checker) che analizza eventuali ricorsi o segnalazioni degli user

Funzionalità principali:

- Analisi automatica del testo prima della pubblicazione.
- Classificazione dei contenuti in *attendibili*, *sospetti* o *falsi*.
- Pannello dedicato ai fact-checker per la revisione e l'addestramento continuo del modello.

3.2. Functional requirements

Funzionalità per l'Utente (Publisher / Reader)

- L'utente può registrarsi e accedere tramite l'apposito form
- Ricevere conferma immediata di ricezione.
- Presentare appello su post bloccati.
- Segnalare altri post (report) con motivo e commento.
- Ricevere notizie sullo stato del post (pubblicato, bloccato, in revisione, esito appello).
- Creare nuovo post: titolo, testo, link e allegati.

Funzionalità dell'AI Service / Processo automatico

- Valutazione automatica per ogni post

- SLA: supporto alla pubblicazione automatica entro 5s quando previsto.

Funzionalità per il FactChecker

- Accedere alla dashboard di revisione con visuale unificata: testo, score IA e segnali utente. Può anche vedere le segnalazioni e i post bloccati
- Azioni possibili su un caso: pubblica, mantieni bloccato, rimuovi, edit.

Operazioni di pubblicazione e stato

- Stati supportati: in attesa, pubblicato, bloccato, rimosso.
- Transizioni determinate da regole: IA + soglie + azione FactChecker.
- Azioni automatiche visibili al pubblico (banner + motivo sintetico) quando applicate.

Requisiti di qualità chiave (breve)

- Decisione e pubblicazione automatica entro 5s quando applicabile.
- Notifica all'utente (es. blocco) entro 5s.

3.3. Nonfunctional requirements

3.3.1. Usability

Interfacce chiare per invio post, appello e coda fact-checker; dashboard unificata per fact-checker; conformità base a standard di accessibilità.

3.3.2. Reliability

Il sistema deve garantire un'accuratezza minima dell'75% nelle classificazioni AI.

3.3.3. Performance

Decisione IA + pubblicazione automatica $\leq 10s$ quando prevista

3.3.4. Supportability

Tutte le modifiche devono essere tracciate e versionate.

3.3.5. Implementation

- Linguaggio principale: Python (per AI e backend).
- Framework web: Flask / FastAPI.
- Database: MySQL.
- Frontend: html/css

3.3.6. Interface

- Utente: interfaccia web responsive.

- Fact-checker: pannello dedicato per gestione dei ricorsi e segnalazioni
- API REST: per comunicazione tra moduli AI e piattaforma social.

3.3.7. Packaging

Il sistema sarà distribuito come piattaforma web modulare, con componenti separati per backend, motore AI, frontend .

Dal lato utente, il sistema sarà accessibile tramite interfaccia web.

3.3.8. Legal

Il trattamento dei dati personali seguirà il Regolamento GDPR (UE 2016/679).

Tutti i contenuti generati dagli utenti saranno soggetti a una policy di responsabilità che vieta la pubblicazione di materiale offensivo, discriminatorio o lesivo di diritti altrui.

3.4. System models

3.4.1. Scenarios

Scenario 1 — userSubmitsNews

Attori Partecipanti

maria:User, AIService

Flusso di eventi

1. Maria (User) apre la form “Nuovo Post” e incolla titolo + testo + eventuale immagine.
2. Maria clicca “Invia”.
3. AIService analizza il testo: estrae features, calcola embedding e produce una stima di attendibilità (score = 0.87).
4. AIService invia la decisione e il log (features principali, score, timestamp) a FakeBusterSystem.
5. FakeBusterSystem, con soglia predefinita (es. $\text{publish_if_score} \geq 0.7$), pubblica automaticamente il post nello stream pubblico.
6. Maria riceve notifica “Post pubblicato” con link.

Precondizione

L’utente è autenticato e non supera vincoli di rate-limit.

Postcondizione

Il post è pubblicato e la decisione dell’IA è registrata.

Quality requirements

- Decisione e pubblicazione entro 5 secondi.

Scenario 2 — aiBlocksSubmission

Attori partecipanti

luca:User, AIService

Flusso di eventi

1. Luca invia una notizia (titolo + testo + immagine).
2. AIService valuta e produce score = 0.18 (basso).
3. FakeBusterSystem cambia stato del post in bloccato e invia notifica a Luca: “Il tuo contenuto non è pubblicato per possibile bassa attendibilità. Puoi modificare e riprovare o presentare appello.”
4. Luca può modificare il testo o allegare fonti; se modifica e reinvia, il ciclo riparte dalla valutazione AI.

Precondizione

Utente autenticato; contenuto analizzabile (testo o testo+link).

Postcondizione

Post rimane bloccato o viene reinviato dopo modifica.

Quality requirements

- Notifica all’utente entro 5s.
- Motivo sintetico (es. “score 0.18 — alta probabilità di contenuto non verificato”) mostrato all’utente; non esporre feature sensibili.

Scenario 3 — userAppeal

Attori partecipanti

anna:User, fact checker:moderator1,AIService

Flusso di eventi

1. Anna trova il suo contenuto bloccato e clicca “Presenta appello”
2. FakeBusterSystem registra l’appello e mette in coda per revisione umana.
3. Un Fact checker(moderator1) apre l’interfaccia che mostra: testo del post, score IA.
4. Moderator1 valuta e sceglie fra: pubblica, mantieni bloccato.
5. Se pubblica → FakeBusterSystem pubblica il post. Se mantieni bloccato → notifica all’utente con motivazione sintetica.

Precondizione

Post in stato revisione e appello inviato.

Postcondizione

Post pubblicato o definitivamente respinto.

Quality requirements

- Fact checker deve visualizzare tutte le informazioni rilevanti (testo, log IA) in un’unica schermata.

Scenario 4 — userReportsPublicPostToFactChecker

Attori partecipanti

user (utente autenticato),AIServicefactChecker

Flusso di eventi

- user clicca **Segnala** su un post
- system crea report.
- system assegna la task a un factChecker e invia notifica.
- factChecker valuta e decide.
- Autore può vedere la decisione del fact checker guardando i suoi post.

Precondizione

utente autenticato + post pubblico visibile.

Postcondizione

post bloccato o pubblico a discrezione del fact checker

Quality requirements

trasparenza pubblica per azioni automatiche (banner + motivo);

Scenario 5 — userViewsPostHistory

Attori partecipanti

user

Flusso di eventi

- user clicca sul proprio profilo → seleziona “I miei post”.
- system recupera l'elenco dei post associati all'utente.
- system mostra l'elenco [titolo, stato (Pubblicato, Bloccato, In Revisione)].

Precondizione

utente autenticato.

Postcondizione

utente visualizza il proprio storico post.

Quality requirements

Accuratezza dello stato (lo stato visualizzato deve rispecchiare lo stato reale del post).

3.4.2. Use case model

UC1 — userSubmitsNews

Nome caso d'uso Pubblicazione di una notizia

Attori principali User , AI

Descrizione L'utente invia una nuova notizia che viene analizzata automaticamente dall'AI e pubblicata se supera la soglia di attendibilità.

Precondizione L'utente è autenticato.

1. L'utente apre la form “Nuovo post” e inserisce titolo, testo e immagine.
2. Clicca “Invia”.

Flusso principale 3. L'AI analizza il contenuto e assegna uno *score* di attendibilità.
4. FakeBusterSystem confronta lo score con la soglia.
5. Se $\text{score} \geq 0.7 \rightarrow$ pubblica il post.

Nome caso d'uso **Pubblicazione di una notizia**

6. L'utente riceve la notifica "Post pubblicato".

Postcondizione Il post è pubblicato e la decisione dell'AI è registrata nel log.

Requisiti di qualità Tempo massimo di analisi e pubblicazione: 5s.

Flusso di eventi alternativi: Flusso Alternativo Questo flusso inizia al punto 5 del Flusso Principale. 5a. Se score < 0.7 (es. 0.18): 6a. FakeBusterSystem blocca la pubblicazione. 7a. FakeBusterSystem invia una notifica pop-up all'utente con il motivo.

UC2 — userAppeal

Nome caso d'uso **Appello di un contenuto bloccato**

Attori principali User , FactChecker, AI

Descrizione L'utente contesta la decisione automatica di blocco e chiede la revisione manuale da parte di un fact checker.

Precondizione Post in stato "bloccato"; appello inviato.

Flusso principale

1. L'utente clicca "Presenta appello".
2. Il sistema registra l'appello e lo invia in coda.
3. Il fact checker apre l'interfaccia di revisione (testo + log AI).
4. Valuta il caso e sceglie "Pubblica" o "Mantieni bloccato".
5. Il sistema esegue la decisione e invia notifica all'utente.

Postcondizione Il post viene pubblicato o respinto definitivamente.

Requisiti di qualità Tutte le info visibili in un'unica schermata.

UC3 — userReportsPublicPostToFactChecker

Nome caso d'uso Segnalazione di un post pubblico

Attori principali User, FactChecker, AIService

Descrizione Un utente segnala un post pubblico sospetto. Il fact checker valuta la segnalazione e decide se mantenere o bloccare il post.

Precondizione L'utente è autenticato; il post è pubblico.

Flusso principale

1. L'utente clicca "Segnala" su un post.
2. Il sistema crea un report e lo assegna a un fact checker.
3. Il fact checker analizza il post e decide.
4. L'autore visualizza l'esito della revisione.

Postcondizione Il post rimane pubblico o viene bloccato.

Requisiti di qualità Trasparenza: banner con motivo della revisione visibile al creatore del post.

UC4 — userReportsPublicPostToFactChecker

Nome caso d'uso Visualizza Cronologia Post

Attori principali User

Descrizione L'utente accede alla propria area personale per visualizzare l'elenco di tutti i post che ha inviato e il loro stato corrente (es. Pubblicato, Bloccato, In revisione)

Precondizione L'utente è autenticato;

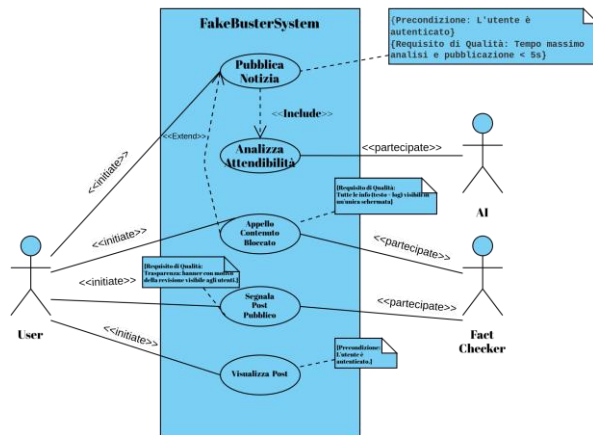
Flusso principale

1. L'utente clicca sul suo profilo e seleziona "I miei post".
2. Compila il motivo della segnalazione.
3. Il sistema mostra l'elenco, indicando per ciascuno il titolo e lo stato (Pubblicato, Bloccato, In Revisione).

Postcondizione L'utente ha visualizzato il suo storico.

Requisiti di qualità Trasparenza: banner con stato post , Accuratezza dello stato (lo stato visualizzato deve rispecchiare lo stato reale del post).

diagramma use case

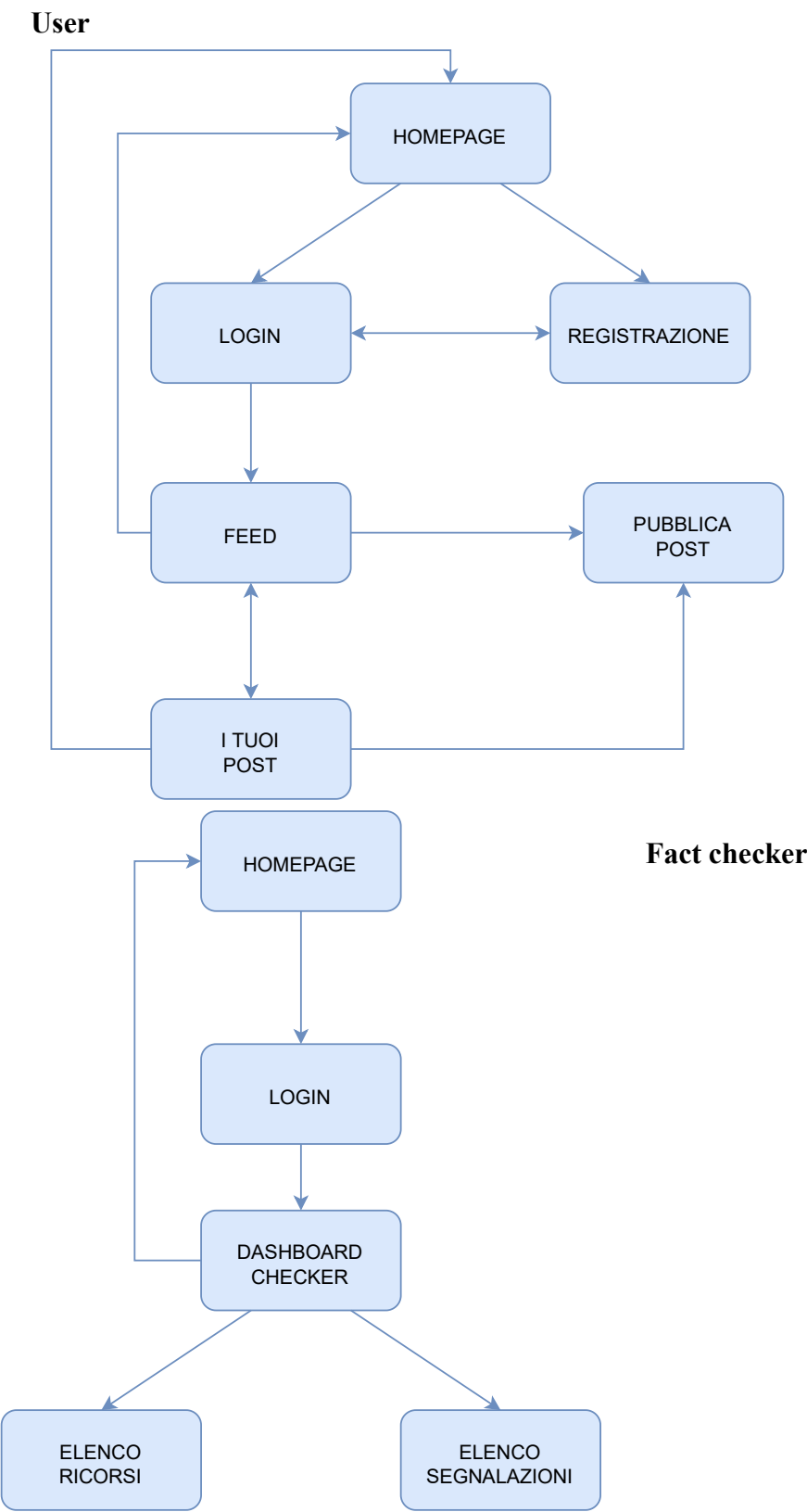


3.4.3. Object model


3.4.4. Dynamic model

3.4.5. User interface-navigational paths and screen mock-ups

Navigation paths



Mock-ups



**Combatti la disinformazione con
Fake Buster Social**

Accedi

Non sei ancora registrato ?

Registrati Ora

Registrati

Email

Password

Registrati

Hai già un account? [accedi](#)


Login

Email

Password

Accedi

Non hai un account? [registrati](#)

**FAKE BUSTER**

Feed | I tuoi post | User

Feed

Autore

img

Titolo news

Testo della new


Segnala

Autore

img

Titolo news

Testo della new

**FAKE BUSTER**

Feed | I tuoi post | User

I tuoi post

Pubblica post

Titolo news

img

Testo della news

Bloccato

Titolo news

img

Testo della news

In revisione

Titolo news

img

Testo della news

Pubblicato

Pubblica post

Allega Foto


Titolo

Testo

Pubblica




Fact-checker



User

Dashboard

Ricorsi

Autore	Titolo	
	Testo della news	<div>Pubblica</div> <div>Blocca</div>

Segnalazioni