

Università degli Studi di Salerno
Corso di Ingegneria del Software

**FakeBuster
System Design Document
Versione 2.0**



Data: 12/01/2026

Progetto: FakeBuster	Versione: 2.0
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
23/11/2025	1.0	Creazione del System Design Document	Bruno Santo & Emiliano Di Giuseppe
12/01/2026	2.0	Revisione Finale	Bruno Santo & Emiliano Di Giuseppe

Indice

1.	Introduction	4
1.1.	Purpose of the system.....	4
1.2.	Design goals	4
1.3.	Definitions, acronyms, and abbreviations	5
1.4.	References	6
1.5.	Overview	6
2.	Current software architecture	7
3.	Proposed software architecture.....	7
3.1.	Overview	7
3.2.	Subsystem decomposition.....	8
3.3.	Hardware/software mapping	9
3.4.	Persistent data management	10
3.5.	Access control and security.....	12
3.6.	Global software control.....	13
3.7.	Boundary conditions	13
3.7.1.	Start up	13
3.7.2.	Shut Down.....	16
3.7.3.	Failures.....	18
3.7.4.	Gestione dei fallimenti	19
4.	Subsystem services	19
5.	Glossary	21

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. Design goals

Obiettivo (Design Goal)	Descrizione (Cosa significa per FakeBuster)	Fonte/Requisito (Da RAD)
DG1: Performance	Il sistema deve essere estremamente rapido. L'analisi dell'IA e la decisione (pubblicazione o blocco) sono il cuore del progetto e devono avvenire in tempo reale.	"Tempo di risposta dell'IA inferiore a 5 secondi". "Decisione e pubblicazione automatica entro 5s". "Notifica all'utente (es. blocco) entro 5s.
DG2: Affidabilità	L'accuratezza del modello AI è un criterio di successo fondamentale. Il sistema <i>deve</i> essere affidabile nel classificare le notizie per essere utile.	"Criteri di successo: Almeno '75% di accuratezza nella classificazione".
DG3: Usabilità	L'interfaccia deve essere intuitiva e senza attriti, sia per l'User (che pubblica e fa appello) sia per il Fact Checker (che gestisce la dashboard di revisione).	"Interfaccia accessibile e facile da usare". "Interfacce chiare per invio post, appello e coda fact-checker". (Implicito da tutti i mock-up).
DG4: Sicurezza	Il sistema deve proteggere i dati degli utenti (come da policy legali) e garantire l'integrità del processo di login.	"Password memorizzata solo in forma hashata". "Il trattamento dei dati personali seguirà il Regolamento GDPR".
DG5: Tracciabilità	Ogni decisione (sia dell'IA che del Fact Checker) deve essere registrata e tracciabile, per permettere la revisione umana e la supervisione.	"la tracciabilità di ogni decisione automatica". "Tutte le modifiche devono essere tracciate e versionate"

1.3. Definitions, acronyms, and abbreviations

Termino	Definizione
Fact-checker	Un utente esperto (moderatore) incaricato di verificare la correttezza delle decisioni automatiche del sistema in caso di appello o segnalazione.
Fake News	Una notizia falsa o manipolata, diffusa con intento ingannevole.
Guest	Un utente non autenticato che può solo accedere alle funzionalità di Login (UC5) e Registrazione (UC6).
Score	Il punteggio numerico (es. 0.87, 0.18) restituito dall'IA che rappresenta la stima di attendibilità di un post.
User	Un utente autenticato che può pubblicare notizie, segnalare post altrui e presentare appelli per i propri post.

Acronimi e Abbreviazioni

- **AI (Artificial Intelligence) / IA (Intelligenza Artificiale):** La tecnologia che consente al sistema di analizzare e valutare automaticamente le notizie.
- **EBC:** Acronimo di **Entity-Boundary-Control**, un pattern architettonale usato per classificare le classi di un sistema.
- **RAD (Requirement Analysis Document):** Il documento che definisce i requisiti funzionali e non funzionali del sistema (questo documento).
- **SDD (System Design Document):** Il documento che descrive l'architettura e il design del sistema (quello che stiamo creando).
- **UC (Use Case):** Caso d'Uso. Descrive una specifica interazione tra un attore e il sistema per raggiungere un obiettivo.

1.4. References

- Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition.
Bernd Bruegge & Allen H. Dutoit
- Documento di RAD relativo a questo progetto
- Documento di Problem Statement

1.5. Overview

Il presente documento (System Design Document) è organizzato nelle seguenti sezioni principali:

- **1. Introduzione:** Descrive in generale lo scopo del sistema, gli obiettivi di design (Design Goals) che il sistema si propone di raggiungere e la struttura di questo documento.
- **2. Architettura:** Sistema Corrente: Analizza lo stato attuale dei sistemi di moderazione dei social network, evidenziando i problemi che FakeBuster si propone di risolvere .
- **3. Architettura Sistema Proposto:** È la sezione centrale del documento. Presenta l'architettura Three-Tier e MVC scelta, la decomposizione in sottosistemi (i nostri Gestori), il mapping hardware/software (Deployment Diagram), la gestione dei dati persistenti (Schema Database) e il controllo degli accessi.
- **4. Servizi dei Sottosistemi:** Descrive in dettaglio le interfacce e i servizi offerti da ciascun sottosistema logico (es. i metodi del GestorePubblicazioni).
- **5. Glossario:** Fornisce una raccolta delle definizioni e degli acronimi utilizzati nel documento (come "Fact-checker", "Score", "UC", ecc.).

2. Current software architecture

Allo stato attuale, le principali piattaforme di social network (come X, Facebook o Instagram) si basano su architetture progettate per la pubblicazione immediata dei contenuti, dove i meccanismi di moderazione intervengono solo in fase successiva (approccio reattivo) tramite segnalazioni manuali o algoritmi asincroni.

Non esiste, al momento, un'architettura software diffusa che integri nativamente un motore di Intelligenza Artificiale come "gatekeeper" bloccante all'interno del flusso di pubblicazione. Poiché i sistemi attuali mancano strutturalmente di questo livello di validazione preventiva sincrona, non esiste una reale architettura di riferimento con cui operare un confronto adeguato rispetto al modello proposto da FakeBuster.

3. Proposed software architecture

3.1. Overview

La piattaforma "FakeBuster Social" è un'applicazione software che interagisce con gli utenti mediante un'interfaccia web responsive e gestisce la persistenza dei dati mediante un **database relazionale**.

Il sistema proposto è basato sullo stile architetturale **Three-Tier** combinato con un'implementazione del design pattern **MVC (Model-View-Controller)**. Il sistema si presenta quindi diviso in tre componenti logici:

- **Model:** Rappresenta la struttura dei dati e le regole di business fondamentali. Contiene i metodi di accesso ai dati persistenti (gestiti tramite MySQL) e le definizioni delle entità principali come User, Post e Segnalazione.
- **View:** Si occupa di visualizzare i dati all'Utente tramite interfaccia web (HTML/CSS) e gestisce l'interazione fra quest'ultimo e l'infrastruttura sottostante.
- **Controller:** Riceve i comandi dell'Utente attraverso la View (es. la richiesta di pubblicazione di un post) e reagisce eseguendo le operazioni necessarie. In questo progetto, il Controller coordina anche la comunicazione con il modulo esterno di Intelligenza Artificiale (AIService) per la validazione dei contenuti.

Questa scelta architetturale ci permette una netta separazione dei concetti (Separation of Concerns) e un'organizzazione pulita del codice. Ciò comporta la semplificazione dei processi di modifica e manutenzione, aspetto cruciale dato che il modulo AI e il backend potrebbero evolvere separatamente.

L'architettura scelta offre importanti vantaggi in termini di:

- **Scalabilità:** I moduli di backend e il database possono essere scalati indipendentemente.
- **Manutenibilità:** La logica di presentazione è separata dalla logica di business.
- **Modularità:** Facilita l'integrazione e l'aggiornamento del componente AI senza impattare l'intera piattaforma.

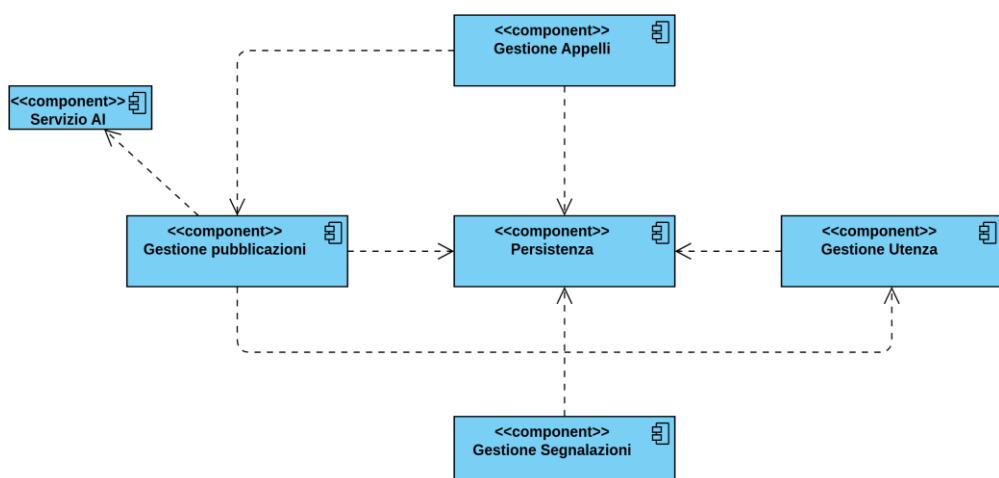
Ovviamente questo pattern architettonico può presentare anche degli svantaggi se non implementato correttamente, come una maggiore complessità iniziale nella configurazione dell'ambiente e una verbosità del codice per gestire il passaggio di dati tra i livelli. Tuttavia, questi costi sono giustificati dalla necessità di garantire robustezza e tracciabilità in un sistema di moderazione automatica.

3.2. Subsystem decomposition

Il sistema è suddiviso in sei sottosistemi principali, che rispecchiano la logica di controllo definita nell'analisi dei requisiti:

- **Gestione utenza:** è responsabile della gestione delle funzioni di autenticazione (registrazione e login), della gestione dei profili utente e dell'assegnazione dei ruoli (User o Fact-checker).
- **Gestione pubblicazioni:** è responsabile dell'intero ciclo di vita di una notizia, dalla sottomissione da parte dell'utente, al coordinamento con il servizio di IA per l'analisi, fino alla decisione automatica di pubblicazione o blocco in base allo score ricevuto.
- **Gestione appelli:** si occupa della gestione del flusso di ricorso per i contenuti bloccati; gestisce la creazione dell'appello da parte dell'utente e la coda di revisione manuale per il Fact-checker, applicando la decisione finale (pubblica o mantieni bloccato).
- **Gestione segnalazioni:** è responsabile della raccolta e gestione dei report inviati dagli utenti su post pubblici ritenuti sospetti, assegnandoli alla coda di moderazione per la verifica da parte del Fact-checker.
- **Servizio IA (Analisi):** si occupa dell'interfacciamento con il motore di Intelligenza Artificiale per l'analisi semantica del testo e il calcolo dello score di attendibilità.
- **Persistenza:** si occupa della gestione della persistenza dei dati (utenti, post, appelli, segnalazioni e log) tramite l'ausilio di un database relazionale.

Component Diagram



3.3.Hardware/software mapping

Il Sistema "FakeBuster Social" utilizza un'architettura **Client/Server**, in cui un Server centrale fornisce servizi a più Client.

Su una macchina Client è eseguito un **web browser** che consente all'Utente (sia esso User o Fact-checker) di interagire con l'Application Server per inoltrare richieste (es. pubblicazione post, invio appello) e visualizzare le risposte ricevute. L'Application Server gestisce la logica applicativa e coordina l'interazione con il modulo di Intelligenza Artificiale, mentre il Database Server gestisce i dati persistenti. La comunicazione tra Client e Server avviene tramite protocollo **HTTP** (interfaccia REST).

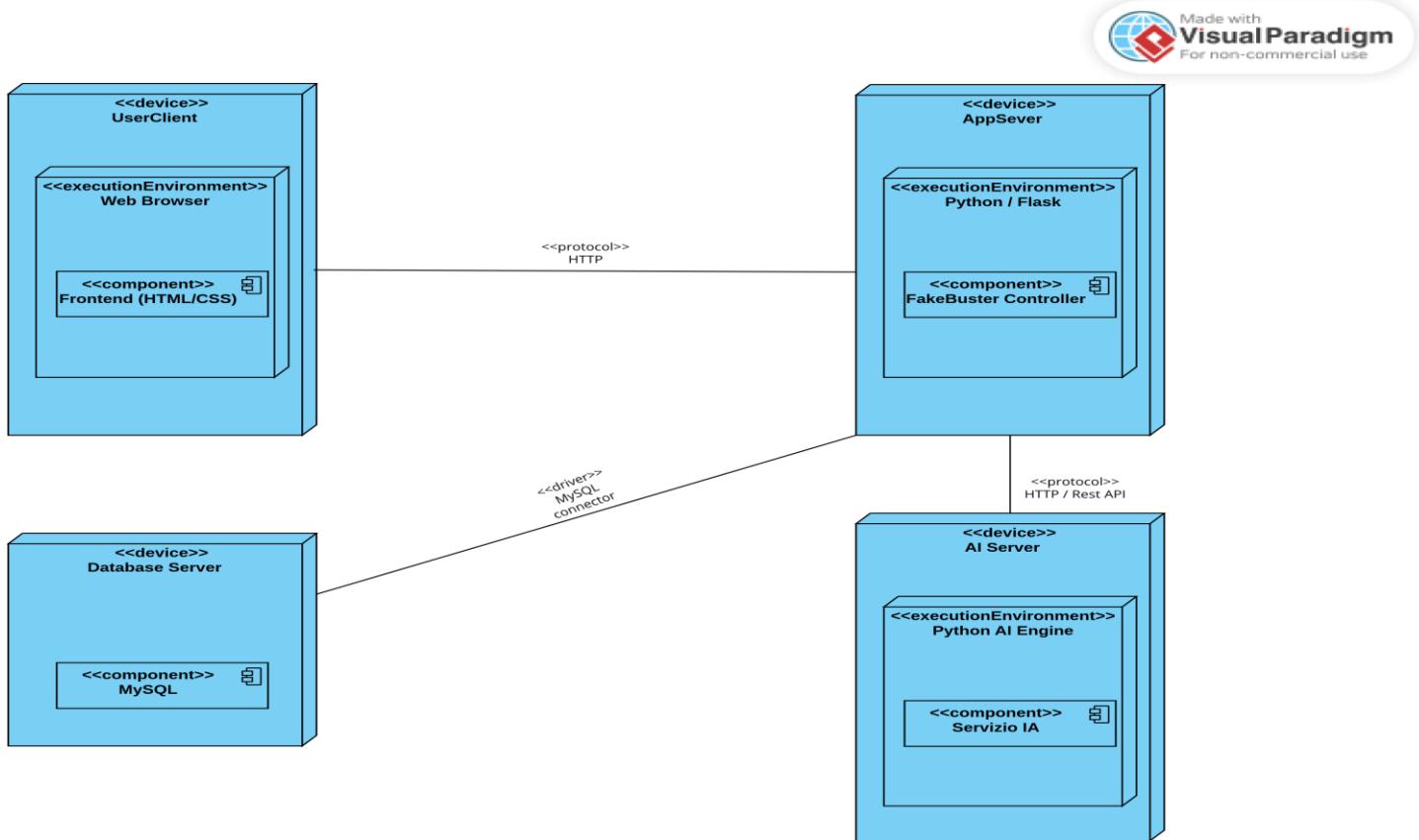
Le specifiche hardware e software necessarie per il **Client** sono rispettivamente una macchina (Desktop o Mobile) dotata di connessione a Internet e un sistema operativo con un web browser moderno installato.

Le specifiche hardware necessarie per il **Server** consistono in una macchina connessa a Internet ad alta disponibilità, la quale sia capace di immagazzinare grandi quantità di dati e disponga di risorse computazionali sufficienti per supportare il carico delle richieste.

Per quanto riguarda le specifiche software necessarie lato Server, esse comprendono:

- Un **Database Management System Relazionale (MySQL)** per la gestione dei dati persistenti.
- Un ambiente di esecuzione **Python** (con framework Flask) per la gestione della logica applicativa, delle API e del modulo di analisi **AI**.

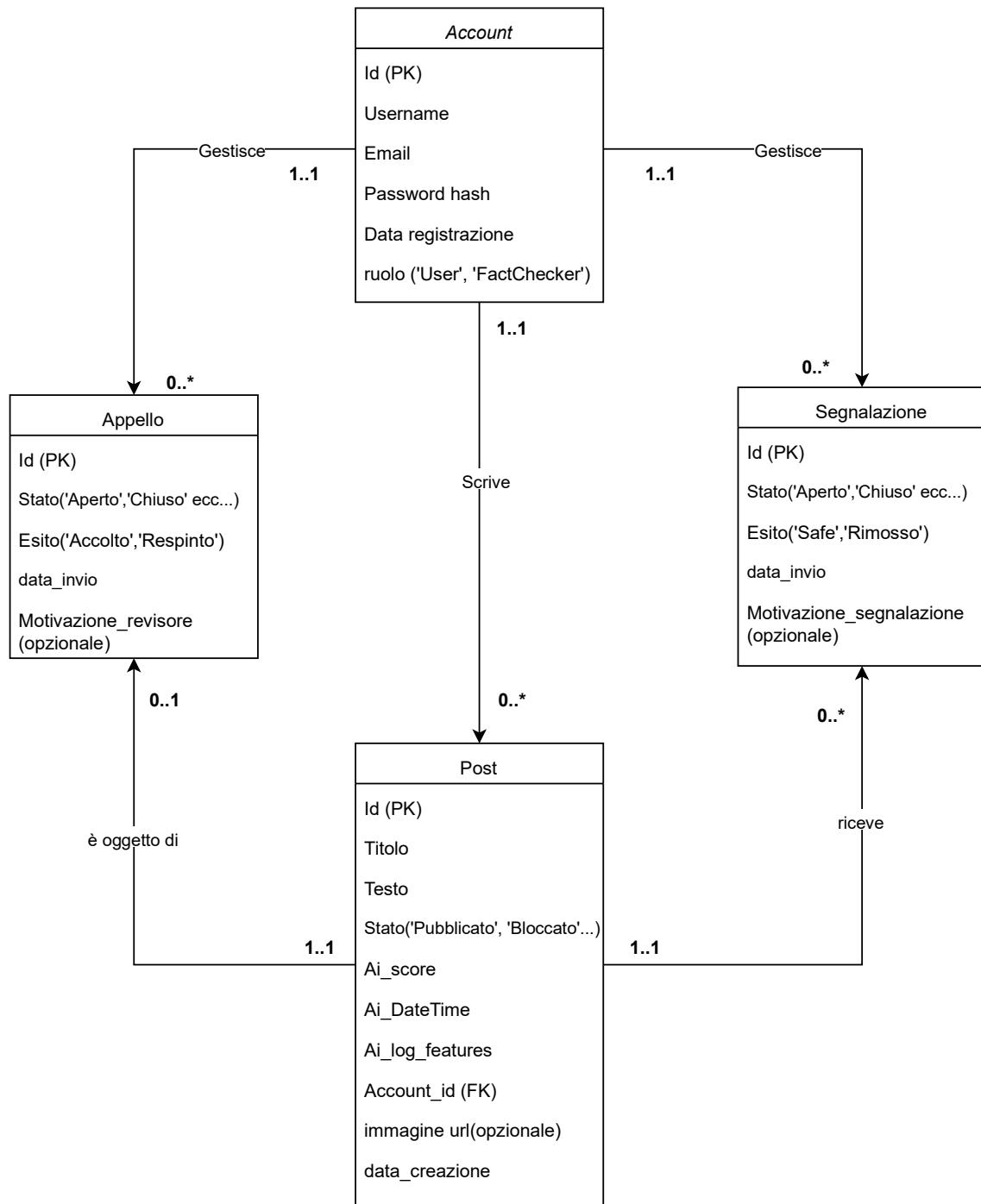
Deployment Diagram



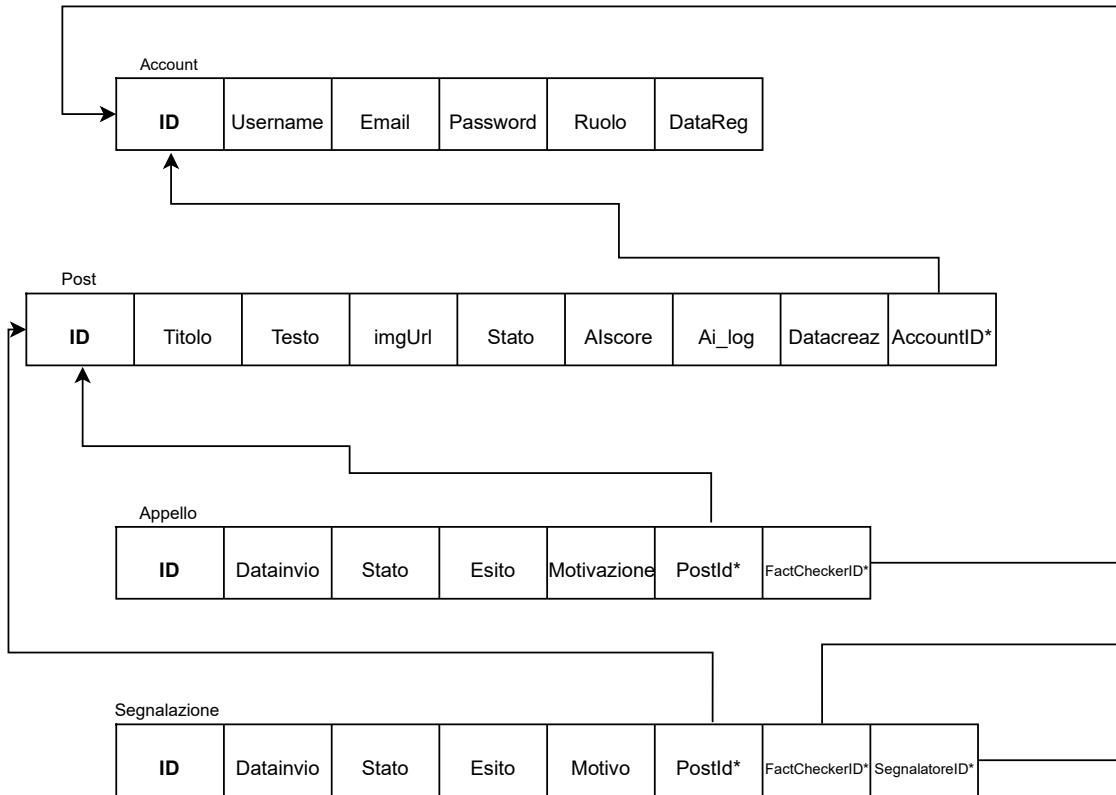
3.4. Persistent data management

Per la gestione dei dati persistenti la scelta è ricaduta su un database relazionale (SQL), nello specifico MySQL. Questa scelta deriva dalla necessità di garantire una rigorosa integrità referenziale e la coerenza dei dati (ACID), requisiti fondamentali per un sistema che gestisce relazioni strette tra utenti, contenuti e azioni di moderazione (segnalazioni e appelli). La ristrutturazione dell'Entity Class Diagram (in vista dello schema logico del database) è avvenuta secondo i seguenti criteri: Le entità "User" e "Fact-checker" (presenti nell'Object Model come entità distinte) sono state accorpate in un'unica tabella "Account" (o Utente), aggiungendo a quest'ultima l'attributo "Ruolo" (ENUM: 'User', 'FactChecker') per gestire i permessi e l'accesso alle diverse aree (Feed vs Dashboard). L'entità logica "AI" (intesa come attore nel RAD) non diventa una tabella a sé stante; i dati relativi alle valutazioni (Score, Timestamp analisi, Log features) sono stati integrati direttamente nell'entità "Post" e/o in una tabella di storico, per garantire un accesso rapido allo stato di attendibilità di ogni notizia.

Entity Class Diagram



Schema logico



3.5. Access control and security

ATTORI / OGGETTI	Gestione Utenza	Gestione Pubblicazioni	Gestione appelli	Gestione segnalazioni
User	<ul style="list-style-type: none"> • Login • Registrazione • Logout • Visualizzazione area personale 	<ul style="list-style-type: none"> • Pubblicare • Vedere news di altri utenti 	<ul style="list-style-type: none"> • appellare 	<ul style="list-style-type: none"> • Segnalare
Fact-checker	<ul style="list-style-type: none"> • Login • Logout • Visualizzazione area Personale • Visualizzazione ricorsi e appelli 	NA	<ul style="list-style-type: none"> • Elaborazione Prendere decisione 	<ul style="list-style-type: none"> • Presa in carico • Elaborazione

3.6.Global software control

Il sistema implementa un modello di controllo del flusso basato sull'elaborazione di richieste asincrone e sincrone originate dal client. Ogni interazione dell'utente con l'interfaccia web genera una richiesta HTTP, la quale viene trasmessa al server applicativo centrale. Quest'ultimo funge da punto di ingresso per tutte le operazioni e provvede a smistare la richiesta verso il modulo applicativo responsabile della gestione dell'evento associato.

All'interno dell'architettura del sistema, ciascuna richiesta viene presa in carico da un componente specializzato (Controller/View/Endpoint), che incapsula la logica di business relativa alla funzionalità richiesta. Il componente gestore si occupa di validare i parametri ricevuti, coordinare eventuali interazioni con i sottosistemi interni (motore di analisi, servizi di persistenza, moduli IA) ed eseguire l'elaborazione necessaria per produrre il risultato atteso.

Una volta ottenuto l'output dall'esecuzione della logica applicativa, il componente responsabile provvede a costruire la risposta secondo il formato previsto dal protocollo (HTML, JSON o altro formato serializzato) e a restituirla al client originario tramite il server web. Tale meccanismo garantisce un flusso di esecuzione deterministico, modulare e facilmente estendibile, mantenendo una netta separazione tra livello di presentazione, logica di controllo e servizi di elaborazione.

3.7.Boundary conditions

Nella seguente sezione vengono riportate le condizioni limite del sistema, come l'avvio, la terminazione e i fallimenti del sistema. In caso di malfunzionamento critico (es. indisponibilità del modulo AI), il Sistema mostrerà un avviso di manutenzione, precludendo all'utente la creazione di nuovi contenuti ma permettendo la visualizzazione di quelli esistenti. Nel caso di un crash dovuto a corruzione dati, si effettuerà un ripristino al backup più recente.

3.7.1. Start up

IDENTIFICATIVO	UC_SU	DATA	23/11/2025
-----------------------	--------------	-------------	-------------------

IDENTIFICATIVO	UC_SU	DATA	23/11/2025
NOME	Avvio del Server FakeBuster	VERSIONE	1.0
AUTORE	Bruno Santo & Emiliano Di Giuseppe		
DESCRIZIONE	Lo UC fornisce la funzionalità di avvio del Server Application e la connessione ai servizi dipendenti (DB e AI).		
ATTORE PRINCIPALE	ADMIN (TECNICO): Vuole avviare il Server per rendere disponibile il Sistema.		
ATTORI SECONDARI	AIService (Sistema Esterno)		
ENTRY CONDITION	Il Server è spento ma alimentato e connesso alla rete.		
EXIT CONDITION ON SUCCESS	Il Server è avviato, connesso al Database e al servizio AI. Il sistema è operativo.		
EXIT CONDITION ON FAILURE	Il Server non è avviato o è in modalità ristretta (errore connessione).		
RILEVANZA	Elevata	FREQUENZA	1/mese (manutenzione)
FLUSSO DI EVENTI PRINCIPALE	<p>Admin:</p> <p>1. Utilizza il comando di avvio sulla shell del server.</p> <p>Sistema:</p>		

IDENTIFICATIVO	UC_SU	DATA	23/11/2025
	<p>2. Inizializza la connessione con il Database MySQL.</p> <p>3. Effettua un <i>handshake</i> (test di connessione) con l'API dell'AIService per verificare la disponibilità del motore di analisi.</p> <p>4. Avvia il web server (Flask) e apre le porte HTTP.</p> <p>5. Mostra a terminale il messaggio "System Online".</p>		
FLUSSO ALTERNATIVO (AI Unreachable)	<p>3.a1 Sistema: Il test di connessione con AIService fallisce (timeout).</p> <p>3.a2 Sistema: Il server si avvia in modalità "Solo Lettura". Gli utenti possono fare login e leggere, ma non pubblicare.</p> <p>3.a3 Sistema: Notifica l'Admin dell'errore critico sul modulo AI.</p>		
FLUSSO DI ERRORE (DB Error)	2.a1 Sistema:		

IDENTIFICATIVO	UC_SU	DATA	23/11/2025
	<p>La connessione al Database fallisce.</p> <p>2.a2 Sistema:</p> <p>Interrompe la procedura di avvio e scrive l'errore nel log di sistema. Il server rimane spento.</p>		

3.7.2. Shut Down

IDENTIFICATIVO	UC_SD	DATA	23/11/2025
NOME	Arresto del Server	VERSIONE	1.0
AUTORE	Bruno Santo & Emiliano Di Giuseppe		
DESCRIZIONE	Lo UC fornisce la funzionalità di arresto controllato (graceful shutdown) del Server.		
ATTORE PRINCIPALE	ADMIN (TECNICO): Vuole arrestare il Server per manutenzione o spegnimento.		
ENTRY CONDITION	Il Server è avviato e operativo.		
EXIT CONDITION ON SUCCESS	Il Server viene arrestato correttamente senza perdita di dati.		
EXIT CONDITION ON FAILURE	Il Server viene arrestato forzatamente (possibile perdita dati in transito).		

IDENTIFICATIVO	UC_SD	DATA	23/11/2025
RILEVANZA	Elevata	FREQUENZA	1/mese
FLUSSO DI EVENTI PRINCIPALE	<p>Admin:</p> <ol style="list-style-type: none"> Utilizza il comando di arresto. <p>Sistema:</p> <ol style="list-style-type: none"> Blocca l'accettazione di nuove richieste HTTP (stop alle nuove sottomissioni di post). Attende il completamento delle analisi AI attualmente in corso (grace period). Effettua il commit finale dei dati sul Database. Chiude le connessioni al Database e arresta il processo Flask. 		
FLUSSO DI ERRORE	<p>3.a1 Sistema:</p> <p>Alcuni processi (es. analisi AI) non terminano entro il tempo limite (timeout).</p> <p>3.a2 Sistema:</p> <p>Forza la chiusura dei thread pendenti, segnalando l'evento nel log.</p>		

IDENTIFICATIVO	UC_SD	DATA	23/11/2025
	<p>3.a3 Sistema:</p> <p>Procede con la chiusura del Database e lo spegnimento.</p>		

3.7.3. Failures

IDENTIFICATIVO	UC_FA	DATA	22/11/2025
NOME	Failure: Indisponibilità AIService	VERSIONE	1.0
AUTORE	Bruno Santo & Emiliano Di Giuseppe		
DESCRIZIONE	Lo UC descrive il comportamento del sistema quando il modulo AI non risponde alle richieste di analisi.		
ATTORE PRINCIPALE	SISTEMA (Automatico)		
ENTRY CONDITION	Il Sistema tenta di inviare un post all'AIService ma riceve un errore (500) o timeout.		
EXIT CONDITION ON SUCCESS	Il Sistema gestisce l'errore informando l'utente senza crashare.		
RILEVANZA	Critica	FREQUENZA	Imprevedibile
FLUSSO DI EVENTI PRINCIPALE	<p>Sistema:</p> <p>1. Rileva il timeout nella chiamata REST verso l'AIService.</p>		

IDENTIFICATIVO	UC_FA	DATA	
	<p>2. Registra l'incidente nel Log degli Errori (Timestamp, ID Post).</p> <p>3. Blocca temporaneamente la pubblicazione del post specifico.</p> <p>4. Restituisce all'Utente un messaggio di errore specifico: "Servizio di verifica momentaneamente non disponibile. Riprova più tardi.".</p> <p>5. Invia una notifica di allerta all'Admin Tecnico.</p>		

3.7.4. Gestione dei fallimenti

In questa sezione vengono definite le strategie per mitigare i rischi hardware e software specifici dell'architettura proposta.

- **Fallimento del Modulo AI:** Nel caso in cui il servizio esterno di IA non sia raggiungibile, il sistema è progettato per **non bloccare la navigazione**. La funzionalità di "Pubblicazione Post" verrà disabilitata temporaneamente (o restituirà errore all'utente), ma la lettura del feed e l'accesso alla dashboard rimarranno attivi.
- **Fallimento Dati Persistenti (Database):** È previsto un backup automatico giornaliero (snapshot) del database MySQL. In caso di corruzione dei dati, l'Admin provvederà al ripristino dell'ultimo backup valido.
- **Interruzione Alimentazione:** Il server è collegato a un gruppo di continuità (UPS). In caso di blackout, l'UPS invia un segnale al sistema operativo che avvia automaticamente la procedura di **Shut down (UC_SD)** descritta sopra, garantendo la chiusura corretta delle connessioni al database prima dello spegnimento fisico.

4. Subsystem services

- **Gestione utenza**

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità permette ad un cittadino/professionista di registrarsi al sistema	GestioneUtenza
Registrazione	Questa funzionalità permette ad un cittadino/professionista di registrarsi al sistema	GestioneUtenza
Logout	Questa funzionalità permette di effettuare l'uscita dal sistema.	GestioneUtenza

- **Gestione Pubblicazioni**

Servizio	Descrizione	Interfaccia
Pubblicazione	Questa funzionalità permette ad un utente di pubblicare news	GestioneNotizie

- **Gestione Appelli**

Servizio	Descrizione	Interfaccia
appellare	Questa funzionalità permette ad un utente di appellarsi alla decisione del blocco della news fatta dall'IA	GestioneAppelli
Presa in carico	Questa funzionalità permette ad un fact-checker di prendere in carico un appello.	GestioneAppelli
Elaborazione	Questa funzionalità permette ad un fact-checker di operare azioni su un appello	GestioneAppelli

- **Gestione Segnalazioni**

Servizio	Descrizione	Interfaccia
Segnalare	Questa funzionalità permette ad un utente di segnalare una news pubblicata da un altro utente	GestioneSegnalazioni
Presa in carico	Questa funzionalità permette ad un fact-checker di prendere in carico una segnalazione	GestioneSegnalazioni
Elaborazione	Questa funzionalità permette ad un admin di operare azioni su una segnalazione	GestioneSegnalazioni

- Gestione persistenza

Servizio	Descrizione	Interfaccia
Salvataggio	Questa funzionalità permette al sistema di salvare i dati necessari per il funzionamento, all'interno del database.	GestionePersistenza
Eliminazione	Questa funzionalità permette al sistema di eliminare i dati all'interno del database.	GestionePersistenza
Modifica	Questa funzionalità permette al sistema di modificare i dati all'interno del database.	GestionePersistenza
Ricerca	Questa funzionalità permette al sistema di ricercare dati all'interno del database.	GestionePersistenza

5. Glossary

TERMINE	SIGNIFICATO
System Design Document (SDD)	Documento che definisce gli obiettivi di progettazione, l'architettura software e la decomposizione in sottosistemi del sistema proposto.
Requirement Analysis Document (RAD)	Documento preliminare che descrive i requisiti funzionali e non funzionali, gli attori e gli scenari del sistema.
Design Goal	Obiettivi di design progettati per il sistema (es. Performance, Affidabilità) che guidano le scelte architettoniche.
Sottosistema	Un sottoinsieme dei servizi del dominio applicativo (es. Gestione Pubblicazioni), formato da componenti legati da una relazione funzionale.
Front-end	La parte visibile all'utente (Interfaccia Web) con cui egli può interagire, realizzata in HTML/CSS.
Back-end	La parte che si occupa di gestire il funzionamento del sistema, la logica applicativa e la comunicazione con il Database e il servizio AI.
Database Relazionale	Architettura software che organizza i dati in tabelle correlate (nel nostro caso MySQL), garantendo l'integrità referenziale.
Persistenza	Caratteristica dei dati di sopravvivere all'esecuzione del programma che li ha creati, venendo salvati in uno storage non volatile (Database).
Framework	Un'architettura logica di supporto che fornisce strumenti per facilitare il lavoro di programmazione (es. Flask per il backend Python).
MVC (Model-View-Controller)	Pattern architettonico che separa la logica di presentazione (View), la logica di business (Controller) e la gestione dei dati (Model).
Model	Componente MVC che contiene i metodi di accesso ai dati e rappresenta le entità del dominio (es. Post, Account).
View	Componente MVC che si occupa di visualizzare i dati all'Utente e gestisce l'interazione fra quest'ultimo e l'infrastruttura.
Controller	Componente MVC che riceve i comandi dell'Utente e reagisce eseguendo operazioni che coinvolgono il Model e il servizio AI.

TERMINE	SIGNIFICATO
Client	Componente (Browser Web) che accede a servizi e risorse forniti dal Server.
Server	Componente che gestisce il traffico di informazioni, ospita l'applicazione Python e fornisce servizi attraverso la rete.
API REST	Interfaccia di programmazione che permette la comunicazione tra componenti software distinti; utilizzata per collegare il Backend al Servizio AI esterno.
AI (Intelligenza Artificiale)	Il modulo software esterno incaricato di analizzare semanticamente il testo delle notizie e produrre una valutazione.
Score (Punteggio)	Valore numerico (decimale) generato dall'IA che rappresenta la stima di attendibilità di un contenuto.
Fact-checker	Ruolo utente con privilegi speciali di moderazione, incaricato di revisionare manualmente gli appelli e le segnalazioni.
Startup	Procedura di avvio del sistema server e inizializzazione delle connessioni (Database e AI).
Shutdown	Procedura di spegnimento controllato del sistema con salvataggio dello stato corrente.
Backup	Procedura di messa in sicurezza dei dati attraverso la creazione di una copia di riserva del Database MySQL.