



Code7Crusaders

Software Development Team

Analisi Dei Requisiti

Membri del Team:

Enrico Cotti Cottini, Gabriele Di Pietro, Tommaso Diviesti
Francesco Lapenna, Matthew Pan, Eddy Pinarello, Filippo Rizzolo

| Ver | Data | Redattore | Verificatore | Descrizione |
|-----|------------|----------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2.1 | 28/02/2025 | Enrico Cotti Cottini | | Aggiornamento requisiti funzionali e tracciamenti relativi, definito U.C.23, correzione stesura U.C.2.1 |
| 2.0 | 19/02/2025 | Enrico Cotti Cottini | Gabriele Di Pietro | Approvazione documento |
| 1.7 | 18/02/2025 | Enrico Cotti Cottini | Eddy Pinarello | Aggiunti requisiti qualitativi e di vincolo per il Piano di Qualifica |
| 1.6 | 18/02/2025 | Enrico Cotti Cottini | Eddy Pinarello | Aggiunti nuovi requisiti funzionali e di vincolo per la gestione del chatbot e delle metriche. Integrazione con modelli di embedding e LangChain. Aggiunti requisiti RFO43, RFD44, RFD45, RFO46, RVO18, RVD19, RVO20 nel tracciamento |
| 1.5 | 17/02/2025 | Eddy Pinarello | Enrico Cotti Cottini | Rivisti requisiti di vincolo, corretti requisiti errati, implementati requisiti sulle tecnologie, tracciati nuovi casi d'uso |
| 1.4 | 15/02/2025 | Gabriele Di Pietro | Enrico Cotti Cottini | Aggiunto tracciamento Caso d'uso - Requisito |
| 1.3 | 14/02/2025 | Filippo Rizzolo | Enrico Cotti Cottini | Correzione del caso d'uso U.C.20 |
| 1.2 | 11/02/2025 | Filippo Rizzolo | Enrico Cotti Cottini | Correzione dei casi d'uso U.C.14, U.C.15 e U.C.17 |
| 1.1 | 10/02/2025 | Gabriele Di Pietro | Filippo Rizzolo | Prima stesura del documento dopo correzioni Colloquio RTB |
| 1.0 | 04/02/2025 | Gabriele Di Pietro | Tommaso Divesti | Correzione imprecisioni e approvazione documento |
| 0.8 | 27/01/2025 | Gabriele Di Pietro | Eddy Pinarello | Correzioni casi d'uso e aggiunti nuovi requisiti |
| 0.7 | 14/01/2025 | Gabriele Di Pietro | Enrico Cotti Cottini | Aggiunte tabelle requisiti |
| 0.6 | 17/12/2024 | Enrico Cotti Cottini | Eddy Pinarello | Ridefinizione dell'architettura dopo un'analisi approfondita dei mezzi |
| 0.5 | 16/12/2024 | Gabriele Di Pietro | Enrico Cotti Cottini | Inizio stesura requisiti |
| 0.3 | 06/12/2024 | Gabriele Di Pietro | Enrico Cotti Cottini | Aggiunti casi d'uso |
| 0.2 | 20/11/2024 | Enrico Cotti Cottini | Gabriele Di Pietro | Prima stesura del documento |

Indice

| | | |
|----------|-------------------------------------------------------------|-----------|
| 1 | Introduzione | 5 |
| 1.1 | Scopo del documento | 5 |
| 1.2 | Scopo del prodotto | 5 |
| 1.3 | Glossario | 5 |
| 1.4 | Approccio Incrementale | 5 |
| 1.5 | Riferimenti | 5 |
| 1.5.1 | Riferimenti normativi | 5 |
| 1.5.2 | Riferimenti informativi | 6 |
| 2 | Descrizione del prodotto | 7 |
| 2.1 | Obiettivi del prodotto | 7 |
| 2.2 | Architettura del prodotto | 7 |
| 2.3 | Implementazione scelta: LLM e strumenti | 7 |
| 2.3.1 | Scelta dell'LLM | 7 |
| 2.3.2 | Pipeline di Implementazione | 8 |
| 2.3.3 | Motivazione degli Strumenti Scelti | 8 |
| 2.4 | Funzionalità del prodotto | 9 |
| 2.5 | Tecnologie utilizzate | 10 |
| 2.6 | Utenti finali | 10 |
| 3 | Casi d'uso | 11 |
| 3.1 | Introduzione | 11 |
| 3.1.1 | Codifica dei casi d'uso | 11 |
| 3.2 | Elenco dei Casi d'uso | 12 |
| 3.2.1 | U.C.1 Scrivi Messaggio | 12 |
| 3.2.2 | U.C.2 Visualizza Risposta | 13 |
| 3.2.3 | U.C.2.1 Prodotto non trovato | 14 |
| 3.2.4 | U.C.3 Seleziona Template | 15 |
| 3.2.5 | U.C.4 Visualizza lista conversazioni precedenti | 16 |
| 3.2.6 | U.C.5 Visualizza Singola Conversazione Precedente | 17 |
| 3.2.7 | U.C.6 Login | 18 |
| 3.2.8 | U.C.6.1 Inserisci Username | 19 |
| 3.2.9 | U.C.6.2 Inserisci Password | 20 |
| 3.2.10 | U.C.6.3 Password Errata | 21 |
| 3.2.11 | U.C.6.4 Username Errato | 22 |
| 3.2.12 | U.C.6.5 Caratteri non Validi | 23 |
| 3.2.13 | U.C.7 Sign Up | 24 |
| 3.2.14 | U.C.7.1 Inserisci Username | 25 |
| 3.2.15 | U.C.7.2 Inserisci Password | 26 |
| 3.2.16 | U.C.7.3 Caratteri non Validi | 27 |
| 3.2.17 | U.C.7.4 Username troppo Lungo | 28 |
| 3.2.18 | U.C.7.5 Password troppo Lunga | 29 |
| 3.2.19 | U.C.7.6 Username già presente | 30 |
| 3.2.20 | U.C.8 Salva Chat | 31 |
| 3.2.21 | U.C.8.1 Memoria Piena | 32 |
| 3.2.22 | U.C.9 Feedback Chat | 33 |
| 3.2.23 | U.C.10: Creazione di un nuovo template | 34 |

| | | |
|----------|------------------------------------------------------------------|-----------|
| 3.2.24 | U.C.11: Modifica di un template esistente | 35 |
| 3.2.25 | U.C.12: Elimina un template esistente | 36 |
| 3.2.26 | U.C.13 Controllo Validità Formato | 37 |
| 3.2.27 | U.C.14: Visualizzazione delle metriche generali | 38 |
| 3.2.28 | U.C.15: Visualizzazione Feedback Utenti | 39 |
| 3.2.29 | U.C.16: Importazione di Dati | 40 |
| 3.2.30 | U.C.16.1: Validazione Formato | 41 |
| 3.2.31 | U.C.17: Esportazione di Metriche nel Sistema | 42 |
| 3.2.32 | U.C.18 Elimina Chat | 43 |
| 3.2.33 | U.C.19: Riprendi Conversazione | 44 |
| 3.2.34 | U.C.20: Risposta alla Richiesta dall'Amministratore | 45 |
| 3.2.35 | U.C.21: Cambio stato Richiesta | 46 |
| 3.2.36 | U.C.22 Invio Richiesta a un Operatore Umano | 47 |
| 3.2.37 | U.C.23: Visualizzazione Template di Domanda e Risposta | 48 |
| 4 | Requisiti | 49 |
| 4.1 | Classificazione dei requisiti | 49 |
| 4.2 | Fonti dei requisiti | 49 |
| 4.3 | Codifica dei requisiti | 50 |
| 4.4 | Requisiti funzionali | 51 |
| 4.5 | Requisiti qualitativi | 54 |
| 4.6 | Requisiti di vincolo | 55 |
| 4.7 | Tracciamento | 56 |
| 4.7.1 | Requisito - Fonte | 56 |
| 4.8 | Caso d'uso - Requisito | 58 |

Elenco delle tabelle

| | | |
|---|----------------------------------|----|
| 1 | Requisiti funzionali | 53 |
| 2 | Requisiti qualitativi | 54 |
| 3 | Requisiti di vincolo | 55 |
| 4 | Requisito - Fonte | 57 |
| 5 | Caso d'uso - Requisito | 58 |

Elenco delle figure

| | | |
|----|-------------------------------------------------------------|----|
| 1 | Architettura del prodotto | 9 |
| 2 | Logo di OpenAI | 10 |
| 3 | Logo di LangChain | 10 |
| 4 | U.C.1 Scrivi Messaggio | 12 |
| 5 | U.C.2 Visualizza Risposta | 13 |
| 6 | U.C.2.1 Prodotto non Trovato | 14 |
| 7 | U.C.3 Seleziona Template | 15 |
| 8 | U.C.4 Visualizza lista conversazioni precedenti | 16 |
| 9 | U.C.5 Visualizza Singola Conversazione Precedente | 17 |
| 10 | U.C.6 Login | 18 |
| 11 | U.C.6.1 Inserisci Username | 19 |
| 12 | U.C.6.2 Inserisci Password | 20 |

| | | |
|----|---------------------------------------------------------------|----|
| 13 | U.C.6.3 Password Errata | 21 |
| 14 | U.C.6.4 Username Errato | 22 |
| 15 | U.C.6.5 Caratteri non Validi | 23 |
| 16 | U.C.7 Sign Up | 24 |
| 17 | U.C.7.1 Inserisci Username | 25 |
| 18 | U.C.7.2 Inserisci Password | 26 |
| 19 | U.C.7.3 Caratteri non Validi | 27 |
| 20 | U.C.7.4 Username troppo Lungo | 28 |
| 21 | U.C.7.5 Password troppo lunga | 29 |
| 22 | U.C.7.6 Username già presente | 30 |
| 23 | U.C.8 Salva Chat | 31 |
| 24 | U.C.8.1 Memoria Piena | 32 |
| 25 | U.C.9 Feedback Chat | 33 |
| 26 | U.C.10: Creazione di un nuovo template | 34 |
| 27 | U.C.11: Modifica di un template esistente | 35 |
| 28 | U.C.12: Elimina un template esistente | 36 |
| 29 | U.C.13 Controllo Validità Formato | 37 |
| 30 | U.C.14: Visualizzazione delle metriche generali | 38 |
| 31 | U.C.15: Visualizzazione Feedback Utenti | 39 |
| 32 | U.C.16: Importazione di Dati | 40 |
| 33 | U.C.16.1: Validazione Formato | 41 |
| 34 | U.C.17: Esportazione di Dati nel Sistema | 42 |
| 35 | U.C.18 Elimina Chat | 43 |
| 36 | U.C.19: Riprendi Conversazione | 44 |
| 37 | U.C.20: Risposta alla Richiesta dall'Amministratore | 45 |
| 38 | U.C.21: Cambio stato Richiesta | 46 |
| 39 | U.C.22 Invio Richiesta a un Operatore Umano | 47 |

1 Introduzione

1.1 Scopo del documento

Questo documento mira a offrire una panoramica dettagliata del prodotto, delineando i bisogni degli utenti in base alle diverse categorie individuate durante l'analisi del capitolato e gli incontri con il committente. L'obiettivo è identificare chiaramente tutti i requisiti e gli attori coinvolti nel sistema software, garantendo una descrizione accurata delle componenti del programma e una visione strutturata delle attività da svolgere.

1.2 Scopo del prodotto

Lo scopo del prodotto è realizzare un **Assistente Virtuale basato su LLM**, per supportare aziende produttrici di bevande nel fornire informazioni dettagliate e personalizzate sui loro prodotti. Il sistema si rivolge principalmente ai proprietari di locali, consentendo loro di ottenere risposte rapide e precise su caratteristiche, disponibilità e dettagli delle bevande, come se interagissero con uno specialista umano.

L'obiettivo è sostituire e migliorare il supporto degli specialisti tradizionali, rendendo le informazioni accessibili 24/7 tramite una **web app** intuitiva. Questa piattaforma permetterà agli utenti di formulare domande in linguaggio naturale e ricevere risposte pertinenti, basate sui dati forniti dalle aziende.

Il sistema sarà progettato per garantire flessibilità e scalabilità^G, integrando dati relativi ai prodotti nei database aziendali e utilizzando le capacità avanzate di un LLM^G per comprendere e generare risposte personalizzate. Questo approccio semplificherà il processo decisionale per i clienti, migliorando l'efficienza operativa delle aziende e offrendo un'esperienza utente fluida e moderna.

1.3 Glossario

Per garantire una chiara comprensione della terminologia utilizzata nel documento, è stato predisposto un *Glossario*^G in un file dedicato. Questo strumento serve a evitare ambiguità nella definizione dei termini impiegati nell'attività progettuale, offrendo descrizioni precise e condivise.

1.4 Approccio Incrementale

Questo documento è stato elaborato seguendo un approccio incrementale, consentendo di apportare modifiche in modo agile nel tempo, in base alle necessità concordate tra il gruppo di lavoro e il proponente. Di conseguenza, la versione attuale non deve essere considerata come definitiva o completa.

1.5 Riferimenti

1.5.1 Riferimenti normativi

- **Capitolato C7 LLM: ASSISTENTE VIRTUALE**
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C7.pdf>
- **Regolamento del progetto didattico**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf>
- **Norme di Progetto v.1.0**
https://code7crusaders.github.io/docs/RTB/documentazione_interna/norme_di_progetto.html

1.5.2 Riferimenti informativi

- **Slide Corso Ingegneria del software: Analisi dei Requisiti**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T05.pdf>
- **Slide Corso Ingegneria del software: Diagrammi delle classi**
<https://www.math.unipd.it/~rcardin/swea/2023/Diagrammi%20delle%20Classi.pdf>
- **Slide Corso Ingegneria del software: Diagrammi dei casi d'uso**
<https://www.math.unipd.it/~rcardin/swea/2022/Diagrammi%20Use%20Case.pdf>
- **Glossario^G v.1.0**
https://code7crusaders.github.io/docs/RTB/documentazione_interna/glossario.html
- **Analisi LLM**
https://code7crusaders.github.io/docs/altri_documenti/analisi_modelli_firmato.html
- **Analisi framework frontend**
https://code7crusaders.github.io/docs/altri_documenti/analisi_frontend_firmato.html
- **Analisi framework backend**
https://code7crusaders.github.io/docs/altri_documenti/analisi_framework_backend.html
- **Analisi database Vettoriale**
https://code7crusaders.github.io/docs/altri_documenti/analisi_dbvettoriale.html
- **LangChain^G**
<https://python.langchain.com/docs/introduction/>
- **OpenAI**
<https://openai.com/>

2 Descrizione del prodotto

2.1 Obiettivi del prodotto

Il progetto ha come obiettivo la realizzazione di una piattaforma che consenta di gestire un assistente virtuale per la conoscenza e la descrizione di bevande, sfruttando un'infrastruttura basata su modelli linguistici di grandi dimensioni. La piattaforma dovrà supportare le richieste degli utenti in modo rapido, preciso e sempre disponibile, eliminando la necessità di uno specialista fisico. Essa permetterà la consultazione di informazioni dettagliate su prodotti come caratteristiche, formati disponibili e suggerimenti d'uso, adattandosi alle esigenze specifiche dei clienti e garantendo un'interazione fluida in linguaggio naturale. L'assistente virtuale sarà progettato per integrarsi con database aziendali, sfruttando le informazioni esistenti per rispondere alle domande in modo contestualizzato e accurato.

2.2 Architettura del prodotto

I componenti del prodotto sono:

- **Database Relazionale^G**: Questo componente memorizza i dati strutturati dell'azienda, come descrizioni di prodotti, ingredienti, specifiche tecniche e altro. È il punto di partenza per acquisire informazioni utili che saranno processate e utilizzate dal sistema. Supporta query SQL per consentire l'accesso rapido e organizzato ai dati.
- **Embedding^G Model**: L'Embedding Model è un modello pre-addestrato in grado di trasformare il testo in rappresentazioni numeriche preservando il significato semantico. Viene utilizzato sia per i dati aziendali durante l'addestramento che per le domande poste dagli utenti. Gli embedding risultanti permettono confronti efficienti nel database vettoriale^G.
- **Database Vettoriale^G**: Questo componente archivia i vettori generati dall'Embedding^G Model. Utilizza indicizzazione ottimizzata per operazioni di *nearest neighbor search*^G, permettendo di trovare rapidamente i vettori più simili a una query. È il cuore della fase di recupero delle informazioni nel sistema.
- **LLM^G**: Il Large Language Model riceve in input il contesto fornito dal database vettoriale^G e la domanda dell'utente. Grazie alla sua capacità generativa, l'LLM^G riceve in input il contesto fornito dal database vettoriale e la domanda dell'utente. Grazie alla sua capacità generativa, il LLM elabora risposte dettagliate e accurate, combinando i dati presenti con la comprensione del linguaggio naturale.
- **Web App**: La Web App è l'interfaccia attraverso la quale gli utenti interagiscono con il sistema. Fornisce un'esperienza semplice e intuitiva per inserire domande e visualizzare risposte. Comunica con il backend tramite API REST^G per garantire un'interazione rapida e scalabile.

2.3 Implementazione scelta: LLM e strumenti

2.3.1 Scelta dell'LLM

Dopo un'attenta analisi comparativa tra i modelli di Huggingface (BLOOM e varianti) e OpenAI (GPT-4o e GPT-4omini) nel file di **analisi modelli.pdf**, si è optato per l'utilizzo di GPT-4omini di OpenAI. La scelta è motivata dai seguenti fattori:

- **Prestazioni**: GPT-4omini offre risultati superiori rispetto ai modelli open-source in termini di accuratezza e capacità di generazione del linguaggio, come evidenziato dai benchmark^G **analisi modelli**.

- **Scalabilità^G**: Le API di OpenAI garantiscono un'infrastruttura cloud scalabile, eliminando i costi e la complessità legati alla gestione locale di modelli di grandi dimensioni.
- **Costi ottimizzati**: Il costo basato sui token^G elaborati permette un utilizzo flessibile e sostenibile, adattandosi alle esigenze di carico variabile del sistema.

2.3.2 Pipeline di Implementazione

L'implementazione sfrutta un'architettura RAG (Retrieval-Augmented Generation), integrando i seguenti componenti:

1. **Generazione degli Embedding**: Le informazioni aziendali (es. cataloghi di prodotti) vengono pre-processate e trasformate in vettori numerici tramite BERT^G Embedding^G: Le informazioni aziendali (es. cataloghi di prodotti) vengono pre-processate e trasformate in vettori numerici tramite BERT di Huggingface.
2. **Archiviazione degli Embedding^G**: I vettori generati vengono memorizzati e indicizzati utilizzando FAISS^G, che consente un recupero efficiente dei dati rilevanti.
3. **Integrazione con LLM**: Le domande degli utenti, trasformate in embedding, vengono confrontate con il database vettoriale^G. Il contesto recuperato viene passato a GPT-4omini per generare risposte accurate e personalizzate.
4. **Web App e API REST^G**: La comunicazione tra frontend (React) e backend avviene tramite API REST^G, garantendo tempi di risposta rapidi.

2.3.3 Motivazione degli Strumenti Scelti

- **BERT + FAISS**: Permette di ottimizzare la fase di retrieval, migliorando l'efficienza della ricerca semantica.
- **GPT-4omini**: Offre risposte di alta qualità con costi prevedibili, bilanciando prestazioni e budget di progetto.
- **LangChain^G**: Facilita l'orchestrazione dell'intera pipeline RAG, riducendo i tempi di sviluppo e semplificando l'integrazione dei componenti.

Questa architettura garantisce un sistema performante, flessibile e scalabile, in grado di soddisfare le esigenze degli utenti finali e delle aziende committenti.

1. Il sistema riceve in ingresso i dati aziendali strutturati (es. descrizioni, ingredienti).
2. I documenti vengono pre-processati e suddivisi in blocchi di dati.
3. I blocchi di testo sono trasformati in vettori numerici tramite l'Embedding^G Model.
4. I vettori generati sono memorizzati nel Database Vettoriale^G e indicizzati.

Flusso di Interazione con l'Utente

1. L'utente invia una domanda tramite la Web App.
2. La domanda viene inoltrata al Web Server tramite API REST^G.
3. L'Embedding^G Model trasforma la domanda in un vettore numerico.
4. Il vettore della domanda viene confrontato con i vettori nel Database Vettoriale^G.
5. Viene restituito il contesto più rilevante, insieme alla domanda, all'LLM^G.
6. L'LLM^G elabora la risposta utilizzando il contesto fornito.
7. La risposta viene inoltrata al dispositivo dell'utente tramite API REST^G.



Figura 1: Architettura del prodotto

2.4 Funzionalità del prodotto

Il prodotto avrà il compito di interagire con i propri utenti attraverso una webapp, rispondendo a domande su cataloghi di bevande. Ogni risposta sarà generata in linguaggio naturale, elaborando i dati tramite LLM^G. Le funzionalità principali includono:

- **Interfaccia utente interattiva:** consente agli utenti di porre domande sul catalogo (*es. descrizione di un prodotto o disponibilità in magazzino*) e di ricevere risposte immediate.
- **Motore di ricerca intelligente:** utilizza un sistema di embedding^G per trovare corrispondenze semantiche tra le domande degli utenti e i dati aziendali, estrae il contesto dai dati aziendali per fornire all'LLM^G dati accurati da elaborare.
- **Gestione dei dati:** accesso ai dettagli dei prodotti memorizzati in database relazionali, garantendo aggiornamenti in tempo reale. Costruzione di un database vettoriale^G per l'embedding^G delle parole.
- **Personalizzazione tramite backend:** gli amministratori possono configurare risposte predefinite (template^G di domanda e risposta), monitorare l'utilizzo e migliorare il sistema tramite feedback utente.

- **Apprendimento continuo:** il sistema evolve grazie ai feedback raccolti dagli utenti, migliorando la qualità delle risposte.
- **Compatibilità multi-dispositivo:** la piattaforma è progettata per essere accessibile 24/7 da mobile e desktop.

Il prodotto garantirà inoltre scalabilità^G e flessibilità, adattandosi a un'ampia gamma di aziende che desiderano offrire ai propri clienti un'esperienza di interazione avanzata e intuitiva.

2.5 Tecnologie utilizzate

Le tecnologie selezionate per la realizzazione del prodotto software sono le seguenti:

- **Embedding Model:** BERT^G **Embedding^G** Model: BERT (Huggingface) oppure modelli di embedding di Openai (es. Embedding-large) per la trasformazione del testo in vettori semantici. Questo modello viene utilizzato per generare embedding^G efficaci nella fase RAG (Retrieval-Augmented Generation).
- **Database Relazionale^G:** *PostgreSQL^G* per memorizzare i dati strutturati come cataloghi di prodotti, descrizioni e metadati.
- **Database Vettoriale^G:** *FAISS^G* per l'archiviazione e il recupero veloce degli embedding e permette un'indicizzazione ottimizzata per la *nearest neighbor search^G*.
- **LLM^G:** OpenAI GPT-4, integrato tramite API OpenAI, per garantire prestazioni elevate e costi scalabili.



Figura 2: Logo di OpenAI

- **Web App:** *React* per lo sviluppo dell'interfaccia utente, combinata con API REST^G per una comunicazione efficiente con il backend^G.
- **LangChain^G:** Libreria di orchestrazione utilizzata per integrare LLM^G, database vettoriali e pipeline di retrieval.



Figura 3: Logo di LangChain

2.6 Utenti finali

Il prodotto è rivolto a aziende che desiderano offrire un servizio di assistenza clienti automatizzato e personalizzato. Gli utenti finali sono quindi i clienti delle aziende che interagiranno con l'assistente virtuale per ottenere informazioni sui prodotti e ricevere supporto.

3 Casi d'uso

3.1 Introduzione

In questa sezione vengono presentati i casi d'uso individuati durante l'attività di analisi, condotta a partire dal capitolato d'appalto e dagli incontri con il proponente. Gli attori vengono identificati in base alla gerarchia trovata e alle funzionalità potenziali rilevate.

3.1.1 Codifica dei casi d'uso

I casi d'uso sono codificati utilizzando la seguente notazione:

- **UC[ID-Principale][ID-Sottocaso]:** Identificativo univoco del caso d'uso, composto da un ID principale che identifica il caso principale e, se necessario, da un ID del sottocaso.
- **Titolo:** Breve descrizione del caso d'uso.
- **Attori:** Elenco degli attori coinvolti nel caso d'uso.
- **Precondizioni:** Condizioni che devono essere vere prima che il caso d'uso possa iniziare.
- **Postcondizioni:** Condizioni che devono essere vere dopo che il caso d'uso è stato completato con successo.
- **Scenario principale:** Descrizione dettagliata del flusso di eventi principale del caso d'uso.
- **Generalizzazioni:** Eventuali casi d'uso generalizzati.
- **Estensioni:** Eventuali casi d'uso estesi.
- **Inclusione:** Eventuali inclusioni.

3.2 Elenco dei Casi d'uso

3.2.1 U.C.1 Scrivi Messaggio

- **Attore:** Utente
- **Precondizioni:** Utente che ha acceduto nel sistema e vuole domandare al bot delle informazioni riguardo un prodotto o una serie di prodotti
- **Postcondizioni:** L'utente ha inviato il messaggio al bot per ricevere una risposta.
- **Scenario principale:** L'utente apre la chat, inserisce la domanda nella casella di testo e invia il messaggio premendo il pulsante "Invia".
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 4: U.C.1 Scrivi Messaggio

3.2.2 U.C.2 Visualizza Risposta

- **Attore:** Utente
- **Attore Secondario:** OpenAi
- **Precondizioni:** L'utente ha inviato una domanda al bot e attende una risposta.
- **Postcondizioni:** L'utente riceve una risposta dal bot nella finestra della chat, basata sui dati caricati nel database vettoriale, fornendo una risposta coerente.
- **Scenario principale:** Dopo aver inviato una domanda, l'utente attende la risposta elaborata dall' LLM^G , che utilizza come contesto le informazioni dei dati caricati nel chatbot. La risposta viene visualizzata nella finestra della chat.
- **Generalizzazioni:** -
- **Estensioni:** U.C.2.1
- **Inclusione:** -



Figura 5: U.C.2 Visualizza Risposta

3.2.3 U.C.2.1 Prodotto non trovato

- **Attore:** Utente
- **Precondizioni:** L'utente richiede informazioni su un prodotto non presente nel database del sistema.
- **Postcondizioni:** L'utente riceve una risposta da parte del LLM^G, non è possibile elaborare l'informazione.
- **Scenario principale:** L'utente invia una domanda su un prodotto specifico, il sistema cerca nel database ma non trova risultati. Un messaggio di errore avvisa l'utente che il prodotto non è presente nel sistema.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 6: U.C.2.1 Prodotto non Trovato

3.2.4 U.C.3 Seleziona Template

- **Attore:** Utente
- **Precondizioni:** L'utente ha effettuato l'accesso ed è nella chat. Vuole selezionare una domanda predefinita tra quelle suggerite dal sistema chiamati *Template*^G.
- **Postcondizioni:** L'utente riceve una risposta templatizzata senza dover formulare una domanda manuale (senza dover chiamare l'*LLM*^G).
- **Scenario principale:** L'utente visualizza un elenco di domande suggerite, ne seleziona una e il sistema fornisce immediatamente una risposta.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 7: U.C.3 Seleziona Template

3.2.5 U.C.4 Visualizza lista conversazioni precedenti

- **Attore:** Utente
- **Precondizioni:** L'utente ha effettuato l'accesso e in passato ha salvato almeno una conversazione.
- **Postcondizioni:** L'utente visualizza l'elenco delle conversazioni salvate.
- **Scenario principale:** L'utente accede alla sezione delle conversazioni e visualizza una lista ordinata delle sue conversazioni passate.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 8: U.C.4 Visualizza lista conversazioni precedenti

3.2.6 U.C.5 Visualizza Singola Conversazione Precedente

- **Attore:** Utente
- **Precondizioni:** L'utente ha effettuato l'accesso e in passato ha salvato almeno una conversazione.
- **Postcondizioni:** L'utente visualizza la conversazione salvata con i messaggi in uscita per il chat bot e in entrata i suoi precedenti messaggi, nella finestra della chat.
- **Scenario principale:** L'utente accede alla sezione delle conversazioni e visualizza una delle sue conversazioni passate.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 9: U.C.5 Visualizza Singola Conversazione Precedente

3.2.7 U.C.6 Login

- **Attore:** Utente
- **Precondizioni:** L'utente è registrato e desidera accedere al sistema.
- **Postcondizioni:** L'utente accede con successo al sistema e può utilizzare le funzionalità disponibili.
- **Scenario principale:** L'utente inserisce il proprio username e password nei campi di accesso ed effettua l'accesso. Il sistema verifica le credenziali e consente l'accesso.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** U.C.6.1, U.C.6.2



Figura 10: U.C.6 Login

3.2.8 U.C.6.1 Inserisci Username

- **Attore:** Utente
- **Precondizioni:** L'utente desidera accedere al sistema.
- **Postcondizioni:** L'username è stato inserito correttamente nel campo corrispondente.
- **Scenario principale:** L'utente digita il proprio username nel campo di testo dedicato e procede con l'autenticazione.
- **Generalizzazioni:** -
- **Estensioni:** U.C.6.4, U.C.6.5
- **Inclusione:** -



Figura 11: U.C.6.1 Inserisci Username

3.2.9 U.C.6.2 Inserisci Password

- **Attore:** Utente
- **Precondizioni:** L'utente desidera accedere al sistema.
- **Postcondizioni:** La password è stata inserita correttamente nel campo corrispondente.
- **Scenario principale:** L'utente digita la password nel campo di testo e procede con l'autenticazione.
- **Generalizzazioni:** -
- **Estensioni:** U.C.6.3, U.C.6.5
- **Inclusione:** -



Figura 12: U.C.6.2 Inserisci Password

3.2.10 U.C.6.3 Password Errata

- **Attore:** Utente
- **Precondizioni:** L'utente ha inserito una password non corrispondente a quella registrata.
- **Postcondizioni:** L'accesso viene negato e il sistema visualizza un messaggio di errore.
- **Scenario principale:** L'utente inserisce una password sbagliata. Il sistema verifica le credenziali, rileva l'errore e mostra un messaggio che informa l'utente dell'errore.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 13: U.C.6.3 Password Errata

3.2.11 U.C.6.4 Username Errato

- **Attore:** Utente
- **Precondizioni:** L'utente ha inserito un username non registrato o con errori di battitura.
- **Postcondizioni:** L'accesso viene negato e il sistema visualizza un messaggio di errore.
- **Scenario principale:** L'utente digita un username inesistente. Il sistema verifica i dati e notifica l'errore.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 14: U.C.6.4 Username Errato

3.2.12 U.C.6.5 Caratteri non Validi

- **Attore:** Utente
- **Precondizioni:** L'utente inserisce dati che contengono caratteri non ammessi.
- **Postcondizioni:** L'accesso viene negato e viene mostrato un messaggio di errore specifico.
- **Scenario principale:** L'accesso viene negato e viene mostrato un messaggio di errore specifico.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -

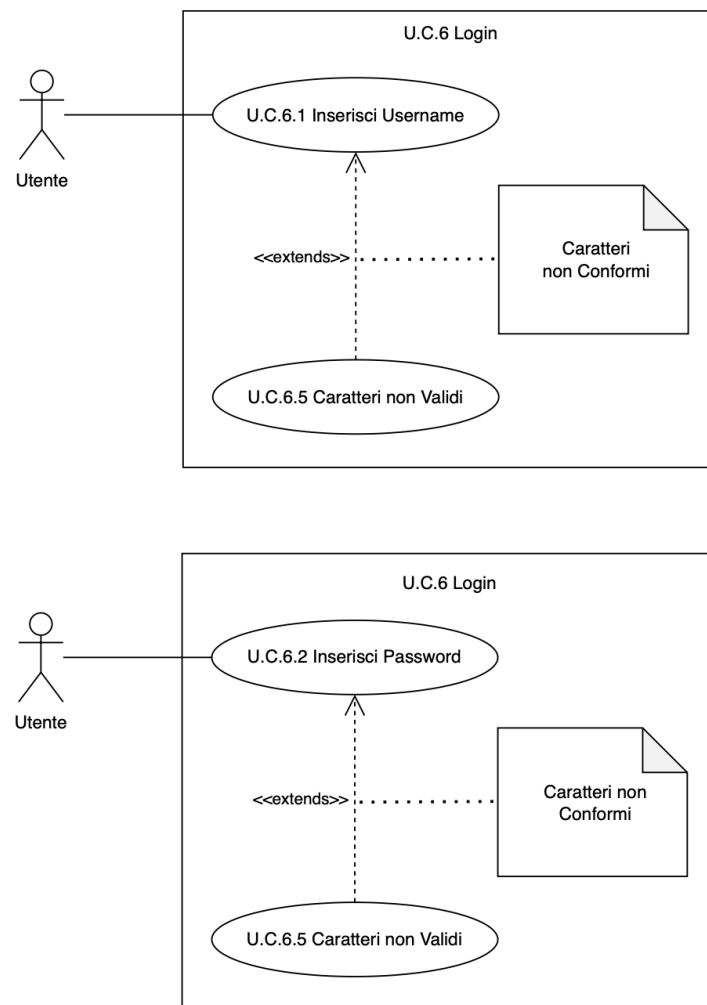


Figura 15: U.C.6.5 Caratteri non Validi

3.2.13 U.C.7 Sign Up

- **Attore:** Utente
- **Precondizioni:** L'utente non è registrato al sistema e desidera accedere ai servizi.
- **Postcondizioni:** L'utente viene registrato con successo e può accedere al sistema.
- **Scenario principale:** L'utente seleziona l'opzione di registrazione, compila i campi richiesti come username e password, e conferma l'operazione. Il sistema verifica i dati e completa la registrazione.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** U.C.7.1, U.C.7.2



Figura 16: U.C.7 Sign Up

3.2.14 U.C.7.1 Inserisci Username

- **Attore:** Utente
- **Precondizioni:** L'utente non è registrato e sta procedendo alla creazione di un account.
- **Postcondizioni:** L'username è stato inserito correttamente nel sistema.
- **Scenario principale:** L'utente compila il campo "Username" durante la registrazione e procede al passaggio successivo.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** U.C.7.3, U.C.7.4, U.C.7.6



Figura 17: U.C.7.1 Inserisci Username

3.2.15 U.C.7.2 Inserisci Password

- **Attore:** Utente
- **Precondizioni:** L'utente non è registrato e sta completando il processo di registrazione.
- **Postcondizioni:** La password viene salvata correttamente nel sistema.
- **Scenario principale:** L'utente inserisce una password nel campo corrispondente e procede con la registrazione.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** U.C.7.3, U.C.7.5



Figura 18: U.C.7.2 Inserisci Password

3.2.16 U.C.7.3 Caratteri non Validi

- **Attore:** Utente
- **Precondizioni:** Durante la registrazione, l'utente inserisce caratteri non consentiti nel campo username o password.
- **Postcondizioni:** La registrazione viene interrotta e l'utente riceve un messaggio di errore.
- **Scenario principale:** L'utente tenta di registrarsi ma utilizza caratteri non validi nei campi obbligatori. Il sistema rileva l'errore e avvisa l'utente.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 19: U.C.7.3 Caratteri non Validi

3.2.17 U.C.7.4 Username troppo Lungo

- **Attore:** Utente
- **Precondizioni:** L'utente inserisce un username che supera il limite massimo di caratteri consentiti.
- **Postcondizioni:** La registrazione non viene completata e l'utente riceve un messaggio di errore.
- **Scenario principale:** Durante la registrazione, il sistema rileva che l'username è troppo lungo e informa l'utente.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 20: U.C.7.4 Username troppo Lungo

3.2.18 U.C.7.5 Password troppo Lunga

- **Attore:** Utente
- **Precondizioni:** L'utente inserisce una password che supera il limite massimo consentito.
- **Postcondizioni:** La registrazione viene bloccata e l'utente viene informato dell'errore.
- **Scenario principale:** L'utente tenta di completare la registrazione ma il sistema respinge la password perché troppo lunga.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 21: U.C.7.5 Password troppo lunga

3.2.19 U.C.7.6 Username già presente

- **Attore:** Utente
- **Precondizioni:** L'utente inserisce un username già esistente nel sistema.
- **Postcondizioni:** La registrazione viene interrotta e il sistema suggerisce di scegliere un altro username.
- **Scenario principale:** L'utente tenta di registrarsi con uno username già utilizzato e il sistema blocca l'operazione con un messaggio informativo.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -

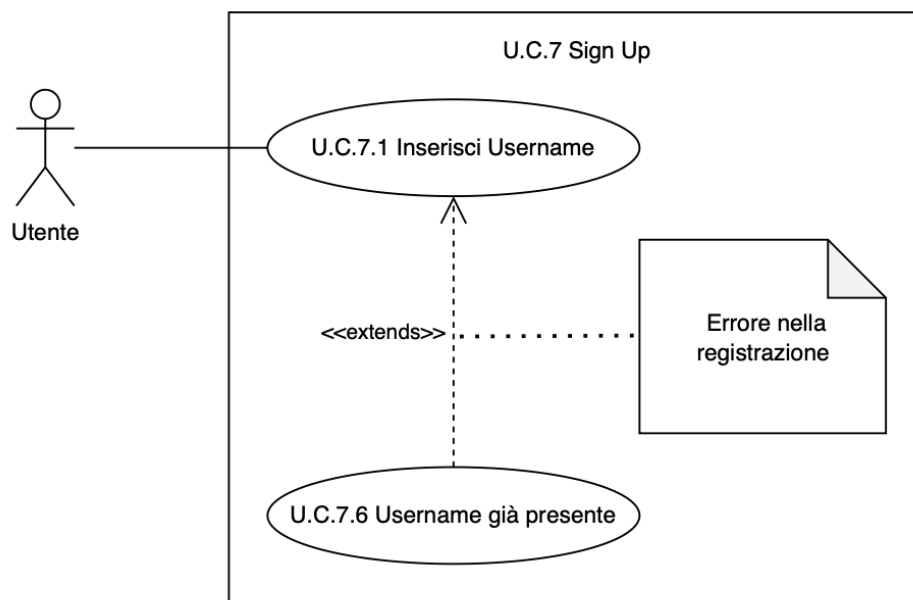


Figura 22: U.C.7.6 Username già presente

3.2.20 U.C.8 Salva Chat

- **Attore:** Utente
- **Precondizioni:** L'utente ha completato una conversazione con il bot e desidera conservarlo per consultazioni future.
- **Postcondizioni:** La conversazione viene salvata e aggiunta all'elenco delle chat salvate.
- **Scenario principale:** Dopo aver terminato la conversazione, l'utente seleziona l'opzione "Salva Chat" e il sistema archivia la conversazione.
- **Generalizzazioni:** -
- **Estensioni:** U.C.8.1
- **Inclusione:** -



Figura 23: U.C.8 Salva Chat

3.2.21 U.C.8.1 Memoria Piena

- **Attore:** Utente
- **Precondizioni:** L'utente ha superato il limite di conversazioni salvabili nel proprio account.
- **Postcondizioni:** La conversazione non viene salvata e l'utente riceve un avviso.
- **Scenario principale:** L'utente tenta di salvare una chat, ma il sistema rileva che lo spazio dedicato alle conversazioni è esaurito. Il sistema invita l'utente a eliminare alcune chat per liberare spazio.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 24: U.C.8.1 Memoria Piena

3.2.22 U.C.9 Feedback Chat

- **Attore:** Utente
- **Precondizioni:** L'utente ha completato una conversazione con il bot e vuole esprimere un giudizio sulla qualità delle risposte ricevute.
- **Postcondizioni:** Il feedback viene registrato nel sistema per analisi future.
- **Scenario principale:** Dopo la conversazione, l'utente valuta la conversazione corrente con il bot tramite due pulsanti di feedback (positivo o negativo). Il sistema registra il giudizio in formato booleano per migliorare le prestazioni future.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 25: U.C.9 Feedback Chat

3.2.23 U.C.10: Creazione di un nuovo template

- **Attore:** Admin
- **Precondizioni:** L'amministratore ha effettuato l'accesso al sistema di gestione e ha selezionato l'opzione per creare un nuovo template^G.
- **Postcondizioni:** Un nuovo template^G con una domanda predefinita e una risposta associata è stato salvato nel sistema.
- **Scenario principale:** L'amministratore accede alla funzione di creazione di un nuovo template^G. In questa sezione, inserisce una domanda predefinita e una risposta predefinita. Dopo aver verificato che i dati inseriti siano corretti, l'amministratore salva il nuovo template^G, che diventa immediatamente disponibile per essere utilizzato dagli utenti nella chat.
- **Generalizzazioni:** -
- **Estensioni:** U.C.13
- **Inclusione:** -



Figura 26: U.C.10: Creazione di un nuovo template

3.2.24 U.C.11: Modifica di un template esistente

- **Attore:** Admin
- **Precondizioni:** L'amministratore ha effettuato l'accesso e ha selezionato un template^G esistente dalla lista.
- **Postcondizioni:** Il template^G selezionato viene aggiornato con i nuovi dati forniti.
- **Scenario principale:** L'amministratore visualizza l'elenco dei template^G disponibili e seleziona quello che desidera modificare. Accede quindi ai dettagli del template^G, dove può modificare sia la domanda predefinita che la risposta associata. Dopo aver apportato le modifiche necessarie, l'amministratore salva i cambiamenti, aggiornando così il template^G nel sistema. Le modifiche apportate sono immediatamente visibili agli utenti quando utilizzano la funzione di selezione delle domande predefinite.
- **Generalizzazioni:** -
- **Estensioni:** U.C.13
- **Inclusione:** -



Figura 27: U.C.11: Modifica di un template esistente

3.2.25 U.C.12: Elimina un template esistente

- **Attore:** Admin
- **Precondizioni:** L'amministratore ha effettuato l'accesso e ha selezionato un template^G esistente dalla lista.
- **Postcondizioni:** Il template^G viene eliminato dal sistema e non è più disponibile per gli utenti.
- **Scenario principale:** L'amministratore, dalla lista dei template^G, individua quello che intende eliminare. Dopo aver selezionato il template^G, conferma l'operazione tramite un'apposita finestra di dialogo. Il sistema procede quindi a rimuovere il template^G, aggiornando l'elenco dei template^G disponibili. Da quel momento, la domanda predefinita associata non sarà più accessibile agli utenti.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 28: U.C.12: Elimina un template esistente

3.2.26 U.C.13 Controllo Validità Formato

- **Attore:** Admin
- **Precondizioni:** L'amministratore sta creando o modificando un template^G, ma inserisce un formato non valido (ad esempio, una domanda vuota o una risposta eccessivamente lunga ecc..)
- **Postcondizioni:** Il sistema non consente di salvare il template^G e informa l'amministratore dell'errore.
- **Scenario principale:** Durante la creazione o modifica di un template^G, l'amministratore inserisce dati non conformi, come una domanda lasciata vuota o una risposta con caratteri non consentiti. Il sistema esegue un controllo sui dati inseriti e rileva l'errore, bloccando il salvataggio del template^G. Viene visualizzato un messaggio di errore chiaro che spiega il problema e invita l'amministratore a correggere i dati prima di procedere con il salvataggio.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 29: U.C.13 Controllo Validità Formato

3.2.27 U.C.14: Visualizzazione delle metriche generali

- **Attore:** Admin
- **Precondizioni:** L'amministratore ha effettuato l'accesso alla dashboard di monitoraggio e ha selezionato i filtri desiderati per la visualizzazione delle metriche.
- **Postcondizioni:** L'amministratore visualizza le metriche principali delle run selezionate in base ai filtri applicati, come id, nome, input, tempo inizio, tempo fine, errore, output, tags, token totali usati, costo totale e feedback. Di default, vengono mostrate solo le prime 10 run.
- **Scenario principale:** L'amministratore accede alla dashboard di monitoraggio. Seleziona i criteri di filtro per limitare il numero di run visualizzate. Il sistema carica e visualizza le metriche delle run filtrate, evitando di caricare tutte le run contemporaneamente.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 30: U.C.14: Visualizzazione delle metriche generali

3.2.28 U.C.15: Visualizzazione Feedback Utenti

- **Attore:** Admin
- **Precondizioni:** L'amministratore ha effettuato l'accesso e ha selezionato la sezione relativa ai feedback degli utenti.
- **Postcondizioni:** I feedback degli utenti sulle risposte del chatbot, espressi in forma positiva o negativa tramite pulsante utente, sono stati visualizzati tramite un'icona e analizzati dall'amministratore.
- **Scenario principale:** L'amministratore accede alla sezione dei feedback, consulta i giudizi degli utenti e utilizza le informazioni raccolte per apportare miglioramenti al sistema.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 31: U.C.15: Visualizzazione Feedback Utenti

3.2.29 U.C.16: Importazione di Dati

- **Attore:** Admin
- **Attore Secondario:** OpenAi
- **Precondizioni:** L'amministratore ha selezionato un file di dati da importare.
- **Postcondizioni:** I dati vengono caricati per la validazione.
- **Scenario principale:** L'amministratore seleziona il file dal proprio dispositivo e avvia il processo di importazione. Il sistema prepara i dati per la validazione.
- **Generalizzazioni:** -
- **Estensioni:** U.C.16.1
- **Inclusione:** -



Figura 32: U.C.16: Importazione di Dati

3.2.30 U.C.16.1: Validazione Formato

- **Attore:** Admin
- **Precondizioni:** L'amministratore vuole caricare un file dati da caricare.
- **Postcondizioni:** I file vengono respinti se il formato dati è sbagliato, altrimenti vengono importati.
- **Scenario principale:** Il sistema analizza il file, controlla la coerenza e il formato dei dati, se incoerente respinge e segnala errore.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 33: U.C.16.1: Validazione Formato

3.2.31 U.C.17: Esportazione di Metriche nel Sistema

- **Attore:** Admin
- **Precondizioni:** L'amministratore ha selezionato i filtri desiderati per l'esportazione delle metriche.
- **Postcondizioni:** I dati vengono esportati in un file JSON contenente le metriche delle run filtrate, evitando di esportare tutte le run contemporaneamente. Di default, vengono esportate solo le prime 10 run.
- **Scenario principale:** L'amministratore accede alla dashboard di monitoraggio. Seleziona i criteri di filtro per limitare il numero di run da esportare. Il sistema genera un file JSON contenente le metriche delle run filtrate. L'amministratore scarica il file esportato.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 34: U.C.17: Esportazione di Dati nel Sistema

3.2.32 U.C.18 Elimina Chat

- **Attore:** Utente
- **Precondizioni:** L'utente ha effettuato l'accesso ed è presente almeno una conversazione salvata.
- **Postcondizioni:** La conversazione selezionata viene eliminata dal sistema.
- **Scenario principale:** L'utente accede all'elenco delle chat salvate, seleziona una conversazione specifica e conferma l'eliminazione.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -

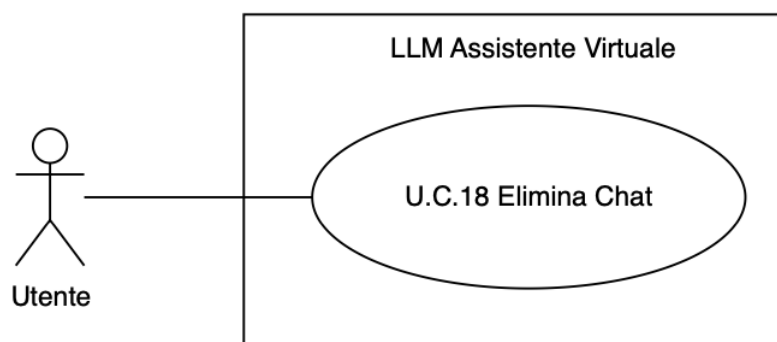


Figura 35: U.C.18 Elimina Chat

3.2.33 U.C.19: Riprendi Conversazione

- **Attore:** Utente
- **Precondizioni:** L'utente ha effettuato l'accesso al sistema ed effettuato una conversazione
- **Postcondizioni:** L'utente riprende la conversazione con l'assistente virtuale.
- **Scenario principale:** L'utente ha per qualche motivo dovuto interrompere la conversazione e la riprende successivamente selezionandola dalla lista conversazioni precedenti.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 36: U.C.19: Riprendi Conversazione

3.2.34 U.C.20: Risposta alla Richiesta dall'Amministratore

- **Attore:** Admin
- **Precondizioni:** L'amministratore dopo aver eseguito l'accesso al sistema procede con l'apertura della dashboard. Al suo interno gli può gestire le richieste di contatto da parte di utenti che necessitano assistenza.
- **Postcondizioni:** L'amministratore risponde alle richieste di assistenza tramite mail.
- **Scenario principale:** L'amministratore dopo avere eseguito l'accesso alla dashboard, seleziona una richiesta da gestire e visualizza che tipologia di assistenza deve fornire.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -

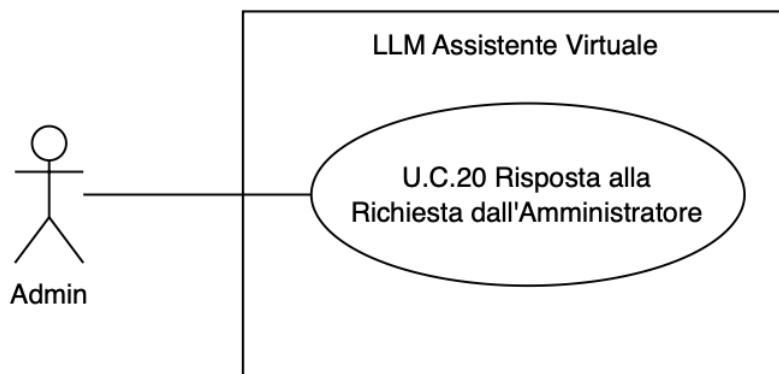


Figura 37: U.C.20: Risposta alla Richiesta dall'Amministratore

3.2.35 U.C.21: Cambio stato Richiesta

- **Attore:** Admin
- **Precondizioni:** L'amministratore ha effettuato l'accesso al sistema e in seguito alla dashboard per la gestione delle richieste di contatto. È presente almeno una richiesta inviata da un utente.
- **Postcondizioni:** La richiesta è stata aggiornata come "gestita" dallo stato di "attesa".
- **Scenario principale:** Dopo la risposta L'amministratore può decidere se lasciare la richiesta in stato di attesa o segnalarla come "gestita".
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 38: U.C.21: Cambio stato Richiesta

3.2.36 U.C.22 Invio Richiesta a un Operatore Umano

- **Attore:** Utente
- **Precondizioni:** L'utente ha ricevuto una risposta non soddisfacente dal sistema basato su *LLM*^G.
- **Postcondizioni:** La richiesta dell'utente è stata inviata agli amministratori ed è visibile nella dashboard.
- **Scenario principale:** L'utente seleziona l'opzione per richiedere assistenza a un operatore umano, compila un modulo opzionale con eventuali dettagli e invia la richiesta. Il sistema registra la richiesta e la rende disponibile nella dashboard degli amministratori.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -



Figura 39: U.C.22 Invio Richiesta a un Operatore Umano

3.2.37 U.C.23: Visualizzazione Template di Domanda e Risposta

- **Attore:** Amministratore
- **Precondizioni:** L'amministratore ha effettuato l'accesso al sistema e ha selezionato la sezione di gestione dei template di domanda e risposta.
- **Postcondizioni:** L'amministratore visualizza la lista dei template di domande e risposte disponibili nel sistema.
- **Scenario principale:** L'amministratore accede alla sezione dei template di domanda e risposta. Il sistema mostra la dashboard con l'elenco dei template registrati. L'amministratore consulta i template disponibili per verificare o modificare la configurazione delle risposte predefinite.
- **Generalizzazioni:** -
- **Estensioni:** -
- **Inclusione:** -

4 Requisiti

In questa sezione vengono presentati i requisiti emersi durante l'attività di analisi, condotta a partire dai casi d'uso, dall'esame del capitolato d'appalto e dagli incontri, sia interni che con il proponente.

4.1 Classificazione dei requisiti

I requisiti sono classificati in tre categorie principali:

- **Funzionali:** riguardano l'usabilità del prodotto finale;
- **Di qualità:** includono gli strumenti e la documentazione da fornire;
- **Di vincolo:** fanno riferimento alle tecnologie da utilizzare.

Ciascun requisito è indicato da:

- **Codice Identificativo:** codice univoco che identifica il requisito;
- **Descrizione:** breve spiegazione del requisito;
- **Fonte:** origine del requisito (es. capitolato, interno, ecc.);
- **Priorità:** importanza del requisito rispetto agli altri;

4.2 Fonti dei requisiti

I requisiti sono stati identificati a partire dalle seguenti fonti:

- **Capitolato:** requisiti individuati tramite analisi del capitolato;
- **Interno:** requisiti individuati durante riunioni interne al gruppo di lavoro;
- **Esterno:** requisiti individuati durante incontri con il proponente;
- **Piano di Qualifica^G:** requisiti necessari per rispettare standard di qualità definiti nel documento Piano di Qualifica^G;
- **Norme di Progetto^G:** requisiti necessari per rispettare le norme di progetto definite nel documento Norme di Progetto^G norme di progetto^G definite nel documento Norme di Progetto;

4.3 Codifica dei requisiti

I requisiti sono codificati come segue: **R[Tipo][Importanza][Numero]**

Dove **Tipo** può essere:

- **F** (funzionale)
- **Q** (di qualità)
- **V** (di vincolo)

Importanza può essere:

- **O** (obbligatorio)
- **D** (desiderabile)
- **F** (facoltativo)

Numero è un numero identificativo univoco del requisito.

Esempio:

- **RFO1**: requisito funzionale obbligatorio numero 1
- **RQD2**: requisito di qualità desiderabile numero 2
- **RVF3**: requisito di vincolo facoltativo numero 3

4.4 Requisiti funzionali

| Codice | Fonte | Descrizione |
|--------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RFO1 | Capitolato, Interno | L'amministratore inserisce dalla pagina di gestione i dati semantici aziendali da cui apprendere la conoscenza da file in formato .pdf. |
| RFO2 | Capitolato, Interno | L'amministratore inserisce dalla pagina di gestione i dati semantici aziendali da cui apprendere la conoscenza da file in formato .txt. |
| RFO3 | Capitolato | I testi recuperati dai documenti verranno suddivisi in blocchi, ovvero pezzi più piccoli di dati che rappresentano una piccola porzione del contesto. |
| RFO4 | Capitolato | I vettori generati verranno memorizzati all'interno di un database vettoriale e opportunamente indicizzati. |
| RFO5 | Capitolato, Esterno | Da un'interfaccia utente della web app, viene catturata una domanda da parte dell'utente. |
| RFO6 | Capitolato | La domanda viene inoltrata al sistema attraverso delle API REST risiedenti in un Web Server. |
| RFO7 | Capitolato | La rappresentazione vettoriale viene utilizzata per effettuare una ricerca all'interno del database vettoriale da dove vengono reperiti i vettori più simili. |
| RFO8 | Capitolato | La domanda viene inviata al sistema LLM tramite API. |
| RFO9 | Capitolato | Viene attesa la risposta dall'LLM tramite API. |
| RFO10 | Capitolato | Attraverso API REST, il sistema inoltra la risposta all'account dell'utente. |
| RFO11 | Interno | L'utente deve essere in grado di ottenere informazioni riguardo un prodotto attraverso la conversazione con il bot. |
| RFO12 | Interno | L'utente deve essere in grado di ottenere informazioni riguardo una serie di prodotti attraverso la conversazione con il bot. |
| RFO13 | Interno | La conversazione tra utente e bot deve essere salvata. |
| RFO14 | Interno | L'utente deve essere in grado di visualizzare una delle conversazioni precedentemente salvate. |
| RFO15 | Interno | L'utente deve essere in grado di riprendere una delle conversazioni precedentemente salvata. |
| RFO16 | Interno | L'utente o l'amministratore devono poter accedere al sistema inserendo Username e Password. |
| RFO17 | Interno | L'utente si registra inserendo Username e Password. |
| RFO18 | Interno | Gli input del form di registrazione devono essere sanificati per prevenire attacchi SQL Injection. |
| RFO19 | Interno | Gli input del form di accesso devono essere sanificati per prevenire attacchi SQL Injection. |
| RFO20 | Interno, Esterno | L'utente deve essere in grado di dare un feedback (thumbsup/thumbsdown) sulla qualità della conversazione dopo averla provata. |
| RFO21 | Esterno | L'accesso alla dashboard dei "template di domanda e risposta" è consentito solo agli utenti con ruolo di amministratore. |

| Codice | Fonte | Descrizione |
|--------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RFO22 | Esterno | Dopo l'accesso da parte dell'amministratore, la pagina di gestione mostra la dashboard dei "template di domanda e risposta". |
| RFO23 | Esterno | Un "template di domanda e risposta" è formato da una domanda (possibilmente una domanda posta frequentemente che l'amministratore decide di inserire per risparmiare una chiamata al modello) associata ad una corrispondente risposta. |
| RFO24 | Esterno | L'amministratore deve essere in grado di creare un template, che è formato da una domanda associata ad una corrispondente risposta. |
| RFO25 | Esterno | L'amministratore deve essere in grado di modificare uno dei template esistenti. |
| RFO26 | Interno | L'amministratore deve essere in grado di eliminare un template esistente. |
| RFO27 | Interno, Esterno | Il sistema deve poter fermare la creazione di un template invalido, ovvero quando il template non rispetta il formato Json. |
| RFO28 | Interno | L'amministratore deve poter accedere alla dashboard di monitoraggio delle metriche. |
| RFO29 | Interno | L'accesso alla dashboard delle metriche delle run è consentito solo agli utenti con ruolo di amministratore. |
| RFO30 | Interno | Dopo l'accesso da parte dell'amministratore, la pagina di gestione mostra la dashboard delle metriche delle run. |
| RFD31 | Interno | L'amministratore deve poter selezionare criteri di filtro per visualizzare solo le run di interesse. |
| RFD32 | Interno | Il sistema deve permettere la selezione di filtri come ID, nome, input, data di inizio e fine, errore, output, tag, numero di token, costo. |
| RFF33 | Interno | Una volta selezionati i filtri, il sistema deve aggiornare la visualizzazione senza ricaricare l'intera pagina. |
| RFO34 | Interno | Se nessun filtro è selezionato, il sistema mostra le prime dieci run per impostazione predefinita. |
| RFD35 | Interno | Dopo aver applicato i filtri, l'amministratore deve poter visualizzare le metriche principali delle run selezionate. |
| RFD36 | Interno | Il sistema deve mostrare le metriche principali delle run filtrate (ID, nome, input, data di inizio e fine, errore, output, tag, token totali, costo totale). |
| RFF37 | Interno | La visualizzazione deve essere chiara e strutturata, con possibilità di ordinare le colonne. |
| RFO38 | Interno | L'amministratore deve poter visualizzare i feedback dati dagli utenti. |
| RFO39 | Esterno | Il sistema deve poter rifiutare l'importazione dati di file non compatibili, ovvero file non nel formato pdf o txt. |
| RFO40 | Interno | L'utente deve poter eliminare una conversazione precedentemente effettuata. |
| RFO41 | Esterno | L'utente deve poter mandare richieste di assistenza per poter parlare con un operatore umano. |

| Codice | Fonte | Descrizione |
|--------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RFO42 | Interno | L'accesso alla dashboard delle richieste di assistenza è consentito solo agli utenti con ruolo di amministratore. |
| RFO43 | Interno | Dopo l'accesso da parte dell'amministratore, la pagina di gestione mostra la dashboard delle richieste di assistenza. |
| RFO44 | Esterno | L'amministratore deve poter visualizzare le richieste di assistenza ricevute da parte dell'utente. |
| RFO45 | Interno | L'amministratore deve poter segnalare ad altri amministratori che una richiesta è stata presa in carico. |
| RFD46 | Esterno, Interno | L'amministratore deve essere in grado di poter rispondere all'utente tramite contatto via e-mail. |
| RFD47 | Interno | Le metriche delle run del chatbot devono essere esportabili in JSON. |
| RFD48 | Interno | Le metriche della run devono includere ID univoco della run, nome assegnato alla sessione, dati di input elaborati dal modello, timestamp di avvio e completamento dell'esecuzione, eventuali errori incontrati, risultato generato dal modello, numero totale di token utilizzati e stima dei costi basata sul consumo di token. |
| RFO49 | Interno | Il bot per rispondere a una domanda deve ricordarsi i messaggi precedenti nella singola conversazione. |
| RFD50 | Interno | Il sistema deve notificare l'utente quando la memoria per le chat salvate è piena e non è possibile salvare ulteriori conversazioni. |
| RFO51 | Interno | L'utente seleziona una delle domande tra quelle predefinite. |
| RFO52 | Interno | L'utente deve essere in grado di visualizzare una lista delle conversazioni precedentemente salvate. |
| RFO53 | Interno | La lunghezza massima dell'username è di 256 caratteri. |
| RFO54 | Interno | La lunghezza massima della password è di 256 caratteri. |
| RFO55 | Interno | Il Sistema rifiuta la registrazione di un nuovo account con username già presente. |

Tabella 1: Requisiti funzionali

4.5 Requisiti qualitativi

| Codice | Fonte | Descrizione |
|-------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| RQO1 | Capitolato, Piano di Qualifica ^G | Schema di progettazione della base di dati |
| RQO2 | Capitolato, Piano di Qualifica ^G | Codice prodotto in formato sorgente reso disponibile tramite repository pubblici |
| RQO3 | Piano di Qualifica ^G | Documentazione riassuntiva delle metriche e dei risultati |
| RQO4 | Piano di Qualifica ^G | Il software deve essere testato con una copertura di codice minima dell'80% e una copertura dei rami dell'80%, con un obiettivo ottimale del 100% |
| RQO5 | Piano di Qualifica ^G | Il 90% dei test deve essere superato come requisito minimo, mentre l'obiettivo ottimale è il 100% |
| RQO6 | Piano di Qualifica ^G | La metodologia di sviluppo deve seguire il paradigma del Test Driven Development (TDD), garantendo che il codice venga scritto partendo dai test |

Tabella 2: Requisiti qualitativi

4.6 Requisiti di vincolo

| Codice | Fonte | Descrizione |
|--------------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| RVO1 | Capitolato | Il chatbot deve rispondere con il contesto dato dai file di allenamento (pdf o file di testo inseriti) |
| RVO2 | Capitolato | LLM ^G deve essere integrato tramite API |
| RVO3 | Interno (Analisi dei modelli) | LLM utilizzato deve essere quello di OpenAI |
| RVO4 | Capitolato | Deve essere usato un database relazionale |
| RVO5 | Interno (Analisi del Database) | Deve essere gestito il salvataggio delle chat precedenti con tutti i messaggi in esse tramite un database relazionale con PostgreSQL |
| RVO6 | Capitolato | Deve essere implementato un database vettoriale |
| RVO7 | Interno (Analisi dei modelli) | Deve essere implementato un database vettoriale FAISS per poter rendere possibile la ricerca con contesto dall'LLM |
| RVO8 | Capitolato | Deve essere implementato un embedding ^G model |
| RVO9 | Interno (Analisi dei modelli) | L'embedding model deve essere quello di OpenAI |
| RVO10 | Capitolato | Deve essere implementata una WebApp che permetta di comunicare con il chatbot |
| RVO11 | Interno (Analisi Frontend ^G) | L'interfaccia deve essere costruita utilizzando componenti funzionali React. |
| RVO12 | Interno | Si deve creare un backend che gestisca le chiamate HTTP, il database vettoriale e il database relazionale con Flask. |
| RVO13 | Interno | La gestione dello stato locale deve essere implementata tramite useState. |
| RVO14 | Interno | La WebApp deve utilizzare React Router per gestire la navigazione tra le pagine. |
| RVO15 | Interno | Gli stili devono essere gestiti tramite CSS inline o con className per garantire modularità. |
| RVO16 | Interno | La comunicazione tra componenti deve essere gestita inviando funzioni come props ^G . |
| RVO17 | Interno | La WebApp deve essere responsiva e adattarsi dinamicamente alle dimensioni della finestra. |
| RVO18 | Interno (Analisi Vettoriale) | La gestione dei blocchi di testo vettorializzati deve essere gestita tramite Faiss |
| RVD19 | Interno (Analisi Backend) | Le metriche delle run del chatbot devono essere recuperate tramite Langsmith |
| RVO20 | Interno (Analisi Backend) | Bisogna usare la libreria LangChain per la interazione con i modelli LLM e Embedding |

Tabella 3: Requisiti di vincolo

4.7 Tracciamento

4.7.1 Requisito - Fonte

| Requisito | Fonte |
|-----------|---------------------|
| RFO1 | Capitolato, Interno |
| RFO2 | Capitolato, Interno |
| RFO3 | Capitolato |
| RFO4 | Capitolato |
| RFO5 | Capitolato, Esterno |
| RFO6 | Capitolato |
| RFO7 | Capitolato |
| RFO8 | Capitolato |
| RFO9 | Capitolato |
| RFO10 | Capitolato |
| RFO11 | Interno |
| RFO12 | Interno |
| RFO13 | Interno |
| RFO14 | Interno |
| RFO15 | Interno |
| RFO16 | Interno |
| RFO17 | Interno |
| RFO18 | Interno |
| RFO19 | Interno |
| RFO20 | Interno, Esterno |
| RFO21 | Esterno |
| RFO22 | Esterno |
| RFO23 | Esterno |
| RFO24 | Esterno |
| RFO25 | Esterno |
| RFO26 | Interno |
| RFO27 | Interno, Esterno |
| RFO28 | Interno |
| RFO29 | Interno |
| RFO30 | Interno |
| RFD31 | Interno |
| RFD32 | Interno |
| RFF33 | Interno |
| RFO34 | Interno |
| RFD35 | Interno |
| RFD36 | Interno |
| RFF37 | Interno |
| RFO38 | Interno |
| RFO39 | Esterno |
| RFO40 | Interno |
| RFO41 | Esterno |
| RFO42 | Interno |

| Requisito | Fonte |
|-----------|---------------------------------------------|
| RFO43 | Interno |
| RFO44 | Esterno |
| RFO45 | Interno |
| RFD46 | Esterno, Interno |
| RFD47 | Interno |
| RFD48 | Interno |
| RFO49 | Interno |
| RFD50 | Interno |
| RFO51 | Interno |
| RFO52 | Interno |
| RFO53 | Interno |
| RFO54 | Interno |
| RFO55 | Interno |
| RQO1 | Capitolato, Piano di Qualifica ^G |
| RQO2 | Capitolato, Piano di Qualifica ^G |
| RQO3 | Piano di Qualifica ^G |
| RQO4 | Piano di Qualifica ^G |
| RQO5 | Piano di Qualifica ^G |
| RQO6 | Piano di Qualifica ^G |
| RVO1 | Capitolato |
| RVO2 | Capitolato |
| RVO3 | Interno (Analisi dei modelli) |
| RVO4 | Capitolato |
| RVO5 | Interno (Analisi Database) |
| RVO6 | Capitolato |
| RVO7 | Interno (Analisi dei modelli) |
| RVO8 | Capitolato |
| RVF9 | Interno (Analisi dei modelli) |
| RVF10 | Capitolato |
| RVD11 | Interno (Analisi Frontend ^G) |
| RVO12 | Interno |
| RVO13 | Interno |
| RVO14 | Interno |
| RVO16 | Interno |
| RVO15 | Interno |
| RVO17 | Interno |
| RVO18 | Interno (Analisi Vettoriale) |
| RVD19 | Interno (Analisi Backend) |
| RVO20 | Interno (Analisi Backend) |

Tabella 4: Requisito - Fonte

4.8 Caso d'uso - Requisito

| Caso d'uso | Requisito |
|------------|-----------------------------------------------------------------------------|
| U.C.1 | RFO5 |
| U.C.2 | RFO3, RFO6, RFO7, RFO8, RFO9, RFO10, RFO11, RFO12, RFO49 |
| U.C.2.1 | RFO7, RFO8, RFO9 |
| U.C.3 | RFO5, RFO51 |
| U.C.4 | RFO52 |
| U.C.5 | RFO14 |
| U.C.6 | RFO16 |
| U.C.6.1 | RFO16 |
| U.C.6.2 | RFO16 |
| U.C.6.3 | RFO16 |
| U.C.6.4 | RFO16 |
| U.C.6.5 | RFO16, RFO19 |
| U.C.7 | RFO17 |
| U.C.7.1 | RFO17 |
| U.C.7.2 | RFO17 |
| U.C.7.3 | RFO17, RFO18 |
| U.C.7.4 | RFO17, RFO53 |
| U.C.7.5 | RFO17, RFO54 |
| U.C.7.6 | RFO17, RFO55 |
| U.C.8 | RFO13 |
| U.C.8.1 | RFD50 |
| U.C.9 | RFO20 |
| U.C.10 | RFO24 |
| U.C.11 | RFO25 |
| U.C.12 | RFO26 |
| U.C.13 | RFF27 |
| U.C.14 | RFO28, RFO29, RFO30, RFD31, RFD32, RFF33, RFO34, RFD35, RFD36, RFF37, RFD48 |
| U.C.15 | RFO38 |
| U.C.16 | RFO1, RFO2, RFO4 |
| U.C.16.1 | RFO1, RFO2, RFO39 |
| U.C.17 | RFD47, RFD48 |
| U.C.18 | RFO40 |
| U.C.19 | RFO15 |
| U.C.20 | RFO42, RFO43, RFO44, RFD46 |
| U.C.21 | RFO45 |
| U.C.22 | RFO41 |
| U.C.23 | RFO21, RFO22, RFO23 |

Tabella 5: Caso d'uso - Requisito