



Analisi dei Requisiti

2025-04-11

V1.1.2

sweetenteam@gmail.com

<https://sweetenteam.github.io>



Destinatari	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	AzzurroDigitale
Redattori	Valeri Mihail Belenkov
	Davide Benedetti
	Orlando Ferazzani
	Nicolas Fracaro
	Mouad Mahdi
	Andrea Santi
Verificatori	Valeri Mihail Belenkov
	Matteo Campagnaro
	Orlando Ferazzani
	Nicolas Fracaro
	Mouad Mahdi
	Andrea Santi

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
1.1.2	2025-04-11	Nicolas Fracaro	Andrea Santi	Aggiornamento documento per la fase PB, UC1.3, UC4.
1.1.1	2025-04-10	Andrea Santi	Nicolas Fracaro	Aggiornamento documento per la fase PB
1.1.0	2025-02-20	Valeri Mihail Belenkov, Orlando Ferazzani, Nicolas Fracaro, Andrea Santi	Mouad Mahdi	Apportate modifiche richieste in RTB
1.0.0	2025-02-10	Nicolas Fracaro	Andrea Santi	Approvazione per RTB con formattazione pagine
0.2.0	2025-02-07	Valeri Mihail Belenkov, Davide Benedetti, Orlando Ferazzani, Nicolas Fracaro, Andrea Santi	Matteo Campagnaro	Inseriti requisiti funzionali, di qualità, di vincolo e completato sezioni tracciamento e riepilogo
0.1.0	2025-02-05	Orlando Ferazzani, Nicolas Fracaro, Mouad Mahdi, Andrea Santi	Valeri Mihail Belenkov	Rimozione UC Github, UC Jira, UC Confluence. Aggiunta precondizioni UC4, Aggiunta errori specifici a UC1, UC3, UC4. Numerazione casi d'uso corretta. Aggiunta spiegazione attori secondari
0.0.9	2025-01-07	Nicolas Fracaro	Andrea Santi	Inseriti UC1, UC2, UC3
0.0.8	2025-01-04	Mouad Mahdi	Orlando Ferazzani	Inserito UC GitHub
0.0.7	2024-12-02	Andrea Santi	Nicolas Fracaro	Sistematì UC Jira, aggiunto diagramma
0.0.6	2024-12-10	Orlando Ferazzani	Nicolas Fracaro	Inserito UC Confluence
0.0.5	2024-12-02	Andrea Santi	Mouad Mahdi	Inseriti primi UC, sistemate immagini e tabelle
0.0.4	2024-11-25	Andrea Santi	Mouad Mahdi	Introdotte sezioni Casi d'Uso (3) + Requisiti (4), impaginazione tabelle
0.0.3	2024-11-22	Andrea Santi	Mouad Mahdi	Terminata la descrizione (punto 2).
0.0.2	2024-11-21	Mouad Mahdi	Andrea Santi	Fine punto 1 e inizio punto 2.
0.0.1	2024-11-19	Valeri Mihail Belenkov	Andrea Santi	Stesura iniziale del documento e breve introduzione.

Indice

1) Introduzione	6
1.1) Scopo del documento	6
1.2) Scopo del progetto	6
1.3) Glossario	7
1.4) Sviluppo e miglioramento	7
1.5) Tecnologie utilizzate	7
1.6) Riferimenti	9
1.6.1) Normativi	9
1.6.2) Informativi	9
1.6.3) Tecnologici:	9
2) Descrizione	10
2.1) Obiettivo del prodotto	10
2.2) Funzionalità del prodotto	10
2.3) Utenti e caratteristiche	10
3) Casi d'uso	11
3.1) Introduzione ai casi d'uso e obbiettivi	11
3.2) Attori	11
3.3) Definizione casi d'uso	12
3.3.1) UC1, Visualizzazione storico chat	12
3.3.1.1) UC1.1, Visualizzazione errore nessun messaggio nello storico della chat	13
3.3.1.2) UC1.2, Errore durante il recupero dello storico della chat	13
3.3.1.3) UC1.3, Visualizzazione errore di timeout	14
3.3.1.4) UC1.5, Visualizzazione errore backend non disponibile	14
3.3.1.5) UC1.6, Visualizzazione caricamento dello Storico della chat	15
3.3.1.6) UC1.4, Visualizzazione singolo messaggio	15
3.3.1.6.1) UC1.4.1, Visualizzazione messaggio da chatbot	17
3.3.1.6.2) UC1.4.5, Visualizzazione contenuto del messaggio	18
3.3.1.6.3) UC1.4.4, Visualizzazione data e ora del messaggio;	18
3.3.1.6.4) UC1.4.3, Visualizzazione data e ora ultimo aggiornamento dati risposta	18
3.3.1.6.5) UC1.4.6, Visualizzazione messaggio con snippet code	19
3.3.1.6.6) UC1.4.7, Visualizzazione snippet di codice	19
3.3.1.6.7) UC1.4.2, Visualizzazione messaggio da utente	20
3.3.2) UC2, Scrittura domanda in linguaggio naturale	21
3.3.3) UC3, Visualizzazione risposta generata	22
3.3.3.1) UC3.1, Visualizzazione errore nella generazione della risposta	23
3.3.3.2) UC3.2, Visualizzazione errore risposta troppo lunga	23
3.3.3.3) UC3.3, Visualizzazione errore domanda troppo lunga	24
3.3.3.4) UC3.4, Visualizzazione risposta a domanda fuori contesto	24
3.3.3.5) UC3.5, Visualizzazione risposta a domanda richiedente documenti non disponibili .	25
3.3.3.6) UC3.6, Visualizzazione caricamento della risposta	26
3.3.4) UC4, Invio richiesta con domanda dell'utente	26
3.3.4.1) UC4.1, Errore durante la generazione della risposta	29
3.3.4.2) UC4.2, Errore risposta troppo lunga	31
3.3.5) UC5, Modifica domanda	32
3.3.6) UC6, Visualizzazione UI	33
3.3.6.0.1) UC6.1, Visualizzazione UI in modalità chiara	33
3.3.6.0.2) UC6.2, Visualizzazione UI in modalità scura	34

3.3.7) UC7, Visualizzazione link - icona	34
3.3.7.1) UC7.1, Visualizzazione link - icona Jira	35
3.3.7.2) UC7.2, Visualizzazione link - icona GitHub	35
3.3.7.3) UC7.3, Visualizzazione link - icona Confluence	36
3.3.8) UC8, Visualizzazione bottone «Load More»	36
4) Requisiti	38
4.1) Requisiti Funzionali	38
4.2) Requisiti di Qualità	42
4.3) Requisiti di Vincolo	43
4.4) Tracciamento	43
4.5) Riepilogo	45

Lista della immagini

Figura 1	Logo di Typescript	7
Figura 2	Logo di Langchain	7
Figura 3	Logo di Node.js	7
Figura 4	Logo di Nest.js	8
Figura 5	Logo di GroqCloud	8
Figura 6	Logo di Nomic	8
Figura 7	Logo di Qdrant	8
Figura 8	Logo di PostgreSQL	8
Figura 9	Logo di Octokit	8
Figura 10	Logo di JiraJs	8
Figura 11	Logo di ConfluenceJs	8
Figura 12	Logo di Docker	9
Figura 13	Logo di React	9
Figura 14	Logo di Tailwind CSS	9
Figura 15	Logo di ReactQuery	9
Figura 16	Logo di Next.js	9
Figura 17	Diagramma UC1, visualizzazione storico chat	12
Figura 18	Diagramma UC1.4, Visualizzazione messaggio singolo	15
Figura 19	Diagramma UC1.4.1, Visualizzazione messaggio chatbot	17
Figura 20	Diagramma UC1.4.2, Visualizzazione messaggio user	20
Figura 21	Diagramma UC2, Scrittura domanda in linguaggio naturale	21
Figura 22	Diagramma UC3, Invio domanda dell'utente	22
Figura 23	Diagramma UC4, Invio richiesta con domanda dell'utente	26
Figura 24	Diagramma UC5, Modifica domanda	32
Figura 25	Diagramma UC6, visualizzazione UI	33
Figura 26	Diagramma UC7, Visualizzazione link - icone	34
Figura 27	Diagramma UC8, Visualizzazione bottone Load More	36

Lista delle tabelle

Tabella 1	Requisiti Funzionali	38
Tabella 2	Requisiti di Qualità	42
Tabella 3	Requisiti di Vincolo	43
Tabella 4	Tracciamento	43
Tabella 5	Riepilogo	45

1) Introduzione

1.1) Scopo del documento

L' *Analisi dei requisiti*_G è un documento fondamentale per tutti i progetti di sviluppo software che vogliono creare un prodotto a regola d'arte.

Lo scopo di questo documento è di definire le funzionalità che il sistema sarà in grado di offrire, ossia i requisiti obbligatori e opzionali che dovranno essere soddisfatti al fine di realizzare le richieste fatte dal *proponente*_G.

Il documento non si pone come una soluzione tecnica al problema, quanto più una definizione chiara e concisa di esso e di come possa essere risolto.

In particolare, le finalità di questo documento possono essere descritte nei seguenti punti:

- **Definire le esigenze del proponente:**

Questo documento si basa principalmente sulle richieste del proponente, ossia le idee che quest'ultimo ha riguardo a come dovrebbe essere il software che verrà sviluppato dal nostro team. Tali idee verranno raccolte tramite i vari documenti e incontri con *azzurro digitale*: che avverranno lungo il percorso dello svolgimento del progetto.

- **Fornire una base per la progettazione del sistema:**

L' *Analisi dei requisiti*_G fornisce una base per la progettazione del sistema, in quanto definisce le funzionalità che il sistema dovrà offrire, permettendo così ai *programmatori*_G di comprendere le esigenze dei proponenti identificando le soluzioni che più si adeguano a tali esigenze.

- **Tracciare i requisiti del sistema:**

Una volta raccolte le richieste del proponente, questo documento si impone di identificare i requisiti e suddividerli in requisiti funzionali e non funzionali.

- **Verificare e validare i requisiti:**

Questo processo garantisce che le attività siano svolte seguendo il *Way of Working*_G del gruppo, controllando la presenza di errori e correggendoli una volta identificati. Ciò permette di accertare che il prodotto finale corrisponda alle aspettative del proponente.

Una volta che i requisiti del sistema saranno stati definiti in maniera chiara in modo tale da permettere al lettore di comprenderli pienamente, allora verrà data una rappresentazione formale grafica del software attraverso l'utilizzo di *diagrammi dei casi d'uso*_G.

1.2) Scopo del progetto

Lo scopo del progetto è la realizzazione di un assistente virtuale sotto forma di *chatbot*_G in grado di assistere gli utenti, rispondendo alle loro domande in linguaggio naturale. Il *chatbot*_G garantirà un accesso rapido alle informazioni interne dell'azienda, aggregando i dati provenienti dalle piattaforme *Jira*_G, *Github*_G e *Confluence*_G ed elaborando questi ultimi con l'uso dell'intelligenza artificiale per dare risposte chiare. Questo strumento sarà utile non solo ai membri già attivi dell'azienda per tagliare i tempi lunghi di ricerca manuale di una determinata informazione, ma anche per il processo di onboarding dei nuovi arrivati rispondendo alle domande più frequenti e guidandoli nel processo di apprendimento delle risorse aziendali senza il bisogno di una figura di supporto.

1.3) Glossario

Al fine di evitare eventuali equivoci o incomprensioni, si è deciso di adottare un Glossario presente come file e nella pagina web (nella sua versione attuale v1.0), in cui vengono riportate tutte le definizioni delle parole ambigue utilizzate nei documenti di questo progetto. Nel documento verranno riportati tutti i termini definiti nel loro ambiente di utilizzo con la descrizione del loro significato. I termini presenti nel glossario sono evidenziati e hanno una piccola “G” alla fine.

1.4) Sviluppo e miglioramento

Questo documento è stato sviluppato in modo graduale e progressivo, con l’obiettivo di facilitare eventuali modifiche future in base alle necessità concordate tra il gruppo di progetto e l’azienda committente. Pertanto è soggetto a un continuo miglioramento.

1.5) Tecnologie utilizzate

Il progetto richiede l’utilizzo di svariate e diverse tecnologie, sia per la parte back-end che per la parte front-end. Queste devono essere scelte in base alle esigenze del progetto e alle richieste del proponente oltre che tenendo conto del grado di qualità che il progetto, e di conseguenza poi il prodotto, dovrà avere.

Come linguaggio di programmazione il team si avvale di Typescript, basato su Javascript ma con sintassi più chiara e tipizzazione statica. Questo linguaggio verrà utilizzato sia nella parte di back-end che di front-end. Questa scelta è stata ponderata in quanto ci permette di avere meno differenze possibili tra l’implementazione delle due parti del nostro sistema.



Figura 1: Logo di Typescript

Per ciò che concerne il back-end sono state utilizzate le seguenti tecnologie:

- Langchain: framework per lo sviluppo di applicazioni supportate da [LLM](#)



Figura 2: Logo di Langchain

- Node.js: ambiente di runtime per eseguire codice Javascript;



Figura 3: Logo di Node.js

- Nest.js: framework per la creazione di applicazioni server-side efficienti e scalabili. L'utilizzo di NestJs permette al team di implementare vari design pattern utili per lo sviluppo;



Figura 4: Logo di Nest.js

- GroqCloud: Piattaforma AI basata su hardware specializzato (LPU) per inferenza ad alte prestazioni, supporta modelli LLM e integrazione con strumenti AI per elaborazione in tempo reale.



Figura 5: Logo di GroqCloud

- Nomic: Servizio che mette a disposizione un modello di AI che permette l'embedding del testo;



Figura 6: Logo di Nomic

- Qdrant: motore di ricerca e analisi di dati non strutturati, supporta l'indicizzazione e la ricerca di dati in tempo reale;



Figura 7: Logo di Qdrant

- PostgreSQL: sistema di gestione di database relazionali;

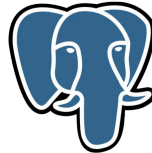


Figura 8: Logo di PostgreSQL

- Octokit: toolkit per l'interazione con l'API di GitHub;



Figura 9: Logo di Octokit

- JiraJs: toolkit per l'interazione con l'API di Jira;



Figura 10: Logo di JiraJs

- ConfluenceJs: toolkit per l'interazione con l'API di Confluence;



Figura 11: Logo di ConfluenceJs

- Docker: piattaforma per lo sviluppo, il deploy e l'esecuzione di applicazioni in container;



Figura 12: Logo di Docker

Successivamente, per la parte front-end sono state utilizzati i seguenti linguaggi e framework:

- React: libreria Javascript (o Typescript) per la creazione di interfacce utente;



Figura 13: Logo di React

- Tailwind CSS: framework CSS per la creazione di interfacce utente;



Figura 14: Logo di Tailwind CSS

- ReactQuery: libreria per la gestione dello stato e delle richieste di dati in React.



Figura 15: Logo di ReactQuery

- NextJs: framework React per la creazione di applicazioni web;



Figura 16: Logo di Next.js

1.6) Riferimenti

1.6.1) Normativi

Presentazione pdf del capitolato C9: [C9p.pdf](#) (*versione disponibile al 2025-03-20*)

Norme di Progetto: [Norme di Progetto_v1.0.0.pdf](#)

Piano di Qualifica: [Piano di Qualifica_v1.0.0.pdf](#)

1.6.2) Informativi

Slide del corso(T5): [T05.pdf](#) (*versione disponibile al 2025-03-20*)

Diagrammi dei Casi d'Uso: [Diagrammi Use Case.pdf](#) (*versione disponibile al 2025-03-20*)

Glossario: [Glossario_v1.0](#)

1.6.3) Tecnologici:

- [Typescript](#)
- [Langchain](#)
- [Node.js](#)

- [Nest.js](#)
- [GrokCloud](#)
- [Qdrant](#)
- [PostgreSQL](#)
- [Octokit](#)
- [JiraJs](#)
- [ConfluenceJs](#)
- [Docker](#)
- [React](#)
- [ReactQuery](#)
- [Tailwind CSS](#)
- [Next.js](#)

2) Descrizione

2.1) Obiettivo del prodotto

Questo progetto consiste nella creazione di un *chatbot text-to-text* per l'azienda *azzurro digitale*, che centralizza le informazioni relative all'azienda provenienti da diverse piattaforme e, con l'uso di un *LLM*, le elabora per creare delle risposte in linguaggio naturale. BuddyBot sarà in grado di rispondere a domande che variano dalla data di un determinato incontro fissato su *Jira*, a cosa è cambiato in un determinato commit su *Github* e a richieste specifiche su documenti presenti in *Confluence*. Inoltre aiuterà i nuovi membri a navigare tra le risorse aziendali, rispondendo alle domande frequenti. Tale prodotto, in conclusione, risponde alla necessità di accedere in modo facile e immediato alle informazioni.

2.2) Funzionalità del prodotto

BuddyBot è un assistente virtuale progettato per reperire informazioni aziendali attraverso un'interfaccia semplice ed intuitiva e basata sul linguaggio naturale. Il punto cardine del progetto è il seguente: il sistema si deve connettere alle piattaforme utilizzate dall'azienda, ossia *Jira*, *GitHub* e *Confluence*, estrapolando informazioni da quest'ultime e fornendo le risposte alle domande poste dall'utente.

L'assistente virtuale utilizza tecnologie di Intelligenza Artificiale per interpretare le richieste degli utenti e restituire informazioni personalizzate e contestualizzate.

A seguire, *BuddyBot* garantisce anche la persistenza dei dati, ossia domande e risposte, con il fine di mantenere lo storico della chat agevolando il recupero di informazioni già richieste.

2.3) Utenti e caratteristiche

Il prodotto si rivolge principalmente al team aziendale:

- **Sviluppatori**, che accedono a informazioni tecniche come codice e documentazione;
- **Project Manager**, che usa *BuddyBot* per monitorare task e risorse;
- **Nuovi membri** del team, che vengono supportati nell'*onboarding* e nella ricerca delle informazioni necessarie, facilitando la loro integrazione nell'azienda;
- Membri del dipartimento delle **risorse umane**, che utilizzano *BuddyBot* per gestire le domande relative ai benefici e alle procedure di *onboarding*.

Questa sezione mette in luce il ruolo centrale che BuddyBot può avere poiché, come spiegato in precedenza, centralizza le informazioni e semplifica i processi aziendali attraverso l'uso di IA, aumentando efficienza e produttività per tutti gli utenti coinvolti e diminuendo perdite di tempo.

3) Casi d'uso

3.1) Introduzione ai casi d'uso e obiettivi

In questa sezione vengono elencati dettagliatamente i ***casi d'uso (UC)_G*** individuati dal gruppo in seguito ad analisi e valutazioni circa le specifiche del capitolato. Gli scenari sottostanti seguono uno schema e può prevedere:

- **Titolo:** titolo del caso d'uso;
- **Attori:** il soggetto che interagisce con il sistema nel contesto del caso d'uso;
- **Precondizioni e Postcondizioni:** stato del sistema prima e dopo il caso d'uso;
- **Scenario principale:** descrizione dettagliata delle azioni che l'attore deve compiere per completare il caso d'uso, vengono formalizzati anche ipotesi e risultati attesi;
- **Estensioni:** relazione tra due casi d'uso; indica la situazione in cui un caso d'uso (estendibile) include opzionalmente un altro caso d'uso (esteso) al verificarsi di condizioni specifiche, gestendo scenari particolari senza complicare il caso principale;
- **Inclusioni:** relazione tra due casi d'uso; indica la situazione in cui, prendendo in esame un caso d'uso specifico, alcune funzionalità o azioni comuni appartengono a un altro caso d'uso, sempre eseguito come parte integrante dello scenario principale;
- **Generalizzazioni:** relazione tra due casi d'uso; indica la situazione in cui un caso d'uso più specifico eredita comportamenti e proprietà da un caso d'uso più generale;
- **User Story:** descrizione di una funzionalità del software dal punto di vista dell'utente; aiuta a comprendere le esigenze dell'utente e a definire i requisiti del sistema.

3.2) Attori

Gli attori coinvolti nei casi d'uso sono:

Attori Primari:

- **User:** utente finale che interagisce direttamente con l'interfaccia di BuddyBot ponendo domande e ricevendo risposte;
- **User Interface:** componente del sistema che gestisce l'interazione con l'utente e la presentazione delle informazioni.

Attori Secondari:

- **Backend:** componente del sistema che elabora le richieste, gestisce la business logic e coordina l'interazione con le altre componenti;
- **Jira:** piattaforma esterna da cui il sistema recupera informazioni sui ticket e la gestione dei progetti;
- **GitHub:** piattaforma esterna da cui il sistema recupera informazioni sui repository e il controllo versione;
- **Confluence:** piattaforma esterna da cui il sistema recupera documentazione e informazioni aziendali;
- **LLM:** Large Language Model che elabora le domande e genera le risposte in linguaggio naturale.

3.3) Definizione casi d'uso

3.3.1) UC1, Visualizzazione storico chat

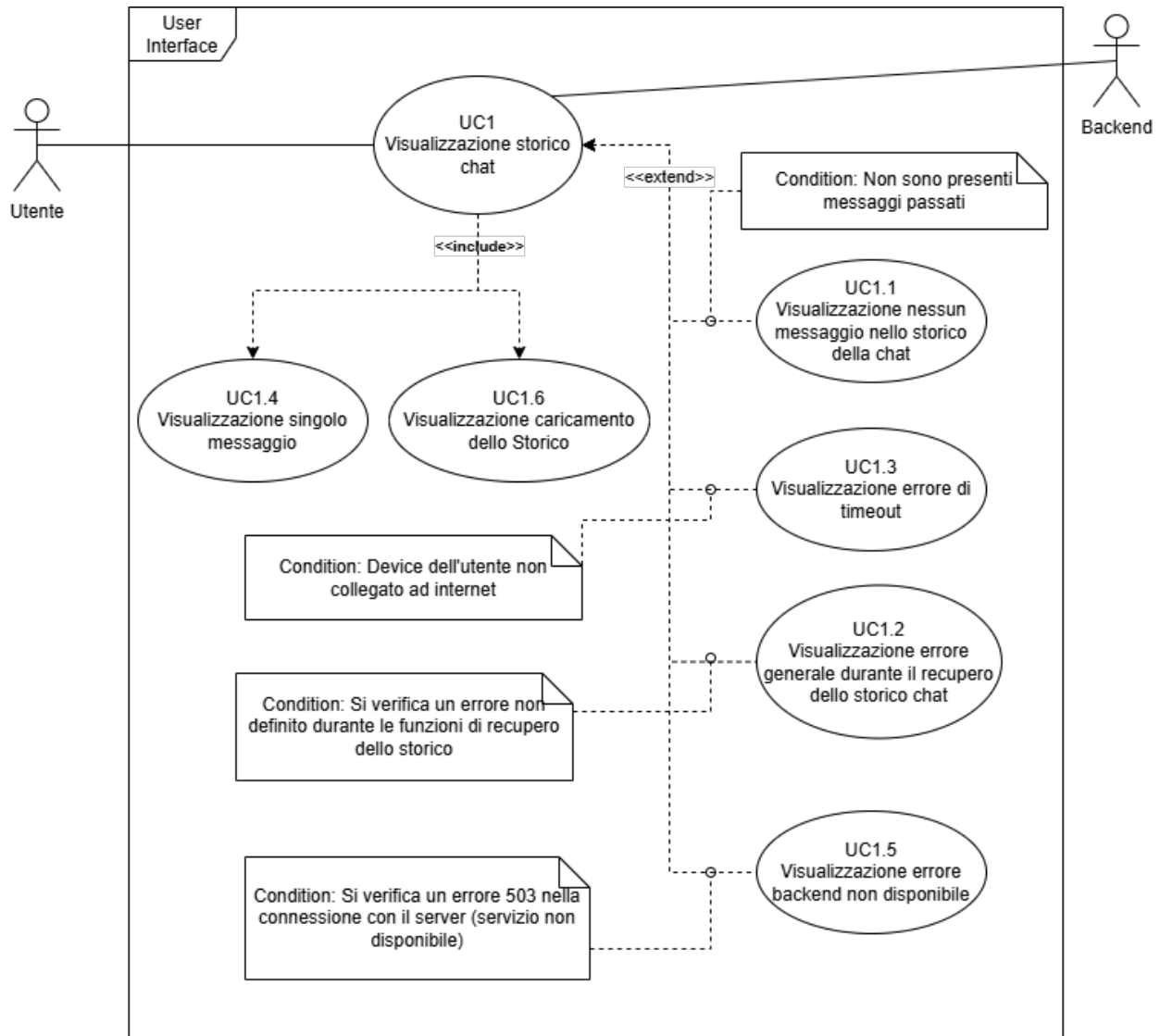


Figura 17: Diagramma UC1, visualizzazione storico chat

Attori coinvolti:

- Primari:
 - User_G.
- Secondari:
 - Backend_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

Postcondizioni

- L'interfaccia utente viene aggiornata con i messaggi delle passate interazioni tra l'utente e Buddy-bot_G.

Scenario principale

- L'utente accede all'interfaccia di Buddybot_G tramite l'applicazione web;
- Vengono recuperati i messaggi delle precedenti interazioni;

- Nella schermata appaiono i messaggi riguardanti le passate interazioni con **Buddybot_G**.

Estensioni

- UC1.1, Visualizzazione errore nessun messaggio nello storico della chat;
- UC1.2, Visualizzazione errore generale durante il recupero dello storico della chat;
- UC1.3, Visualizzazione errore di timeout;
- UC1.5, Visualizzazione errore backend non disponibile;

Inclusioni

- UC1.4, Visualizzazione singolo messaggio.

User story associata

- «Come utente, voglio poter vedere i messaggi delle passate interazioni con Buddybot, in modo da avere una conferma delle informazioni ricevute e poter approfondire eventuali dubbi o richiedere ulteriori dettagli.»

3.3.1.1) UC1.1, Visualizzazione errore nessun messaggio nello storico della chat

Attori coinvolti:

- Primari:
 - **User_G**.
- Secondari:
 - **Backend_G**.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

Postcondizioni

- L'interfaccia utente viene aggiornata con un avviso che informa l'utente che non ci sono messaggi precedenti disponibili.

Scenario principale

- L'utente accede all'interfaccia di **Buddybot_G**
- Viene tentato il recupero dei messaggi delle precedenti interazioni ma non è presente nessun messaggio nello storico della chat;
- L'utente viene informato che non sono presenti messaggi precedenti tramite un avviso.

User story associata

- «Quando l'utente utilizza **Buddybot_G** per la prima volta, viene informato che, non essendoci interazioni pregresse, non sono disponibili messaggi nello storico».

3.3.1.2) UC1.2, Errore durante il recupero dello storico della chat

Attori coinvolti: **User_G**.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

Postcondizioni

- L'interfaccia utente mostra un avviso che informa l'utente dell'impossibilità di recuperare i messaggi precedenti a causa di un problema.

Scenario principale

- L'utente accede all'interfaccia di **Buddybot_G**
- Viene tentato il recupero dei messaggi delle precedenti interazioni ma si verifica un errore durante il processo che impedisce il recupero di tali messaggi;

- L'utente viene informato che si è verificato un errore durante il recupero dei messaggi precedenti tramite un avviso.

User story associata

- «Come utente, voglio essere informato in modo chiaro se si verifica un errore durante il recupero dei messaggi precedenti per poter agire di conseguenza senza confusione».

3.3.1.3) UC1.3, Visualizzazione errore di timeout**Attori coinvolti:**

- Primari:
 - User_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

Postcondizioni

- L'interfaccia utente viene aggiornata con un messaggio di errore che informa l'utente del fallimento dell'operazione a causa di un timeout della richiesta.

Scenario principale

- L'utente accede all'interfaccia di Buddybot_G e richiede l'esecuzione di un'operazione che coinvolge una comunicazione esterna;
- Il sistema tenta di eseguire la richiesta, ma non riesce a completarla entro il tempo massimo previsto;
- L'utente viene informato del fallimento dell'operazione tramite un messaggio di errore che segnala un timeout.

User story associata

- «Come utente, voglio essere informato se il motivo del fallimento della richiesta è un timeout, così da poter decidere se riprovare o attendere».

3.3.1.4) UC1.5, Visualizzazione errore backend non disponibile**Attori coinvolti:**

- Primari:
 - User_G.
- Secondari:
 - Backend_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

Postcondizioni

- L'interfaccia utente viene aggiornata con un messaggio di errore che informa l'utente che non è stato possibile eseguire l'operazione richiesta perchè il backend_G non è al momento disponibile.

Scenario principale

- L'utente accede all'interfaccia di Buddybot_G e chiede all'interfaccia utente di eseguire un'operazione che richiede una risposta al backend_G
- Viene tentata l'esecuzione della richiesta dell'utente ma non viene portata a termine perchè il backend_G non è al momento disponibile;
- L'utente viene informato che si è verificato un errore durante l'esecuzione della richiesta tramite un avviso.

User story associata

- «Come utente, voglio essere informato se il motivo del fallimento della richiesta è il *backend_G* non disponibile così da poter contattare il supporto tecnico».

3.3.1.5) UC1.6, Visualizzazione caricamento dello Storico della chat

Attori coinvolti:

- Primari:
 - User

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

Postcondizioni

- Nella *UI_G* viene visualizzata un'animazione di caricamento circolare (loading), a indicare che il sistema sta recuperando lo storico della chat.

Scenario principale

- L'utente accede a BuddyBot e viene inviata al backend la richiesta di recupero storico chat.
- Mentre è in corso tutta la procedura legata al recupero dei messaggi fino alla apparizione a video, compare a schermo un'animazione di caricamento circolare.

User story associata

- «Come utente, voglio vedere un'animazione di caricamento mentre attendo le interazioni passate, in modo da avere un feedback visivo che mi confermi che il sistema sta elaborando la richiesta.»

3.3.1.6) UC1.4, Visualizzazione singolo messaggio

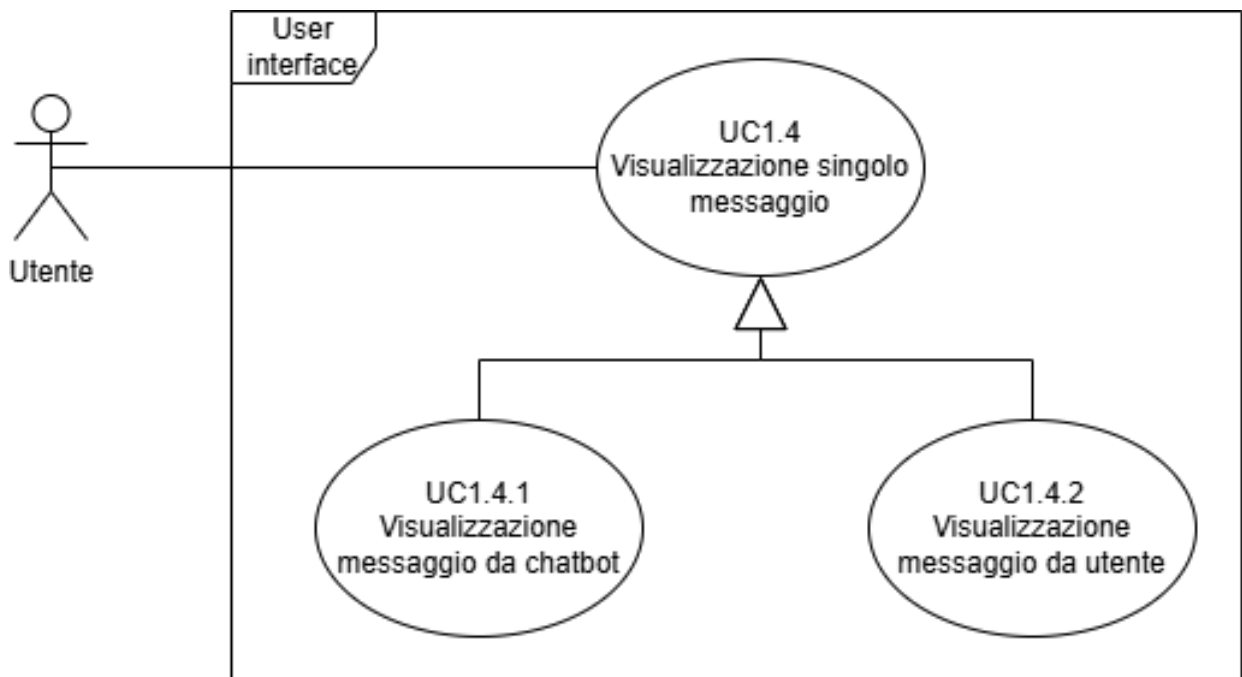


Figura 18: Diagramma UC1.4, Visualizzazione messaggio singolo

Attori coinvolti:

- Primari:
 - *User_G*.

Precondizioni

- La UI di *Buddybot_G* è funzionante e pronta per l'utilizzo;
- Esiste almeno un messaggio proveniente da una passata interazione tra l'utente e *Buddybot_G*

Postcondizioni

- La user interface di *Buddybot_G* viene aggiornata mostrando il messaggio.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza il messaggio della passata interazione con *Buddybot_G*.

Generalizzazioni

- UC1.4.1, Visualizzazione messaggio da chatbot;
- UC1.4.2, Visualizzazione messaggio da utente.

User story associata

- «Come utente voglio poter visualizzare un singolo messaggio della chat in modo chiaro e comprensibile in modo da avere sempre una chiara idea del contesto e dei dettagli delle passate interazioni con *Buddybot_G*».

3.3.1.6.1) UC1.4.1, Visualizzazione messaggio da chatbot

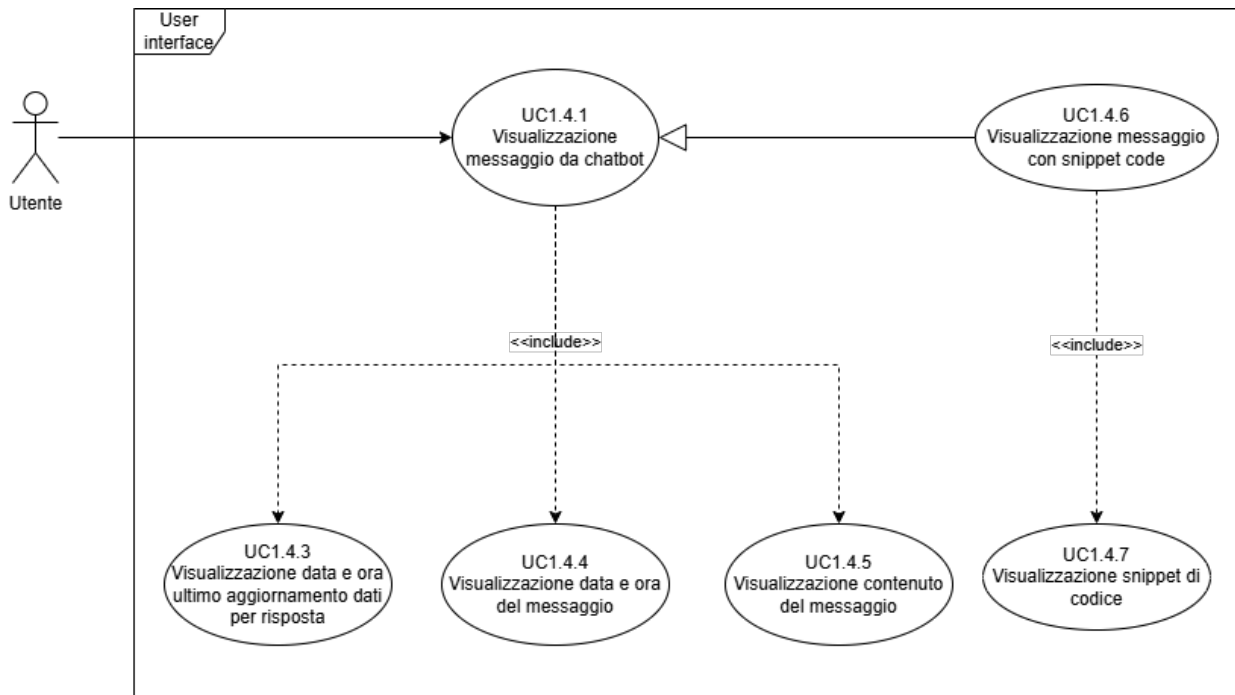


Figura 19: Diagramma UC1.4.1, Visualizzazione messaggio chatbot

Attori coinvolti:

- Primari:
 - *User_G*.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- Esiste almeno un messaggio di risposta di *Buddybot_G* proveniente da una passata interazione tra l'utente e *Buddybot_G*

Postcondizioni

- L'interfaccia utente di *Buddybot_G* viene aggiornata mostrando il messaggio inviato da *Buddybot_G*.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza il messaggio inviato da *Buddybot_G* nella parte sinistra dell'interfaccia utente e con lo sfondo del messaggio di colore blu;

Inclusioni

- UC1.4.5, Visualizzazione contenuto del messaggio;
- UC1.4.4, Visualizzazione data e ora del messaggio;
- UC1.4.3, Visualizzazione ultimo aggiornamento dati per risposta;

Generalizzazioni

- UC1.4.6, Visualizzazione messaggio con snippet code.

User story associata

- «Come utente, desidero che i messaggi inviati da BuddyBot siano visivamente distinti da quelli inviati dagli utenti, in modo da poter identificare rapidamente l'origine di ciascun messaggio e mantenere una chiara comprensione del flusso conversazionale durante le interazioni con *Buddybot_G*».

3.3.1.6.2) UC1.4.5, Visualizzazione contenuto del messaggio

Attori coinvolti:

- Primari:
 - User_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- Esiste almeno un messaggio proveniente da una passata interazione tra l'utente e Buddybot_G.

Postcondizioni

- L'interfaccia utente di Buddybot_G viene aggiornata mostrando il contenuto del messaggio in formato markdown.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza il contenuto del messaggio;

User story associata

- «Come utente, voglio poter visualizzare il contenuto di un messaggio della chat in modo da comprendere le informazioni scambiate durante le passate interazioni con Buddybot».

3.3.1.6.3) UC1.4.4, Visualizzazione data e ora del messaggio;

Attori coinvolti:

- Primari:
 - User_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- Esiste almeno un messaggio proveniente da una passata interazione tra l'utente e Buddybot_G.

Postcondizioni

- L'interfaccia utente di Buddybot_G viene aggiornata mostrando la data e l'ora del messaggio.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza la data e l'ora del messaggio.

User story associata

- «Come utente, voglio poter visualizzare la data e l'ora di un messaggio della chat in modo da sapere quando è stato inviato e contestualizzarlo all'interno delle mie interazioni con Buddybot_G».

3.3.1.6.4) UC1.4.3, Visualizzazione data e ora ultimo aggiornamento dati risposta

Attori coinvolti:

- Primari:
 - User_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- Esiste almeno un messaggio proveniente da una passata interazione tra l'utente e Buddybot_G.

Postcondizioni

- L'interfaccia utente di Buddybot_G viene aggiornata mostrando la data e l'ora dell'ultimo aggiornamento dei dati utilizzati per generare la risposta.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza la data e l'ora dell'ultimo aggiornamento dei dati usati per generare la risposta;

User story associata

- «Come utente, voglio poter visualizzare quando è avvenuto l'ultimo aggiornamento dei dati utilizzati per generare la risposta così da essere sicuro che la risposta ricevuta sia accurata».

3.3.1.6.5) UC1.4.6, Visualizzazione messaggio con snippet code**Attori coinvolti**

- Primari:
 - User_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- Esiste almeno un messaggio di risposta di Buddybot_G proveniente da una passata interazione tra l'utente e Buddybot_G

Postcondizioni

- L'interfaccia utente di Buddybot_G viene aggiornata mostrando il messaggio inviato da Buddybot_G al cui interno è presente almeno uno snippet code.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza il messaggio inviato da Buddybot_G in cui è presente uno snippet code.

Inclusioni

- UC1.4.7, Visualizzazione snippet di codice.

User story associata

- «Come utente, nei messaggi inviati da BuddyBot_G vorrei vedere con un formato differente gli eventuali snippet di codice.»

3.3.1.6.6) UC1.4.7, Visualizzazione snippet di codice**Attori coinvolti:**

- Primari:
 - User_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- Esiste almeno un messaggio proveniente da una passata interazione tra l'utente e Buddybot_G

Postcondizioni

- L'interfaccia utente di Buddybot_G viene aggiornata mostrando lo snippet di codice del messaggio.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza lo snippet di codice del messaggio.

User story associata

- «Come utente, voglio poter visualizzare uno snippet di codice che BuddyBot_G inserisce all'interno di un suo messaggio».

3.3.1.6.7) UC1.4.2, Visualizzazione messaggio da utente

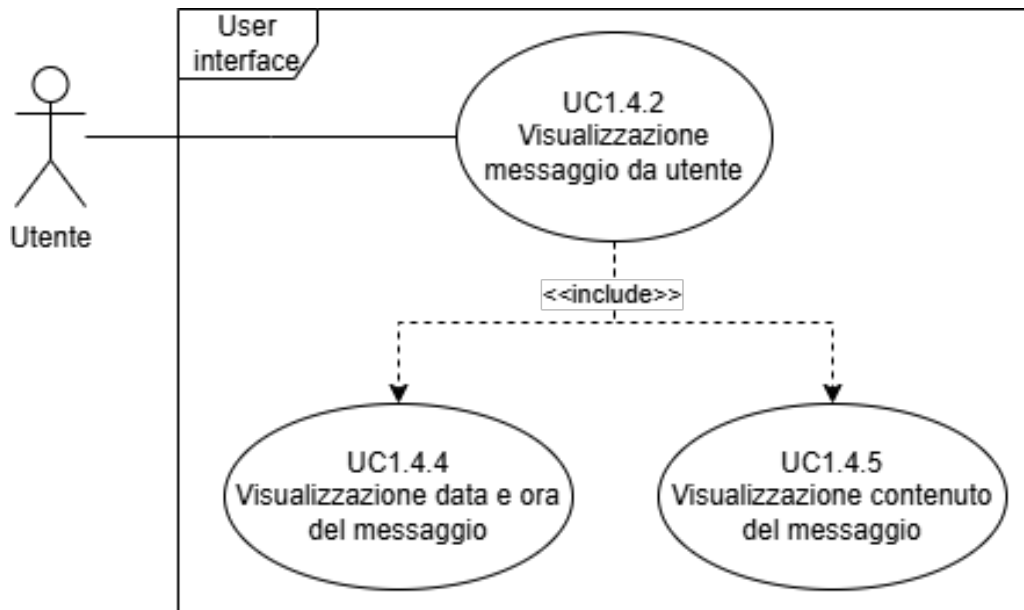


Figura 20: Diagramma UC1.4.2, Visualizzazione messaggio user

Attori coinvolti

- Primari:
 - *User_G*.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- Esiste almeno un messaggio inviato dall'utente in una passata interazione con *Buddybot_G*

Postcondizioni

- L'interfaccia utente di *Buddybot_G* viene aggiornata mostrando il messaggio inviato dall'utente.

Scenario principale

- L'utente vuole visualizzare il contenuto della chat;
- L'utente visualizza il messaggio inviato da sé stesso nella parte destra dell'Interfaccia utente e con lo sfondo del messaggio di colore grigio.

Inclusioni

- UC1.4.5, Visualizzazione contenuto del messaggio;
- UC1.4.4, Visualizzazione data e ora del messaggio.

User story associata

- «Come utente, desidero che i messaggi inviati da me siano visivamente distinti da quelli inviati da BuddyBot, in modo da poter identificare rapidamente l'origine di ciascun messaggio e mantenere una chiara comprensione del flusso conversazionale durante le interazioni con *Buddybot_G*».

3.3.2) UC2, Scrittura domanda in linguaggio naturale

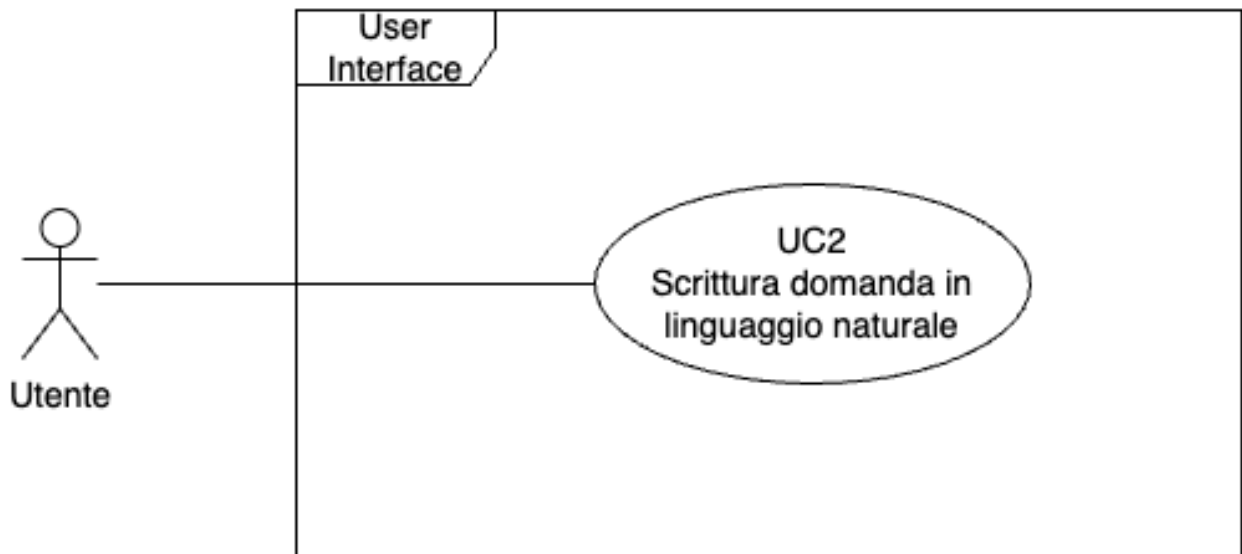


Figura 21: Diagramma UC2, Scrittura domanda in linguaggio naturale

Attori coinvolti:

- Primari:
 - User_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

Postcondizioni

- L'interfaccia utente viene aggiornata con la domanda scritta dall'utente.

Scenario principale

- L'utente accede all'interfaccia di Buddybot_G tramite l'applicazione web;
- L'utente scrive la propria domanda nel campo di input dedicato con l'opzione di incollare il contenuto copiato in precedenza;
- Il campo di input viene aggiornato con la domanda scritta dall'utente.

User story associata

- «Come utente, voglio poter scrivere la mia domanda in linguaggio naturale in modo da poter interagire con Buddybot in modo naturale e intuitivo.»

3.3.3) UC3, Visualizzazione risposta generata

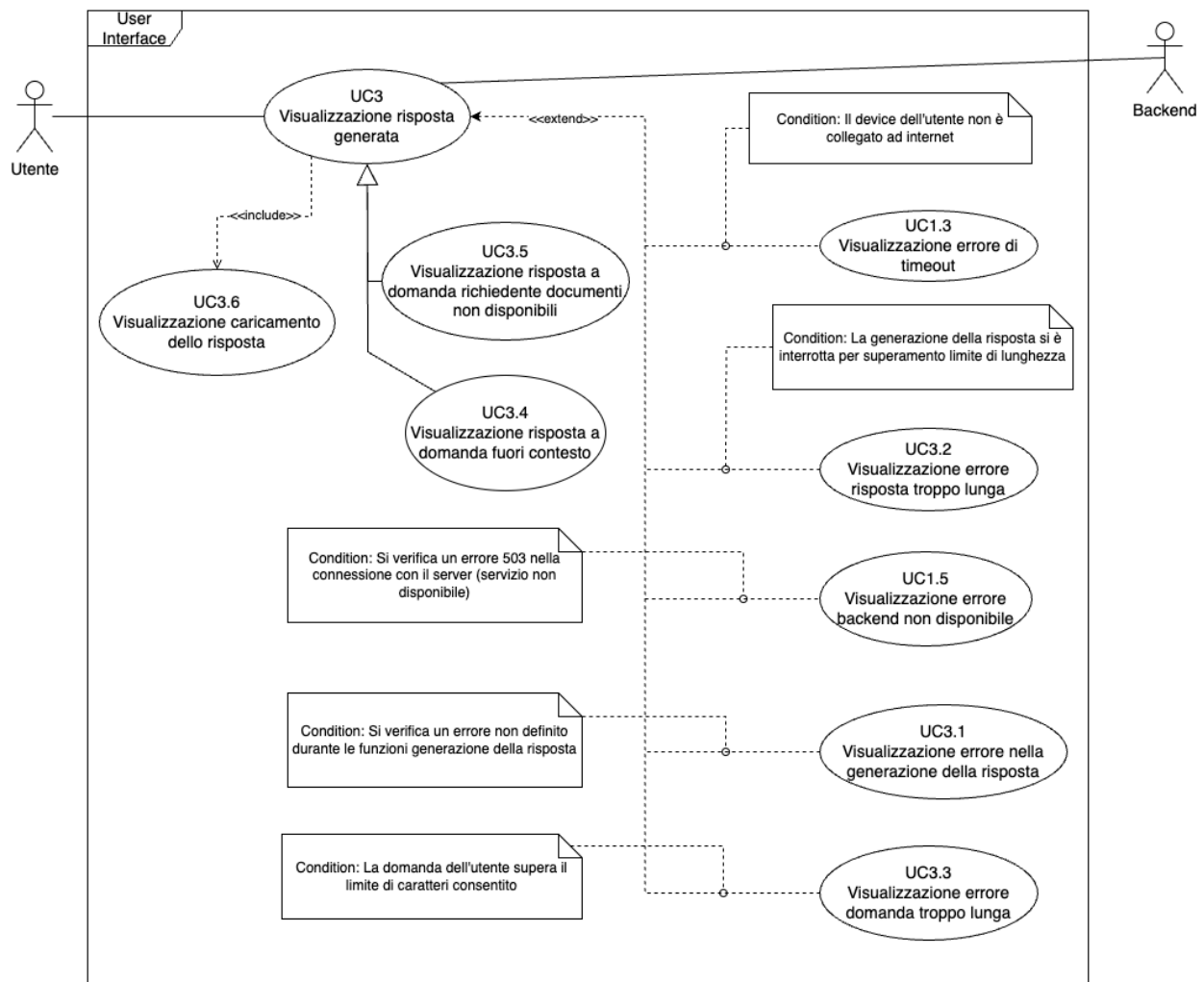


Figura 22: Diagramma UC3, Invio domanda dell'utente

Attori coinvolti:

- Primari:
 - User_G.
- Secondari:
 - Backend_G.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- L'utente ha scritto la propria domanda nel campo di input dedicato.

Postcondizioni

- L'interfaccia utente viene aggiornata con il contenuto, la data, l'ora e il mittente del messaggio di risposta di Buddybot_G alla domanda dell'utente e la domanda posta dall'utente.

Scenario principale

- L'utente dopo aver scritto la propria domanda, preme il tasto di invio;
- L'interfaccia utente invia la richiesta dell'utente al backend_G per generare una risposta alla domanda;
- L'interfaccia utente riceve la risposta alla domanda dell'utente;
- L'interfaccia utente viene aggiornata con il messaggio inviato dall'utente;

- L'interfaccia utente viene aggiornata con il messaggio di risposta di **Buddybot_G** alla domanda dell'utente.

Estensioni

- UC1.3, Visualizzazione errore di timeout;
- UC1.5, Visualizzazione errore backend non disponibile;
- UC3.1, Visualizzazione errore nella generazione della risposta;
- UC3.2, Visualizzazione errore risposta troppo lunga;
- UC3.3, Visualizzazione errore domanda troppo lunga.

Generalizzazioni

- UC3.4, Visualizzazione risposta a domanda fuori contesto
- UC3.5, Visualizzazione risposta a domanda richiedente documenti non disponibili

User story associata

- «Come utente voglio interagire con **Buddybot_G** attraverso una chat per porre domande e ricevere risposte in modo rapido e intuitivo. Inoltre, desidero visualizzare tutti i dettagli di ogni messaggio, inclusi contenuto, data, ora e mittente, in modo da avere un'esperienza chiara e completa durante l'interazione.»

3.3.3.1) UC3.1, Visualizzazione errore nella generazione della risposta

Attori coinvolti:

- Primari:
 - **User_G**.
- Secondari:
 - **Backend_G**.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente;
- L'utente ha scritto la propria domanda nel campo di input dedicato.

Postcondizioni

- L'interfaccia utente viene aggiornata con un messaggio di errore generico che informa l'utente che non è stato possibile generare la risposta alla sua domanda.

Scenario principale

- L'utente dopo aver scritto la propria domanda, preme il tasto di invio;
- L'interfaccia utente invia la richiesta dell'utente al **backend_G** per generare una risposta alla domanda;
- Si verifica un errore durante la generazione della risposta;
- L'interfaccia utente viene aggiornata con un messaggio di errore che informa l'utente che non è stato possibile generare la risposta alla sua domanda.

User story associata

- «Come utente voglio essere informato tramite un messaggio di errore chiaro e comprensibile se si verifica un problema durante la generazione della risposta da parte di **Buddybot_G**, in modo da sapere che la mia richiesta non è stata elaborata e poter eventualmente riprovare».

3.3.3.2) UC3.2, Visualizzazione errore risposta troppo lunga

Attori coinvolti:

- Primari:
 - **User_G**.
- Secondari:

▸ **Backend_G**.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente;
- L'utente ha scritto la propria domanda nel campo di input dedicato.

Postcondizioni

- L'interfaccia utente viene aggiornata con un messaggio di errore che informa l'utente che la risposta da generare per quella domanda è troppo lunga e quindi di riprovare con una domanda più specifica.

Scenario principale

- L'utente dopo aver scritto la propria domanda, preme il tasto di invio;
- L'interfaccia utente invia la richiesta dell'utente al **backend_G** per generare una risposta alla domanda;
- La generazione della risposta si interrompe perchè la risposta ha superato la lunghezza massima consentita;
- L'interfaccia utente viene aggiornata con un messaggio di errore che informa l'utente che la risposta da generare per quella domanda è troppo lunga e quindi di riprovare con una domanda più specifica.

User story associata

- «Come utente voglio essere informato tramite un messaggio di errore chiaro e comprensibile se la risposta da generare per quella domanda è troppo lunga così da poter riprovare con una domanda più specifica».

3.3.3.3) UC3.3, Visualizzazione errore domanda troppo lunga**Attori coinvolti:**

- Primari:
 - **User_G**.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente;
- L'utente ha scritto la propria domanda nel campo di input dedicato.

Postcondizioni

- L'interfaccia utente viene aggiornata con un messaggio di errore che informa l'utente che la domanda è troppo lunga e quindi di riprovare con una domanda più specifica.

Scenario principale

- L'utente dopo aver scritto la propria domanda, preme il tasto di invio;
- L'interfaccia utente controlla la lunghezza della domanda;
- La lunghezza della domanda supera la lunghezza massima consentita;
- L'interfaccia utente viene aggiornata con un messaggio di errore che informa l'utente che la domanda è troppo lunga e quindi di riprovare con una domanda più specifica.

User story associata

- «Come utente voglio essere informato tramite un messaggio di errore chiaro e comprensibile se la domanda è troppo lunga così da poter riprovare con una domanda più specifica».

3.3.3.4) UC3.4, Visualizzazione risposta a domanda fuori contesto**Attori coinvolti:**

- Primari:
 - **User_G**.
- Secondari:

▸ **Backend_G.****Precondizioni**

- L'interfaccia utente è pronta all'interazione con l'utente;
- L'utente ha scritto la propria domanda nel campo di input dedicato.

Postcondizioni

- L'interfaccia utente viene aggiornata con la domanda dell'utente e con un messaggio che indica che la domanda è al di fuori delle competenze di **BuddyBot_G**.

Scenario principale

- L'utente dopo aver scritto la propria domanda, preme il tasto di invio;
- L'interfaccia utente invia la richiesta dell'utente al **backend_G** per generare una risposta alla domanda;
- L'interfaccia utente riceve la risposta alla domanda dell'utente;
- L'interfaccia utente viene aggiornata con il messaggio inviato dall'utente;
- L'interfaccia utente viene aggiornata con un messaggio di risposta di **BuddyBot_G** che spiega che non è stato possibile rispondere adeguatamente alla domanda, poiché riguarda temi fuori dalle sue competenze.

User story associata

- «Come utente, voglio che l'interfaccia utente mostri un messaggio di risposta quando la mia domanda riguarda temi fuori dalle competenze di **BuddyBot_G**, in modo da essere informato che non è stato possibile ottenere una risposta adeguata.»

3.3.3.5) UC3.5, Visualizzazione risposta a domanda richiedente documenti non disponibili**Attori coinvolti:**

- Primari:
 - **User_G.**
- Secondari:
 - **Backend_G.**

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente;
- L'utente ha scritto la propria domanda nel campo di input dedicato.

Postcondizioni

- L'interfaccia utente viene aggiornata con la domanda dell'utente e con un messaggio di risposta di **BuddyBot_G**, che spiega che la domanda riguarda dati presenti in documenti non ancora disponibili.

Scenario principale

- L'utente dopo aver scritto la propria domanda, preme il tasto di invio;
- L'interfaccia utente invia la richiesta dell'utente al **backend_G** per generare una risposta alla domanda;
- L'interfaccia utente riceve la risposta alla domanda dell'utente;
- L'interfaccia utente viene aggiornata con il messaggio inviato dall'utente;
- L'interfaccia utente viene aggiornata con un messaggio di risposta di **BuddyBot_G** che spiega che non è stato possibile rispondere adeguatamente alla domanda, poiché riguarda dati presenti in documenti non ancora disponibili.

User story associata

- «Come utente, voglio che l'interfaccia utente mostri un messaggio di risposta quando la mia domanda riguarda dati presenti in documenti non ancora disponibili, in modo da essere informato che la risposta non può essere fornita al momento.»

3.3.3.6) UC3.6, Visualizzazione caricamento della risposta

Attori coinvolti:

- Primari:
 - User

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente
- L'utente invia un messaggio
- La risposta deve ancora essere visualizzata a schermo

Postcondizioni

- Nella **UI_G** viene visualizzata un'animazione di caricamento composta da tre puntini, indicativa dell'elaborazione in corso.

Scenario principale

- L'utente accede all'interfaccia di **Buddybot_G** e chiede all'interfaccia utente di eseguire un'operazione che richiede una risposta al **backend_G**
- Mentre è in corso tutta la procedura legata all'elaborazione della risposta fino alla sua apparizione a video, compaiono a schermo tre puntini di caricamento.

User story associata

- «Come utente, voglio vedere un'animazione di caricamento mentre attendo la risposta, in modo da avere un feedback visivo che mi confermi che il sistema sta elaborando la mia richiesta.»

3.3.4) UC4, Invio richiesta con domanda dell'utente

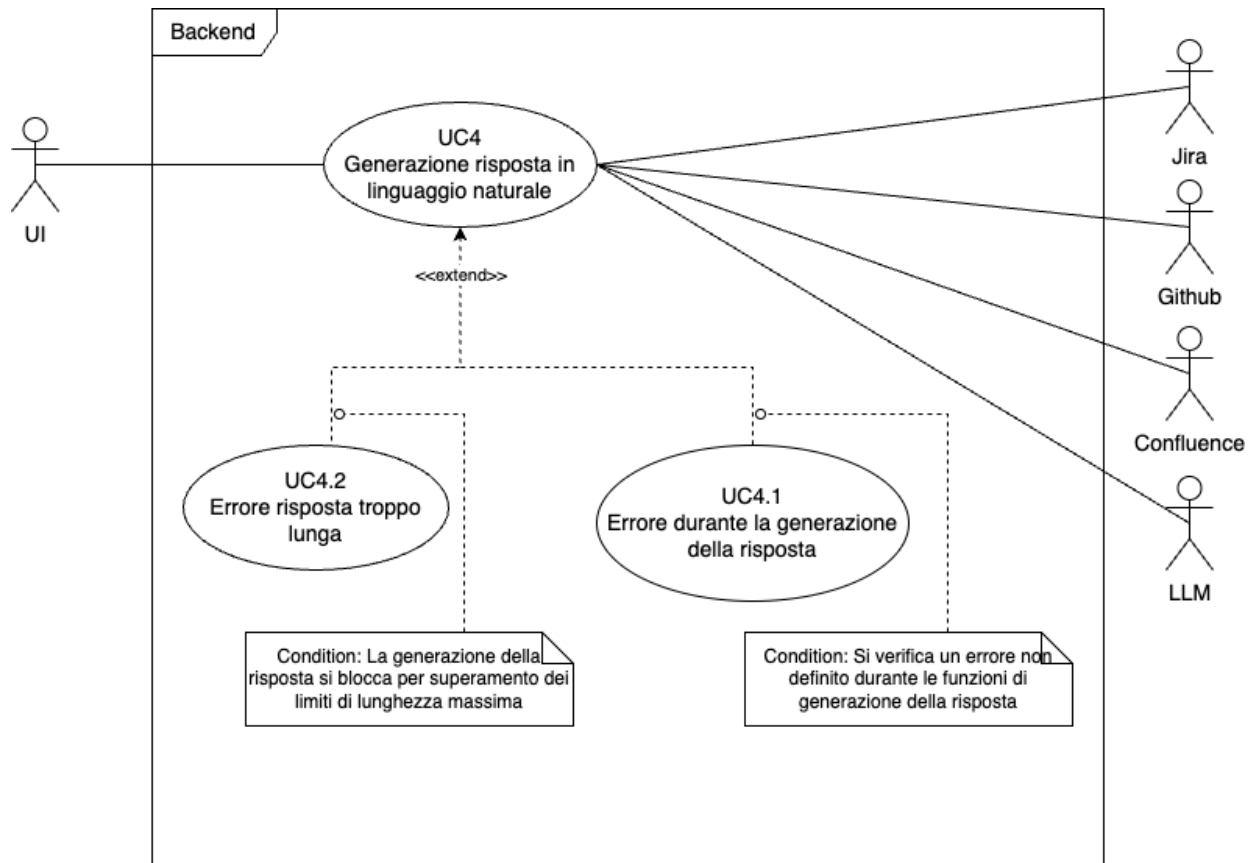


Figura 23: Diagramma UC4, Invio richiesta con domanda dell'utente

Attori coinvolti:

- Primari:
 - *User Interface_G*.
- Secondari:
 - *Jira_G*;
 - *Github_G*;
 - *Confluence_G*;
 - *LLM_G*.

Precondizioni

- La UI e il *backend_G* di *Buddybot_G* comunicano correttamente tramite una connessione stabile e funzionante;
- La User Interface ha acquisito una domanda dall'utente;
- Il *backend_G* ha accesso ai seguenti dati provenienti dalle fonti *Jira_G*, *Github_G*, *Confluence_G*:
 - *Github_G*:

Per ogni repository:

- Nome della repository;
- Id della repository;
- Descrizione della repository;
- Data di creazione della repository;
- Ultima data di aggiornamento della repository;
- Linguaggio principale della repository.

Per ogni commit:

- Hash del commit;

- Messaggio del commit;
- Data e ora dell'ultimo commit;
- Branch associato al commit;
- File modificati nel commit;
- Autore dell'ultimo commit;
- Nome della repository di appartenenza del commit;
- Nome del branch di appartenenza del commit.

Per ogni pull request:

- Id della pull request;
- Titolo della pull request;
- Descrizione della pull request;
- Stato della pull request;
- Assegnatario della pull request;
- Reviewers della pull request;
- Commenti della pull request;
- File modificati nella pull request;
- Branch di origine della pull request;
- Branch di destinazione della pull request;
- Nome repository di appartenenza.

Per ogni workflow:

- Id del workflow;
- Nome del workflow;
- Stato del workflow;
- Nome repository di appartenenza;
- Lista delle run per il workflow.

Per ogni workflow run:

- Id della run;
- Stato della run;
- Durata in secondi della run;
- Link del log della run;
- Trigger della run;
- Id del workflow di appartenenza;
- Nome del workflow di appartenenza.

Per ogni file:

- Path del file;
- SHA del file;
- Nome repository di appartenenza;
- Nome branch di appartenenza;
- Contenuto del file.

► *Confluence*_G:

- Id di una pagina;
- Titolo di una pagina;
- Status di una pagina;
- Autore di una pagina;
- Owner di una pagina;
- Spazio di una pagina;
- Contenuto di una pagina.

► **Jira_G:**

- Id di un ticket;
- Titolo di un ticket;
- Descrizione di un ticket;
- Assegnatario di un ticket;
- Stato di un ticket;
- Sprint di appartenenza di un ticket;
- Story point estimate di un ticket;
- Creatore di un ticket;
- Priorità;
- Data di scadenza;
- Ticket collegati;
- Commenti del ticket.

Per ogni commento del ticket:

- Autore del commento;
- Data di creazione;
- Contenuto del commento.

Postcondizioni

- Il **backend_G** genera e restituisce una risposta accurata in linguaggio naturale alla domanda ricevuta dall'interfaccia utente.
- La domanda dell'utente e la relativa risposta vengono salvate nello storico della chat per future consultazioni.

Scenario principale

- La UI invia al **backend_G** di **Buddybot_G** la domanda formulata dall'utente;
- Il **backend_G** analizza la domanda e recupera tutti i documenti rilevanti di **Jira_G**, **GitHub_G** e **Confluence_G**, necessari per elaborare una risposta accurata;
- La domanda dell'utente, insieme ai documenti recuperati, viene inviata al **LLM_G** per la generazione della risposta in linguaggio naturale;
- La risposta generata dal **LLM_G** insieme alla data e ora dell'ultimo aggiornamento dei dati utilizzati per la generazione della risposta viene restituita dal **backend_G** alla User Interface;
- La domanda dell'utente e la risposta generata vengono aggiunte allo storico della chat, rendendole disponibili per la successiva visualizzazione.

Estensioni

- UC4.1, Errore durante la generazione della risposta.
- UC4.2, Errore risposta troppo lunga;

User story associata

- «Come utente di **Buddybot_G**, voglio poter porre domande attraverso l'interfaccia utente e ricevere risposte accurate in linguaggio naturale, basate sui documenti rilevanti recuperati da **Jira_G**, **GitHub_G** e **Confluence_G**, così da ottenere informazioni utili e coerenti con il contesto del sistema. Voglio inoltre poter visualizzare la data e l'ora dell'ultimo aggiornamento dei dati utilizzati per la generazione della risposta così da poter sapere se i dati sono stati aggiornati recentemente o meno.»

3.3.4.1) UC4.1, Errore durante la generazione della risposta**Attori coinvolti:**

- Primari:
 - **User Interface_G**.
- Secondari:

- *LLM_G*.
- *Jira_G*.
- *Github_G*.
- *Confluence_G*.

Precondizioni

- La user interface e il *backend_G* di *Buddybot_G* comunicano correttamente tramite una connessione stabile e funzionante;
- L'interfaccia utente ha acquisito una domanda dall'utente.

Postcondizioni

- Il *backend_G* ritorna un errore informando che non è stato possibile generare la risposta alla domanda richiesta.

Scenario principale

- L'interfaccia utente invia al *backend_G* di *Buddybot_G* la domanda formulata dall'utente.
- Il *backend_G* tenta di analizzare la domanda, di recuperare i documenti pertinenti e di generare una risposta in linguaggio naturale ma non riesce a terminare per un errore durante uno di questi processi;
- Il *backend_G* ritorna un errore generico informando l'interfaccia utente che non è stato possibile generare la risposta alla domanda richiesta.

User story associata

- «Come utente voglio essere informato in modo chiaro e immediato nel caso in cui BuddyBot non riesca a generare una risposta alla mia domanda, così da poter eventualmente riformulare la domanda o contattare un supporto alternativo».

3.3.4.2) UC4.2, Errore risposta troppo lunga

Attori coinvolti:

- Primari:
 - *User Interface_G* .
- Secondari:
 - *LLM_G* .
 - *Jira_G* .
 - *Github_G* .
 - *Confluence_G* .

Precondizioni

- La user interface e il *backend_G* di *Buddybot_G* comunicano correttamente tramite una connessione stabile e funzionante;
- L'interfaccia utente ha acquisito una domanda dall'utente.

Postcondizioni

- Il *backend_G* ritorna un errore informando che non è stato possibile generare la risposta alla domanda richiesta in quanto la risposta supererebbe la lunghezza massima consentita.

Scenario principale

- L'interfaccia utente invia al *backend_G* di *Buddybot_G* la domanda formulata dall'utente;
- Il *backend_G* analizza la domanda e recupera tutti i documenti rilevanti di *Jira_G*, *GitHub_G* e *Confluence_G*, necessari per elaborare una risposta accurata;
- La domanda dell'utente, insieme ai documenti recuperati, viene inviata al *LLM_G* per la generazione della risposta in linguaggio naturale;
- La generazione della risposta si interrompe perchè la risposta ha superato la lunghezza massima consentita;
- Il *backend_G* ritorna un errore informando che non è stato possibile generare la risposta alla domanda richiesta in quanto la risposta supererebbe la lunghezza massima consentita.

User story associata

- «Come utente voglio essere informato in modo chiaro e immediato nel caso in cui BuddyBot non riesca a generare una risposta alla mia domanda perchè la risposta supererebbe la lunghezza massima consentita in modo da poter eventualmente riformulare la domanda».

3.3.5) UC5, Modifica domanda

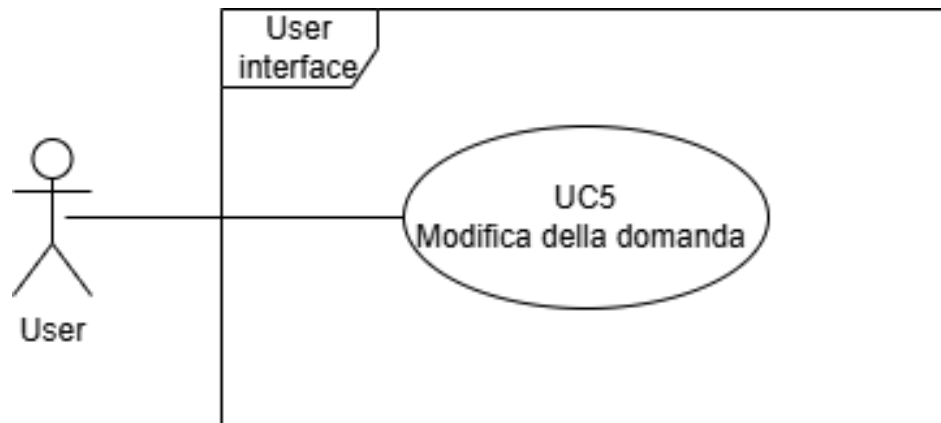


Figura 24: Diagramma UC5, Modifica domanda

Attori coinvolti:

- Primari:
 - *User_G*.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente;
- L'utente ha già fatto una domanda, quindi c'è almeno un messaggio.

Postcondizioni

- L'interfaccia utente viene aggiornata con la nuova domanda scritta dall'utente.

Scenario principale

- L'utente accede all'interfaccia di *Buddybot_G* tramite l'applicazione web;
- L'utente scrive la propria domanda nel campo di input dedicato;
- L'utente si accorge di un errore o vuole modificare la domanda;
- L'utente schiaccia sull'icona di modifica e modifica la domanda;
- La vecchia domanda viene sostituita con quella nuova.

User story associata

- «Come utente, voglio modificare la mia domanda per renderla più chiara o correggere un errore dopo averla mandata.»

3.3.6) UC6, Visualizzazione UI

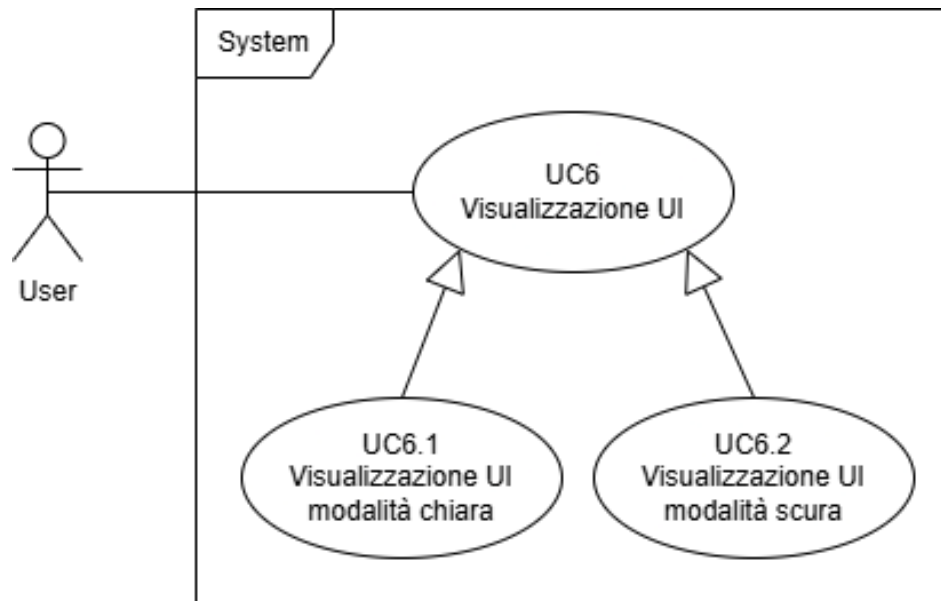


Figura 25: Diagramma UC6, visualizzazione UI

Attori coinvolti:

- Primari:
 - *User_G*.

Precondizioni

- La UI di *Buddybot_G* è funzionante e pronta per l'utilizzo;

Postcondizioni

- Lo user visualizza la UI e tutte le funzionalità sono pronte all'uso.

Scenario principale

- L'utente vuole visualizzare la UI di *Buddybot_G* e usare le sue funzionalità;
- L'utente visualizza la UI di *Buddybot_G* e può usare tutte le sue funzionalità.

Generalizzazioni

- UC6.1, visualizzazione UI modalità chiara;
- UC6.2, visualizzazione UI modalità scura.

User story associata

- «Come utente voglio poter visualizzare e usare la UI di *Buddybot_G*».

3.3.6.0.1) UC6.1, Visualizzazione UI in modalità chiara

Attori coinvolti:

- Primari:
 - *User_G*.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- La UI è in modalità scura.

Postcondizioni

- L'interfaccia utente di *Buddybot_G* è visualizzata in modalità chiara.

Scenario principale

- L'utente vuole visualizzare la UI in modalità chiara;
- L'utente usa il tasto per cambiare la modalità;
- La UI passa in modalità chiara.

User story associata

- «Come utente, desidero visualizzare la UI in modalità chiara anziché scura».

3.3.6.0.2) UC6.2, Visualizzazione UI in modalità scura

Attori coinvolti:

- Primari:
 - *User_G*.

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- La UI è in modalità chiara.

Postcondizioni

- L'interfaccia utente di *Buddybot_G* è visualizzata in modalità scura.

Scenario principale

- L'utente vuole visualizzare la UI in modalità scura;
- L'utente usa il tasto per cambiare la modalità;
- La UI passa in modalità scura.

User story associata

- «Come utente, desidero visualizzare la UI in modalità scura anziché chiara».

3.3.7) UC7, Visualizzazione link - icona

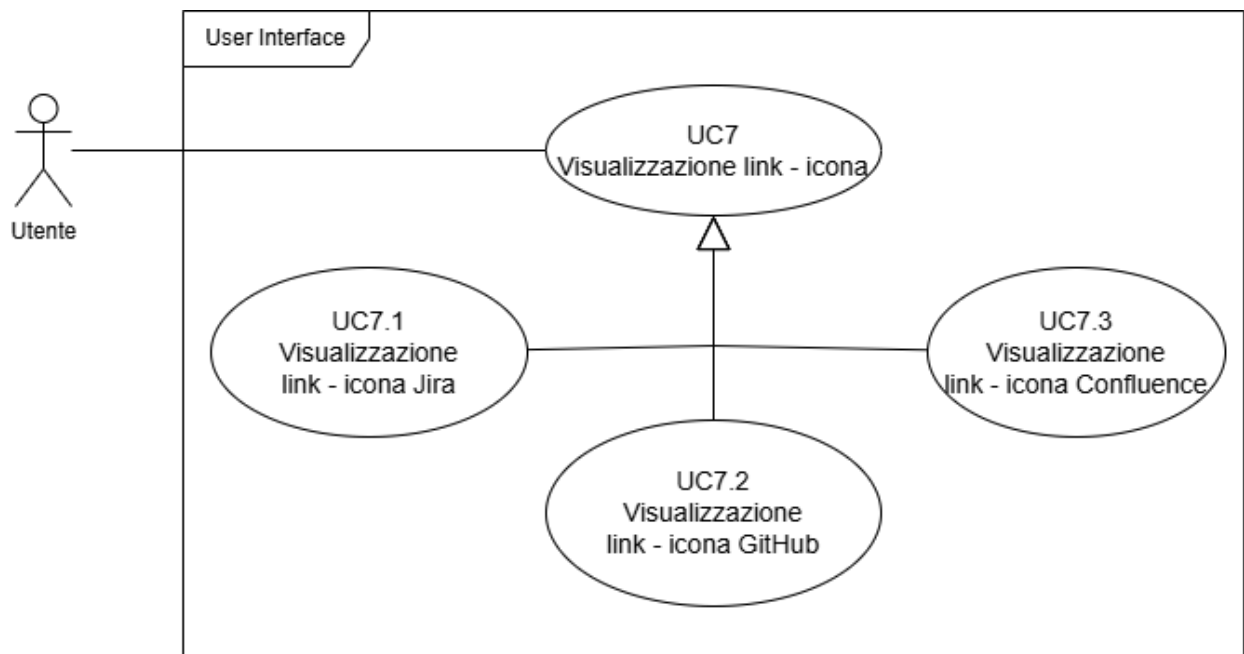


Figura 26: Diagramma UC7, Visualizzazione link - icone

Attori coinvolti

- Utente

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.

- La connessione a internet dev'essere attiva e funzionante.

Postcondizioni

- L'utente viene reindirizzato a una nuova pagina web corrispondente al link associato all'icona selezionata, aperta in una nuova finestra o scheda del browser.

Scenario principale

- L'utente accede all'interfaccia di *BuddyBot_G*
- Il sistema mostra una o più icone che fungono da link verso risorse esterne
- L'utente individua e clicca sull'icona desiderata
- L'utente viene reindirizzato su un'altra pagina web associata all'icona

User story associata

- «Come utente, voglio poter visualizzare e cliccare su icone che fungono da link verso risorse esterne in maniera tale da poter essere reindirizzato alla pagina web associata in una nuova finestra o scheda del browser.»

3.3.7.1) UC7.1, Visualizzazione link - icona Jira**Attori coinvolti**

- Utente

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- La connessione a internet dev'essere attiva e funzionante.

Postcondizioni

- La pagina web della documentazione di *Jira_G* associata all'icona è aperta in una nuova finestra o scheda del browser.

Scenario principale

- L'utente accede all'interfaccia di *BuddyBot_G*
- Il sistema mostra l'icona relativa alla documentazione di Jira
- L'utente posiziona il cursore su tale icona e la clicca
- L'utente viene reindirizzato alla pagina web di documentazione associata all'icona di *Jira_G*, aperta in una nuova finestra o scheda del browser.

User story associata

- «Come utente, voglio cliccare sull'icona di Jira presente nell'interfaccia del *chatbot_G* per aprire la documentazione in una nuova scheda o finestra, così da poter consultare facilmente le informazioni necessarie.»

3.3.7.2) UC7.2, Visualizzazione link - icona GitHub**Attori coinvolti**

- Utente

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- La connessione a internet dev'essere attiva e funzionante.

Postcondizioni

- La pagina web della documentazione di *GitHub_G* associata all'icona è aperta in una nuova finestra o scheda del browser.

Scenario principale

- L'utente accede all'interfaccia di *BuddyBot_G*
- Il sistema mostra l'icona relativa alla documentazione di GitHub
- L'utente posiziona il cursore su tale icona e la clicca
- L'utente viene reindirizzato alla pagina web di documentazione associata all'icona di *GitHub_G*, aperta in una nuova finestra o scheda del browser.

User story associata

- «Come utente, voglio cliccare sull'icona di GitHub presente nell'interfaccia del *chatbot_G* per aprire la documentazione in una nuova scheda o finestra, così da poter consultare facilmente le informazioni necessarie.»

3.3.7.3) UC7.3, Visualizzazione link - icona Confluence

Attori coinvolti

- Utente

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- La connessione a internet dev'essere attiva e funzionante.

Postcondizioni

- La pagina web della documentazione di *Confluence_G* associata all'icona è aperta in una nuova finestra o scheda del browser.

Scenario principale

- L'utente accede all'interfaccia di *BuddyBot_G*
- Il sistema mostra l'icona relativa alla documentazione di Confluence
- L'utente posiziona il cursore su tale icona e la clicca
- L'utente viene reindirizzato alla pagina web di documentazione associata all'icona di *Confluence_G*, aperta in una nuova finestra o scheda del browser.

User story associata

- «Come utente, voglio cliccare sull'icona di Confluence presente nell'interfaccia del *chatbot_G* per aprire la documentazione in una nuova scheda o finestra, così da poter consultare facilmente le informazioni necessarie.»

3.3.8) UC8, Visualizzazione bottone «Load More»

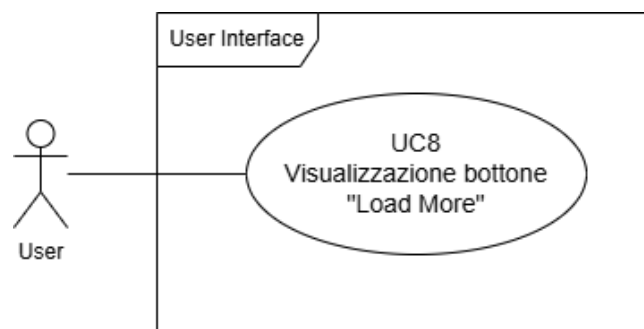


Figura 27: Diagramma UC8, Visualizzazione bottone Load More

Attori coinvolti

- *User_G*

Precondizioni

- L'interfaccia utente è pronta all'interazione con l'utente.
- L'utente deve avere già avuto delle interazioni pregresse con BuddyBot

Postcondizioni

- L'utente clicca il pulsante per caricare 10 nuovi messaggi che non erano stati ancora visualizzati nella chat

Scenario principale

- L'utente accede all'interfaccia di chat con BuddyBot.
- Vengono caricati e recuperati dei messaggi legati a passate interazioni con BuddyBot
- Nella parte superiore della chat, dopo aver trascorso tutti i messaggi già caricati, compare un pulsante di «Load More» per caricare altre interazioni passate
- L'utente clicca sul bottone e vengono caricati nuove 10 interazioni avute con BuddyBot seguendo l'ordine cronologico

User story associata

- «Come utente, voglio poter cliccare su un pulsante "Load More" per caricare messaggi più vecchi, in modo da poter consultare facilmente lo storico delle conversazioni passate.»

4) Requisiti

In questa sezione vengono esposti i requisiti esposti nel capitolato e individuati a seguito delle analisi effettuate dal gruppo e dai *casi d'uso (UC)_G* esaminati in precedenza. Per garantire la maggior chiarezza, i requisiti verranno identificati da un codice univoco con questa struttura:

R[Tipo] - [Identificativo]

Dove:

- **Tipo:** indica il tipo di requisito, che può essere funzionale, di qualità o di vincolo.

I tipi sono indicati da una sigla:

- **F** per i requisiti funzionali;
- **Q** per i requisiti di qualità;
- **V** per i requisiti di vincolo.

- **Identificativo:** rappresenta il numero progressivo del requisito;

Inoltre, in ogni tabella saranno indicate la descrizione del requisito in oggetto, la sua fonte (capitolato o caso d'uso) e la sua priorità.

Si ricorda che come indicato nelle Norme di Progetto, i requisiti vengono classificati con 3 livelli di priorità:

- **Obbligatorio:** requisito irrinunciabile per il committente;
- **Desiderabile:** requisito che porta valore aggiunto al prodotto ma non è strettamente necessario;
- **Opzionale:** requisito che può essere soddisfatto o meno senza che il prodotto risulti compromesso.

4.1) Requisiti Funzionali

I *Requisiti Funzionali_G* definiscono cosa il sistema deve fare, ovvero le sue funzionalità principali e i comportamenti attesi, concentrandosi quindi sulle operazioni che gli utenti/attori devono poter eseguire.

Si osservi che, come indicato sopra, i requisiti funzionali verranno identificati con i seguenti codici:

Codice	Descrizione	Fonti	Priorità
RF-001	L'utente deve accedere all'applicazione senza necessità di autenticazione	Capitolato	Obbligatorio
RF-002	Il sistema deve archiviare in modo persistente le domande degli utenti e le risposte generate	UC1	Obbligatorio
RF-003	L'utente deve poter visualizzare lo storico della chat in ordine cronologico inverso (dal più recente al più vecchio).	UC1	Obbligatorio
RF-004	L'utente deve visualizzare un messaggio informativo che spiega che non ci sono messaggi nello storico	UC1.1	Obbligatorio
RF-005	L'utente deve visualizzare un messaggio di errore se il sistema non riesce a recuperare lo storico	UC1.2	Obbligatorio

RF-006	L'utente deve visualizzare un messaggio di errore se la richiesta non è stata completata a causa di un timeout	UC1.3	Obbligatorio
RF-007	L'utente deve visualizzare un messaggio di errore se il backend non è disponibile	UC1.5	Obbligatorio
RF-008	L'utente deve visualizzare per ogni messaggio: il contenuto, la data e ora di invio	UC1.4, UC1.4.4, UC1.4.5	Obbligatorio
RF-009	L'utente deve visualizzare lo sfondo di un messaggio inviato da un utente di colore grigio	UC1.4.1	Obbligatorio
RF-010	L'utente deve visualizzare lo sfondo di un messaggio inviato da <i>Buddybot_G</i> di colore blu	UC1.4.2	Obbligatorio
RF-011	L'utente deve visualizzare per ogni messaggio inviato da <i>Buddybot_G</i> la data e l'ora dell'ultimo aggiornamento dei dati usati per generare la risposta	UC1.4.3	Obbligatorio
RF-012	L'utente deve poter scrivere una domanda in linguaggio naturale	UC2	Obbligatorio
RF-013	L'utente deve poter inviare la domanda scritta al sistema	UC3	Obbligatorio
RF-014	L'utente deve poter visualizzare la risposta generata da <i>Buddybot_G</i>	UC3	Obbligatorio
RF-015	L'utente deve essere informato se la domanda che ha posto non rientra nelle competenze specifiche del sistema tramite una risposta generata da <i>Buddybot_G</i>	UC3.4	Obbligatorio
RF-016	L'utente deve essere informato se i documenti richiesti nella domanda non sono disponibili all'interno del sistema tramite una risposta generata da <i>Buddybot_G</i>	UC3.5	Obbligatorio
RF-017	L'utente deve poter visualizzare un messaggio di errore se si è verificato un errore generico nella generazione della risposta da parte del <i>backend_G</i>	UC3.1	Obbligatorio
RF-018	L'utente deve poter visualizzare un messaggio di errore se la risposta non è stata generata perchè supera la lunghezza massima consentita	UC3.2	Obbligatorio
RF-019	L'utente deve poter visualizzare un messaggio di errore se la domanda supera la lunghezza massima consentita	UC3.3	Obbligatorio
RF-020	Il sistema deve generare una risposta appropriata alla domanda posta dell'utente	UC4	Obbligatorio
RF-021	Il sistema deve recuperare da GitHub le seguenti informazioni: Per ogni repository: • Nome della repository	UC4, Capitolato	Obbligatorio

	<ul style="list-style-type: none"> • Id della repository • Descrizione della repository • Data di creazione della repository • Ultima data di aggiornamento della repository • Linguaggio principale della repository <p>Per ogni commit:</p> <ul style="list-style-type: none"> • Hash del commit • Messaggio del commit • Data e ora dell'ultimo commit • Branch associato al commit • File modificati nel commit • Autore dell'ultimo commit • Nome della repository di appartenenza del commit • Nome del branch di appartenenza del commit <p>Per ogni pull request:</p> <ul style="list-style-type: none"> • Id della pull request • Titolo della pull request • Descrizione della pull request • Stato della pull request • Assegnatario della pull request • Reviewers della pull request • Commenti della pull request • File modificati nella pull request • Branch di origine della pull request • Branch di destinazione della pull request • Nome repository di appartenenza <p>Per ogni workflow:</p> <ul style="list-style-type: none"> • Id del workflow • Nome del workflow • Stato del workflow • Nome repository di appartenenza • Lista delle run per il workflow <p>Per ogni workflow run:</p> <ul style="list-style-type: none"> • Id della run • Stato della run • Durata in secondi della run • Link del log della run • Trigger della run • Id del workflow di appartenenza • Nome del workflow di appartenenza <p>Per ogni file:</p> <ul style="list-style-type: none"> • Path del file • SHA del file • Nome repository di appartenenza 		
--	---	--	--

	<ul style="list-style-type: none"> Nome branch di appartenenza Contenuto del file 		
RF-022	<p>Il sistema deve recuperare da Confluence le seguenti informazioni:</p> <ul style="list-style-type: none"> Id di una pagina Titolo di una pagina Stato di una pagina Autore di una pagina Owner di una pagina Spazio di una pagina Contenuto di una pagina 	UC4, Capitolato	Obbligatorio
RF-023	<p>Il sistema deve recuperare da Jira le seguenti informazioni:</p> <ul style="list-style-type: none"> Id di un ticket Titolo di un ticket Descrizione di un ticket Assegnatario di un ticket Stato di un ticket Sprint di appartenenza di un ticket Story point estimate di un ticket Creatore di un ticket Priorità Data di scadenza Ticket collegati Commenti del ticket <p>Per ogni commento del ticket:</p> <ul style="list-style-type: none"> Autore del commento Data di creazione Contenuto del commento 	UC4, Capitolato	Obbligatorio
RF-024	Il sistema deve informare l'utente in caso di errore durante la generazione della risposta	UC4.1	Obbligatorio
RF-025	Il sistema deve informare l'utente se la risposta supera la lunghezza massima consentita	UC4.2	Obbligatorio
RF-026	Il sistema deve fornire la data e l'ora dell'ultimo aggiornamento dei dati utilizzati	UC4	Obbligatorio
RF-027	Il sistema deve aggiornare i dati dei documenti provenienti da GitHub, Confluence e Jira ogni 24 ore	Capitolato	Obbligatorio
RF-028	Il sistema deve salvare i dati provenienti dalle fonti (Githbu, Jira, Confluence) in un database vettoriale	Verbale interno	Obbligatorio
RF-029	Il sistema deve convertire i dati provenienti dalle fonti (Githbu, Jira, Confluence) da forma testuale a forma vettoriale	Verbale interno	Obbligatorio
RF-030	L'utente deve poter modificare una domanda già inviata	UC5	Opzionale

RF-031	L'utente deve poter selezionare il tema chiaro o scuro per visualizzare l'interfaccia utente	UC6, UC6.1, UC6.2	Opzionale
RF-032	Il sistema deve visualizzare un'icona identificativa (cliccabile ed interattiva) per l'accesso a una risorsa esterna, aprendo la pagina web associata in una nuova finestra o scheda del browser.	UC7	Opzionale
RF-033	Il sistema deve visualizzare un'icona identificativa (cliccabile ed interattiva) per l'accesso al sito-documentazione di Jira	UC7.1	Opzionale
RF-034	Il sistema deve visualizzare un'icona identificativa (cliccabile ed interattiva) per l'accesso al sito-documentazione di GitHub	UC7.2	Opzionale
RF-035	Il sistema deve visualizzare un'icona identificativa (cliccabile ed interattiva) per l'accesso al sito-documentazione di Confluence	UC7.3	Opzionale
RF-036	Il sistema deve visualizzare un'animazione di caricamento circolare durante il recupero dello storico della chat	UC1.6	Opzionale
RF-037	Il sistema deve visualizzare un'animazione di caricamento composta da tre puntini, durante l'elaborazione della risposta da parte del backend	UC3.6	Opzionale
RF-038	Il sistema deve visualizzare un pulsante "Load More" nella parte superiore della chat, che consenta all'utente di caricare 10 messaggi precedenti non ancora visualizzati	UC8	Opzionale
RF-039	L'utente deve visualizzare il contenuto del messaggio in formato markdown	UC1.4.5	Desiderabile
RF-040	L'utente deve poter incollare nell'input di testo il contenuto copiato in precedenza	UC2	Desiderabile
RF-041	L'interfaccia utente deve scrollare verso il basso mostrando l'ultimo messaggio inviato ogni volta che l'utente invia un nuovo messaggio	Verbale interno	Opzionale

Tabella 1: Requisiti Funzionali

4.2) Requisiti di Qualità

I **Requisiti di Qualità** riguardano **come** il sistema deve funzionare, definendo quindi caratteristiche cruciali per garantire un'ottima «*User Experience*», soddisfacendo le sue esigenze.

Come i requisiti funzionali, i requisiti di qualità seguono la struttura definita al [Capitolo 4](#).

Codice	Descrizione	Fonti	Priorità
RQ-001	Occorre realizzare e consegnare un documento di analisi dei requisiti con un diagramma dei casi d'uso in formato UML.	Capitolato	Obbligatorio

RQ-002	Occorre realizzare e consegnare un documento denominato «Specifica Tecnica» che fornisce una visione dettagliata riguardante l'architettura del sistema, i design pattern utilizzati, le tecnologie adottate e le scelte progettuali effettuate.	Capitolato	Obbligatorio
RQ-003	Occorre che il codice sorgente sia presente in un singolo repository su Github accessibile all'azienda.	Capitolato	Obbligatorio
RQ-004	Occorre usufruire di un sistema di bug reporting per segnalare e tracciare eventuali errori o malfunzionamenti riscontrati nell'applicazione.	Capitolato	Obbligatorio
RQ-005	Occorre che venga creata una suite di test automatizzati per garantire il corretto funzionamento del codice	Capitolato	Obbligatorio
RQ-006	Occorre che sia assicurata una coverage di almeno 80% del codice, per garantire che la maggior parte delle funzionalità siano testate e prive di bug.	Capitolato	Obbligatorio

Tabella 2: Requisiti di Qualità

4.3) Requisiti di Vincolo

I **Requisiti di Vincolo_G** definiscono i limiti tecnici e/o progettuali entro i quali il sistema deve essere sviluppato.

Anche i requisiti di vincolo seguono la struttura del **Capitolo 4**

Codice	Descrizione	Fonti	Priorità
RV-001	Occorre che l'applicazione venga sviluppata suddividendola in frontend e backend	Capitolato	Obbligatorio
RV-002	Occorre che l'applicazione sia compatibile con la versione 135.0 di Firefox, che è l'ultima disponibile al momento della stesura di questo documento, e che quindi sia accessibile da ogni sistema operativo che supporta Firefox	Verbale interno	Obbligatorio
RV-003	Occorre che l'applicazione sia compatibile con la versione 134.0.6998.23 di Google Chrome, che è l'ultima disponibile al momento della stesura di questo documento, e che quindi sia accessibile da ogni sistema operativo che supporta Chrome	Verbale interno	Obbligatorio
RV-004	Occorre usare NestJS	Capitolato	Desiderabile
RV-005	Occorre usare Langchain	Capitolato	Desiderabile
RV-006	Occorre usare OpenAI	Capitolato	Desiderabile
RV-007	Occorre usare Angular	Capitolato	Desiderabile

Tabella 3: Requisiti di Vincolo

4.4) Tracciamento

Il **Tracciamento_G** è il processo che mira a collegare ogni requisito del progetto alle sue «fonti» di origine (capitolato, incontri con l'azienda, incontri interni) e alle sue implementazioni, come i **Casi d'Uso (UC)_G**.

Fonte	Requisito
Capitolato	RF-001
UC1	RF-002
UC1	RF-003
UC1.1	RF-004
UC1.2	RF-005
UC1.3	RF-006
UC1.5	RF-007
UC1.6	RF-036
UC1.4, UC1.4.4, UC1.4.5	RF-008
UC1.4.1	RF-009
UC1.4.2	RF-010
UC1.4.3	RF-011
UC1.4.5	RF-039
UC2	RF-012
UC2	RF-040
UC3	RF-013
UC3	RF-014
UC3.4	RF-015
UC3.5	RF-016
UC3.6	RF-037
UC3.1	RF-017
UC3.2	RF-018
UC3.3	RF-019
UC4	RF-020
UC4, Capitolato	RF-021
UC4, Capitolato	RF-022
UC4, Capitolato	RF-023
UC4.1	RF-024
UC4.2	RF-025
UC4	RF-026
Capitolato	RF-027
Verbale interno	RF-028
Verbale interno	RF-029
UC5	RF-030
UC6, UC6.1, UC6.2	RF-031
UC7	RF-032
UC7.1	RF-033
UC7.2	RF-034
UC7.3	RF-035

UC8	RF-038
Verbale interno	RF-041
Capitolato	RQ-001
Capitolato	RQ-002
Capitolato	RQ-003
Capitolato	RQ-004
Capitolato	RQ-005
Capitolato	RQ-006
Capitolato	RV-001
Capitolato	RV-002
Capitolato	RV-003
Capitolato	RV-004
Capitolato	RV-001
Capitolato	RV-002
Capitolato	RV-003
Capitolato	RV-004
Capitolato	RV-005
Capitolato	RV-006
Capitolato	RV-007

Tabella 4: Tracciamento

4.5) Riepilogo

Tipologia	Obbligatorio	Desiderabile	Opzionale	Totale
Funzionali	29	2	10	41
Qualità	6	0	0	6
Vincolo	3	4	0	7

Tabella 5: Riepilogo

Il totale dei Requisiti è: 56