



Analisi dei Requisiti

Versione: 1.1.0 23/11/2024

Redattori

Ion Cainareanu
Luca Parise
Marco Perazzolo
Malik Giafar Mohamed

Verifica

Ion Cainareanu
Luca Parise
Marco Perazzolo
Malik Giafar Mohamed
Stefano Baso
Maria Fuensanta Trigueros Hernandez

Approvazione

Luca Parise

Uso

Esterno

nextsoftpadova@gmail.com

Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica
1.1.0	05/04/2025	Malik Giafar Mohamed, Luca Parise, Marco Perazzolo	Modifiche generali agli use cases e aggiornamento del relativo tracciamento	
1.0.1	27/03/2025	Marco Perazzolo	Correzione di casi d'uso e requisiti post-incontro RTB	
1.0.0	08/03/2025	Ion Cainareanu	Aggiunti i termini del glossario e rilasciata la versione 1.0.0 del documento	Marco Perazzolo
0.6.3	07/03/2025	Marco Perazzolo	Affinamento degli extension points nei diagrammi UML	Stefano Baso
0.6.2	26/02/2025	Ion Cainareanu	Outline automatico per le tabelle	Malik Giafar Mohamed
0.6.1	26/02/2025	Malik Giafar Mohamed	Aggiornamento data di creazione documento	Ion Cainareanu
0.6.0	25/02/2025	Luca Parise, Ion Cainareanu, Marco Perazzolo	Modifiche post incontro con l'azienda	Malik Giafar Mohamed
0.5.0	18/02/2025	Luca Parise	Inserimento requisiti	Ion Cainareanu
0.4.0	11/02/2025	Marco Perazzolo	Inserimento dei diagrammi Use Case	Ion Cainareanu

Versione	Data	Autore	Descrizione	Verifica
0.3.1	06/02/2025	Marco Perazzolo	Finalizzazione Use Case testuali	Malik Giafar Mohamed
0.3.0	06/01/2025	Ion Cainareanu	Stesura iniziale degli Use Case	Marco Perazzolo, Luca Parise
0.2.0	30/12/2024	Ion Cainareanu	Stesura dell'Introduzione e Descrizione	Stefano Baso, Malik Giafar Mohamed
0.1.1	04/12/2024	Luca Parise	Aggiunta indice e creazione struttura tabella per use case	Malik Giafar Mohamed
0.1.0	23/11/2024	Malik Giafar Mohamed	Creazione Documento	Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Glossario	5
2	Descrizione	6
2.1	Obiettivi del prodotto	6
2.2	Funzionalità del prodotto	6
2.3	Utenti e caratteristiche	6
3	Use Case	7
3.1	Obiettivi	7
3.2	Attori	7
3.3	UC_1 - Importazione dei requisiti	7
3.4	UC_1.1 - Selezione del file	8
3.5	UC_1.2 - Visualizzazione errore file importazione	8
3.6	UC_2 - Importazione dei requisiti con tracciamento	10
3.7	UC_3 - Analisi dei requisiti e dell'implementazione	12
3.8	UC_3.1 - Visualizzazione errore tracciamento mancante	13
3.9	UC_3.2 - Visualizzazione errore di connessione	14
3.10	UC_3.3 - Visualizzazione avviso performance ridotte	14
3.11	UC_3.4 - Visualizzazione errore codice sorgente non disponibile	14
3.12	UC_4 - Esportazione del tracciamento	15

3.13	UC_4.1 - Visualizzazione errore di salvataggio	16
3.14	UC_4.2 - Visualizzazione errore tracciamento mancante	16
3.15	UC_5 - Esportazione del tracciamento e dei risultati	17
3.16	UC_5.1 - Visualizzazione errore risultati non disponibili	18
3.17	UC_6 - Visualizzazione elenco requisiti	19
3.18	UC_6.1 - Visualizzazione singolo requisito in elenco	20
3.19	UC_6.1.1 - Visualizzazione ID requisito	20
3.20	UC_6.1.2 - Visualizzazione icona ripetizione analisi	21
3.21	UC_7 - Visualizzazione dettaglio requisito	21
3.22	UC_7.1 - Visualizzazione testo requisito	22
3.23	UC_7.2 - Visualizzazione tracciamento requisito	23
3.24	UC_7.2.1 - Visualizzazione nome file	24
3.25	UC_7.2.2 - Visualizzazione riga inizio	24
3.26	UC_7.2.3 - Visualizzazione riga fine	24
3.27	UC_8 - Visualizzazione risultati requisito	25
3.28	UC_8.1 - Visualizzazione stato di conformità	27
3.29	UC_8.1.1 - Visualizzazione stato conforme	27
3.30	UC_8.1.2 - Visualizzazione stato non conforme	28
3.31	UC_8.2 - Visualizzazione valutazione requisito	28
3.32	UC_8.3 - Visualizzazione valutazione codice	28
3.33	UC_8.4 - Visualizzazione lista suggerimenti	29
3.34	UC_8.4.1 - Visualizzazione singolo suggerimento	29
3.35	UC_8.5 - Visualizzazione lista problemi	30
3.36	UC_8.5.1 - Visualizzazione singolo problema	31
3.37	UC_9 - Filtraggio dei requisiti	32
3.38	UC_10 - Analisi di un singolo requisito	33
3.39	UC_11 - Tracciamento dei requisiti nel codice	34
3.40	UC_12 - Configurazione dei path da ignorare	35
3.41	UC_12.1 - Visualizzazione errore path non valido	36
3.42	UC_13 - Configurazione del modello LLM	37
3.43	UC_14 - Configurazione dell'endpoint del server Ollama	38
3.44	UC_15 - Configurazione della soglia di conformità	39
3.45	UC_15.1 - Visualizzazione errore valore soglia non valido	40
4	Requisiti	41
4.1	Introduzione	41
4.2	Requisiti Funzionali	41
4.3	Requisiti di qualità	47
4.4	Requisiti di vincolo	48
4.5	Requisiti Prestazionali	48
4.6	Tracciamento dei Requisiti	48
4.7	Riepilogo	50

1 Introduzione

1.1 Scopo del documento

Lo scopo del presente documento è fornire una descrizione completa e dettagliata degli obiettivi, delle funzionalità e delle caratteristiche tecniche del progetto **Requirement Tracker - Visual Studio Code Plug-in**, con particolare attenzione all'utilizzo dell'*UML*^G per la modellazione dei *casi d'uso*^G. Il documento funge da riferimento per tutti gli *stakeholder*^G coinvolti, descrivendo il contesto operativo, i requisiti funzionali e non funzionali, nonché le linee guida tecnologiche necessarie per lo sviluppo del *plug-in*^G. I casi d'uso saranno descritti utilizzando una struttura standardizzata, che includerà il nominativo del caso, gli attori principali, le *precondizioni*^G, le *postcondizioni*^G, lo *scenario principale*^G e gli eventuali *scenari alternativi*^G o sottocasi. Questa struttura garantisce chiarezza e coerenza, facilitando la comprensione e la tracciabilità delle funzionalità principali del sistema. Il documento intende inoltre fornire una visione condivisa del progetto, ponendo le basi per una *pianificazione*^G e un'implementazione efficaci.

1.2 Scopo del prodotto

Lo scopo di **Requirement Tracker - Visual Studio Code Plug-in** è affrontare il problema della complessità nella gestione e nel *tracciamento dei requisiti*^G nei progetti software di grandi dimensioni. Nei *codebase*^G estesi, la verifica manuale della copertura e dell'implementazione dei requisiti è un processo lungo e soggetto a errori, spesso complicato dalla qualità insufficiente con cui i requisiti stessi vengono definiti. Questo può portare a malintesi e problemi durante l'implementazione, compromettendo l'allineamento tra specifiche e funzionalità sviluppate. Il plug-in mira a risolvere queste difficoltà automatizzando il tracciamento dei requisiti nel codice sorgente, migliorando la qualità della loro definizione e semplificando l'identificazione delle aree di mancata o errata implementazione. In particolare, offre strumenti per integrare requisiti tecnici derivati da manuali e datasheet di componenti hardware, fornendo analisi automatizzate e suggerimenti per rendere i requisiti più chiari, specifici e strutturati. Grazie a questo, sviluppatori potranno garantire una gestione più efficace dei requisiti, riducendo errori e aumentando la coerenza tra specifiche e implementazione.

1.3 Glossario

Per evitare ambiguità relative alle terminologie utilizzate è stato creato un documento denominato **Glossario_v1.0.0**. Questo documento comprende tutti i termini tecnici scelti dai membri del gruppo e utilizzati nei vari documenti con le relative definizioni. Tutti i termini inclusi in questo glossario vengono segnalati all'interno del documento con l'apice ^G accanto alla parola.

2 Descrizione

2.1 Obiettivi del prodotto

L'obiettivo del progetto è realizzare un plug-in per *Visual Studio Code*^G che consenta di tracciare e verificare l'implementazione dei requisiti di progetto, basandosi su analisi automatizzate del codice sorgente e sui requisiti tecnici espressi in documenti di riferimento, mediante l'utilizzo di tecnologie avanzate come modelli *LLM*^G di *AI*^G. Il plug-in sarà supportato da *API REST*^G che si interfacciano con *Ollama*^G, fornendo un'infrastruttura flessibile e scalabile per l'integrazione di modelli di AI e garantendo un'elaborazione efficiente e sicura delle analisi richieste.

2.2 Funzionalità del prodotto

Il plug-in sarà utilizzato dal *programmatore*^G per analizzare i requisiti implementati nel codice sorgente. Sia i requisiti che il codice saranno analizzati da vari modelli LLM reperibili attraverso la piattaforma di Ollama, grazie alle API REST che interagiscono con Ollama.

Le funzionalità implementate nell'applicazione includono:

- Importazione del file dei requisiti in formato *CSV*^G
- Richiesta di *analisi dei requisiti*^G tramite un modello LLM;
- Valutazione qualitativa dei requisiti;
- Visualizzazione grafica dei risultati dell'analisi;
- Filtraggio dei risultati dell'analisi;
- Possibilità di eseguire l'analisi su un *requisito*^G specifico;
- Esportazione dei risultati dell'analisi in formato CSV;
- Ricerca dell'implementazione dei requisiti nel codice sorgente;
- *Analisi semantica*^G dei requisiti e del codice sorgente;
- Suggerimenti per migliorare la qualità dei requisiti e del codice;
- Possibilità di modificare il modello LLM che analizza i requisiti ed il codice;
- Possibilità di modificare l'*endpoint*^G di collegamento al server Ollama;
- Possibilità di modificare la soglia di accettazione relativa alla qualità del codice.

2.3 Utenti e caratteristiche

In seguito a un incontro con il proponente, è stato discusso come il plug-in possa essere utilizzato principalmente da un utente che ricopre il ruolo di programmatore. Di conseguenza, si è deciso di focalizzare le funzionalità del plug-in per rispondere alle esigenze di questa categoria di utenti. È stato inoltre specificato che non devono essere fatte

assunzioni sulle competenze tecniche dell'utente riguardo all'uso di *Visual Studio Code*^G. Pertanto, il plug-in deve essere progettato per essere il più intuitivo possibile, con un processo di installazione semplice e accessibile.

3 Use Case

3.1 Obiettivi

Questa sezione si propone di identificare e descrivere i casi d'uso derivati dall'analisi del *capitolato*^G d'appalto selezionato dal gruppo. In particolare, vengono definiti gli attori principali e le funzionalità ad essi associate.

3.2 Attori

L'applicazione è progettata con un unico *attore*^G, il **Programmatore**, esso rappresenta un utente che utilizza il plug-in *Requirement Tracker - Visual Studio Code Plug-in* per importare, analizzare e tracciare l'implementazione dei requisiti software all'interno del codice sorgente di un progetto.

3.3 UC_1 - Importazione dei requisiti

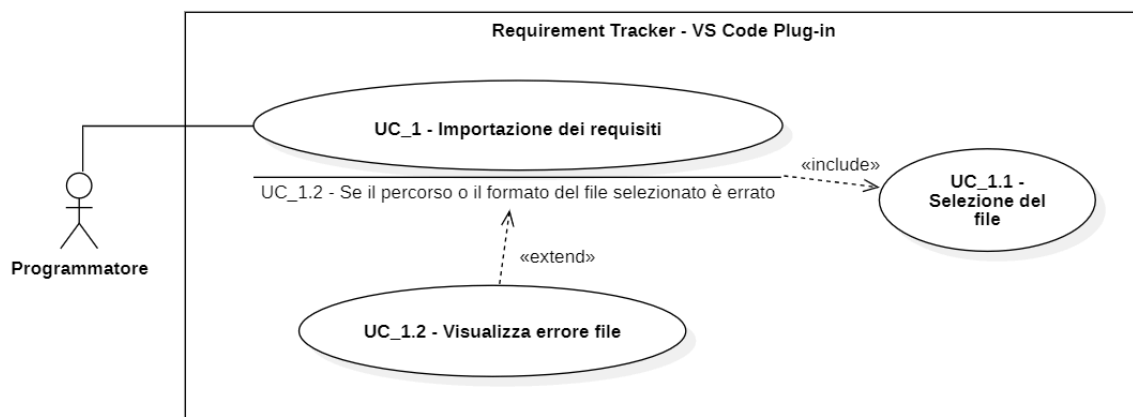


Figure 1: UC_1 - Importazione dei requisiti da file CSV

Attore primario: Programmatore.

Precondizioni:

- L'utente ha aperto un progetto in Visual Studio Code.

- Il file dei requisiti è in formato CSV valido.
- Il plug-in è installato e attivo in Visual Studio Code.

Postcondizioni:

- I requisiti vengono importati e sono visualizzabili nel sistema.

Scenario principale:

1. L'utente seleziona l'opzione "Importa requisiti".
2. Il sistema apre un *file explorer*^G.
3. L'utente seleziona il file CSV da importare [UC_1.1]
4. Il sistema verifica la validità del file e importa i dati (ID e testo di ogni requisito).
5. I requisiti importati vengono mostrati in una vista strutturata [UC_6].

Estensioni:

- **UC_1.2 - Visualizzazione errore file** : Se il file non rispetta il formato previsto o risulta malformato, il sistema notifica l'errore all'utente e richiede di selezionare un file corretto.

3.4 UC_1.1 - Selezione del file

Attore primario: Programmatore.

Precondizioni:

- Il file explorer è stato aperto dal sistema.

Postcondizioni:

- Il file CSV scelto dall'utente viene registrato per l'importazione.

Scenario principale:

1. Il sistema apre il file explorer.
2. L'utente naviga tra le cartelle e individua il file CSV desiderato.
3. L'utente seleziona il file CSV.
4. Il sistema registra la scelta e procede con l'importazione scelta [UC_1] oppure [UC_2].

3.5 UC_1.2 - Visualizzazione errore file importazione

Attore primario: Programmatore.

Precondizioni:

- L'utente ha selezionato un file CSV da importare [UC_1.1].

Postcondizioni:

- L'utente viene informato che il file non è valido.

Scenario principale:

1. Il sistema verifica il file e rileva che è malformato o non valido (per esempio: mancano le colonne ID e descrizione).
2. L'importazione del file fallisce
3. Il sistema mostra un messaggio di errore esplicativo e richiede di selezionare un file valido.

3.6 UC_2 - Importazione dei requisiti con tracciamento

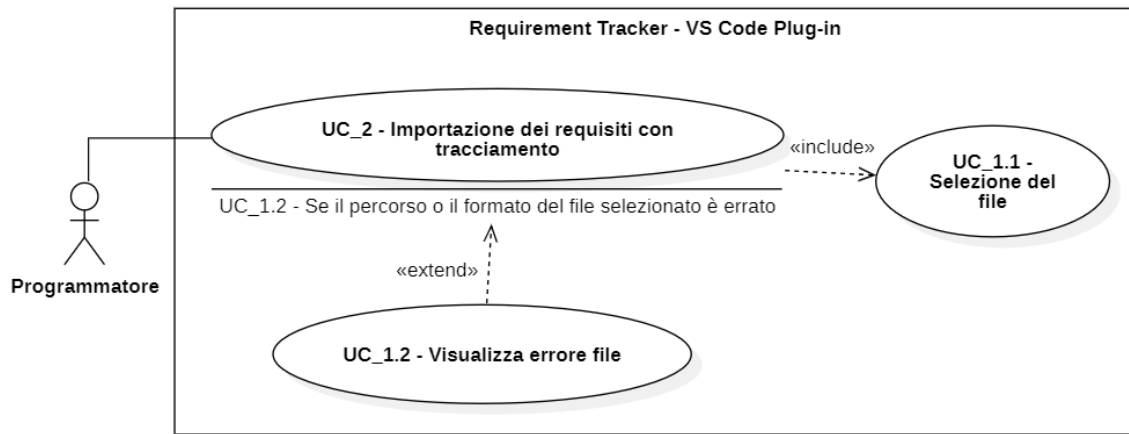


Figure 2: UC_2 - Importazione dei requisiti e del tracciamento da file CSV

Attore primario: Programmatore.

Precondizioni:

- L'utente ha aperto un progetto in Visual Studio Code.
- Il file dei requisiti è in formato CSV valido, e contiene le informazioni di tracciamento.
- Il plug-in è installato e attivo in Visual Studio Code.

Postcondizioni:

- I requisiti e le relative informazioni di tracciamento sono importate e visualizzabili nel sistema.

Scenario principale:

1. L'utente seleziona l'opzione "Importa requisiti con tracciamento".
2. Il sistema apre un file explorer.
3. L'utente seleziona il file CSV da importare [UC_1.1]
4. Il sistema verifica la validità del file e importa i dati (ID, testo, file, intervallo righe di ogni requisito).
5. I requisiti importati vengono mostrati in una vista strutturata [UC_6].

Estensioni:

- **UC_1.2 - Visualizzazione errore file** : Se il file non rispetta il formato previsto o risulta malformato, il sistema notifica l'errore all'utente e richiede di selezionare un file corretto.

3.7 UC_3 - Analisi dei requisiti e dell'implementazione

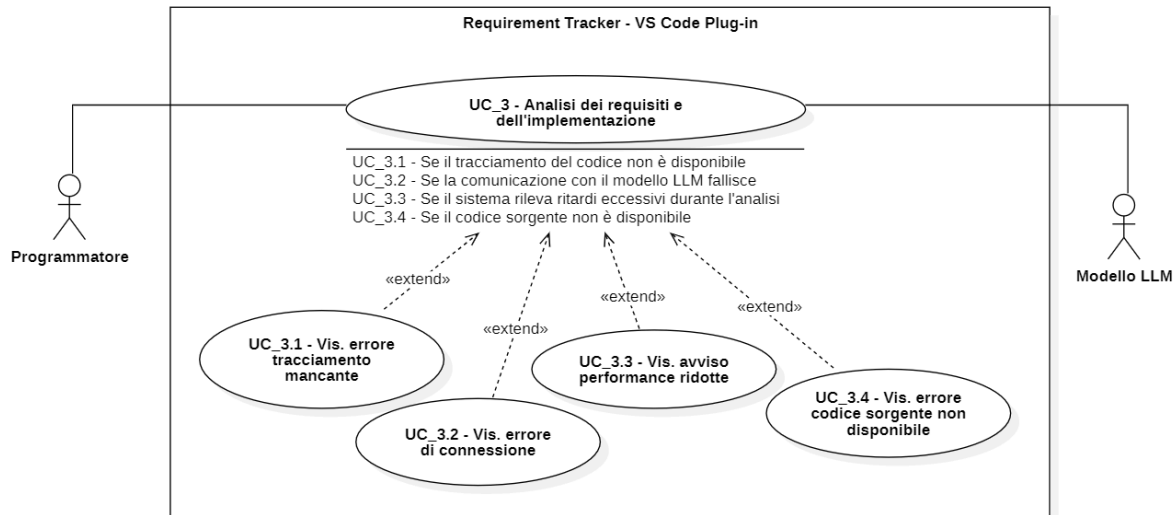


Figure 3: UC_3 - Analisi dei requisiti e della loro implementazione

Attore primario: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- I requisiti sono stati importati [UC_1] oppure [UC_2].
- Il tracciamento dei requisiti nel codice è disponibile, ottenuto direttamente da [UC_2] oppure dopo [UC_11].
- La connessione con le API REST di Ollama è attiva e disponibile.

Postcondizioni:

- Il sistema fornisce una valutazione complessiva per ciascun requisito, integrando l'analisi semantica del testo e la verifica dell'implementazione nel codice.

Scenario principale:

1. L'utente seleziona "Analisi requisiti".
2. Il sistema verifica che siano disponibili sia i requisiti che il relativo tracciamento (da [UC_2] o [UC_11]).
3. Il sistema estrae il testo di ciascun requisito e lo invia al modello LLM per l'analisi semantica.

4. Il modello LLM restituisce una valutazione del requisito e dei suggerimenti per migliorarne la chiarezza e la completezza.
5. Il sistema raccoglie le informazioni di tracciamento per ciascun requisito.
6. Il sistema estrae la porzione di codice di ogni requisito, identificata dalle informazioni di tracciamento.
7. Il sistema invia il testo del requisito e la relativa porzione di codice al modello LLM.
8. Il modello confronta il comportamento del codice con quanto richiesto dal requisito e restituisce una valutazione di conformità, suggerimenti per migliorarne l'implementazione e un elenco di problemi o anomalie nel codice.
9. Il sistema registra i dati e li rende disponibili per la visualizzazione ([UC_8]).

Estensioni:

- **UC_3.1 - Visualizzazione errore tracciamento mancante:** Se il *mapping*^G del codice non è disponibile, il sistema visualizza un messaggio d'errore specifico.
- **UC_3.2 - Visualizzazione errore di connessione:** Se la comunicazione con il modello LLM fallisce (es. timeout o connessione interrotta), il sistema informa l'utente e consente di riprovare.
- **UC_3.3 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.
- **UC_3.4 - Visualizzazione errore codice sorgente non disponibile:** Se il progetto non contiene il file sorgente o non è configurato correttamente, il sistema mostra un messaggio di errore.

3.8 UC_3.1 - Visualizzazione errore tracciamento mancante

Attore primario: Programmatore.

Precondizioni:

- Durante l'analisi [UC_3], il sistema rileva che il mapping (tracciamento del codice) non è disponibile.

Postcondizioni:

- Il sistema visualizza un messaggio d'errore che informa l'utente dell'assenza del tracciamento.

Scenario principale:

1. Durante l'esecuzione di [UC_3], il sistema verifica la presenza del mapping.

2. Se il mapping risulta mancante, il sistema mostra un messaggio d'errore specifico.

3.9 UC_3.2 - Visualizzazione errore di connessione

Attore primario: Programmatore.

Precondizioni:

- Durante l'analisi [UC_3], la comunicazione con il modello LLM fallisce (es. timeout, connessione interrotta).

Postcondizioni:

- Il sistema informa l'utente dell'errore di connessione.

Scenario principale:

1. Durante l'invio dei dati al modello LLM, il sistema rileva un problema di connessione.
2. Il sistema visualizza un messaggio d'errore dettagliato e consente all'utente di riprovare.

3.10 UC_3.3 - Visualizzazione avviso performance ridotte

Attore primario: Programmatore.

Precondizioni:

- Durante l'analisi [UC_3], il sistema rileva tempi di risposta eccessivi dal modello LLM.

Postcondizioni:

- Il sistema mostra un avviso che informa l'utente delle prestazioni ridotte.

Scenario principale:

1. Il sistema monitora il tempo di risposta del modello LLM.
2. Se il tempo supera una soglia prestabilita, il sistema visualizza un avviso informativo.

3.11 UC_3.4 - Visualizzazione errore codice sorgente non disponibile

Attore primario: Programmatore.

Precondizioni:

- Il progetto non contiene file sorgente o non è configurato correttamente.

Postcondizioni:

- Il sistema informa l'utente della mancanza di codice.

Scenario principale:

1. L'utente avvia la verifica dell'implementazione dei requisiti [UC_3].
2. Il sistema verifica la presenza del codice sorgente.
3. Il sistema rileva che non è configurato correttamente o non è presente.
4. Il sistema mostra un messaggio di errore.

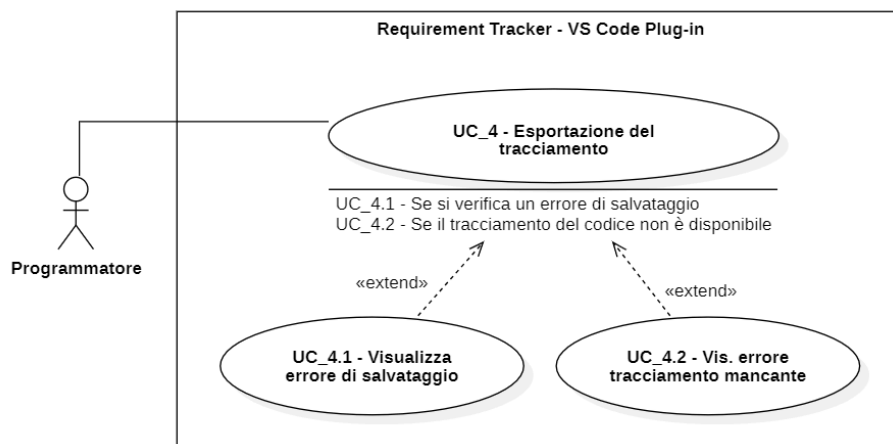
3.12 UC_4 - Esportazione del tracciamento

Figure 4: UC_4 - Esportazione del tracciamento su file CSV

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono disponibili, importati da [UC_1] oppure [UC_2].
- È disponibile il tracciamento dei requisiti nel codice (ottenuto da [UC_2] oppure dopo [UC_11]).

Postcondizioni:

- I requisiti ed il tracciamento vengono esportati correttamente in un file CSV nel percorso specificato dall'utente.

Scenario principale:

1. L'utente seleziona la voce "Esporta tracciamento" nell'interfaccia di esportazione.
2. Il sistema apre un file explorer per la scelta del percorso di salvataggio.

3. L'utente conferma il percorso e il nome del file.
4. Il sistema salva un file CSV contenente esclusivamente i requisiti ed i relativi dati di tracciamento.

Estensioni:

- **UC_4.1 - Visualizzazione errore di salvataggio:** Se il salvataggio fallisce (es. permessi insufficienti o spazio esaurito), il sistema notifica l'errore all'utente e permette di riprovare.
- **UC_4.2 - Visualizzazione errore tracciamento mancante:** Se il mapping del codice non è disponibile, il sistema visualizza un messaggio d'errore specifico.

3.13 UC_4.1 - Visualizzazione errore di salvataggio

Attore primario: Programmatore.

Precondizioni:

- L'utente tenta di esportare i risultati, ma il salvataggio fallisce.

Postcondizioni:

- Il sistema informa l'utente dell'errore e consente di riprovare o di selezionare un percorso alternativo.

Scenario principale:

1. L'utente seleziona "Esporta risultati".
2. Il sistema tenta di salvare il file CSV.
3. Si verifica un errore durante il salvataggio.
4. Il sistema mostra un messaggio d'errore e consente di riprovare.

3.14 UC_4.2 - Visualizzazione errore tracciamento mancante

Attore primario: Programmatore.

Precondizioni:

- Durante l'esportazione [UC_4], il sistema rileva che il mapping (tracciamento del codice) non è disponibile.

Postcondizioni:

- Il sistema visualizza un messaggio d'errore che informa l'utente dell'assenza del tracciamento.

Scenario principale:

1. Durante l'esecuzione di [UC_4], il sistema verifica la presenza del mapping.
2. Se il mapping risulta mancante, il sistema mostra un messaggio d'errore specifico.

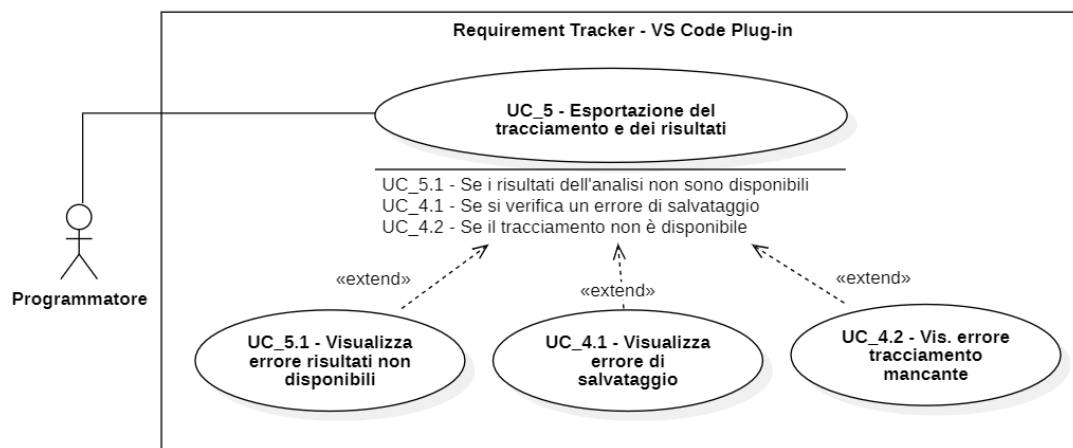
3.15 UC_5 - Esportazione del tracciamento e dei risultati

Figure 5: UC_5 - Esportazione del tracciamento e dei risultati su file CSV

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono disponibili, importati da [UC_1] oppure [UC_2].
- Il tracciamento dei requisiti è disponibile (ottenuto da [UC_2] o da [UC_11]).
- I risultati dell'analisi sono stati generati [UC_3].

Postcondizioni:

- Il sistema esporta in un file CSV tutti i dati: requisiti, tracciamento e risultati dell'analisi.

Scenario principale:

1. L'utente seleziona la voce "Esporta tracciamento e risultati".
2. Il sistema apre un file explorer per la scelta del percorso di salvataggio.
3. L'utente conferma il percorso e il nome del file.

4. Il sistema salva un file CSV contenente i dati relativi ai requisiti, al mapping e ai risultati.

Estensioni:

- **UC_5.1 - Visualizzazione errore risultati non disponibili:** Se i risultati dell'analisi non sono disponibili, il sistema mostra un messaggio d'errore specifico.
- **UC_4.1 - Visualizzazione errore di salvataggio:** Se il salvataggio fallisce (es. permessi insufficienti o spazio esaurito), il sistema notifica l'errore all'utente e permette di riprovare.
- **UC_4.2 - Visualizzazione errore tracciamento mancante:** Se il mapping del codice non è disponibile, il sistema visualizza un messaggio d'errore specifico.

3.16 UC_5.1 - Visualizzazione errore risultati non disponibili

Attore primario: Programmatore.

Precondizioni:

- Durante l'esportazione del tracciamento e dei risultati [UC_5], il sistema rileva che i risultati non sono disponibili.

Postcondizioni:

- Il sistema visualizza un messaggio d'errore che informa l'utente dell'assenza dei risultati.

Scenario principale:

1. Durante l'esecuzione di [UC_5], il sistema verifica la presenza dei risultati dell'analisi.
2. Se i risultati sono mancanti, il sistema mostra un messaggio d'errore specifico.

3.17 UC_6 - Visualizzazione elenco requisiti

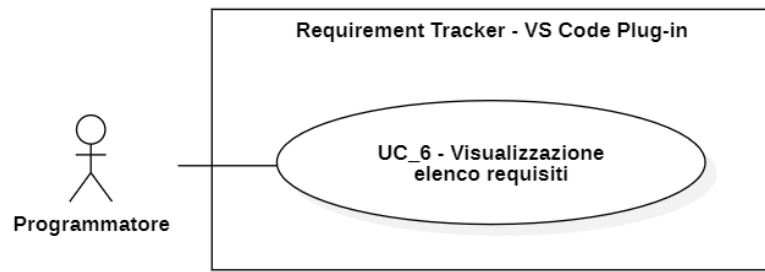


Figure 6: UC_6 - Visualizzazione dell'elenco dei requisiti

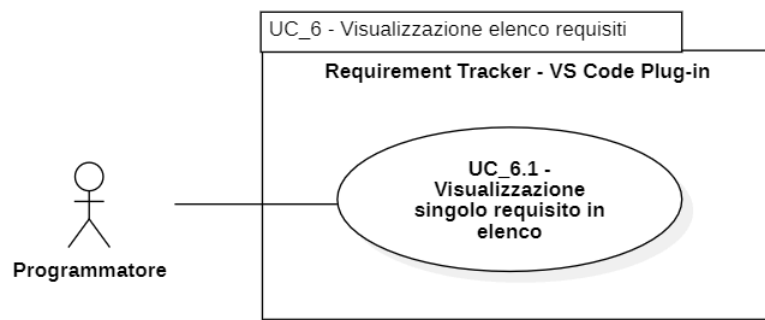


Figure 7: UC_6 - Dettaglio della visualizzazione elenco requisiti

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono stati importati (da [UC_1] oppure [UC_2]) e sono disponibili per la visualizzazione.

Postcondizioni:

- I requisiti vengono visualizzati in un elenco, in cui ciascun requisito è rappresentato da un ID.

Scenario principale:

1. Il sistema visualizza una lista dei requisiti.
2. L'utente può selezionare un requisito per visualizzarne il dettaglio [UC_7].

3.18 UC_6.1 - Visualizzazione singolo requisito in elenco

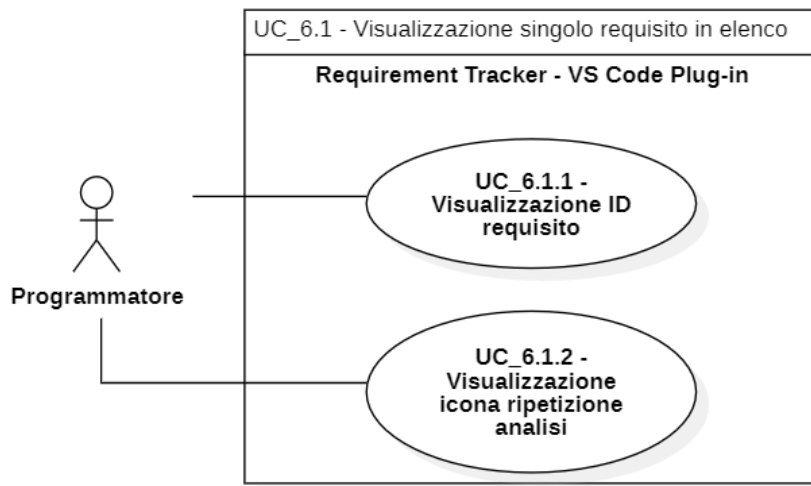


Figure 8: UC_6.1 - Visualizzazione singolo requisito in elenco

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono stati importati (da [UC_1] oppure [UC_2]) e sono disponibili per la visualizzazione.

Postcondizioni:

- Il sistema mostra il proprio identificativo (ID) che, se premuto, permette la visualizzazione in dettaglio di tutte le informazioni relative al requisito [UC_7], e un'icona che permette la ripetizione della funzione di analisi su quel singolo requisito.

Scenario principale:

1. Il sistema visualizza il requisito nell'elenco, mostrandone l'identificativo (ID) e l'icona di ripetizione analisi.

3.19 UC_6.1.1 - Visualizzazione ID requisito

Attore primario: Programmatore.

Precondizioni:

- Il requisito selezionato è stato importato (da [UC_1] o [UC_2]) ed è visualizzabile in elenco [UC_6.1].

Postcondizioni:

- Il sistema mostra il campo “ID” del requisito.

Scenario principale:

1. Il sistema visualizza l'identificativo univoco del requisito.

3.20 UC_6.1.2 - Visualizzazione icona ripetizione analisi

Attore primario: Programmatore.

Precondizioni:

- Il requisito selezionato è stato importato (da [UC_1] o [UC_2]) ed è visualizzabile in elenco [UC_6.1].

Postcondizioni:

- Il sistema mostra l'icona di una freccia, che permette di ripetere l'analisi del requisito [UC_10].

Scenario principale:

1. Il sistema visualizza l'icona di ripetizione analisi requisito.

3.21 UC_7 - Visualizzazione dettaglio requisito

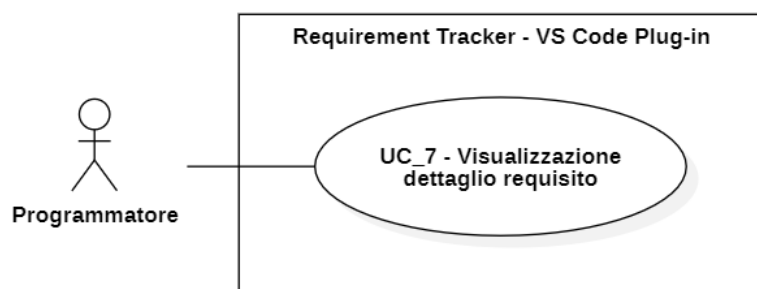


Figure 9: UC_7 - Visualizzazione dettaglio di un requisito

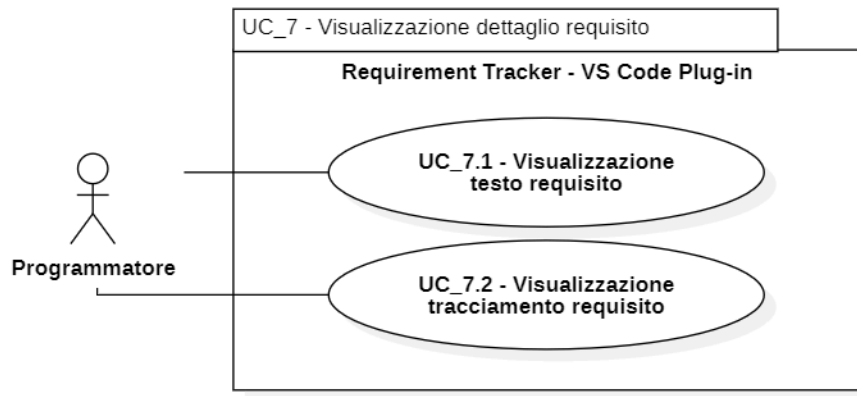


Figure 10: UC_7 - Sottocaso d'uso della visualizzazione in dettaglio di un requisito

Attore primario: Programmatore.

Precondizioni:

- L'utente ha premuto sul requisito in elenco, visualizzato in [UC_6.1].

Postcondizioni:

- Il sistema mostra il dettaglio completo del requisito, includendo:
 - Testo descrittivo del requisito [UC_7.1]
 - (Opzionale) Se disponibili, le informazioni di tracciamento [UC_7.2], quali file sorgente e intervallo di righe.

Scenario principale:

1. L'utente preme sul requisito in elenco [UC_6.1].
2. Il sistema espande il sottomenu di dettaglio, visualizzando il testo descrittivo del requisito [UC_7.1].
3. Se sono disponibili dati di tracciamento, il sistema visualizza anche il menu delle informazioni di tracciamento [UC_7.2].

3.22 UC_7.1 - Visualizzazione testo requisito

Attore primario: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_7].

Postcondizioni:

- Il sistema mostra il campo testuale descrittivo del requisito.

Scenario principale:

1. Il sistema visualizza il testo descrittivo del requisito.

3.23 UC_7.2 - Visualizzazione tracciamento requisito

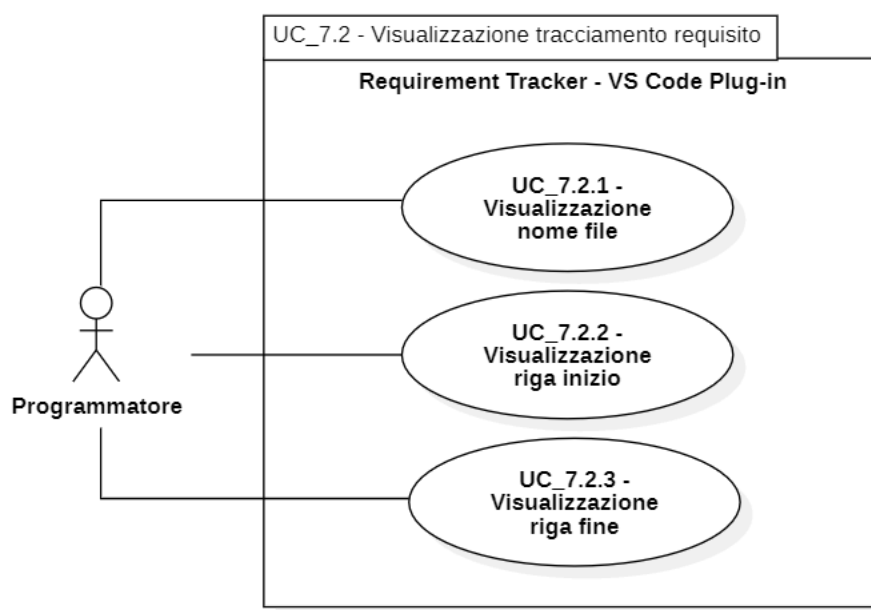


Figure 11: UC_7.2 - Visualizzazione del tracciamento di un requisito

Attore primario: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_7].
- Il requisito selezionato dispone di informazioni di tracciamento, importate da [UC_2] oppure generate da [UC_11].

Postcondizioni:

- Il sistema visualizza una menu di tracciamento espandibile che comprende i dettagli del tracciamento, quali file sorgente e intervallo di righe.

Scenario principale:

1. Nella vista di dettaglio del requisito [UC_7], il sistema verifica la presenza di dati di tracciamento.
2. Se presenti, il sistema espande la sezione “Tracciamento” mostrando i dettagli attraverso i sottocasi [UC_7.2.1], [UC_7.2.2] e [UC_7.2.3].

3.24 UC_7.2.1 - Visualizzazione nome file

Attore primario: Programmatore.

Precondizioni:

- La sezione “Tracciamento” del requisito è disponibile ed espansa [UC_7.2].

Postcondizioni:

- Il sistema mostra il campo “Nome File” relativo al file che contiene il codice relativo all’implementazione del requisito.

Scenario principale:

1. Il sistema visualizza il nome del file associato al tracciamento del requisito.

3.25 UC_7.2.2 - Visualizzazione riga inizio

Attore primario: Programmatore.

Precondizioni:

- La sezione “Tracciamento” del requisito è disponibile ed espansa [UC_7.2].

Postcondizioni:

- Il sistema mostra il campo “Riga Inizio” dell’intervallo di codice nel file indicato ([UC_7.2.1]) relativo all’implementazione del requisito.

Scenario principale:

1. Il sistema visualizza la riga di inizio dell’intervallo di tracciamento del requisito.

3.26 UC_7.2.3 - Visualizzazione riga fine

Attore primario: Programmatore.

Precondizioni:

- La sezione “Tracciamento” del requisito è disponibile ed espansa [UC_7.2].

Postcondizioni:

- Il sistema mostra il campo “Riga Fine” dell’intervallo di codice nel file indicato [UC_7.2.1] relativo all’implementazione del requisito.

Scenario principale:

1. Il sistema visualizza la riga di fine dell’intervallo di tracciamento del requisito.

3.27 UC_8 - Visualizzazione risultati requisito

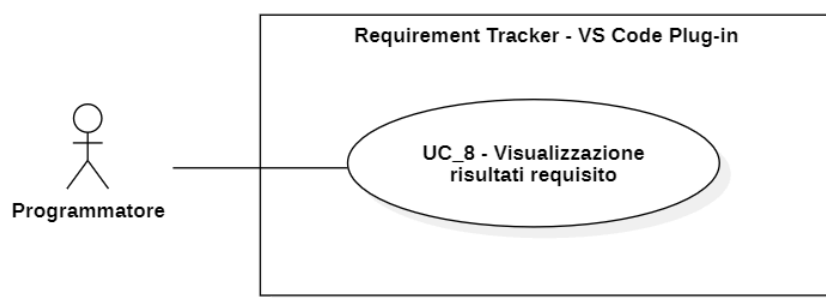


Figure 12: UC_8 - Visualizzazione dei risultati di un requisito

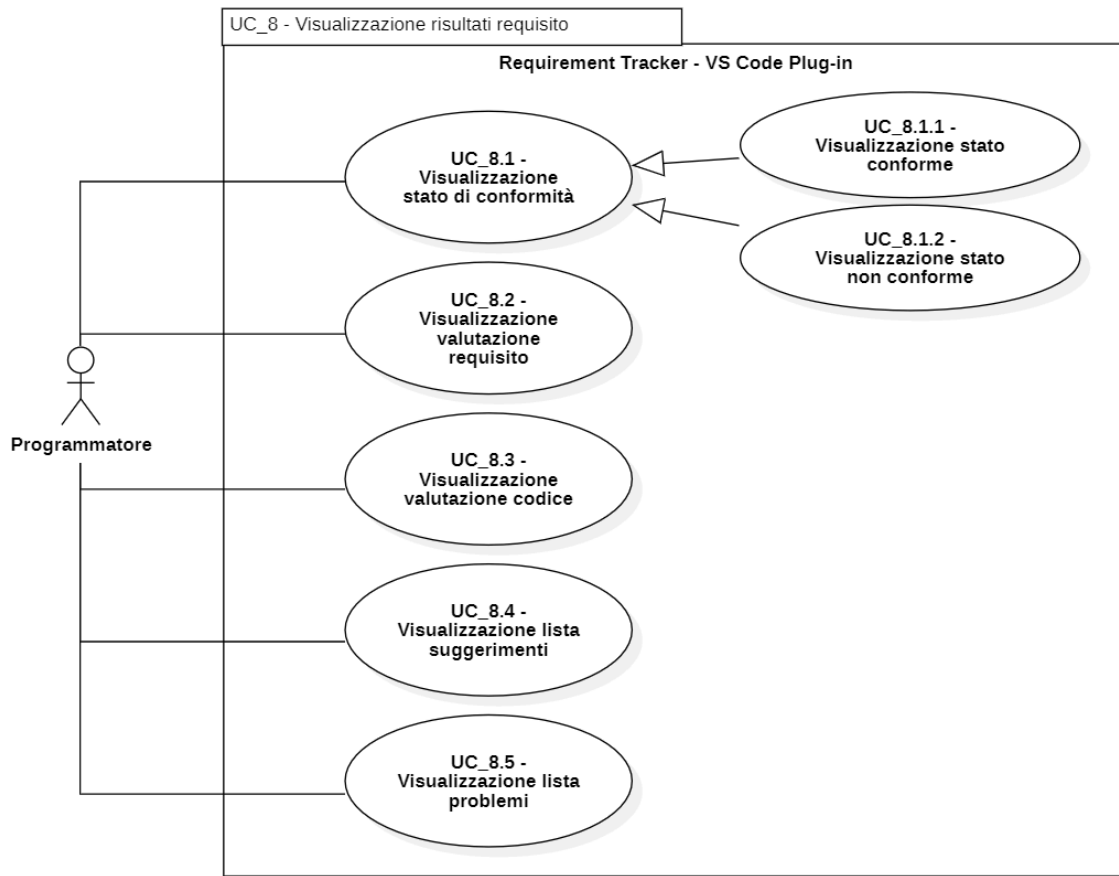


Figure 13: UC_8 - Dettaglio della visualizzazione dei risultati di un requisito

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_3].
- La sezione di visualizzazione in dettaglio del requisito selezionato è stata espansa [UC_7].

Postcondizioni:

- I risultati dell'analisi sono integrati nel menu di dettaglio requisito [UC_7] e sono visualizzabili.

Scenario principale:

1. L'utente seleziona la voce relativa alla visualizzazione del risultato.

2. Il sistema espande la visualizzazione dei risultati che mostra:

- [UC_8.1] Stato di conformità (*passed/not passed*).
- [UC_8.2] Valutazione del requisito in centesimi (0-100).
- [UC_8.3] Valutazione dell'aderenza del codice in centesimi (0-100).
- [UC_8.4] Lista dei suggerimenti generati.
- [UC_8.5] Lista dei problemi riscontrati.

3.28 UC_8.1 - Visualizzazione stato di conformità

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_3].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato espanso [UC_8].

Postcondizioni:

- Viene visualizzato lo stato "*passed*" se il requisito è corretto semanticamente e la sua implementazione nel codice soddisfa il requisito, "*not passed*" altrimenti.

Scenario principale:

1. Il sistema mostra, per ogni requisito, lo stato di conformità basato sui punteggi ottenuti:
 - Se il risultato è *passed*, visualizza lo stato conforme [UC_8.1.1]
 - Se il risultato è *not-passed*, visualizza lo stato non conforme [UC_8.1.2]

3.29 UC_8.1.1 - Visualizzazione stato conforme

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_3].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato espanso [UC_8].
- Il requisito e il codice sono stati valutati come conformi.

Postcondizioni:

- Il sistema mostra lo stato di conformità del requisito.

Scenario principale:

1. Il sistema visualizza lo stato di conformità come “*passed*”.

3.30 UC_8.1.2 - Visualizzazione stato non conforme

Attore primario: Programmatore.

Precondizioni:

- I risultati dell’analisi sono stati generati [UC_3].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato espanso [UC_8].
- Il requisito e il codice sono stati valutati come non conformi.

Postcondizioni:

- Il sistema mostra lo stato di non conformità del requisito.

Scenario principale:

1. Il sistema visualizza lo stato di conformità come “*not passed*”.

3.31 UC_8.2 - Visualizzazione valutazione requisito

Attore primario: Programmatore.

Precondizioni:

- I risultati dell’analisi sono stati generati [UC_3].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_8].

Postcondizioni:

- Il sistema mostra il punteggio numerico della valutazione semantica del requisito selezionato, espresso in centesimi.

Scenario principale:

1. Il sistema visualizza il punteggio relativo alla correttezza semantica e logica del requisito.

3.32 UC_8.3 - Visualizzazione valutazione codice

Attore primario: Programmatore.

Precondizioni:

- I risultati dell’analisi sono stati generati [UC_3].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_8].

Postcondizioni:

- Il sistema mostra il punteggio numerico della valutazione dell'implementazione del codice rispetto al requisito, espresso in centesimi.

Scenario principale:

1. Il sistema visualizza il punteggio relativo all'aderenza del codice al requisito.

3.33 UC_8.4 - Visualizzazione lista suggerimenti

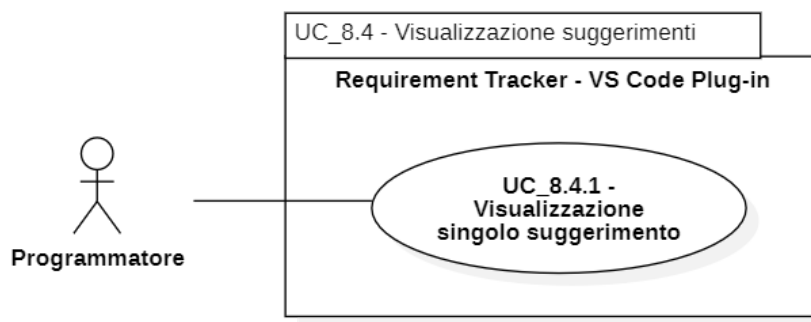


Figure 14: UC_8.4 - Visualizzazione lista dei suggerimenti

Attore primario: Programmatore.

Precondizioni:

- I suggerimenti relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_3].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_8].

Postcondizioni:

- Il sistema mostra un elenco strutturato dei suggerimenti relativi al requisito e al codice.

Scenario principale:

1. L'utente seleziona la voce relativa alla visualizzazione dei suggerimenti ("Suggestions").
2. Il sistema raccoglie i suggerimenti generati e li visualizza in forma di elenco.

3.34 UC_8.4.1 - Visualizzazione singolo suggerimento

Attore primario: Programmatore.

Precondizioni:

- I suggerimenti relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_3].
- Il menu ad albero relativo alla visualizzazione dei suggerimenti è stato aperto [UC_8.4].

Postcondizioni:

- Il sistema mostra il singolo suggerimento.

Scenario principale:

1. Il sistema visualizza il singolo suggerimento, che può essere un suggerimento di miglioramento del requisito o del codice.

3.35 UC_8.5 - Visualizzazione lista problemi

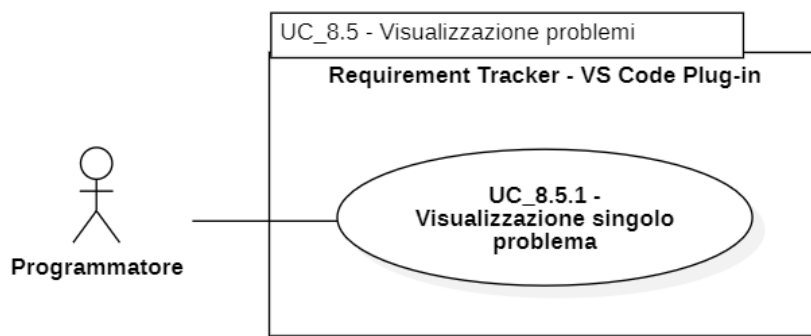


Figure 15: UC_8.5- Visualizzazione lista dei problemi

Attore primario: Programmatore.

Precondizioni:

- I problemi relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_3].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_8].

Postcondizioni:

- Il sistema mostra un elenco strutturato dei problemi relativi al requisito e al codice.

Scenario principale:

1. L'utente seleziona la voce relativa alla visualizzazione dei problemi ("Issues").
2. Il sistema raccoglie i problemi generati e li visualizza in forma di elenco.

3.36 UC_8.5.1 - Visualizzazione singolo problema

Attore primario: Programmatore.

Precondizioni:

- I problemi relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_3].
- Il menu ad albero relativo alla visualizzazione dei problemi è stato aperto [UC_8.5].

Postcondizioni:

- Il sistema mostra il singolo problema.

Scenario principale:

1. Il sistema visualizza il singolo problema, che può essere un problema di implementazione o di coerenza del requisito.

3.37 UC_9 - Filtraggio dei requisiti

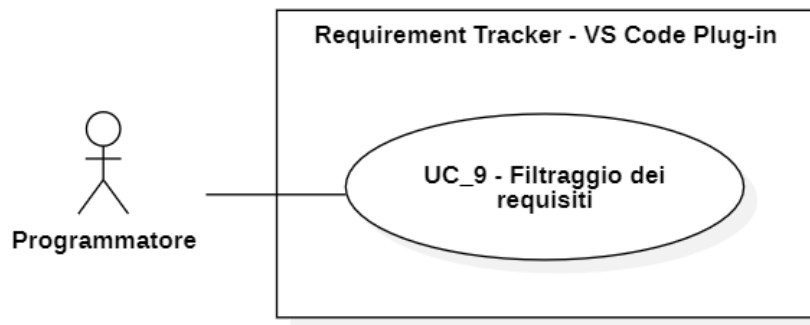


Figure 16: UC9 - Filtraggio dei requisiti

Attore primario: Programmatore.

Precondizioni:

- I requisiti importati sono visualizzati [UC_6].

Postcondizioni:

- I risultati vengono filtrati in base ai campi *ID*, *descrizione*, *file sorgente*.

Scenario principale:

1. L'utente inserisce il testo da ricercare tramite la barra di ricerca.
2. Il sistema filtra la lista dei requisiti in base al campo inserito.
3. Il sistema visualizza la lista dei requisiti filtrati.

3.38 UC_10 - Analisi di un singolo requisito

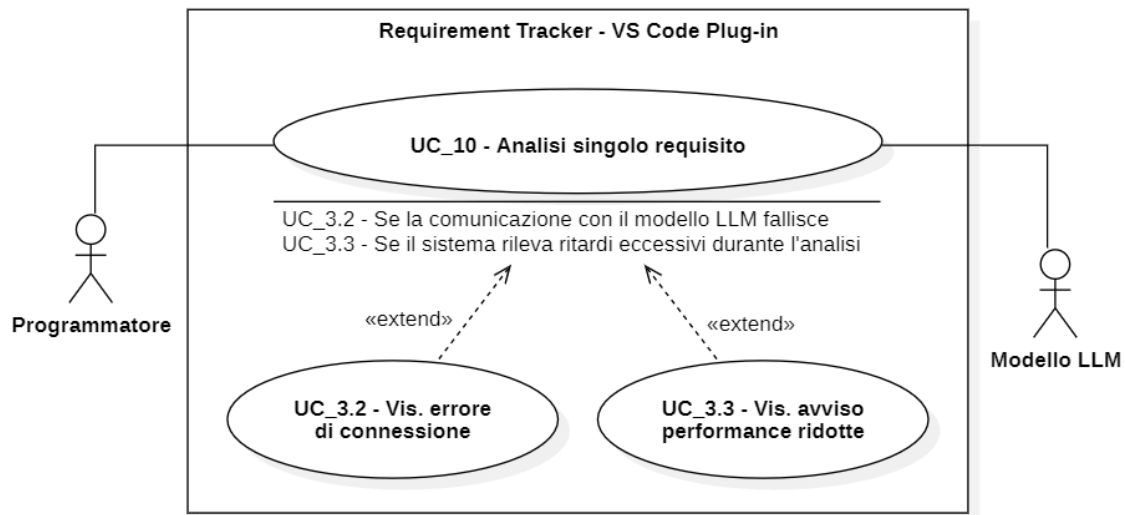


Figure 17: UC_10 - Analisi di un singolo requisito

Attore primario: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- È stata selezionata la funzione di ripetizione dell'analisi

Postcondizioni:

- Viene fornita una nuova valutazione per il requisito selezionato.

Scenario principale:

1. L'utente clicca sull'icona relativa alla funzione di ripetizione analisi, inserita nella riga del singolo requisito in elenco [UC_6.1.2].
2. Il sistema invia il requisito ed il relativo codice al modello LLM per una nuova analisi, analogamente ad [UC_3].
3. I risultati aggiornati vengono registrati e visualizzati per il requisito selezionato [UC_8].

Estensioni:

- **UC_3.2 - Visualizzazione errore di connessione:** Se la comunicazione con il modello LLM fallisce (es. timeout o connessione interrotta), il sistema informa l'utente e consente di riprovare.
- **UC_3.3 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.

3.39 UC_11 - Tracciamento dei requisiti nel codice

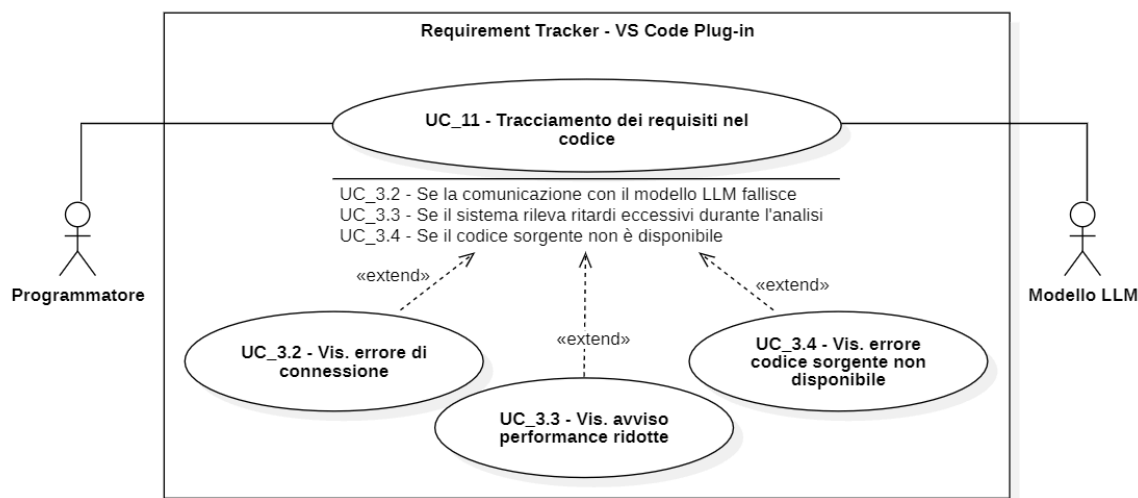


Figure 18: UC_11 - Funzione di tracciamento dei requisiti

Attore primario: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- I requisiti sono stati importati senza informazioni di tracciamento [UC_1].
- È disponibile il codice sorgente nel progetto.

Postcondizioni:

- Il sistema esegue una ricerca nel codice sorgente per associare, a ciascun requisito, la porzione di codice che lo implementa.
- Il tracciamento ottenuto viene registrato e reso disponibile nella vista dei requisiti [UC_7.2].

Scenario principale:

1. L'utente, notando l'assenza del tracciamento, seleziona l'opzione "Tracciamento dei requisiti".
2. Il sistema invia il codice ed i requisiti al modello LLM per la mappatura.
3. Il modello confronta il contenuto del codice con i requisiti inviati e individua le porzioni che li implementano.
4. Il modello restituisce una possibile mappatura del codice, che include il nome del file e l'intervallo di righe, per ogni requisito.
5. Il mapping risultante viene registrato e visualizzato insieme ai requisiti.

Estensioni:

- **UC_3.2 - Visualizzazione errore di connessione:** Se la comunicazione con il modello LLM fallisce (es. timeout o connessione interrotta), il sistema informa l'utente e consente di riprovare.
- **UC_3.3 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.
- **UC_3.4 - Visualizzazione errore codice non disponibile:** Se il progetto non contiene il file sorgente o non è configurato correttamente.

3.40 UC_12 - Configurazione dei path da ignorare

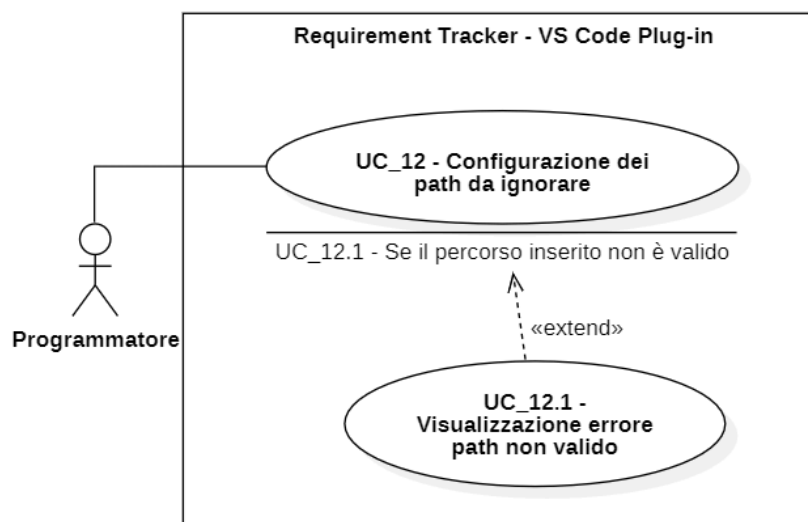


Figure 19: UC_12 - Configurazione dei path da ignorare

Attore primario: Programmatore.

Precondizioni:

- Il progetto è stato configurato in Visual Studio Code.
- Il plug-in è attivo e funzionante.

Postcondizioni:

- I path specificati nel file .reqignore vengono esclusi dall'analisi [UC_3] e dal tracciamento dei requisiti [UC_11].

Scenario principale:

1. L'utente crea o modifica un file .reqignore nel progetto.
2. L'utente inserisce nel file .reqignore i path o *pattern*^G relativi ai file o directory da escludere.
3. Il sistema rileva automaticamente le modifiche apportate al file .reqignore.
4. Durante la l'analisi [UC_3] ed il tracciamento [UC_11], il sistema esclude i path specificati nel file .reqignore.
5. L'utente avvia l'analisi o il tracciamento e i path ignorati non vengono considerati.

Estensioni:

- **UC_12.1 - Visualizzazione errore path non valido:** Se il path specificato non è valido, il sistema notifica l'utente e ignora l'*entry*^G errata mantenendo valide le altre.

3.41 UC_12.1 - Visualizzazione errore path non valido

Attore primario: Programmatore.

Precondizioni:

- L'utente inserisce un path o pattern non valido nel file .reqignore.

Postcondizioni:

- Il sistema ignora il path non valido e prosegue con le configurazioni valide.

Scenario principale:

1. L'utente modifica il file .reqignore e inserisce un path o pattern non valido.
2. Il sistema rileva l'entry non valida durante la verifica del file.
3. Il sistema notifica l'utente dell'errore e fornisce dettagli sul path non valido.
4. Il sistema ignora l'entry non valida e considera solo i path configurati correttamente.

3.42 UC_13 - Configurazione del modello LLM

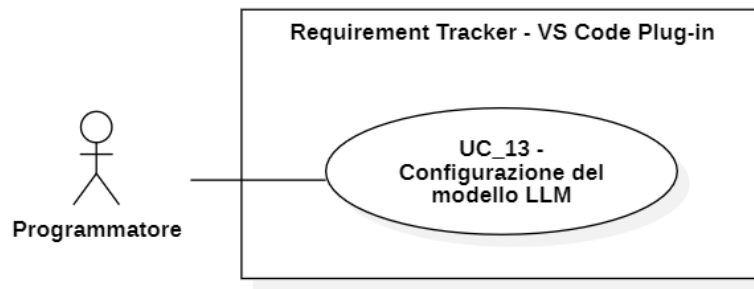


Figure 20: UC_13 - Configurazione del modello LLM per l'analisi

Attore primario: Programmatore.

Precondizioni:

- Il plug-in “Requirement Tracker Plug-in” è installato e attivo in Visual Studio Code.
- L'utente ha accesso alle impostazioni di Visual Studio Code nel menu “Extensions”.

Postcondizioni:

- Il modello LLM configurato viene salvato e verrà utilizzato dal plug-in per le analisi dei requisiti e del codice [UC_3].

Scenario principale:

1. L'utente apre le impostazioni di Visual Studio Code e naviga nel menu “Extensions”.
2. L'utente individua il plug-in “Requirement Tracker Plug-in”.
3. All'interno delle impostazioni del plug-in, l'utente seleziona il campo affianco la voce “Model”.
4. L'utente inserisce il nome del modello LLM desiderato (es. “llama3.2:3b” o “deepseek-coder:7b”).
5. Il sistema salva la configurazione e la utilizza per le analisi successive.

3.43 UC_14 - Configurazione dell'endpoint del server Ollama

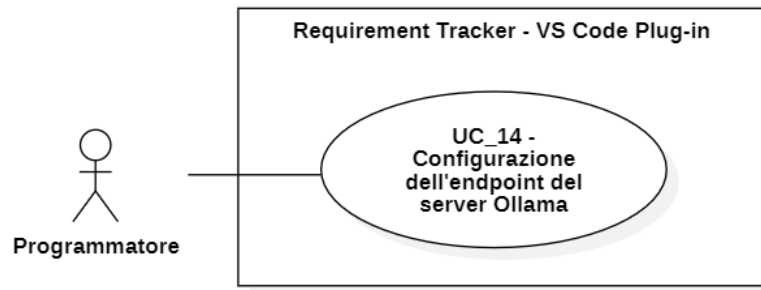


Figure 21: UC_14 - Configurazione dell'endpoint di Ollama

Attore primario: Programmatore.

Precondizioni:

- Il plug-in “Requirement Tracker Plug-in” è installato e attivo in Visual Studio Code.
- L’utente ha accesso alle impostazioni nel menu “Extensions” di Visual Studio Code.

Postcondizioni:

- L’endpoint del server Ollama viene salvato e utilizzato dal plug-in per le chiamate API.

Scenario principale:

1. L’utente apre le impostazioni di Visual Studio Code e naviga nel menu “Extensions”.
2. L’utente individua il plug-in “Requirement Tracker Plug-in”.
3. All’interno delle impostazioni, l’utente seleziona la voce “Ollama Endpoint”.
4. L’utente inserisce l’indirizzo IP o il link del server Ollama.
5. Il sistema salva l’endpoint e lo utilizza per le chiamate API durante le analisi.

3.44 UC_15 - Configurazione della soglia di conformità

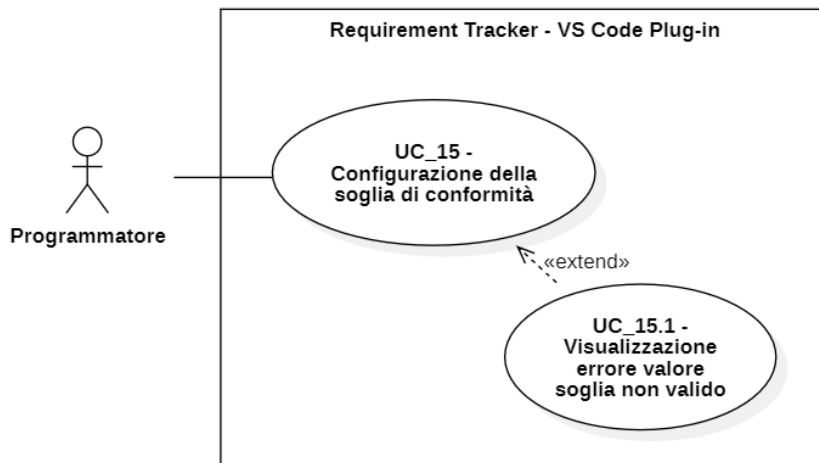


Figure 22: UC_15 - Configurazione della soglia del quality score

Attore primario: Programmatore.

Precondizioni:

- Il plug-in “Requirement Tracker Plug-in” è installato e attivo in Visual Studio Code.
- L’utente ha accesso alle impostazioni nel menu “Extensions” di Visual Studio Code.

Postcondizioni:

- La soglia di qualità viene salvata e utilizzata dal plug-in per determinare lo stato “passed” (true/false) dei requisiti.

Scenario principale:

1. L’utente apre le impostazioni di Visual Studio Code e naviga nel menu “Extensions”.
2. L’utente individua il plug-in “Requirement Tracker Plug-in”.
3. All’interno delle impostazioni, l’utente seleziona la voce “Quality threshold”.
4. L’utente inserisce il valore soglia desiderato (es. 80).
5. Il sistema salva la soglia e la utilizza per valutare i risultati dell’analisi dei requisiti.

Estensioni:

- **UC_15.1 - Visualizza errore valore soglia non valido:** Se il valore inserito per la soglia non è numerico o non rientra nei limiti previsti, il sistema notifica l’errore all’utente e richiede di inserire un valore corretto.

3.45 UC_15.1 - Visualizzazione errore valore soglia non valido

Attore primario: Programmatore.

Precondizioni:

- Durante la configurazione della soglia di conformità in [UC_15], l'utente inserisce un valore non numerico o un valore che non rientra nei limiti previsti.

Postcondizioni:

- Il sistema visualizza un messaggio d'errore che informa l'utente dell'inserimento non valido e richiede la correzione del valore.

Scenario principale:

1. L'utente inserisce il valore per la soglia di conformità nella sezione "Quality threshold" delle impostazioni del plug-in.
2. Il sistema verifica il valore inserito.
3. Se il valore non è numerico o non rientra nei limiti previsti, il sistema visualizza un messaggio d'errore specifico.
4. Il sistema richiede all'utente di inserire un valore corretto.

4 Requisiti

4.1 Introduzione

Il gruppo NextSoft, a seguito di una attenta analisi dichiara che i requisiti che il prodotto finale andrà a soddisfare sono i seguenti. Questi vengono mostrati di seguito in forma tabellare, seguendo quanto detto all'interno del documento *Norme di Progetto*

4.2 Requisiti Funzionali

Questi requisiti descrivono cosa il sistema deve fare

Codice	Classificazione	Descrizione	Fonti
RFO001	Obbligatorio	Il sistema deve comunicare con un modello LLM tramite una REST API per ottenere delle valutazioni	UC_3
RFO002	Obbligatorio	Il sistema deve visualizzare graficamente i risultati delle analisi integrandoli nella lista dei requisiti	UC_8, UC_8.1, UC_8.1.1, UC_8.1.2, UC_8.2, UC_8.3, UC_8.4, UC_8.4.1, UC_8.5, UC_8.5.1
RFO003	Obbligatorio	Il sistema deve informare l'utente nel caso, a seguito di un'analisi, non ci siano risultati	UC_5.1
RFF004	Facoltativo ^G	Il codice relativo a un requisito da analizzare può essere presente in file diversi	Proponente
RFF005	Facoltativo	L'utente deve essere in grado di configurare il modello utilizzato per l'analisi	Proponente
RFO007	Obbligatorio	L'estensione di Visual Studio Code deve essere in inglese	Proponente
RFD008	Desiderabile ^G	Il sistema deve informare l'utente in caso di rallentamenti dovuti a una connessione lenta (risposte con tempo di attesa >20s) o a un modello troppo grande	DA METTERE

Codice	Classificazione	Descrizione	Fonti
		(<i>prompt</i> ^G maggiore di 6000 <i>token</i> ^G e/o velocità di risposta < 20 token/s)	
RFO009	Obbligatorio	Il sistema deve permettere l'importazione di file CSV	UC_1, UC_2
RFO010	Obbligatorio	Il sistema deve restituire un errore in caso il file CSV importato non sia valido	UC_1.2
RFO011	Obbligatorio	Il sistema deve essere in grado di verificare la validità del file CSV da importare, effettuando controlli sui campi del file	UC_1, UC_1.2, UC_2
RFO012	Obbligatorio	Il sistema deve essere in grado di caricare il file dei requisiti in formato CSV dal <i>filesystem</i> ^G	Capitolato, UC_1, UC_2, UC_1.1, UC_1.3, Proponente
RFO013	Obbligatorio	Il sistema deve avvisare l'utente in caso il file CSV caricato non sia valido e permettere all'utente di riprovare	UC_1.2
RFO014	Obbligatorio	Il sistema deve visualizzare le informazioni relative ai requisiti da analizzare, con o senza tracciamento, all'interno di un elenco annidato (nome del file in cui il codice sorgente è stato tracciato e le relative righe di codice, se presenti)	UC_1, UC_2, UC_6, UC_6.1, UC_6.1.1, UC_7
RFO015	Obbligatorio	Per ogni requisito presente nella lista caricata sarà presente un'icona predisposta ad avviare l'analisi del	UC_6.1.2

Codice	Classificazione	Descrizione	Fonti
		singolo requisito quando necessario	
RFO016	Obbligatorio	Il sistema deve essere in grado di reperire le porzioni di codice tracciate all'interno del file CSV importato, e deve essere in grado di utilizzarle nell'analisi dei requisiti	UC_2, UC_3, UC_11
RFO017	Obbligatorio	Il sistema deve fornire una valutazione generica (per esempio, "Result: passed/not passed") per indicare se il requisito è stato rispettato	UC_3, UC_8.1, UC_8.1.2, UC_8.1.1
RFO018	Obbligatorio	Il sistema deve fornire una valutazione da 0 a 100 sull'aderenza della porzione di codice analizzata al rispettivo requisito	UC_3, UC_8.2
RFO019	Obbligatorio	Il sistema deve fornire una valutazione testuale del codice, fornendo i problemi trovati e gli eventuali suggerimenti per migliorare il codice (se presenti)	UC_8.4, UC_8, UC_3, UC_8.4.1, UC_8.5, UC_8.5.1
RFO020	Obbligatorio	Il sistema deve essere in grado di aggiornare la vista dei requisiti, aggiungendo i risultati delle analisi	UC_8
RFO021	Obbligatorio	Il sistema, nel caso vengano inseriti dei requisiti non tracciati, deve mostrare un messaggio che informi l'utente sul loro mancato tracciamento	UC_3.1, UC_4.2

Codice	Classificazione	Descrizione	Fonti
RFO022	Obbligatorio	Il sistema, in caso la comunicazione con il modello LLM venga interrotta (es. timeout, connessione interrotta), deve informare l'utente	UC_3.2
RFO023	Obbligatorio	Il sistema deve essere in grado di misurare i tempi di risposta di Ollama, e restituire un errore in caso questi superino una soglia limite prestabilita	UC_3.3
RFO024	Obbligatorio	Il sistema, nel caso il progetto non contenga il file con il codice sorgente associato a uno specifico requisito, informa l'utente di tale mancanza	UC_3.4
RFO025	Obbligatorio	Il sistema, una volta eseguita l'analisi, deve essere in grado di esportare il tracciamento dei requisiti in formato CSV, i quali conterranno esclusivamente l'ID, la descrizione e il tracciamento	UC_4
RFO026	Obbligatorio	Il sistema, nel caso fallisse nell'operazione di esportazione dei requisiti (es. spazio esaurito o permessi insufficienti), informerà l'utente con un messaggio di errore e permetterà di riprovare	UC_4.1
RFO027	Obbligatorio	Il sistema, una volta eseguita l'analisi, deve essere in grado	UC_5

Codice	Classificazione	Descrizione	Fonti
		di esportare sia il tracciamento dei requisiti sia i risultati dell'analisi	
RFO028	Obbligatorio	Il sistema, nel caso si voglia esportare il tracciamento dei requisiti (e dell'analisi) quando queste non sono presenti, informerà l'utente con un messaggio di errore	UC_5.1, UC_3.1
RFO029	Obbligatorio	Il sistema, una volta selezionato un requisito dalla lista, deve mostrarlo in una sottolista dove vengono specificati la descrizione del requisito stesso e, se presente, il suo tracciamento nel codice sorgente	UC_7, UC_7.1, UC_7.2
RFO030	Obbligatorio	Il sistema, quando mostra la visualizzazione di dettaglio di un requisito, deve mostrare il nome del file in cui si trova il codice sorgente associato al requisito	UC_7.2.1
RFO031	Obbligatorio	Il sistema deve permettere all'utente di filtrare i requisiti in base ai campi ID, descrizione e file sorgente	UC_9
RFO032	Obbligatorio	Il sistema, in presenza di più requisiti caricati, deve essere in grado di eseguire l'analisi di un requisito specifico	UC_10
RFO033	Obbligatorio	Il sistema deve essere in grado di associare ai requisiti non mappati il relativo codice sorgente che lo implementi	UC_11

Codice	Classificazione	Descrizione	Fonti
		attraverso una richiesta di analisi all'LLM, quindi registrare le righe di codice relative e mostrarle nella vista insieme ai requisiti	
RFO034	Obbligatorio	Il sistema deve essere in grado di escludere dall'analisi e dal tracciamento dei requisiti tutti i file elencati in un apposito file di configurazione (chiamato .reqignore) contenente i percorsi dei file associati al progetto. Se il file di configurazione include percorsi non validi, il sistema deve notificare l'errore all'utente, senza però interrompere l'analisi, che deve comunque procedere sui file validi	UC_12, UC_12.1
RFO035	Obbligatorio	Il sistema deve permettere la configurazione del modello o dei modelli LLM che saranno utilizzati per l'analisi dei requisiti	UC_13, Proponente
RFO036	Obbligatorio	Il sistema deve permettere di configurare l'endpoint del server Ollama	UC_14, Proponente
RFO037	Obbligatorio	Il sistema deve essere in grado di ripetere l'analisi di uno o più requisiti	UC_10
RFO038	Obbligatorio	Il sistema deve permettere all'utente di modificare la soglia di accettazione di un	UC_15, Proponente

Codice	Classificazione	Descrizione	Fonti
		requisito, la quale permette ad esso di essere identificato come rispettato	
RFO039	Obbligatorio	Il sistema, nel caso venga inserito un valore “soglia” (di accettazione) non valido, ossia che non rientra nei parametri stabiliti, deve visualizzare un messaggio di errore specifico e permettere all’utente di modificarlo	UC_15.1, UC_15

Table 1: Requisiti Funzionali

4.3 Requisiti di qualità

Questi requisiti riguardano le caratteristiche qualitative del sistema

Codice	Classificazione	Descrizione	Fonti
RQO001	Obbligatorio	Il plug-in deve essere modulare per consentire e facilitare l’aggiunta di nuove <i>feature</i> ^G in base a esigenze o aggiornamenti futuri del progetto.	Capitolato
RQF001	Facoltativo	Il sistema deve essere compatibile con Visual Studio Code con versione ≥ 1.98 e Ollama con versione $\geq 0.6.4$	DA METTERE
RQO002	Obbligatorio	Il prodotto deve essere sviluppato secondo quanto detto all’interno del file <i>Norme_di_Progetto_v1.0.0</i>	<i>Norme_di_Progetto_v1.0.0</i>
RQO003	Obbligatorio	Il prodotto deve essere sviluppato secondo quanto detto all’interno del file <i>Piano_di_Progetto_v1.0.0</i>	<i>Piano_di_Progetto_v1.0.0</i>

Table 2: Requisiti di Qualità

4.4 Requisiti di vincolo

Questi requisiti specificano limiti tecnici o di conformità

Codice	Classificazione	Descrizione	Fonti
RVO001	Obbligatorio	il sistema deve essere in grado di analizzare codice nei linguaggi C e C++	Capitolato
RVF002	Facoltativo	il sistema deve poter supportare altri linguaggi oltre a C e C++	Capitolato
RVF003	Facoltativo	Il sistema deve fornire valutazioni conformi alle normative sulla sicurezza funzionale (ISO 26262 o IEC 61508)	Capitolato

Table 3: Requisiti di Vincolo

4.5 Requisiti Prestazionali

Questi requisiti descrivono aspetti legati alla velocità e alle prestazioni del sistema.

Codice	Classificazione	Descrizione	Fonti
RPF001	Facoltativo	Il sistema deve essere in grado di gestire file CSV di grandi dimensioni (ad esempio >100MB) senza subire rallentamenti	Proponente

Table 4: Requisiti Prestazionali

4.6 Tracciamento dei Requisiti

Fonte	Requisiti
UC_1	RFO009, RFO010, RFO011, RFO012, RFO013
UC_1.1	RFO012
UC_1.2	RFO010, RFO013
UC_2	RFO009, RFO016

Fonte	Requisiti
UC_3	RFO001, RFO017, RFO018, RFO019, RFO022, RFO023, RFO024
UC_3.1	RFO021
UC_3.2	RFO022
UC_3.3	RFO023
UC_3.4	RFO024
UC_4	RFO025
UC_4.1	RFO026
UC_4.2	RFO021
UC_5	RFO027, RFO028
UC_5.1	RFO028
UC_6	RFO014
UC_6.1	RFO014, RFO015
UC_6.1.1	RFO014
UC_6.1.2	RFO015
UC_7	RFO029
UC_7.1	RFO029
UC_7.2	RFO029
UC_7.2.1	RFO030
UC_8	RFO002, RFO020
UC_8.1	RFO017
UC_8.2	RFO018
UC_8.4	RFO019
UC_8.5	RFO019
UC_9	RFO031
UC_10	RFO032, RFO037
UC_11	RFO033
UC_12	RFO034
UC_12.1	RFO034
UC_13	RFO035
UC_14	RFO036
UC_15	RFO038

Fonte	Requisiti
UC_15.1	RFO039
Proponente	RFF004, RFF005, RFO007, RFD008, RFO012, RFO035, RFO036, RFO038, RQO001, RVF001, RVF002, RVF003, RFO001
Capitolato	RVO001, RVF002, RVF003, RQO001, RQO002, RQO003

Table 5: Tracciamento dei Requisiti

4.7 Riepilogo

Tipologia	Obbligatorio	Desiderabile	Facoltativo	Totale
<i>Funzionale</i> ^G	36	1	2	39
<i>Di Qualità</i> ^G	3		1	4
<i>Di Vincolo</i> ^G	1		2	3
Prestazionale	0	0	1	1

Table 6: Riepilogo

Elenco delle immagini

Figure 1	UC_1 - Importazione dei requisiti da file CSV	7
Figure 2	UC_2 - Importazione dei requisiti e del tracciamento da file CSV	10
Figure 3	UC_3 - Analisi dei requisiti e della loro implementazione	12
Figure 4	UC_4 - Esportazione del tracciamento su file CSV	15
Figure 5	UC_5 - Esportazione del tracciamento e dei risultati su file CSV	17
Figure 6	UC_6 - Visualizzazione dell'elenco dei requisiti	19
Figure 7	UC_6 - Dettaglio della visualizzazione elenco requisiti	19
Figure 8	UC_6.1 - Visualizzazione singolo requisito in elenco	20
Figure 9	UC_7 - Visualizzazione dettaglio di un requisito	21
Figure 10	UC_7 - Sottocaso d'uso della visualizzazione in dettaglio di un requisito	22
Figure 11	UC_7.2 - Visualizzazione del tracciamento di un requisito	23
Figure 12	UC_8 - Visualizzazione dei risultati di un requisito	25
Figure 13	UC_8 - Dettaglio della visualizzazione dei risultati di un requisito	26
Figure 14	UC_8.4 - Visualizzazione lista dei suggerimenti	29
Figure 15	UC_8.5 - Visualizzazione lista dei problemi	30
Figure 16	UC9 - Filtraggio dei requisiti	32
Figure 17	UC_10 - Analisi di un singolo requisito	33
Figure 18	UC_11 - Funzione di tracciamento dei requisiti	34
Figure 19	UC_12 - Configurazione dei path da ignorare	35
Figure 20	UC_13 - Configurazione del modello LLM per l'analisi	37
Figure 21	UC_14 - Configurazione dell'endpoint di Ollama	38
Figure 22	UC_15 - Configurazione della soglia del quality score	39

Elenco delle tabelle

Table 1	Requisiti Funzionali	41
Table 2	Requisiti di Qualità	47
Table 3	Requisiti di Vincolo	48
Table 4	Requisiti Prestazionali	48
Table 5	Tracciamento dei Requisiti	48
Table 6	Riepilogo	50