



Analisi dei Requisiti

Versione: 0.4.0

11/02/2025

Redattori

Malik Giafar Mohamed

Verifica

Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

Approvazione

Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

Uso

Esterno

nextsoftpadova@gmail.com

Registro dei cambiamenti

| Versione | Data | Autore | Descrizione | Verifica | Approvazione |
|----------|------------|----------------------|--|---|---|
| 0.4.0 | 11/02/2025 | Marco Perazzolo | Inserimento dei diagrammi Use Case | | |
| 0.3.1 | 06/02/2025 | Marco Perazzolo | Finalizzazione Use Case testuali | | |
| 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 | 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 | Malik Giafar Mohamed |
| 0.1.0 | 23/11/2024 | Malik Giafar Mohamed | Creazione Documento | Ion Cainareanu, Maria Fuensanta Trigueros Hernandez | Ion Cainareanu, Maria Fuensanta Trigueros Hernandez |

Indice

| | | |
|-----|--|---|
| 1 | Introduzione | 4 |
| 1.1 | Scopo del documento | 4 |
| 1.2 | Scopo del prodotto | 4 |
| 1.3 | Glossario | 4 |
| 2 | Descrizione | 5 |
| 2.1 | Obiettivi del prodotto | 5 |
| 2.2 | Funzionalità del prodotto | 5 |
| 2.3 | Utenti e caratteristiche | 5 |
| 3 | Use Case | 6 |
| 3.1 | Obiettivi | 6 |
| 3.2 | Attori | 6 |
| 3.3 | UC_1 - Importazione da file CSV | 7 |
| 3.4 | UC_1.2 - Selezione del file | 7 |
| 3.5 | UC_1.3 - Importazione dei requisiti da file CSV | 8 |
| 3.6 | UC_1.4 - Importazione dei requisiti e del tracciamento da file CSV | 8 |
| 3.7 | UC_1.5 - Visualizza errore file importazione | 8 |
| 3.8 | UC_2 - Analisi dei requisiti e della loro implementazione | 9 |

| | |
|--|----|
| 3.9 UC_2.1 - Analisi semantica dei requisiti | 9 |
| 3.10 UC_2.2 - Analisi implementazione requisiti | 10 |
| 3.11 UC_2.3 - Visualizza errore tracciamento mancante | 10 |
| 3.12 UC_2.4 - Visualizzazione errore di connessione | 10 |
| 3.13 UC_2.5 - Visualizzazione avviso performance ridotte | 11 |
| 3.14 UC_2.6 - Visualizzazione errore codice sorgente non disponibile | 11 |
| 3.15 UC_3 - Esportazione dei risultati in formato CSV | 12 |
| 3.16 UC_3.1 - Visualizza errore di salvataggio | 12 |
| 3.17 UC_4 - Visualizzazione dei risultati | 13 |
| 3.18 UC_4.1 - Visualizzazione singolo risultato | 14 |
| 3.19 UC_4.1.1 - Visualizzazione dettaglio singolo risultato | 15 |
| 3.20 UC_4.1.2 - Visualizzazione stato di conformità | 15 |
| 3.21 UC_4.1.5 - Visualizzazione punteggio in centesimi | 16 |
| 3.22 UC_4.1.6 - Visualizzazione suggerimenti | 16 |
| 3.23 UC_4.1.10 - Visualizzazione problemi | 17 |
| 3.24 UC_5 - Filtraggio dei requisiti | 18 |
| 3.25 UC_5.1 - Visualizzazione lista vuota | 18 |
| 3.26 UC_5.2 - Visualizzazione lista requisiti filtrati | 18 |
| 3.27 UC_6 - Ripetizione analisi singolo requisito | 20 |
| 3.28 UC_7 - Tracciamento dei requisiti nel codice | 21 |
| 3.29 UC_8 - Configurazione dei path da ignorare | 22 |
| 3.30 UC_8.1 - Visualizzazione errore path non valido | 22 |
| 3.31 UC_9 - Visualizzazione dei requisiti | 23 |
| 3.32 UC_9.1 - Visualizzazione singolo requisito | 24 |
| 3.33 UC_9.1.1 - Visualizzazione dettaglio singolo requisito | 24 |
| 3.34 UC_9.1.2 - Visualizzazione ID requisito | 25 |
| 3.35 UC_9.1.3 - Visualizzazione titolo requisito | 25 |
| 3.36 UC_9.1.4 - Visualizzazione testo requisito | 25 |
| 3.37 UC_9.1.5 - Visualizzazione tracciamento singolo requisito | 26 |
| 3.38 UC_9.1.6 - Visualizzazione nome file | 26 |
| 3.39 UC_9.1.7 - Visualizzazione riga inizio | 26 |
| 3.40 UC_9.1.8 - Visualizzazione riga fine | 27 |
| 3.41 UC_9.2 - Visualizzazione lista vuota | 27 |
| 4 Elenco delle immagini | 28 |
| 5 Elenco delle tabelle | 28 |

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 - VS Code Plug-in**, con particolare attenzione all'utilizzo dell'UML per la modellazione dei casi d'uso. Il documento funge da riferimento per tutti gli stakeholder coinvolti, descrivendo il contesto operativo, i requisiti funzionali e non funzionali, nonché le linee guida tecnologiche necessarie per lo sviluppo del plug-in. I casi d'uso saranno descritti utilizzando una struttura standardizzata, che includerà il nominativo del caso, gli attori principali, le precondizioni, le postcondizioni, lo scenario principale e gli eventuali scenari alternativi 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 e un'implementazione efficaci.

1.2 Scopo del prodotto

Lo scopo di **Requirement Tracker - VS Code Plug-in** è affrontare il problema della complessità nella gestione e nel tracciamento dei requisiti nei progetti software di grandi dimensioni. Nei codebase 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**. 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 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 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 tramite un modello LLM;
- Valutazione qualitativa dei requisiti;
- Visualizzazione grafica dei risultati dell'analisi;
- Filtraggio dei risultati dell'analisi;
- Possibilità di ripetere l'analisi di un requisito specifico;
- Esportazione dei risultati dell'analisi in formato CSV;
- Verifica dell'implementazione dei requisiti nel codice sorgente;
- Analisi semantica dei requisiti e del codice sorgente;
- Suggerimenti per migliorare la qualità dei requisiti e 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. Pertanto, il plug-in deve essere progettato per essere il più intuitivo possibile, con un processo di installazione semplice e accessibile.

3 Use Case

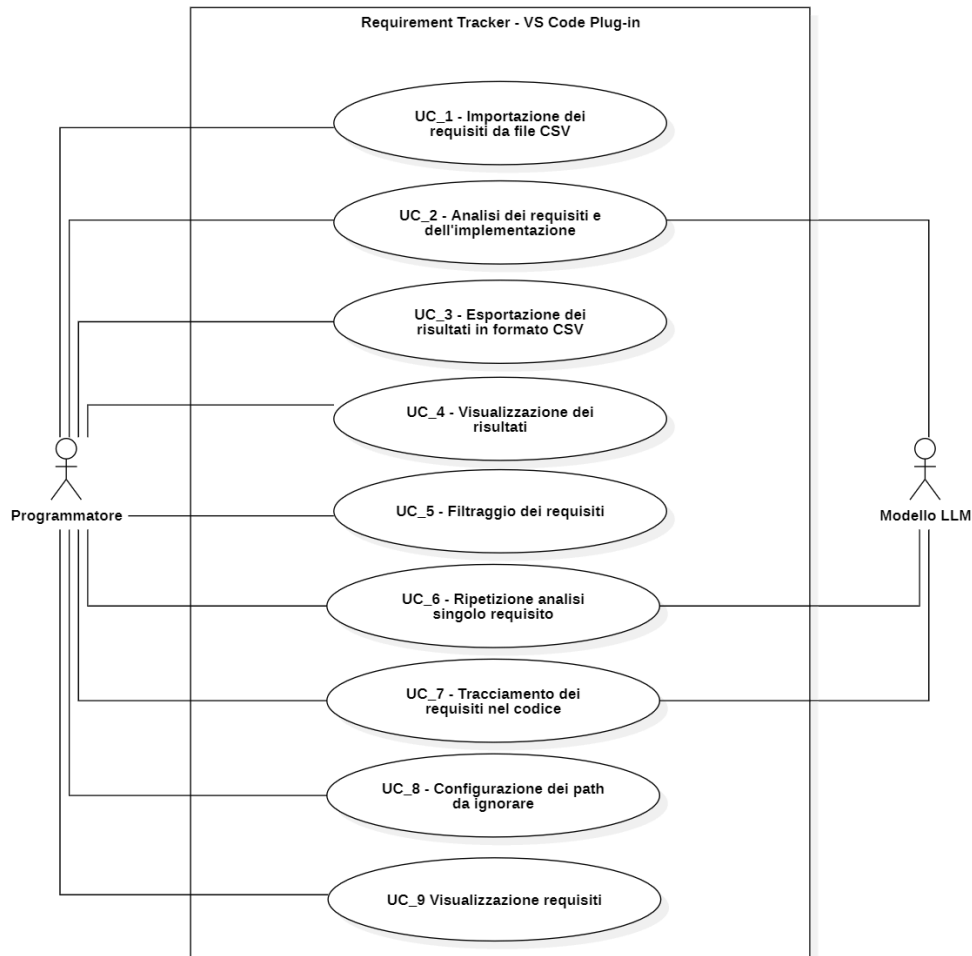


Figure 1: Panoramica delle funzionalità principali del plugin.

3.1 Obiettivi

Questa sezione si propone di identificare e descrivere i casi d'uso derivati dall'analisi del capitolato 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, il **Programmatore**, esso rappresenta un utente che utilizza il plug-in *Requirement Tracker - VS 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 da file CSV

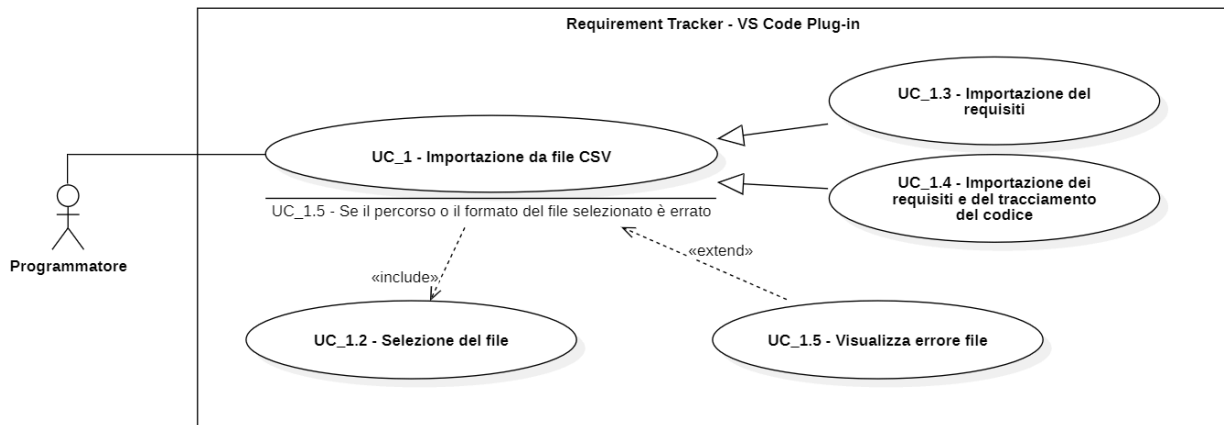


Figure 2: UC_1 - Importazione dei requisiti da file

Attori: 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 VS Code.

Postcondizioni:

- I requisiti (e, se presente, il tracciamento) vengono importati e sono visualizzabili nel sistema.

Scenario principale:

1. In base all'opzione di importazione selezionata dall'utente, il sistema applica una delle seguenti specializzazioni:
 - Se l'utente seleziona la voce "Importa requisiti", il flusso procede con [UC_1.3].
 - Se l'utente seleziona la voce "Importa requisiti con tracciamento", il flusso procede con [UC_1.4].
2. Il sistema apre un file explorer per la selezione del file.
3. L'utente seleziona il file CSV da importare [UC_1.2].
4. Il sistema verifica la validità del file.
5. In base al contenuto del file, il sistema importa i requisiti ed eventualmente il relativo tracciamento.
6. I requisiti importati vengono mostrati in una vista strutturata [UC_9].

Estensioni:

- **UC_1.5 - Visualizza 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_1.2 - Selezione del file

Attori: 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_1.3] oppure [UC_1.4].

3.5 UC_1.3 - Importazione dei requisiti da file CSV

Attori: 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 VS Code.

Postcondizioni:

- I requisiti sono importati e sono visualizzabili nel sistema.

Scenario principale:

1. L'utente seleziona l'opzione "Importa requisiti".
2. Il sistema apre un file explorer.
3. L'utente seleziona il file CSV da importare [UC_1.2]
4. Il sistema verifica la validità del file e importa i dati (ID, titolo, testo, di ogni requisito).
5. I requisiti importati vengono mostrati in una vista strutturata [UC_9].

3.6 UC_1.4 - Importazione dei requisiti e del tracciamento da file CSV

Attori: 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 VS 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_1.2]
4. Il sistema verifica la validità del file e importa i dati (ID, titolo, testo, file, intervallo righe, di ogni requisito).
5. I requisiti importati vengono mostrati in una vista strutturata [UC_9].

3.7 UC_1.5 - Visualizza errore file importazione

Attori: Programmatore.

Precondizioni:

- L'utente ha selezionato un file CSV da importare [UC_1.2].

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.
2. L'importazione del file fallisce
3. Il sistema mostra un messaggio di errore esplicativo e richiede di selezionare un file valido.

3.8 UC_2 - Analisi dei requisiti e della loro implementazione

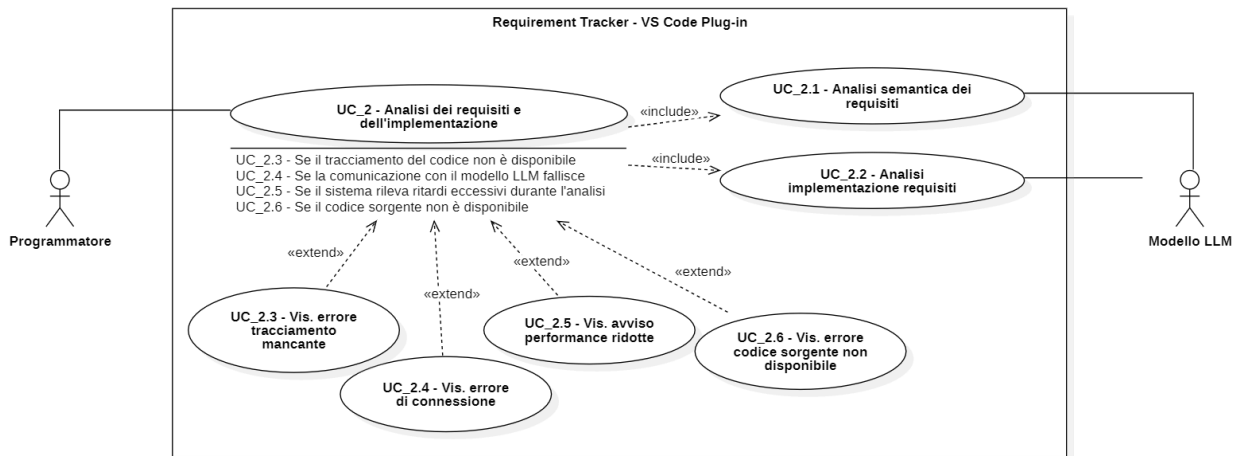


Figure 3: UC_2 - Analisi dei requisiti e dell'implementazione

Attori: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- I requisiti sono stati importati [UC_1].
- Il tracciamento dei requisiti nel codice è disponibile, ottenuto direttamente da [UC_1.4] oppure dopo [UC_7].
- 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_1.4] o [UC_7]).
3. Il sistema esegue l'analisi semantica di ogni requisito [UC_2.1] inviandoli al modello.
4. Il sistema esegue la verifica e la valutazione dell'implementazione del requisito nel codice [UC_2.2] inviandoli al modello.
7. Il modello restituisce i risultati complessivi delle precedenti analisi.
8. Il sistema registra i dati e li rende disponibili per la visualizzazione ([UC_4]).

Estensioni:

- **UC_2.3 - Visualizzazione errore tracciamento mancante:** Se il mapping del codice non è disponibile, il sistema visualizza un messaggio d'errore specifico.
- **UC_2.4 - 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_2.5 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.
- **UC_2.6 - Visualizzazione errore codice sorgente non disponibile:** Se il progetto non contiene il file sorgente o non è configurato correttamente, il sistema informa l'utente e consente di riprovare.

3.9 UC_2.1 - Analisi semantica dei requisiti

Attori: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- I requisiti sono stati importati [UC_1].

Postcondizioni:

- Ogni requisito viene valutato semanticamente, in termini di completezza, chiarezza, correttezza e non ambiguità.
- I risultati dell'analisi semantica sono stati generati e verranno integrati nel report finale.

Scenario principale:

1. Il sistema estrae il testo di ciascun requisito.
2. Il sistema invia il testo al modello LLM per l'analisi semantica.
3. Il Modello LLM restituisce una valutazione del requisito e dei suggerimenti.
4. Il sistema registra e integra i risultati dopo l'analisi dell'implementazione [UC_2.3].

3.10 UC_2.2 - Analisi implementazione requisiti

Attori: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- I requisiti sono stati importati [UC_1].
- È disponibile il mapping dei requisiti nel codice (ottenuto da [UC_1.4] oppure dopo [UC_7]).

Postcondizioni:

- Il sistema verifica se le porzioni di codice associate implementano correttamente i requisiti, assegnando un punteggio, dei suggerimenti ed eventualmente dei problemi.
- I risultati della verifica sono integrati nel report finale.

Scenario principale:

1. Il sistema raccoglie le informazioni di tracciamento per ciascun requisito.
2. Il sistema estrae la porzione di codice di ogni requisito, identificata dalle informazioni di tracciamento.
3. Il sistema invia il testo del requisito e la relativa porzione di codice al modello LLM.
4. Il modello confronta il comportamento del codice con quanto richiesto dal requisito e restituisce una valutazione.
5. Il sistema registra e integra i risultati nel report finale.

3.11 UC_2.3 - Visualizza errore tracciamento mancante

Attori: Programmatore.

Precondizioni:

- Durante l'analisi [UC_2], 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_2], il sistema verifica la presenza del mapping.
2. Se il mapping risulta mancante, il sistema mostra un messaggio d'errore specifico.

3.12 UC_2.4 - Visualizzazione errore di connessione

Attori: Programmatore.

Precondizioni:

- Durante l'analisi [UC_2], 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.

3.13 UC_2.5 - Visualizzazione avviso performance ridotte

Attori: Programmatore.

Precondizioni:

- Durante l'analisi [UC_2], 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.

3.14 UC_2.6 - Visualizzazione errore codice sorgente non disponibile

Attori: 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_2].
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.

3.15 UC_3 - Esportazione dei risultati in formato CSV

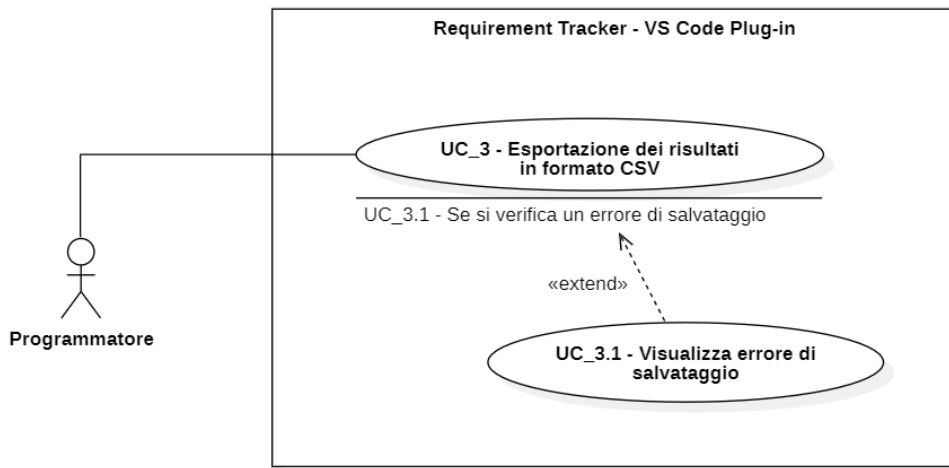


Figure 4: UC_3 - Analisi dei requisiti e dell'implementazione

Attori: Programmatore.

Precondizioni:

- I risultati dell'analisi [UC_2] sono stati generati e registrati.

Postcondizioni:

- I risultati vengono esportati correttamente in un file CSV nel percorso specificato dall'utente.

Scenario principale:

1. L'utente seleziona "Esporta risultati".
2. Il sistema apre un file explorer per scegliere il percorso di salvataggio.
3. L'utente conferma la posizione e il nome del file.
4. Il sistema salva i risultati nel formato CSV.

Estensioni:

- **UC_3.1 - Visualizza errore di salvataggio:** Se il salvataggio fallisce (es. permessi insufficienti o spazio esaurito), il sistema notifica l'errore all'utente e permette di riprovare.

3.16 UC_3.1 - Visualizza errore di salvataggio

Attori: 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.

3.17 UC_4 - Visualizzazione dei risultati

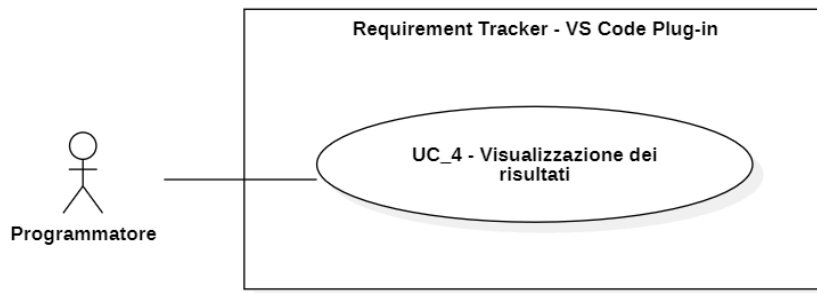


Figure 5: UC_4 - Visualizzazione dei risultati

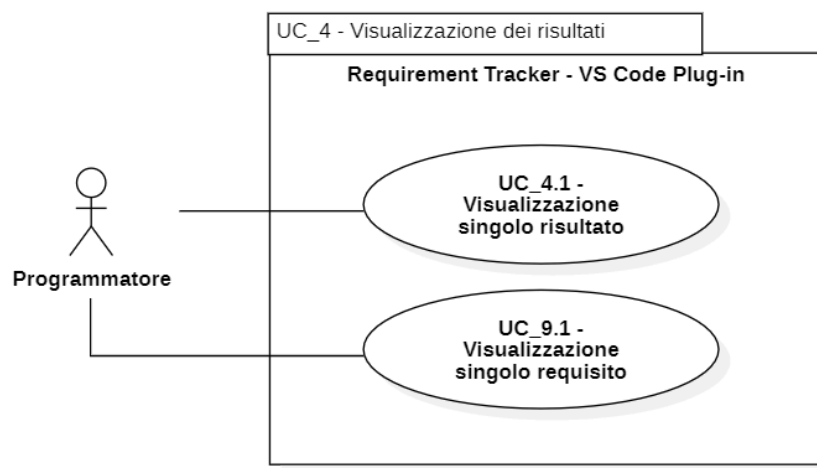


Figure 6: UC_4 - Diagramma di dettaglio sulla visualizzazione dei risultati

Attori: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_2].
- La sezione di visualizzazione del requisito selezionato è stata espansa [UC_9.1].

Postcondizioni:

- I risultati dell'analisi vengono integrati nel menu ad albero di ogni requisito [UC_9.1] in una sezione dedicata e sono visualizzabili.

Scenario principale:

1. L'utente seleziona un requisito dal menu ad albero [UC_9].
2. Il sistema espande il menu con la visualizzazione del requisito [UC_9.1] e la visualizzazione del risultato [UC_4.1].
3. L'utente seleziona la voce relativa alla visualizzazione del risultato [UC_4.1]
4. Il sistema espande la visualizzazione in dettaglio del risultato [UC_4.1.1] contenente i seguenti sottocasi:
 - [UC_4.1.2] Stato di conformità (*passed/not passed*).
 - [UC_4.1.5] Valutazione del codice in centesimi (0-100).
 - [UC_4.1.6] Suggerimenti generati.
 - [UC_4.1.10] Problemi riscontrati.

3.18 UC_4.1 - Visualizzazione singolo risultato

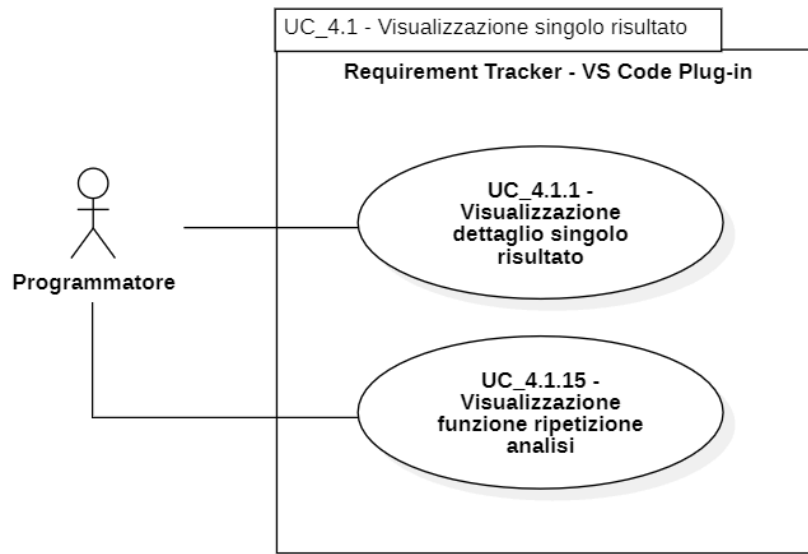


Figure 7: UC_4.1 - Visualizzazione di un singolo risultato

Attori: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_2].
- La sezione di visualizzazione del requisito selezionato è stata espansa [UC_9.1].

Postcondizioni:

- I risultati dell'analisi vengono integrati nel menu di dettaglio requisito [UC_4.1.1] e sono visualizzabili.

Scenario principale:

1. Il sistema espande il menu con la visualizzazione del requisito [UC_9.1] e la visualizzazione del risultato [UC_4.1].
2. L'utente seleziona la voce relativa alla visualizzazione del risultato [UC_4.1]
3. Il sistema espande la visualizzazione in dettaglio del risultato [UC_4.1.1].

3.19 UC_4.1.1 - Visualizzazione dettaglio singolo risultato

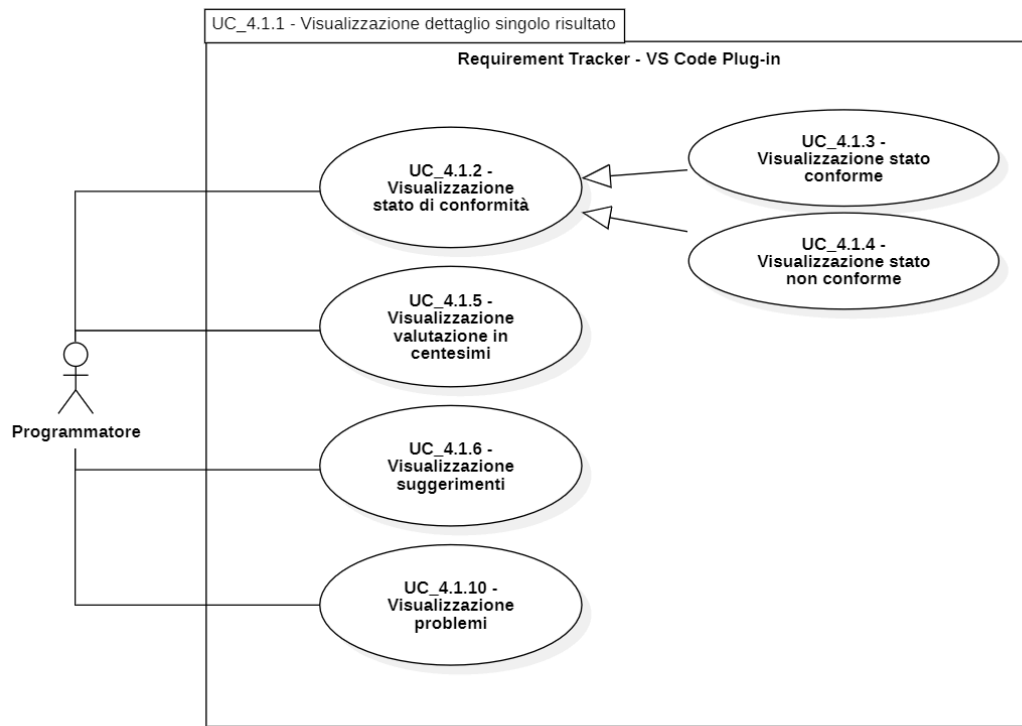


Figure 8: UC_4.1 - Visualizzazione di un singolo risultato

Attori: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_2].
- Il menu di visualizzazione dettagliata dei risultati è stato aperto da [UC_4.1].

Postcondizioni:

- Il sistema mostra una lista di sotto-elementi relativi al risultato dell'analisi per il requisito selezionato.

Scenario principale:

1. Il sistema visualizza i seguenti elementi nel dettaglio:
 - [UC_4.1.2] Stato di conformità (*passed/not passed*).
 - [UC_4.1.5] Valutazione del codice in centesimi (0-100).
 - [UC_4.1.6] Suggerimenti generati.
 - [UC_4.1.10] Problemi riscontrati.

3.20 UC_4.1.2 - Visualizzazione stato di conformità

Attori: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_2].
- Il menu ad albero relativo alla visualizzazione in dettaglio dei risultati è stato espanso [UC_4.1.1]

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 sul punteggio ottenuto:
 - Se il risultato è *passed*, visualizza lo stato conforme [UC_4.1.3]

- Se il risultato è *not-passed*, visualizza lo stato non conforme [UC_4.1.4]

3.21 UC_4.1.5 - Visualizzazione punteggio in centesimi

Attori: Programmatore.

Precondizioni:

- La valutazione del codice è disponibile [UC_2.2].
- Il menu ad albero relativo alla visualizzazione in dettaglio dei risultati è stato aperto [UC_4.1.1]

Postcondizioni:

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

Scenario principale:

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

3.22 UC_4.1.6 - Visualizzazione suggerimenti

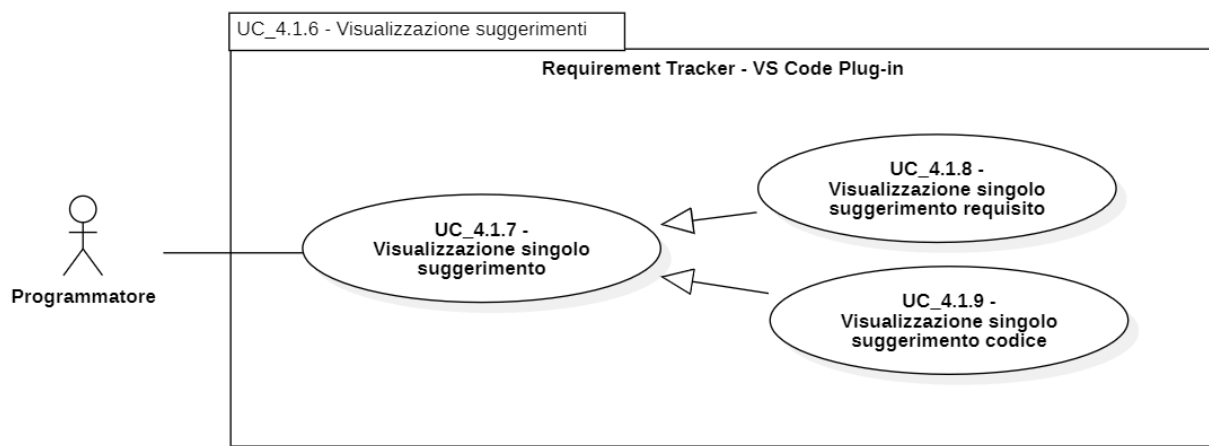


Figure 9: UC_4.1.6 - Visualizzazione dei suggerimenti

Attori: Programmatore.

Precondizioni:

- I suggerimenti relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_2].
- Il menu ad albero relativo alla visualizzazione in dettaglio dei risultati è stato aperto [UC_4.1.1]

Postcondizioni:

- Il sistema mostra un elenco strutturato dei suggerimenti relativi al requisito [UC_4.1.8] e al codice [UC_4.1.9].

Scenario principale:

1. Il sistema raccoglie i suggerimenti generati e li visualizza in forma di elenco.

3.23 UC_4.1.10 - Visualizzazione problemi

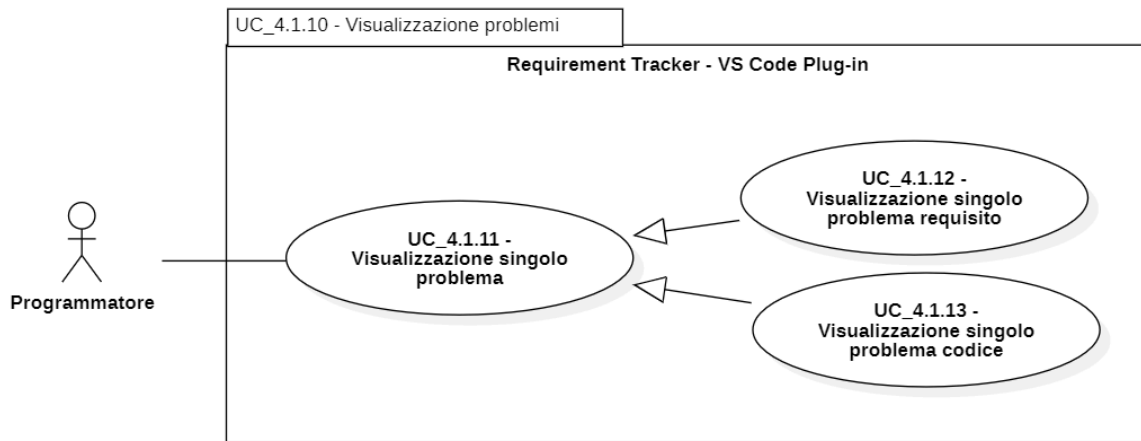


Figure 10: UC_4.1.10 - Visualizzazione dei problemi

Attori: Programmatore.

Precondizioni:

- I problemi relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_2].
- Il menu ad albero relativo alla visualizzazione in dettaglio dei risultati è stato aperto [UC_4.1.1]

Postcondizioni:

- Il sistema mostra un elenco strutturato dei problemi relativi al requisito [UC_4.1.12] e al codice [UC_4.1.13].

Scenario principale:

1. Il sistema raccoglie i problemi generati e li visualizza in forma di elenco.

3.24 UC_5 - Filtraggio dei requisiti

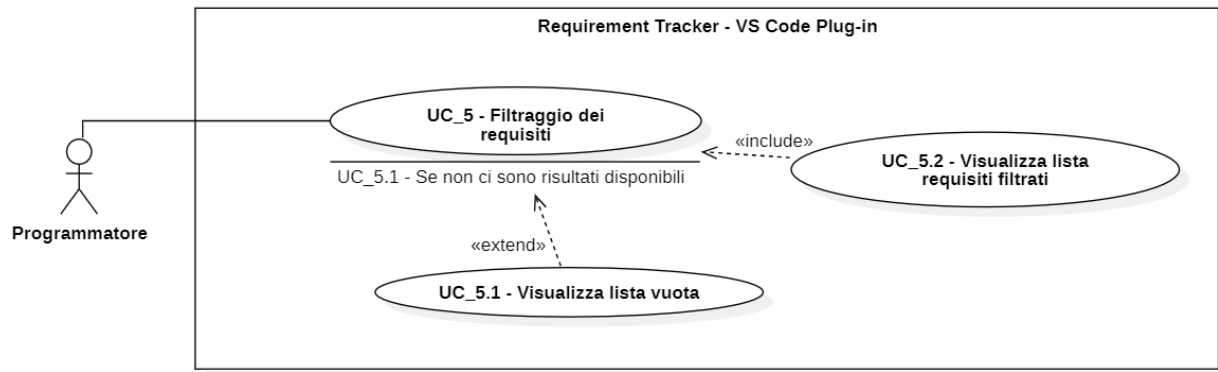


Figure 11: UC_5 - Filtraggio dei requisiti

Attori: Programmatore.

Precondizioni:

- I requisiti importati sono visualizzati [UC_9].

Postcondizioni:

- I risultati vengono filtrati in base al campo *ID*.

Scenario principale:

1. L'utente inserisce l'*ID* del requisito 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 [UC_5.2].

Estensioni:

- **UC_5.1 - Visualizzazione lista vuota:** Se nessun requisito soddisfa i criteri, il sistema visualizza una lista vuota e consente di modificare il filtro.

3.25 UC_5.1 - Visualizzazione lista vuota

Attori: Programmatore.

Precondizioni:

- I criteri di filtro inseriti nel filtraggio [UC_5] non corrispondono a nessun requisito.

Postcondizioni:

- Il sistema visualizza una lista di requisiti vuota.

Scenario principale:

1. L'utente applica un filtro utilizzando la barra di ricerca.
2. Il sistema verifica i risultati e non trova corrispondenze.
3. Viene visualizzata una lista vuota.
4. L'utente ha la possibilità di modificare il filtro o di rimuoverlo completamente, tornando alla visualizzazione di tutti i requisiti [UC_9].

3.26 UC_5.2 - Visualizzazione lista requisiti filtrati

Attori: Programmatore.

Precondizioni:

- I requisiti sono stati importati e visualizzati.
- I criteri di filtro inseriti nel filtraggio [UC_5] generano una lista di corrispondenze.

Postcondizioni:

- Il sistema mostra la lista dei requisiti che soddisfano i criteri di filtro.

Scenario principale:

1. Il sistema applica i filtri inseriti dall'utente.
2. La lista dei requisiti viene aggiornata per mostrare solo quelli filtrati.

3.27 UC_6 - Ripetizione analisi singolo requisito

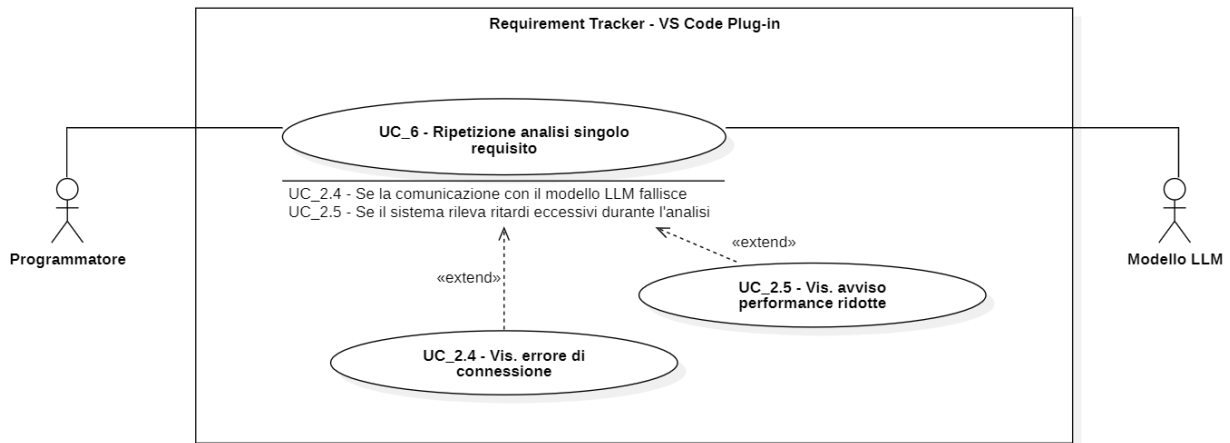


Figure 12: UC_6 - Ripetizione dell'analisi di un singolo requisito

Attori: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_9].
- I risultati dei requisiti sono stati generati [UC_2].
- È stata selezionata la funzione di ripetizione dell'analisi [UC_4.1.15] visualizzata accanto al menu di visualizzazione dettaglio risultato [UC_4.1.1]

Postcondizioni:

- Viene fornita una nuova valutazione per il requisito selezionato.

Scenario principale:

1. L'utente apre la visualizzazione del singolo requisito.
2. L'utente clicca sull'icona relativa alla funzione di ripetizione analisi [UC_4.1.15].
3. Il sistema invia il requisito ed il relativo codice al modello LLM per una nuova analisi, analogamente ad [UC_2].
4. I risultati aggiornati vengono registrati e visualizzati per il requisito selezionato [UC_4.1].

Estensioni:

- **UC_2.4 - 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_2.5 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.

3.28 UC_7 - Tracciamento dei requisiti nel codice

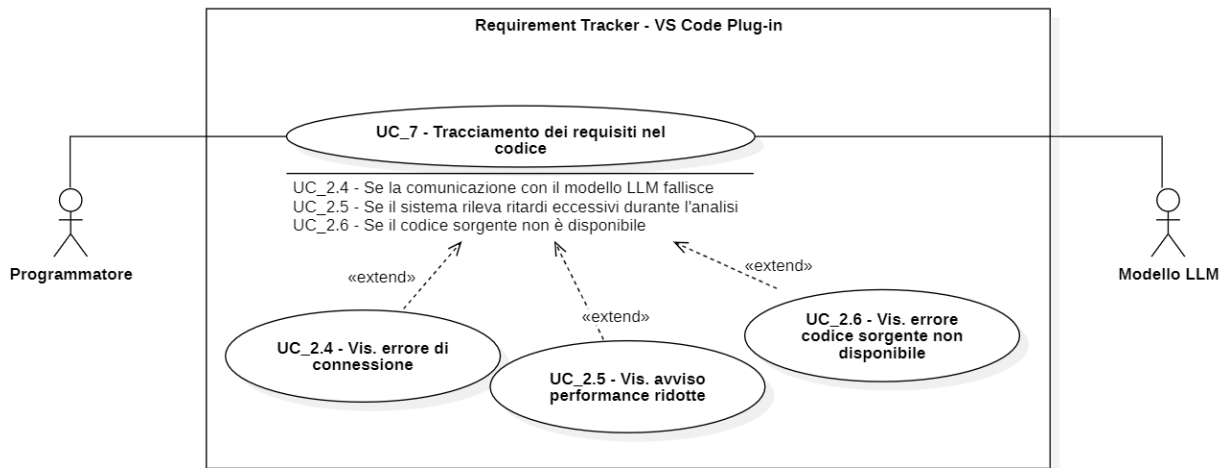


Figure 13: UC_7 - Funzione di tracciamento dei requisiti

Attori: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- I requisiti sono stati importati senza informazioni di tracciamento [UC_1.3].
- 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_9.1.5].

Scenario principale:

1. L'utente, notando l'assenza del mapping, 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 implementative.
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_2.4 - 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_2.5 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.
- **UC_2.6 - Visualizzazione errore codice non disponibile:** Se il progetto non contiene il file sorgente o non è configurato correttamente.

3.29 UC_8 - Configurazione dei path da ignorare



Figure 14: UC_8 - Configurazione dei path da ignorare

Attori: Programmatore.

Precondizioni:

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

Postcondizioni:

- I path specificati nel file .ignore vengono esclusi dall'analisi [UC_2] e dal tracciamento dei requisiti [UC_7].

Scenario principale:

1. L'utente crea o modifica un file .ignore nel progetto.
2. L'utente inserisce nel file .ignore i path o pattern relativi ai file o directory da escludere.
3. Il sistema rileva automaticamente le modifiche apportate al file .ignore.
4. Durante la l'analisi [UC_2] ed il tracciamento [UC_7], il sistema esclude i path specificati nel file .ignore.
5. L'utente avvia l'analisi o il tracciamento e i path ignorati non vengono considerati.

Estensioni:

- **UC_8.1 - Visualizzazione errore path non valido:** Se il path specificato non è valido, il sistema notifica l'utente e ignora l'entry errata mantenendo valide le altre.

3.30 UC_8.1 - Visualizzazione errore path non valido

Attori: Programmatore.

Precondizioni:

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

Postcondizioni:

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

Scenario principale:

1. L'utente modifica il file .ignore 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.

3.31 UC_9 - Visualizzazione dei requisiti

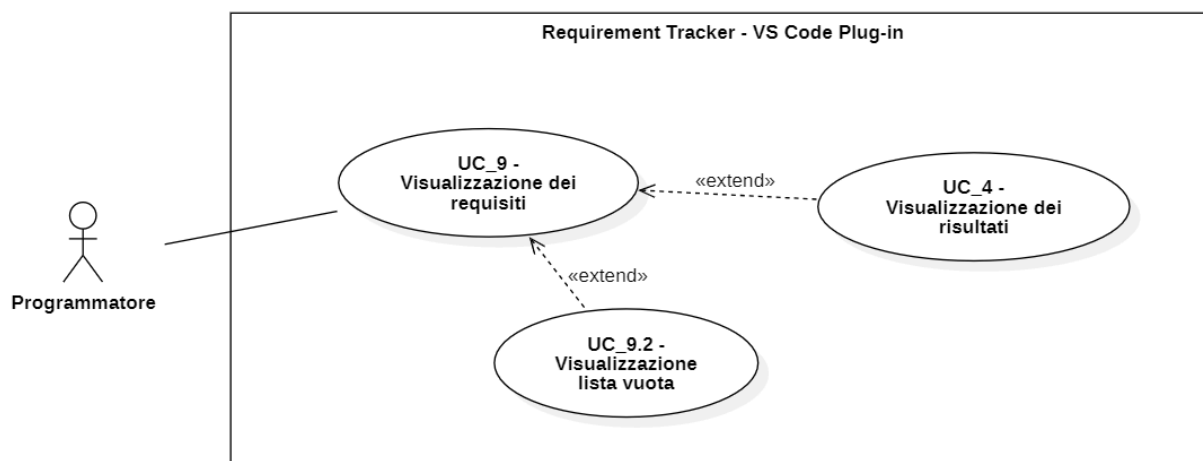


Figure 15: UC_9 - Visualizzazione dei requisiti

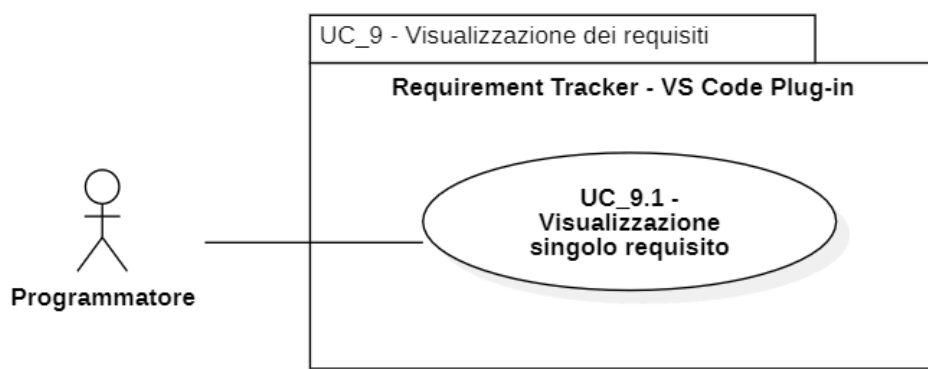


Figure 16: UC_9 - Deettaglio sulla visualizzazione dei requisiti

Attori: Programmatore.

Precondizioni:

- I requisiti sono stati importati [UC_1] e sono disponibili per la visualizzazione.

Postcondizioni:

- I requisiti vengono visualizzati in una lista ad albero, in cui ciascun requisito mostra le informazioni di base (e, se disponibili, il mapping e i risultati dell'analisi).

Scenario principale:

1. Il sistema visualizza una lista dei requisiti.
2. L'utente può selezionare un requisito per visualizzarne il dettaglio [UC_9.1.1].

Estensioni:

- **UC_9.2 - Visualizzazione lista vuota:** Se non sono presenti requisiti da visualizzare, il sistema visualizza una lista vuota.

3.32 UC_9.1 - Visualizzazione singolo requisito

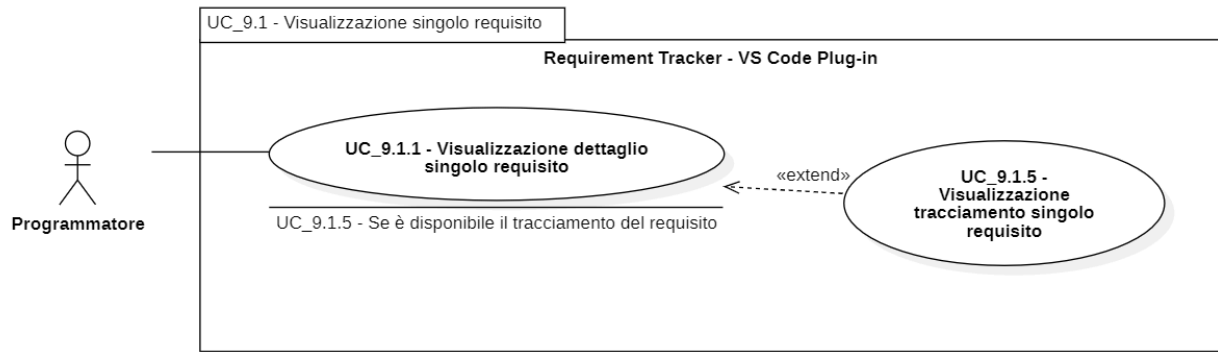


Figure 17: UC_4.1.6 - Visualizzazione di un singolo requisito

Attori: Programmatore.

Precondizioni:

- Un requisito è stato selezionato dalla lista [UC_9].

Postcondizioni:

- Il sistema mostra la voce “Requisito” che, se premuto, permette la visualizzazione in dettaglio di tutte le informazioni relative al requisito [UC_9.1.1].

Scenario principale:

1. L'utente seleziona un requisito dalla lista dei requisiti visualizzati in [UC_9].
2. Il sistema apre un sottomenù contenente la voce “Requisito” che, se premuto, mostra i dettagli del requisito selezionato [UC_9.1.1].

3.33 UC_9.1.1 - Visualizzazione dettaglio singolo requisito

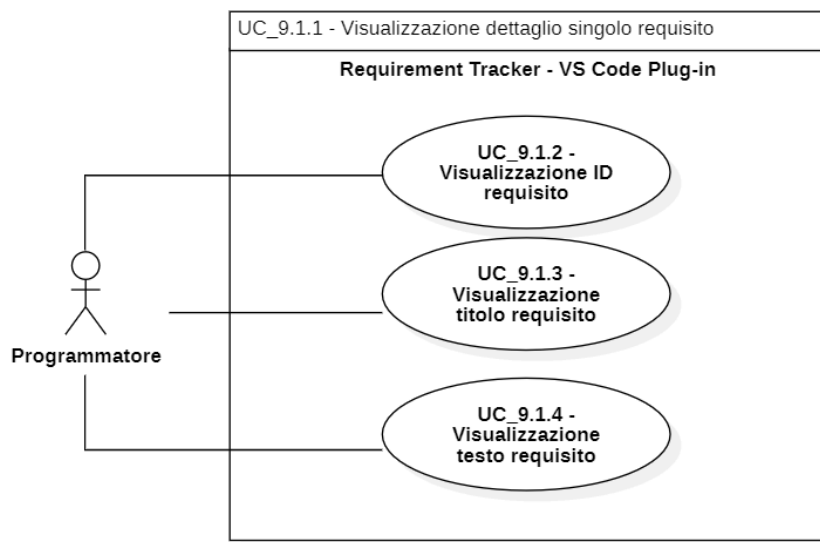


Figure 18: UC_9.1.1 - Visualizzazione in dettaglio di un singolo requisito

Attori: Programmatore.

Precondizioni:

- L'utente ha premuto sulla voce “Requisito”

Postcondizioni:

- Il sistema mostra il dettaglio completo del requisito, includendo:
 - Identificativo del requisito [UC_9.1.2]
 - Titolo del requisito [UC_9.1.3]
 - Testo descrittivo del requisito [UC_9.1.4]
 - (Opzionale) Informazioni di tracciamento [UC_9.1.5]

Scenario principale:

1. L'utente preme sulla voce "Requisito"
2. Il sistema espande il sottomenu di dettaglio, visualizzando tutte le informazioni relative al requisito.
3. Se sono disponibili dati di tracciamento, il sistema visualizza anche il dettaglio del tracciamento [UC_9.1.5].

3.34 UC_9.1.2 - Visualizzazione ID requisito

Attori: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_9.1.1].

Postcondizioni:

- Il sistema mostra il campo "ID" del requisito.

Scenario principale:

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

3.35 UC_9.1.3 - Visualizzazione titolo requisito

Attori: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_9.1.1].

Postcondizioni:

- Il sistema mostra il campo "Titolo" del requisito.

Scenario principale:

1. Il sistema visualizza il titolo del requisito.

3.36 UC_9.1.4 - Visualizzazione testo requisito

Attori: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_9.1.1].

Postcondizioni:

- Il sistema mostra il campo "Testo" completo del requisito.

Scenario principale:

1. Il sistema visualizza il testo descrittivo del requisito.

3.37 UC_9.1.5 - Visualizzazione tracciamento singolo requisito

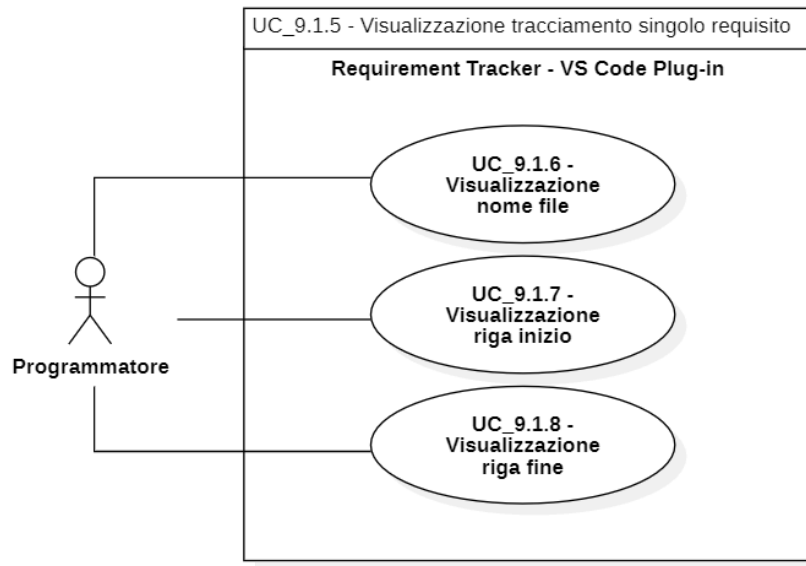


Figure 19: UC_9.1.5 - Visualizzazione delle informazioni di tracciamento di un requisito

Attori: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_9.1.1].
- Il requisito selezionato dispone di informazioni di tracciamento, importate da [UC_1.4] oppure generate da [UC_7].

Postcondizioni:

- Il sistema visualizza una menu di tracciamento espandibile che comprende i dettagli del tracciamento.

Scenario principale:

1. Nella vista di dettaglio del requisito [UC_9.1.1], 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_9.1.6], [UC_9.1.7] e [UC_9.1.8].

3.38 UC_9.1.6 - Visualizzazione nome file

Attori: Programmatore.

Precondizioni:

- La sezione “Tracciamento” del requisito è disponibile ed espansa [UC_9.1.5].

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.

3.39 UC_9.1.7 - Visualizzazione riga inizio

Attori: Programmatore.

Precondizioni:

- La sezione “Tracciamento” del requisito è disponibile ed espansa [UC_9.1.5].

Postcondizioni:

- Il sistema mostra il campo “Riga Inizio” dell’intervallo di codice nel file indicato [UC_9.1.6] relativo all’implementazione del requisito.

Scenario principale:

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

3.40 UC_9.1.8 - Visualizzazione riga fine

Attori: Programmatore.

Precondizioni:

- La sezione “Tracciamento” del requisito è disponibile ed espansa [UC_9.1.5].

Postcondizioni:

- Il sistema mostra il campo “Riga Fine” dell’intervallo di codice nel file indicato [UC_9.1.6] relativo all’implementazione del requisito.

Scenario principale:

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

3.41 UC_9.2 - Visualizzazione lista vuota

Attori: Programmatore.

Precondizioni:

- Non sono presenti requisiti da visualizzare.

Postcondizioni:

- Il sistema visualizza una lista vuota.

Scenario principale:

1. Il sistema rileva l’assenza di dati e visualizza una lista vuota.

4 Elenco delle immagini

- Figure 1: Panoramica delle funzionalità principali del plugin.
- Figure 2: UC_1 - Importazione dei requisiti da file
- Figure 3: UC_2 - Analisi dei requisiti e dell'implementazione
- Figure 4: UC_3 - Analisi dei requisiti e dell'implementazione
- Figure 5: UC_4 - Visualizzazione dei risultati
- Figure 6: UC_4 - Diagramma di dettaglio sulla visualizzazione dei risultati
- Figure 7: UC_4.1 - Visualizzazione di un singolo risultato
- Figure 8: UC_4.1 - Visualizzazione di un singolo risultato
- Figure 9: UC_4.1.6 - Visualizzazione dei suggerimenti
- Figure 10: UC_4.1.10 - Visualizzazione dei problemi
- Figure 11: UC_5 - Filtraggio dei requisiti
- Figure 12: UC_6 - Ripetizione dell'analisi di un singolo requisito
- Figure 13: UC_7 - Funzione di tracciamento dei requisiti
- Figure 14: UC_8 - Configurazione dei path da ignorare
- Figure 15: UC_9 - Visualizzazione dei requisiti
- Figure 16: UC_9 - Deettaglio sulla visualizzazione dei requisiti
- Figure 17: UC_4.1.6 - Visualizzazione di un singolo requisito
- Figure 18: UC_9.1.1 - Visualizzazione in dettaglio di un singolo requisito
- Figure 19: UC_9.1.5 - Visualizzazione delle informazioni di tracciamento di un requisito

5 Elenco delle tabelle