



Analisi dei Requisiti

Versione: 1.2.2 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
1.2.2	28/04/2025	Marco Perazzolo	Aggiunti UC_3, UC_20, UC_21 e modello di embedding come attore secondario	
1.2.1	24/04/2025	Marco Perazzolo	Aggiunti UC_12.1, UC_12.2 e UC di visualizzazione icone	Luca Parise
1.2.0	15/04/2025	Marco Perazzolo	Modifiche agli UC post incontro con Bluewind	Luca Parise
1.1.0	05/04/2025	Malik Giafar Mohamed, Luca Parise, Marco Perazzolo	Modifiche generali agli use cases e aggiornamento del relativo tracciamento	Maria Fuensanta Trigueros Hernandez
1.0.1	27/03/2025	Marco Perazzolo	Correzione di casi d'uso e requisiti post-incontro RTB	Maria Fuensanta Trigueros Hernandez
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

Versione	Data	Autore	Descrizione	Verifica
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
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	6
1.1	Scopo del documento	6
1.2	Scopo del prodotto	6
1.3	Glossario	6
2	Descrizione	7
2.1	Obiettivi del prodotto	7

2.2	Funzionalità del prodotto	7
2.3	Utenti e caratteristiche	8
3	Use Case	8
3.1	Obiettivi	8
3.2	Attori	8
3.3	UC_1 - Importazione dei requisiti	9
3.4	UC_1.1 - Selezione del file	10
3.5	UC_1.2 - Visualizzazione errore file importazione	10
3.6	UC_2 - Importazione dei requisiti con tracciamento	11
3.7	UC_3 - Importazione dei requisiti con tracciamento e risultati	12
3.8	UC_4 - Analisi dei requisiti e dell'implementazione	13
3.9	UC_4.1 - Visualizzazione errore tracciamento non valido o mancante	14
3.10	UC_4.2 - Visualizzazione errore di connessione	15
3.11	UC_4.3 - Visualizzazione avviso performance ridotte	15
3.12	UC_4.4 - Visualizzazione errore codice sorgente non disponibile	16
3.13	UC_5 - Esportazione del tracciamento e dei risultati	16
3.14	UC_5.1 - Visualizzazione errore di salvataggio	17
3.15	UC_6 - Visualizzazione elenco requisiti	19
3.16	UC_6.1 - Visualizzazione singolo requisito in elenco	20
3.17	UC_6.1.1 - Visualizzazione ID requisito	21
3.18	UC_6.1.2 - Visualizzazione indicatore di stato requisito	21
3.19	UC_6.1.2.1 - Visualizzazione stato tracciato	22
3.20	UC_6.1.2.2 - Visualizzazione stato conforme	22
3.21	UC_6.1.2.3 - Visualizzazione stato non conforme	22
3.22	UC_6.1.2.4 - Visualizzazione stato approvato	23
3.23	UC_6.1.3 - Visualizzazione icona tracciamento singolo requisito	23
3.24	UC_6.1.4 - Visualizzazione icona analisi singolo requisito	23
3.25	UC_6.1.5 - Visualizzazione icona approvazione requisito	24
3.26	UC_6.1.6 - Visualizzazione icona marcatura non conformità	24
3.27	UC_7 - Ordinamento dei requisiti	25
3.28	UC_7.1 - Ordinamento per ID crescente	26
3.29	UC_7.2 - Ordinamento per stato conforme	26
3.30	UC_7.3 - Ordinamento per stato non conforme	26
3.31	UC_8 - Visualizzazione dettaglio requisito	27
3.32	UC_8.1 - Visualizzazione testo requisito	28
3.33	UC_8.2 - Visualizzazione tracciamento requisito	29
3.34	UC_8.2.1 - Visualizzazione percorso file	30
3.35	UC_8.2.2 - Visualizzazione riga inizio	30
3.36	UC_8.2.3 - Visualizzazione riga fine	30
3.37	UC_8.2.4 - Visualizzazione icona modifica tracciamento	31
3.38	UC_9 - Visualizzazione risultati requisito	32
3.39	UC_9.1 - Visualizzazione risultato globale	33
3.40	UC_9.1.1 - Visualizzazione risultato conforme	34
3.41	UC_9.1.2 - Visualizzazione risultato non conforme	34
3.42	UC_9.2 - Visualizzazione valutazione requisito	34

3.43	UC_9.3 - Visualizzazione valutazione codice	35
3.44	UC_9.4 - Visualizzazione lista suggerimenti	35
3.45	UC_9.4.1 - Visualizzazione singolo suggerimento	36
3.46	UC_9.4.1.1 - Visualizzazione numero suggerimento	37
3.47	UC_9.4.1.2 - Visualizzazione testo suggerimento	37
3.48	UC_9.5 - Visualizzazione lista problemi	38
3.49	UC_9.5.1 - Visualizzazione singolo problema	39
3.50	UC_9.5.1.1 - Visualizzazione numero problema	39
3.51	UC_9.5.1.2 - Visualizzazione testo problema	40
3.52	UC_10 - Filtraggio dei requisiti	40
3.53	UC_10.1 - Filtraggio per ID	41
3.54	UC_10.2 - Filtraggio per descrizione	41
3.55	UC_10.3 - Filtraggio per file tracciamento	42
3.56	UC_11 - Analisi di un singolo requisito	42
3.57	UC_12 - Gestione di un requisito	43
3.58	UC_12.1 - Approvazione di un requisito	44
3.59	UC_12.2 - Marcatura di un requisito come non conforme	45
3.60	UC_13 - Tracciamento dei requisiti nel codice	46
3.61	UC_14 - Tracciamento di un singolo requisito	48
3.62	UC_15 - Impostazione manuale del tracciamento	49
3.63	UC_15.1 - Inserimento percorso file	50
3.64	UC_15.2 - Inserimento riga inizio	51
3.65	UC_15.3 - Inserimento riga fine	51
3.66	UC_16 - Configurazione dei path da ignorare	53
3.67	UC_16.1 - Visualizzazione errore path non valido	54
3.68	UC_17 - Configurazione del modello di AI	54
3.69	UC_17.1 - Configurazione del modello per il tracciamento	55
3.70	UC_17.2 - Configurazione del modello per l'analisi semantica	56
3.71	UC_17.3 - Configurazione del modello per l'analisi del codice	56
3.72	UC_18 - Configurazione dell'endpoint del server Ollama	57
3.73	UC_19 - Configurazione della soglia di conformità del codice	58
3.74	UC_19.1 - Visualizzazione errore valore soglia non valido	59
3.75	UC_20 - Visualizzazione della soglia di conformità del requisito	59
3.76	UC_21 - Visualizzazione dello stato di progressione	60
4	Requisiti	61
4.1	Introduzione	61
4.2	Requisiti Funzionali	61
4.3	Requisiti di qualità	67
4.4	Requisiti di vincolo	68
4.5	Requisiti Prestazionali	68
4.6	Tracciamento dei Requisiti	68
4.7	Riepilogo	70

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 d'uso, gli attori principali, le *precondizioni*^G, le *postcondizioni*^G, lo *scenario principale*^G e gli eventuali *scenari alternativi*^G o sottocasi d'uso. 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, fornendo suggerimenti per migliorare la qualità della loro definizione e semplificando l'identificazione delle aree di mancata o errata implementazione. In particolare, fornisce analisi automatizzate del codice mediante l'utilizzo di modelli *LLM*^G e suggerimenti per migliorare il codice e 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 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 essa.

Le funzionalità implementate nell'applicazione includono:

- Importazione del file sdei requisiti in formato *CSV*^G
- Ricerca dell'implementazione dei requisiti nel codice sorgente tramite un modello di embedding
- *Analisi semantica*^G dei requisiti tramite un modello LLM
- Analisi del codice sorgente per verificare l'implementazione dei requisiti tramite un modello LLM
- Valutazione qualitativa dei requisiti e del codice
- Visualizzazione dei risultati dell'analisi
- Filtraggio dei risultati dell'analisi
- Possibilità di eseguire l'analisi su un *requisito*^G specifico
- Possibilità di cercare l'implementazione nel codice sorgente di un requisito specifico
- Esportazione dei risultati dell'analisi in formato CSV
- Fornire suggerimenti per migliorare la qualità dei requisiti e del codice
- Evidenziare problemi o anomalie nei requisiti o nel codice sorgente
- 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 ad 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

- **Programmatore:** l'utente che utilizza il plug-in per importare, analizzare e tracciare i requisiti software all'interno del codice sorgente di un progetto.
- **Modello LLM:** il modello di intelligenza artificiale che esegue l'analisi semantica dei requisiti e verifica l'implementazione nel codice sorgente. Il modello LLM è reperibile tramite le API REST di Ollama.
- **Modello di embedding:** il modello di intelligenza artificiale che esegue la ricerca dell'implementazione dei requisiti nel codice sorgente. Il modello di embedding è reperibile tramite le API REST di Ollama.

3.3 UC_1 - Importazione dei requisiti

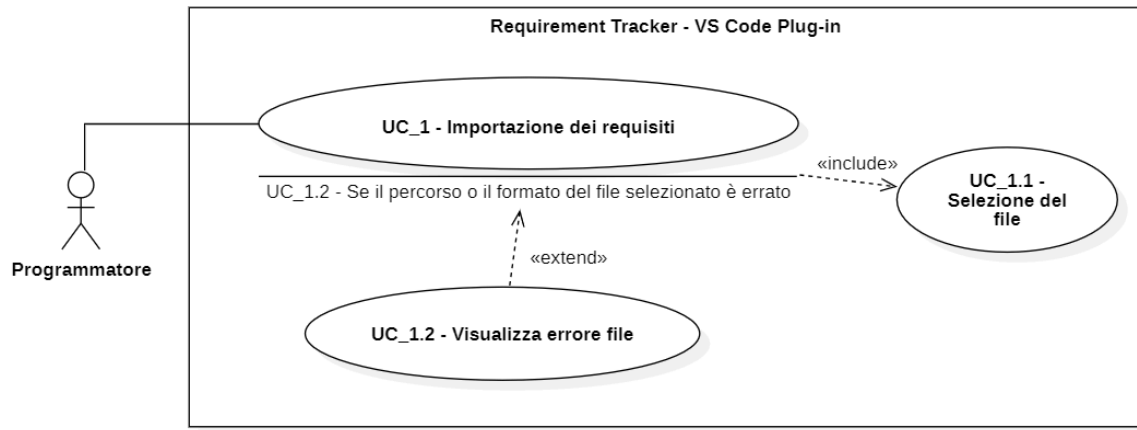


Figure 1: UC_1 - Importazione dei requisiti da file CSV

Attore primario: Programmatore.

Precondizioni:

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

Postcondizioni:

- I requisiti vengono importati e sono visualizzabili nel sistema.

Scenario principale:

1. L'utente seleziona l'opzione "Load CSV".
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 ([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.

3.6 UC_2 - Importazione dei requisiti con tracciamento

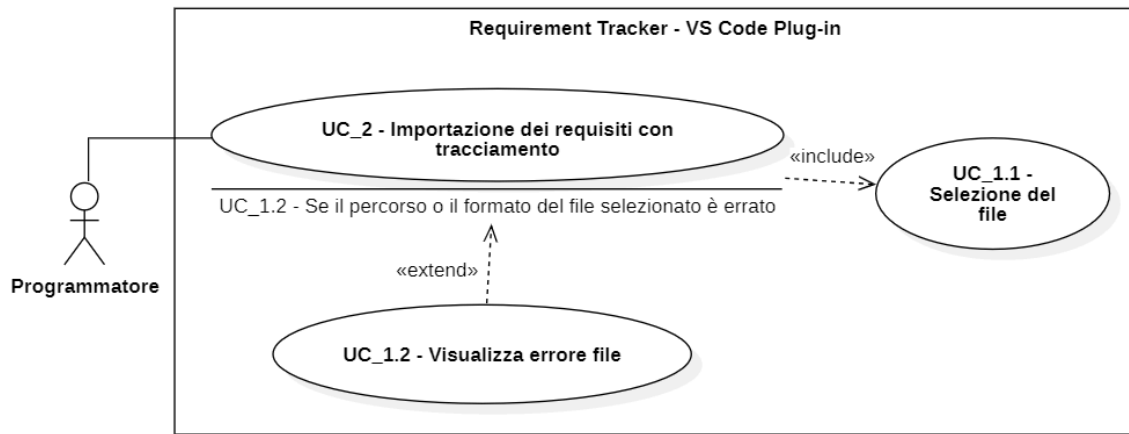


Figure 2: UC_2 - Importazione dei requisiti e del tracciamento da file CSV

Attore primario: Programmatore.

Precondizioni:

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

Postcondizioni:

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

Scenario principale:

1. L'utente seleziona l'opzione "Load CSV".
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 - Importazione dei requisiti con tracciamento e risultati

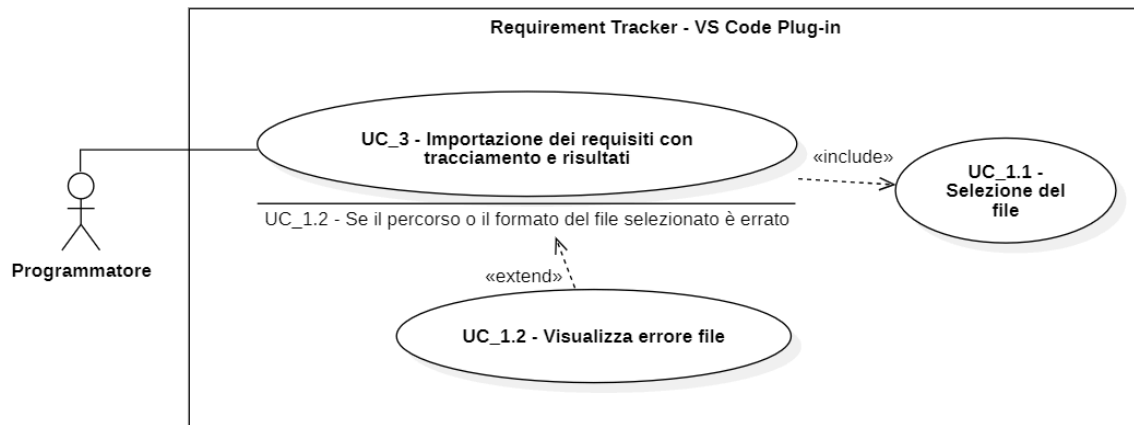


Figure 3: UC_3 - Importazione dei requisiti, del tracciamento e dei risultati da file CSV

Attore primario: Programmatore.

Precondizioni:

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

Postcondizioni:

- I requisiti, le relative informazioni di tracciamento e i risultati delle precedenti analisi sono importati e visualizzabili nel sistema.

Scenario principale:

1. L'utente seleziona l'opzione "Load CSV".
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 e risultati di ogni requisito).

5. I requisiti importati vengono mostrati in una vista strutturata [UC_6].

3.8 UC_4 - Analisi dei requisiti e dell'implementazione

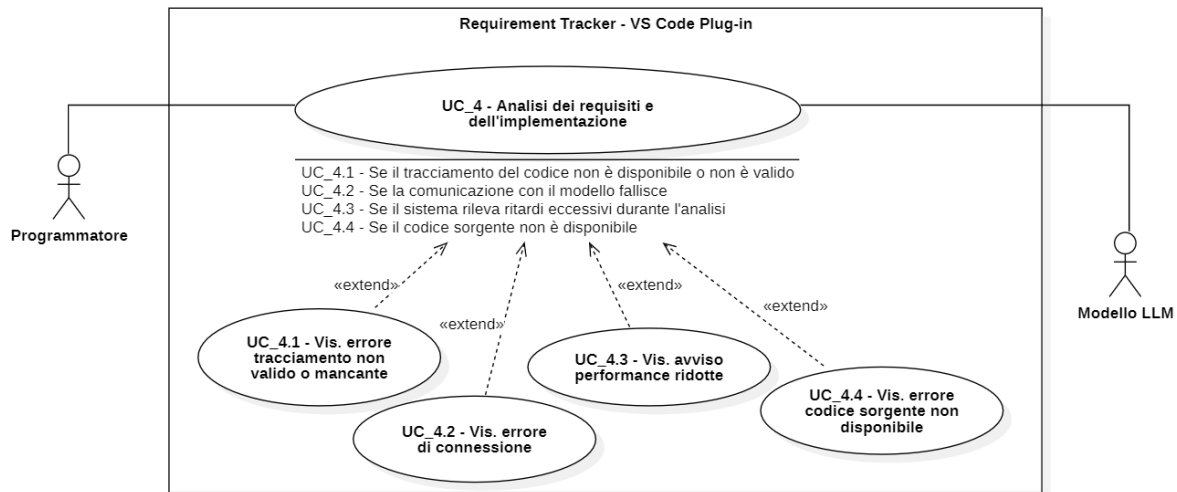


Figure 4: UC_4 - Analisi dei requisiti e della loro implementazione

Attore primario: Programmatore.

Attore secondario: Modello LLM.

Precondizioni:

- I requisiti sono stati importati [UC_1] oppure [UC_2].
- Il tracciamento di ogni requisito è disponibile, ottenuto direttamente da [UC_2] oppure dopo [UC_13] o [UC_14].
- 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 "Analyze".
2. Il sistema verifica, per ogni requisito, che siano disponibili sia la descrizione che il relativo tracciamento (da [UC_2] o [UC_13] o [UC_14]).
 - Se il requisito non ha informazioni di tracciamento, il sistema lo esclude dall'analisi.

- Se il requisito ha informazioni di tracciamento, il sistema lo include nell'analisi.
- 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_9]).

Estensioni:

- **UC_4.1 - Visualizzazione errore tracciamento non valido o mancante:** Se il *mapping*^G del codice non è disponibile o non è valido, il sistema visualizza un messaggio d'errore specifico.
- **UC_4.2 - 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_4.3 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.
- **UC_4.4 - 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.

3.9 UC_4.1 - Visualizzazione errore tracciamento non valido o mancante

Attore primario: Programmatore.

Precondizioni:

- Il sistema rileva che il mapping (tracciamento del codice) non è disponibile o non è valido.

Postcondizioni:

- Il sistema visualizza un messaggio d'errore che informa l'utente dell'assenza del tracciamento o della sua errata configurazione.

Scenario principale:

1. Il sistema verifica la presenza del mapping.
2. Se il mapping risulta mancante o non valido, il sistema mostra un messaggio d'errore specifico.

3.10 UC_4.2 - Visualizzazione errore di connessione

Attore primario: Programmatore.

Precondizioni:

- Durante l'analisi [UC_4], la comunicazione con il modello di AI 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 di intelligenza artificiale (LLM o di embedding) il sistema rileva un problema di connessione.
2. Il sistema visualizza un messaggio d'errore dettagliato e consente all'utente di riprovare.

3.11 UC_4.3 - Visualizzazione avviso performance ridotte

Attore primario: Programmatore.

Precondizioni:

- Durante l'analisi [UC_4], il sistema rileva tempi di risposta eccessivi dal modello di AI (LLM o di embedding).

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 di AI.
2. Se il tempo supera una soglia prestabilita, il sistema visualizza un avviso informativo.

3.12 UC_4.4 - 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 l'analisi dei requisiti [UC_4].
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.

3.13 UC_5 - Esportazione del tracciamento e dei risultati

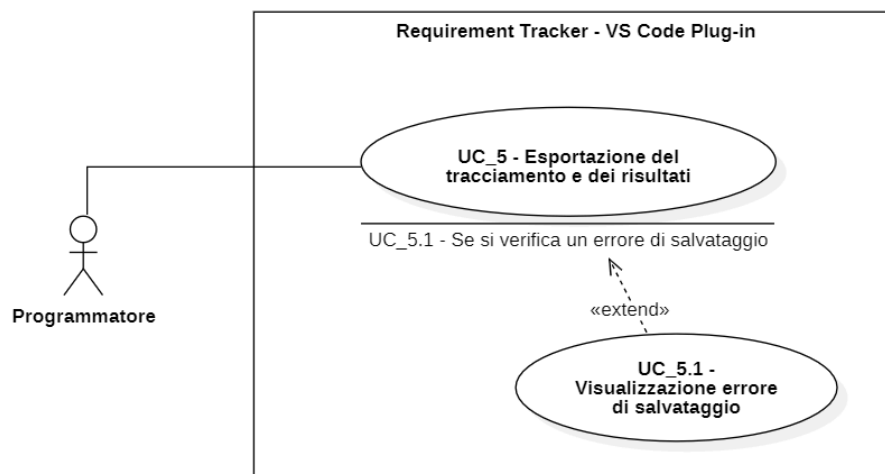


Figure 5: UC_5 - Esportazione del tracciamento su file CSV

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono disponibili, importati da [UC_1] oppure [UC_2].

Postcondizioni:

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

Scenario principale:

1. L'utente seleziona la voce "Export CSV".
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:
 - Tutti i requisiti importati (ID e testo)
 - Lo stato e le informazioni di tracciamento (file e intervallo righe) per ogni requisito, se disponibili
 - I risultati dell'analisi (valutazione e suggerimenti) per ogni requisito, se disponibili.

Estensioni:

- **UC_5.1 - 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.

Nota: L'esportazione è univoca e automatica: il sistema adatta i dati esportati in base alle informazioni disponibili per ciascun requisito.

- Se un requisito ha solo ID e testo, vengono esportati solo questi campi.

- Se un requisito ha anche informazioni di tracciamento, vengono esportati ID, testo e tracciamento.

- Se un requisito ha anche i risultati di analisi, vengono esportati ID, testo, tracciamento e risultati.

3.14 UC_5.1 - 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 "Export CSV".
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.

3.15 UC_6 - Visualizzazione elenco requisiti

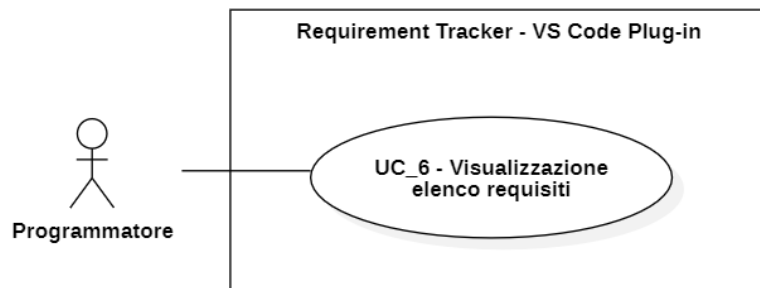


Figure 6: UC_6 - Visualizzazione dell'elenco dei requisiti

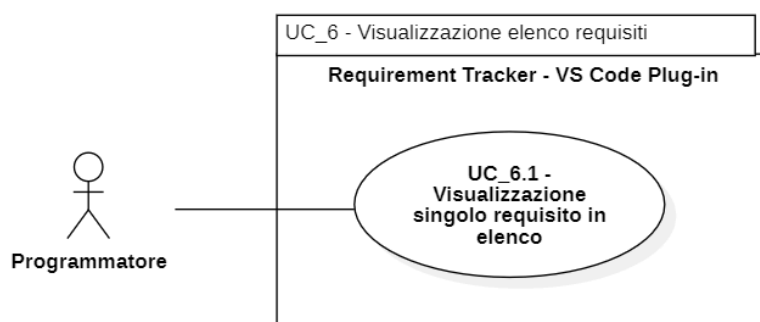


Figure 7: UC_6 - Dettaglio della visualizzazione elenco requisiti

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_8] oppure i risultati (se disponibili) [UC_9].

3.16 UC_6.1 - Visualizzazione singolo requisito in elenco

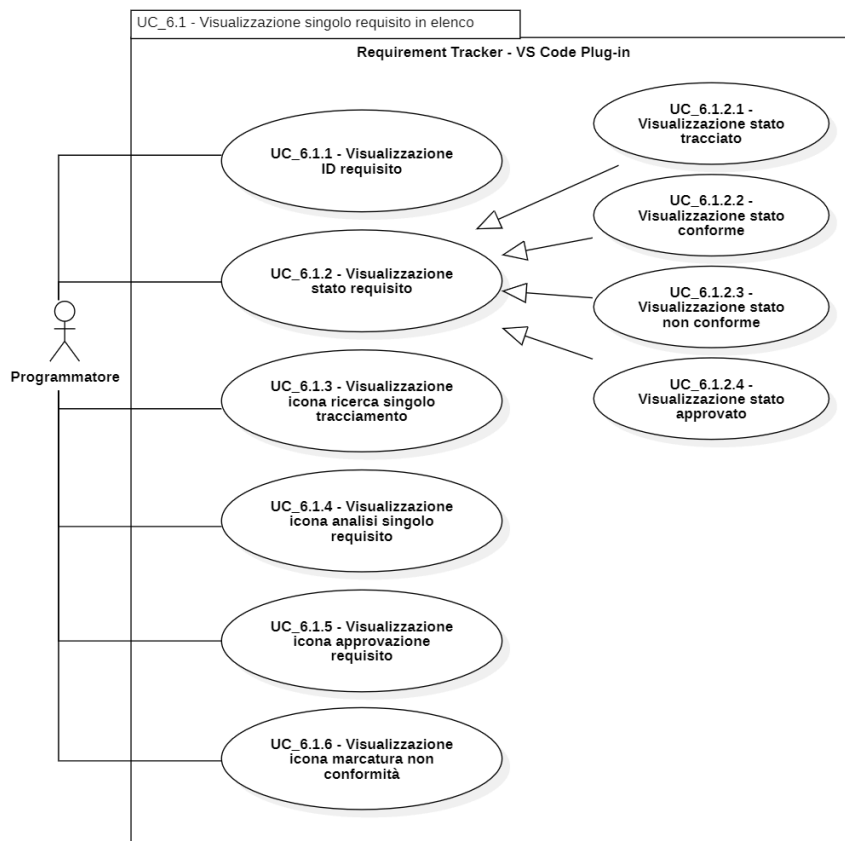


Figure 8: UC_6.1 - Visualizzazione singolo requisito in elenco

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 requisito in elenco con: ID, indicatore di stato e le icone per il tracciamento, analisi, approvazione e non conformità.

Scenario principale:

1. Il sistema visualizza il requisito nell'elenco, mostrandone l'identificativo (ID) e l'indicatore di stato. Inoltre, mostra le icone per il tracciamento, per l'analisi del singolo requisito, per l'approvazione del requisito e per la marcatura del requisito come "non conforme".

3.17 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.

3.18 UC_6.1.2 - Visualizzazione indicatore di stato 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 l'indicatore di stato del requisito.

Scenario principale:

1. Il sistema visualizza un indicatore che comunica lo stato del requisito.

Generalizzazioni

- **UC_6.1.2.1 - Visualizzazione stato tracciato:** Se il requisito è stato tracciato ma non ancora analizzato, il sistema mostra un indicatore giallo.
- **UC_6.1.2.2 - Visualizzazione stato conforme:** Se il requisito è stato analizzato ed il risultato globale è positivo, il sistema mostra un indicatore verde.
- **UC_6.1.2.3 - Visualizzazione stato non conforme:** Se il requisito è stato analizzato ed il risultato globale è negativo, il sistema mostra un indicatore rosso.

- **UC_6.1.2.4 - Visualizzazione stato approvato:** Se il requisito è stato analizzato ed il risultato è stato approvato dall'utente, il sistema mostra un indicatore blu.

3.19 UC_6.1.2.1 - Visualizzazione stato tracciato

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'indicatore di stato del requisito come "tracciato".

Scenario principale:

1. Il sistema visualizza un indicatore giallo per il requisito, che indica che è stato tracciato ma non ancora analizzato.

3.20 UC_6.1.2.2 - Visualizzazione stato conforme

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'indicatore di stato del requisito come "conforme".

Scenario principale:

1. Il sistema visualizza un indicatore verde per il requisito, che indica che è stato analizzato e il risultato globale è positivo.

3.21 UC_6.1.2.3 - Visualizzazione stato non conforme

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'indicatore di stato del requisito come "non conforme".

Scenario principale:

1. Il sistema visualizza un indicatore rosso per il requisito, che indica che è stato analizzato e il risultato globale è negativo.

3.22 UC_6.1.2.4 - Visualizzazione stato approvato

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'indicatore di stato del requisito come "approvato".

Scenario principale:

1. Il sistema visualizza un indicatore blu per il requisito, che indica che è stato analizzato e il risultato è stato approvato dall'utente.

3.23 UC_6.1.3 - Visualizzazione icona tracciamento singolo 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 l'icona di una lente di ingrandimento, che permette di tracciare un singolo requisito.

Scenario principale:

1. Il sistema visualizza l'icona di tracciamento requisito che, se premuto, avvia la ricerca dell'implementazione del singolo requisito nel codice[UC_14].

3.24 UC_6.1.4 - Visualizzazione icona analisi singolo 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 l'icona di un triangolo, che permette di eseguire l'analisi del singolo requisito [UC_11].

Scenario principale:

1. Il sistema visualizza l'icona di ripetizione analisi requisito che, se premuto, avvia l'analisi del singolo requisito [UC_11].

3.25 UC_6.1.5 - Visualizzazione icona approvazione 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 l'icona di una spunta, che permette di eseguire l'approvazione del requisito [UC_12.1].

Scenario principale:

1. Il sistema visualizza l'icona di approvazione requisito che, se premuto, imposta lo stato del requisito come approvato [UC_12.1].

3.26 UC_6.1.6 - Visualizzazione icona marcatura non conformità

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 croce, che permette di marcare il requisito come non conforme [UC_12.2].

Scenario principale:

1. Il sistema visualizza l'icona di marcatura di non conformità che, se premuto, imposta lo stato del requisito come non conforme [UC_12.2].

3.27 UC_7 - Ordinamento dei requisiti

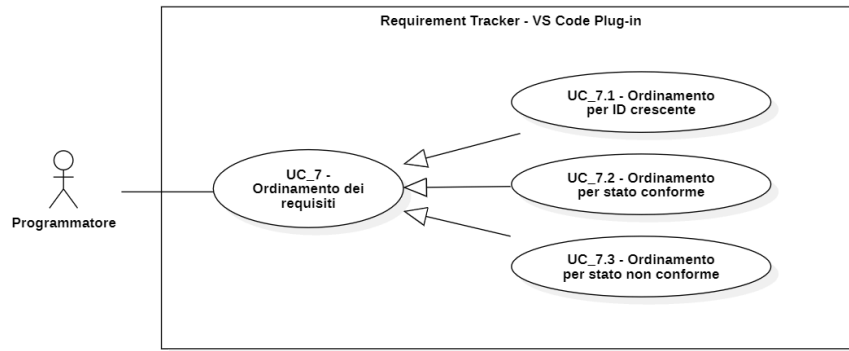


Figure 9: UC_7 - Ordinamento dell'elenco dei requisiti

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono stati importati (da [UC_1] o [UC_2]) ed è visualizzato l'elenco dei requisiti.

Postcondizioni:

- Il sistema mostra l'elenco dei requisiti ordinato in base al criterio selezionato.

Scenario principale:

1. L'utente seleziona il criterio di ordinamento desiderato (ID crescente, stato conforme, stato non conforme) per l'elenco dei requisiti.
2. Il sistema ordina i requisiti in base al criterio selezionato e aggiorna la visualizzazione dell'elenco.

Generalizzazioni:

- **UC_7.1 - Ordinamento per ID crescente:** Il sistema ordina i requisiti in ordine crescente in base all'ID.
- **UC_7.2 - Ordinamento per stato conforme:** Il sistema ordina i requisiti in base allo stato, mostrando prima quelli conformi e poi quelli non conformi.
- **UC_7.3 - Ordinamento per stato non conforme:** Il sistema ordina i requisiti in base allo stato, mostrando prima quelli non conformi e poi quelli conformi.

3.28 UC_7.1 - Ordinamento per ID crescente

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono stati importati (da [UC_1] o [UC_2]) ed è visualizzato l'elenco dei requisiti.

Postcondizioni:

- Il sistema visualizza l'elenco dei requisiti ordinato in ordine crescente in base all'ID.

Scenario principale:

1. L'utente seleziona l'opzione di ordinamento per ID crescente.
2. Il sistema ordina i requisiti in ordine crescente in base all'ID e aggiorna la visualizzazione dell'elenco.

3.29 UC_7.2 - Ordinamento per stato conforme

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono stati importati (da [UC_1] o [UC_2]) ed è visualizzato l'elenco dei requisiti.

Postcondizioni:

- Il sistema visualizza l'elenco dei requisiti ordinato in base allo stato, mostrando prima quelli conformi e poi quelli non conformi.

Scenario principale:

1. L'utente seleziona l'opzione di ordinamento per stato conforme.
2. Il sistema ordina i requisiti in base allo stato, mostrando prima quelli conformi e poi quelli non conformi e aggiorna la visualizzazione dell'elenco.

3.30 UC_7.3 - Ordinamento per stato non conforme

Attore primario: Programmatore.

Precondizioni:

- I requisiti sono stati importati (da [UC_1] o [UC_2]) ed è visualizzato l'elenco dei requisiti.

Postcondizioni:

- Il sistema visualizza l'elenco dei requisiti ordinato in base allo stato, mostrando prima quelli non conformi e poi quelli conformi.

Scenario principale:

1. L'utente seleziona l'opzione di ordinamento per stato non conforme.
2. Il sistema ordina i requisiti in base allo stato, mostrando prima quelli non conformi e poi quelli conformi e aggiorna la visualizzazione dell'elenco.

3.31 UC_8 - Visualizzazione dettaglio requisito

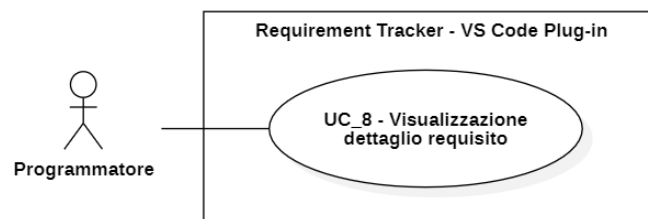


Figure 10: UC_8 - Visualizzazione dettaglio di un requisito

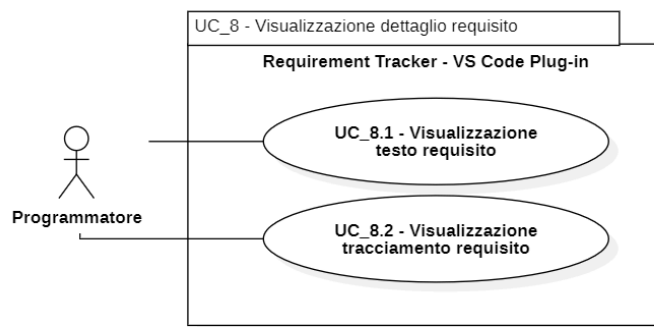


Figure 11: UC_8 - Sottocaso d'uso della visualizzazione in dettaglio di un requisito

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_8.1]
- (Opzionale) Se disponibili, le informazioni di tracciamento [UC_8.2], 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_8.1].
3. Se sono disponibili dati di tracciamento, il sistema visualizza anche il menu delle informazioni di tracciamento [UC_8.2].

3.32 UC_8.1 - Visualizzazione testo requisito

Attore primario: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_8].

Postcondizioni:

- Il sistema mostra il campo testuale descrittivo del requisito.

Scenario principale:

1. Il sistema visualizza il testo descrittivo del requisito.

3.33 UC_8.2 - Visualizzazione tracciamento requisito

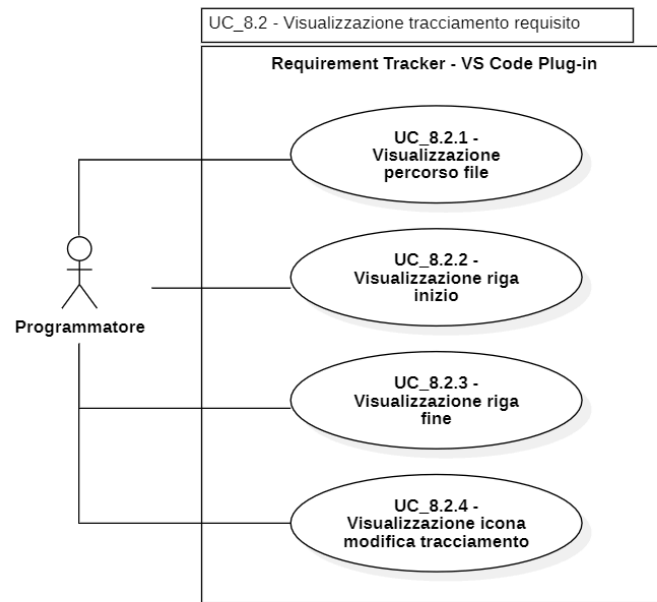


Figure 12: UC_8.2 - Visualizzazione del tracciamento di un requisito

Attore primario: Programmatore.

Precondizioni:

- Il requisito selezionato è espanso nella vista di dettaglio [UC_8].
- Il requisito selezionato dispone di informazioni di tracciamento, importate da [UC_2] oppure generate da [UC_13] o [UC_14].

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_8], il sistema verifica la presenza di dati di tracciamento.
2. Se presenti, il sistema espande la sezione “Tracciamento” mostrando:
 - Il percorso del file sorgente [UC_8.2.1].
 - La riga di inizio dell’intervallo di codice [UC_8.2.2].
 - La riga di fine dell’intervallo di codice [UC_8.2.3].

- Un'icona per la modifica del tracciamento [UC_8.2.4].

3.34 UC_8.2.1 - Visualizzazione percorso file

Attore primario: Programmatore.

Precondizioni:

- La sezione "Tracciamento" del requisito è disponibile ed espansa [UC_8.2].

Postcondizioni:

- Il sistema mostra il percorso relativo al file che contiene il codice relativo all'implementazione del requisito.

Scenario principale:

1. Il sistema visualizza il percorso, incluso il nome e l'estensione, del file associato al tracciamento del requisito.

3.35 UC_8.2.2 - Visualizzazione riga inizio

Attore primario: Programmatore.

Precondizioni:

- La sezione "Tracciamento" del requisito è disponibile ed espansa [UC_8.2].

Postcondizioni:

- Il sistema mostra il campo "Riga Inizio" dell'intervallo di codice nel file indicato ([UC_8.2.1]) relativo all'implementazione del requisito.

Scenario principale:

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

3.36 UC_8.2.3 - Visualizzazione riga fine

Attore primario: Programmatore.

Precondizioni:

- La sezione "Tracciamento" del requisito è disponibile ed espansa [UC_8.2].

Postcondizioni:

- Il sistema mostra il campo "Riga Fine" dell'intervallo di codice nel file indicato [UC_8.2.1] relativo all'implementazione del requisito.

Scenario principale:

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

3.37 UC_8.2.4 - Visualizzazione icona modifica tracciamento

Attore primario: Programmatore.

Precondizioni:

- La sezione "Tracciamento" del requisito è disponibile ed espansa [UC_8.2].

Postcondizioni:

- Il sistema mostra l'icona di modifica del tracciamento, che consente di modificare le informazioni di tracciamento del requisito.

Scenario principale:

1. Il sistema visualizza la riga di fine dell'intervallo di tracciamento del requisito.
2. Se l'icona di modifica viene premuta, il sistema apre un menu di modifica del tracciamento [UC_15].

3.38 UC_9 - Visualizzazione risultati requisito

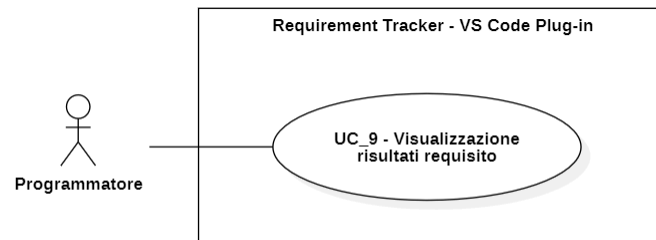


Figure 13: UC_9 - Visualizzazione dei risultati di un requisito

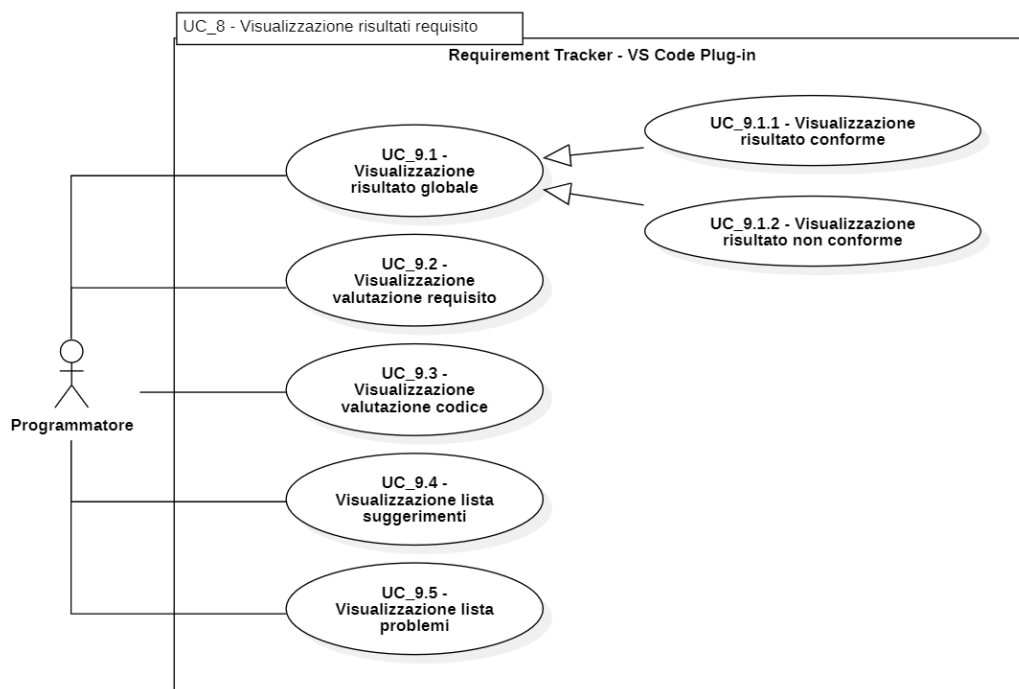


Figure 14: UC_9 - Dettaglio della visualizzazione dei risultati di un requisito

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_4].

- La sezione di visualizzazione in dettaglio del requisito selezionato è stata espansa [UC_8].

Postcondizioni:

- I risultati dell'analisi sono integrati nel menu di dettaglio requisito [UC_8] 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_9.1] Risultato globale (*passed/not passed*).
 - [UC_9.2] Valutazione del requisito in centesimi (0-100).
 - [UC_9.3] Valutazione dell'aderenza del codice in centesimi (0-100).
 - [UC_9.4] Lista dei suggerimenti generati.
 - [UC_9.5] Lista dei problemi riscontrati.

3.39 UC_9.1 - Visualizzazione risultato globale

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_4].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato espanso [UC_9].

Postcondizioni:

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

Scenario principale:

1. Il sistema mostra una valutazione globale basata sui punteggi del requisito e del codice ottenuti durante l'analisi.

Generalizzazioni:

- **UC_9.1.1 - Visualizzazione risultato conforme:** Il sistema mostra il risultato globale come "*passed*".
- **UC_9.1.2 - Visualizzazione risultato non conforme:** Il sistema mostra il risultato globale come "*not passed*".

3.40 UC_9.1.1 - Visualizzazione risultato conforme

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_4].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato espanso [UC_9].
- Il requisito è stato valutato positivamente sia dall'analisi semantica che dall'analisi dell'implementazione nel codice.

Postcondizioni:

- Il sistema mostra lo il risultato conforme del requisito.

Scenario principale:

1. Il sistema visualizza lo stato di conformità come "*Global result: passed*".

3.41 UC_9.1.2 - Visualizzazione risultato non conforme

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_4].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato espanso [UC_9].
- Il requisito è stato valutato negativamente sia dall'analisi semantica che dall'analisi dell'implementazione nel codice.

Postcondizioni:

- Il sistema mostra il risultato non conforme del requisito.

Scenario principale:

1. Il sistema visualizza lo stato di conformità come "*Global result: not passed*".

3.42 UC_9.2 - Visualizzazione valutazione requisito

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_4].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_9].

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.

3.43 UC_9.3 - Visualizzazione valutazione codice

Attore primario: Programmatore.

Precondizioni:

- I risultati dell'analisi sono stati generati [UC_4].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_9].

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.

3.44 UC_9.4 - Visualizzazione lista suggerimenti

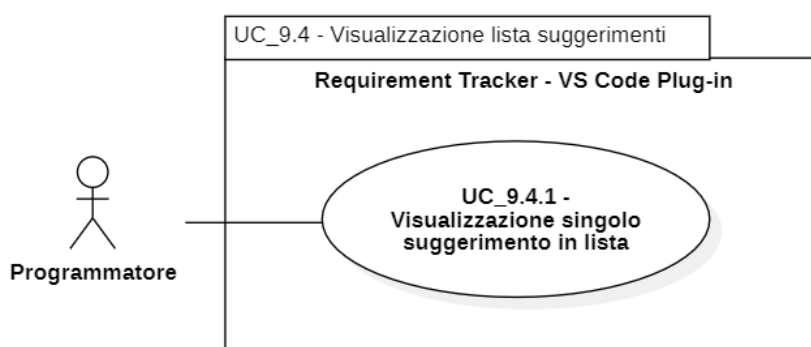


Figure 15: UC_9.4 - Visualizzazione lista dei suggerimenti

Attore primario: Programmatore.

Precondizioni:

- I suggerimenti relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_9].

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.

3.45 UC_9.4.1 - Visualizzazione singolo suggerimento

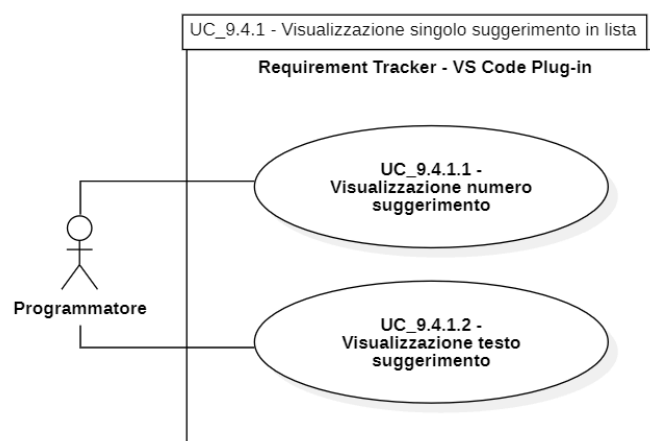


Figure 16: UC_9.4.1 - Dettaglio sulla visualizzazione del singolo suggerimento

Attore primario: Programmatore.

Precondizioni:

- I suggerimenti relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei suggerimenti è stato aperto [UC_9.4].

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.

3.46 UC_9.4.1.1 - Visualizzazione numero suggerimento**Precondizioni:**

- Il suggerimento selezionato è stato generato durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei suggerimenti è stato aperto [UC_9.4].

Postcondizioni:

- Il sistema mostra il numero progressivo del suggerimento.

Scenario principale:

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

3.47 UC_9.4.1.2 - Visualizzazione testo suggerimento

Attore primario: Programmatore.

Precondizioni:

- Il suggerimento selezionato è stato generato durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei suggerimenti è stato aperto [UC_9.4].

Postcondizioni:

- Il sistema mostra il testo del suggerimento.

Scenario principale:

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

3.48 UC_9.5 - Visualizzazione lista problemi

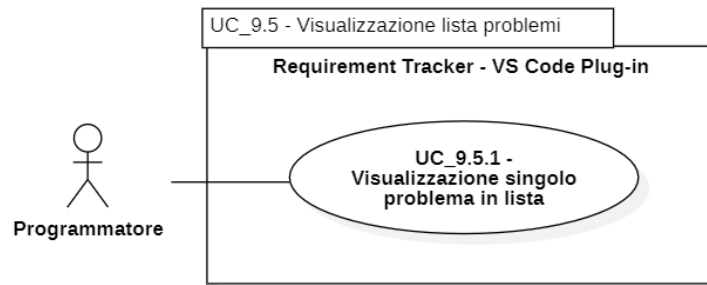


Figure 17: UC_9.5- Visualizzazione lista dei problemi

Attore primario: Programmatore.

Precondizioni:

- I problemi relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei risultati è stato aperto [UC_9].

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.

3.49 UC_9.5.1 - Visualizzazione singolo problema

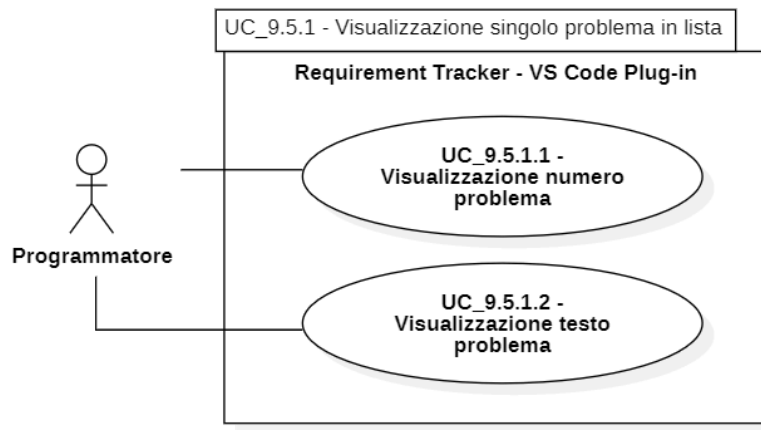


Figure 18: UC_9.5.1 - Dettaglio sulla visualizzazione del singolo problema

Attore primario: Programmatore.

Precondizioni:

- I problemi relativi al requisito e/o al codice sono stati generati durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei problemi è stato aperto [UC_9.5].

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.

3.50 UC_9.5.1.1 - Visualizzazione numero problema

Attore primario: Programmatore.

Precondizioni:

- Il problema selezionato è stato generato durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei problemi è stato aperto [UC_9.5].

Postcondizioni:

- Il sistema mostra il numero progressivo del problema.

3.51 UC_9.5.1.2 - Visualizzazione testo problema

Attore primario: Programmatore.

Precondizioni:

- Il problema è stato generato durante l'analisi [UC_4].
- Il menu ad albero relativo alla visualizzazione dei problemi è stato aperto [UC_9.5].

Postcondizioni:

- Il sistema mostra il testo del problema.

Scenario principale:

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

3.52 UC_10 - Filtraggio dei requisiti

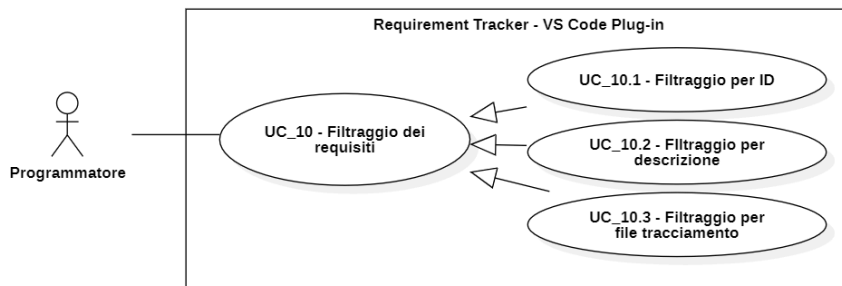


Figure 19: UC9 - Filtraggio dei requisiti

Attore primario: Programmatore.

Precondizioni:

- I requisiti importati sono visualizzati [UC_6].

Postcondizioni:

- I risultati vengono filtrati in base al campo di ricerca inserito dall'utente.
- Il sistema mostra la lista dei requisiti filtrati in base al campo di ricerca.

Scenario principale:

1. L'utente inserisce un testo nel campo di ricerca.
2. Il sistema filtra i requisiti in base al testo inserito e mostra solo quelli che corrispondono al filtro.

Generalizzazioni:

- **UC_10.1 - Filtraggio per ID:** Il sistema filtra i requisiti in base all'ID.
- **UC_10.2 - Filtraggio per descrizione:** Il sistema filtra i requisiti in base alla descrizione.
- **UC_10.3 - Filtraggio per file tracciamento:** Il sistema filtra i requisiti in base al file sorgente del tracciamento.

3.53 UC_10.1 - Filtraggio per ID

Attore primario: Programmatore.

Precondizioni:

- I requisiti importati sono visualizzati [UC_6].

Postcondizioni:

- Il sistema mostra la lista dei requisiti filtrati in base all'ID.

Scenario principale:

1. L'utente inserisce un ID nel campo di ricerca.
2. Il sistema filtra i requisiti in base all'ID inserito e mostra solo quelli che corrispondono al filtro.

3.54 UC_10.2 - Filtraggio per descrizione

Attore primario: Programmatore.

Precondizioni:

- I requisiti importati sono visualizzati [UC_6].

Postcondizioni:

- Il sistema mostra la lista dei requisiti filtrati in base alla descrizione.

Scenario principale:

1. L'utente inserisce una parte della descrizione nel campo di ricerca.
2. Il sistema filtra i requisiti in base alla descrizione inserita e mostra solo quelli che corrispondono al filtro.

3.55 UC_10.3 - Filtraggio per file tracciamento

Attore primario: Programmatore.

Precondizioni:

- I requisiti importati sono visualizzati [UC_6].

Postcondizioni:

- Il sistema mostra la lista dei requisiti filtrati in base al file sorgente del tracciamento.

Scenario principale:

1. L'utente inserisce il nome del file sorgente nel campo di ricerca.
2. Il sistema filtra i requisiti in base al file sorgente inserito e mostra solo quelli che corrispondono al filtro.

3.56 UC_11 - Analisi di un singolo requisito

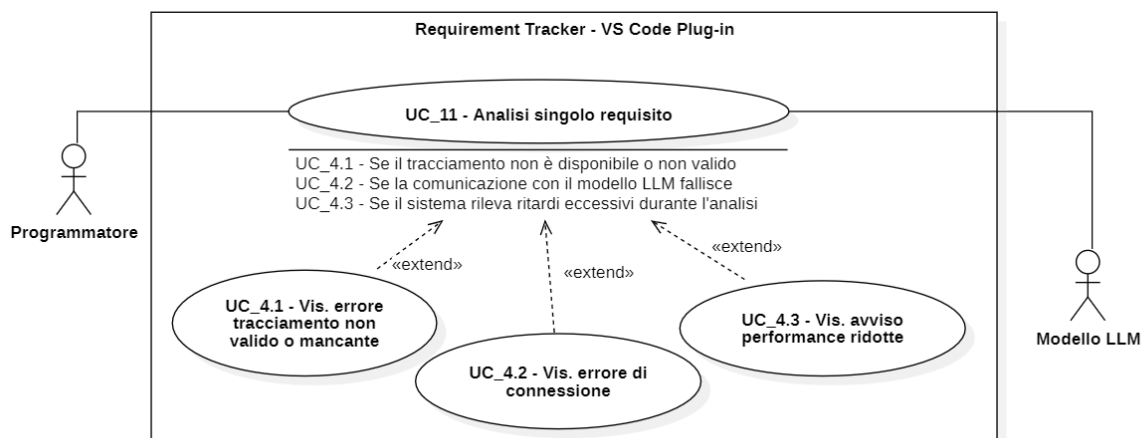


Figure 20: UC_11 - Analisi di un singolo requisito

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 [UC_6.1.4].

Postcondizioni:

- Viene fornita una nuova valutazione per il requisito selezionato.

Scenario principale:

1. L'utente clicca sull'icona relativa alla funzione di ripetizione analisi [UC_6.1.4], inserita nella riga del singolo requisito in elenco [UC_6.1].
2. Il sistema invia il requisito ed il relativo codice al modello LLM per una nuova analisi, analogamente ad [UC_4].
3. I risultati aggiornati vengono registrati e visualizzati per il requisito selezionato [UC_9].

Estensioni:

- **UC_4.1 - Visualizzazione errore tracciamento non valido o mancante:** Se il *mapping*^G del codice non è disponibile o non è valido, il sistema visualizza un messaggio d'errore specifico.
- **UC_4.2 - 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_4.3 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.

3.57 UC_12 - Gestione di un requisito

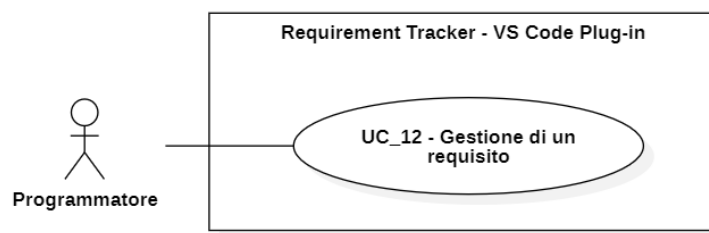


Figure 21: UC_12 - Gestione dello stato di un requisito

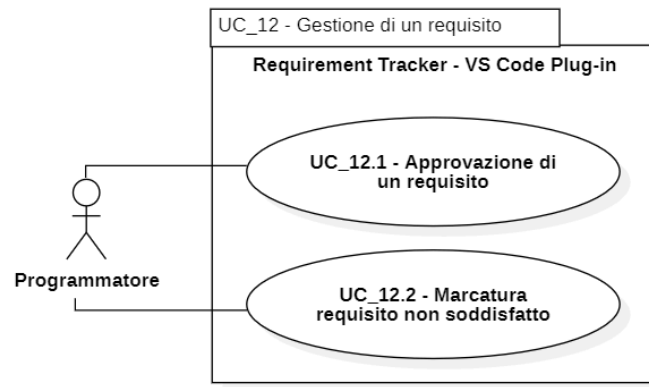


Figure 22: UC_12 - Dettaglio sulla gestione dello stato di un requisito

Attore primario: Programmatore.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- Il requisito è stato tracciato tramite [UC_13] o [UC_14].

Postcondizioni:

- Il requisito selezionato è stato approvato o marcato come non conforme.

Scenario principale:

1. L'utente seleziona l'icona di approvazione del requisito [UC_6.1.5] o l'icona di marcatura di non conformità [UC_6.1.6].
2. Il sistema aggiorna lo stato del requisito in base all'azione selezionata.
3. Il sistema mostra il requisito aggiornato nell'elenco dei requisiti [UC_6].

3.58 UC_12.1 - Approvazione di un requisito

Attore primario: Programmatore.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- Il requisito è stato tracciato tramite [UC_13] o [UC_14] o le sue informazioni di tracciamento sono state importate tramite [UC_2].

Postcondizioni:

- Il requisito selezionato è stato approvato e il suo stato è aggiornato di conseguenza.

Scenario principale:

1. L'utente seleziona l'icona di approvazione del requisito [UC_6.1.5].
2. Il sistema aggiorna lo stato del requisito come approvato.
3. Il sistema mostra il requisito aggiornato nell'elenco dei requisiti [UC_6].

3.59 UC_12.2 - Marcatura di un requisito come non conforme

Attore primario: Programmatore.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- Il requisito è stato tracciato tramite [UC_13] o [UC_14] o le sue informazioni di tracciamento sono state importate tramite [UC_2].

Postcondizioni:

- Il requisito selezionato è stato marcato come non conforme e il suo stato è aggiornato di conseguenza.

Scenario principale:

1. L'utente seleziona l'icona di marcatura di non conformità del requisito [UC_6.1.6].
2. Il sistema aggiorna lo stato del requisito come non conforme.
3. Il sistema mostra il requisito aggiornato nell'elenco dei requisiti [UC_6].

3.60 UC_13 - Tracciamento dei requisiti nel codice

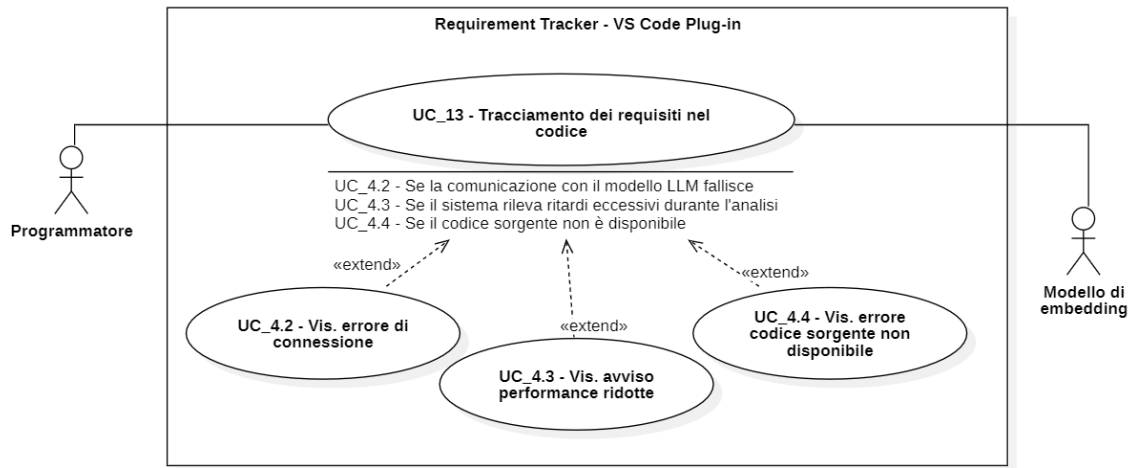


Figure 23: UC_13 - Funzione di tracciamento dei requisiti

Attore primario: Programmatore.

Attore secondario: Modello di embedding.

Precondizioni:

- I requisiti sono stati importati (da [UC_1] o [UC_2]) ed è visualizzato l'elenco dei requisiti [UC_6].
- 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 più verosimilmente lo implementa.
- Il tracciamento ottenuto viene registrato e reso disponibile nella vista dei requisiti [UC_8.2].

Scenario principale:

1. L'utente, notando l'assenza del tracciamento, seleziona l'opzione "Traceability".
2. Il sistema invia il codice ed i requisiti al modello di embedding per la mappatura.
3. Il modello analizza il codice sorgente e i requisiti, restituendo una mappatura del codice che include il nome del file e l'intervallo di righe per ogni requisito.
4. Il sistema registra il tracciamento e lo visualizza nella sezione "Tracciamento" del requisito [UC_8.2].

Estensioni:

- **UC_4.2 - Visualizzazione errore di connessione:** Se la comunicazione con il modello di embedding fallisce (es. timeout o connessione interrotta), il sistema informa l'utente e consente di riprovare.
- **UC_4.3 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.
- **UC_4.4 - Visualizzazione errore codice non disponibile:** Se il progetto non contiene il file sorgente o non è configurato correttamente.

3.61 UC_14 - Tracciamento di un singolo requisito

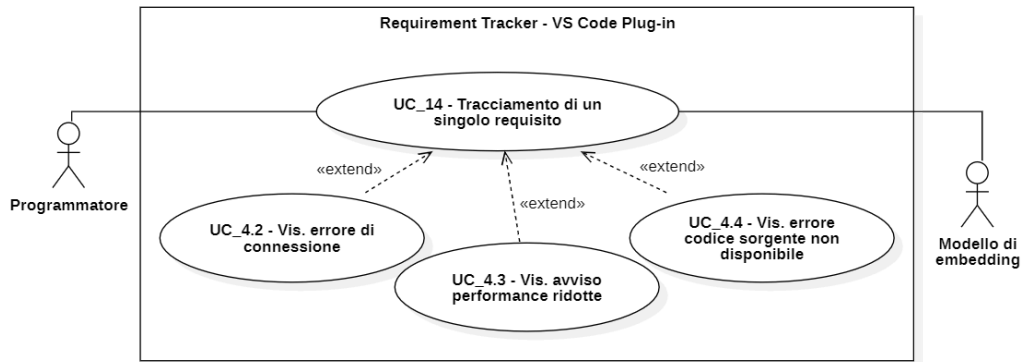


Figure 24: UC_14 - Ricerca del tracciamento nel codice di un singolo requisito

Attore primario: Programmatore.

Attore secondario: Modello di embedding.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].

Postcondizioni:

- Il requisito selezionato è stato tracciato nel codice e il suo stato è aggiornato di conseguenza.

Scenario principale:

1. L'utente seleziona l'icona di tracciamento del singolo requisito (lente di ingrandimento).
2. Il sistema invia il requisito al modello di embedding per il tracciamento.
3. Il modello restituisce il tracciamento del requisito, che include il nome del file e l'intervallo di righe.
4. Il sistema registra il tracciamento e aggiorna lo stato del requisito come tracciato.

Estensioni:

- **UC_4.2 - Visualizzazione errore di connessione:** Se la comunicazione con il modello di embedding fallisce (es. timeout o connessione interrotta), il sistema informa l'utente e consente di riprovare.
- **UC_4.3 - Visualizzazione avviso performance ridotte:** Se la risposta del modello risulta particolarmente lenta, il sistema mostra un avviso all'utente.

- **UC_4.4 - Visualizzazione errore codice non disponibile:** Se il progetto non contiene il file sorgente o non è configurato correttamente.

3.62 UC_15 - Impostazione manuale del tracciamento

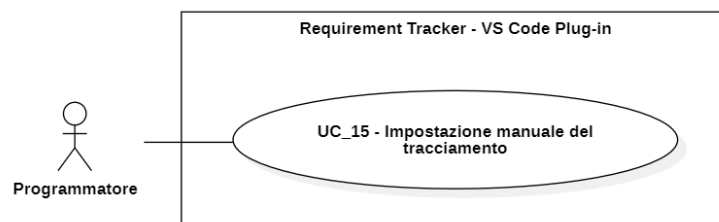


Figure 25: UC_15 - Configurazione manuale delle informazioni di tracciamento

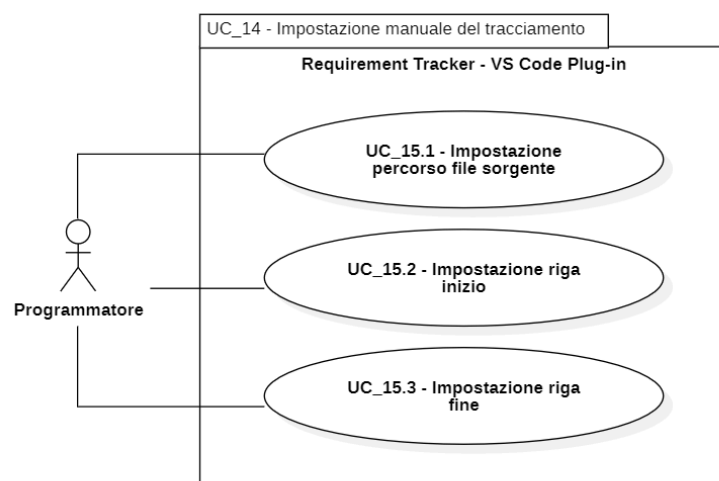


Figure 26: UC_15 - Dettaglio sulle informazioni di tracciamento modificabili

Attore primario: Programmatore.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- Il requisito è stato tracciato automaticamente ed il tracciamento è disponibile ([UC_13] o [UC_14]).

Postcondizioni:

- Le informazioni di tracciamento del requisito sono state aggiornate manualmente.
- Il sistema mostra il tracciamento aggiornato del requisito.

Scenario principale:

1. L'utente entra nel menu di visualizzazione del dettaglio del requisito [UC_8].
2. L'utente espande il menu di tracciamento del requisito [UC_8.2].
3. Il sistema mostra le informazioni di tracciamento, inclusi il percorso del file e l'intervallo di righe.
4. L'utente seleziona l'icona di modifica del tracciamento [UC_8.2.4].
5. Il sistema chiede all'utente di inserire manualmente il percorso del file [UC_15.1] e l'intervallo di righe ([UC_15.2] e [UC_15.3]).
6. L'utente inserisce le informazioni di tracciamento manualmente.
7. Il sistema aggiorna le informazioni di tracciamento del requisito e mostra il tracciamento aggiornato.

3.63 UC_15.1 - Inserimento percorso file

Attore primario: Programmatore.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- Il requisito è stato tracciato automaticamente ed il tracciamento è disponibile ([UC_13] o [UC_14]).
- Il menu di tracciamento del requisito è stato espanso [UC_8.2].
- L'interfaccia di modifica del tracciamento è stata aperta [UC_15].

Postcondizioni:

- Il sistema registra il percorso del file con quello inserito dall'utente.

Scenario principale:

1. Il sistema mostra il campo di input per il percorso del file.
2. L'utente inserisce il percorso del file sorgente che implementa il requisito, incluso il nome del file e la sua estensione.
3. Il sistema registra il percorso del file e lo mostra nel menu di tracciamento del requisito.

3.64 UC_15.2 - Inserimento riga inizio

Attore primario: Programmatore.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- Il requisito è stato tracciato automaticamente ed il tracciamento è disponibile ([UC_13] o [UC_14]).
- Il menu di tracciamento del requisito è stato espanso [UC_8.2].
- L'interfaccia di modifica del tracciamento è stata aperta [UC_15].

Postcondizioni:

- Il sistema aggiorna la riga di inizio con quella inserita dall'utente.

Scenario principale:

1. Il sistema mostra il campo di input per la riga di inizio.
2. L'utente inserisce la riga di inizio dell'intervallo di righe del file sorgente che implementa il requisito.
3. Il sistema registra la riga di inizio e la mostra nel menu di tracciamento del requisito.

3.65 UC_15.3 - Inserimento riga fine

Attore primario: Programmatore.

Precondizioni:

- Il requisito è visualizzato nell'elenco dei requisiti [UC_6].
- Il requisito è stato tracciato automaticamente ed il tracciamento è disponibile ([UC_13] o [UC_14]).
- Il menu di tracciamento del requisito è stato espanso [UC_8.2].
- L'interfaccia di modifica del tracciamento è stata aperta [UC_15].

Postcondizioni:

- Il sistema aggiorna la riga di fine con quella inserita dall'utente.

Scenario principale:

1. Il sistema mostra il campo di input per la riga di fine.
2. L'utente inserisce la riga di fine dell'intervallo di righe del file sorgente che implementa il requisito.

3. Il sistema registra la riga di fine e la mostra nel menu di tracciamento del requisito.

3.66 UC_16 - Configurazione dei path da ignorare

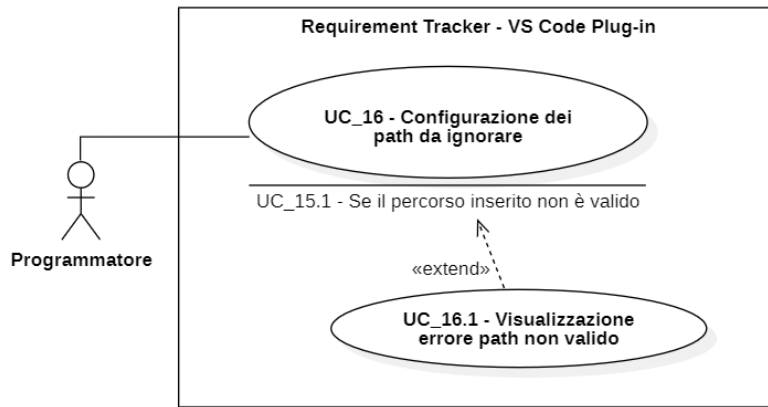


Figure 27: UC_16 - Configurazione dei path da ignorare

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 e dal tracciamento.

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 l'analisi ([UC_4] o [UC_11]) ed il tracciamento ([UC_13] o [UC_14]), 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_16.1 - 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.

3.67 UC_16.1 - 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.

3.68 UC_17 - Configurazione del modello di AI

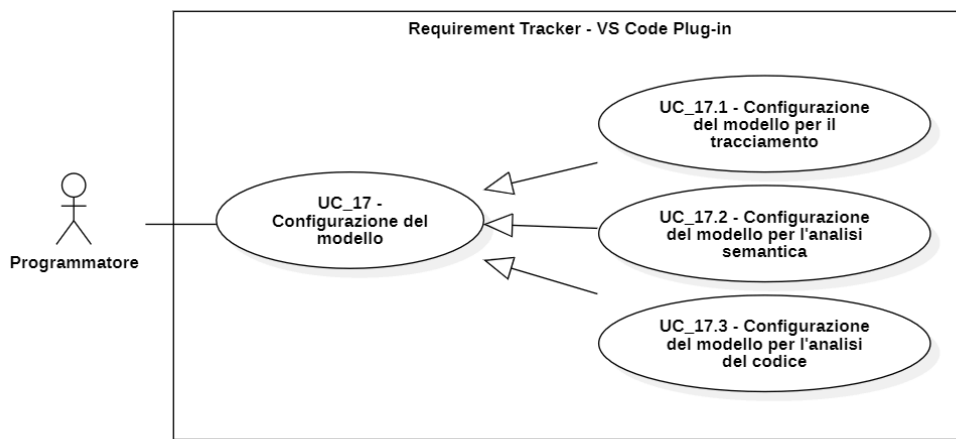


Figure 28: UC_17 - Configurazione del modello per l'analisi e tracciamento dei requisiti

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 o di embedding) configurato viene salvato e verrà utilizzato dal plug-in per le analisi ([UC_4] e [UC_11]) o per il tracciamento ([UC_13] e [UC_14]).

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 che vuole modificare tra:
 - Implementation Model: il modello di embedding per il tracciamento del codice.
 - Requirement Model: il modello LLM per l'analisi semantica del requisito.
 - Code Model: il modello LLM per l'analisi del codice sorgente.
4. L'utente inserisce il nome del modello desiderato (es. "llama3.2:3b" o "deepseek-coder:6.7b").
5. Il sistema salva la configurazione e la utilizza per le operazioni successive.

Generalizzazioni:

- **UC_17.1 - Configurazione del modello per il tracciamento:** L'utente seleziona il modello di embedding per il tracciamento del codice.
- **UC_17.2 - Configurazione del modello per l'analisi semantica:** L'utente seleziona il modello LLM per l'analisi semantica del requisito.
- **UC_17.3 - Configurazione del modello per l'analisi del codice:** L'utente seleziona il modello LLM per l'analisi del codice sorgente.

3.69 UC_17.1 - Configurazione del modello per il tracciamento

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 di embedding per il tracciamento del codice viene salvato e verrà utilizzato dal plug-in per il tracciamento dei requisiti([UC_13] e [UC_14]).

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 "Implementation Model".
4. L'utente inserisce il nome del modello di embedding desiderato (es. "mxbai-embed-large:latest").
5. Il sistema salva la configurazione e la utilizza per il tracciamento del codice.

3.70 UC_17.2 - Configurazione del modello per l'analisi semantica

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 per l'analisi semantica del requisito viene salvato e verrà utilizzato dal plug-in per l'analisi semantica ([UC_4] e [UC_11]).

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 "Requirement Model".
4. L'utente inserisce il nome del modello LLM desiderato (es. "llama3.2:3b").
5. Il sistema salva la configurazione e la utilizza per l'analisi semantica del requisito.

3.71 UC_17.3 - Configurazione del modello per l'analisi del codice

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 per l'analisi del codice viene salvato e verrà utilizzato dal plug-in per l'analisi del codice sorgente ([UC_4] e [UC_11]).

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 "Code Model".
4. L'utente inserisce il nome del modello LLM desiderato (es. "deepseek-coder:6.7b").
5. Il sistema salva la configurazione e la utilizza per l'analisi del codice sorgente.

3.72 UC_18 - Configurazione dell'endpoint del server Ollama

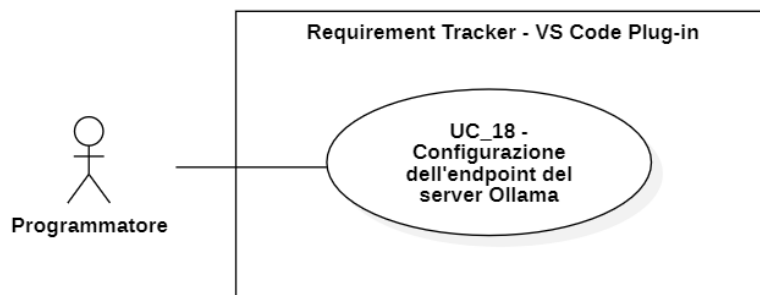


Figure 29: UC_18 - Configurazione dell'endpoint di Ollama

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.

3.73 UC_19 - Configurazione della soglia di conformità del codice

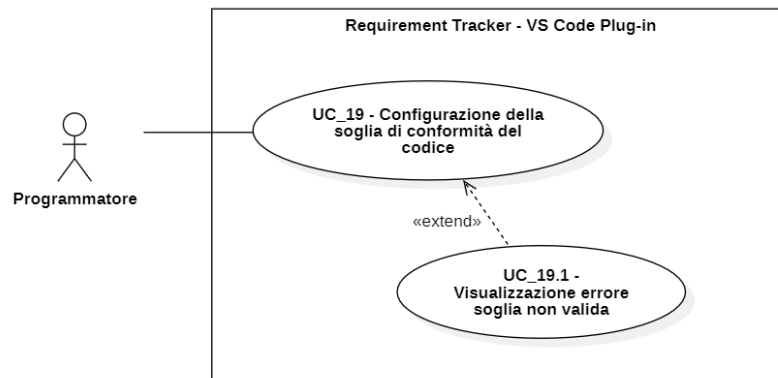


Figure 30: UC_19 - Configurazione della soglia del quality score

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 viene salvata e utilizzata dal plug-in per determinare la conformità dell’implementazione di ciascun requisito: se la valutazione (code score) restituita dall’analisi del codice è maggiore o uguale alla soglia impostata, il requisito viene considerato “passed” (conforme); altrimenti viene considerato “not passed” (non conforme).

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 “Requirement 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 del codice associato ai requisiti.

3.74 UC_19.1 - Visualizzazione errore valore soglia non valido

Attore primario: Programmatore.

Precondizioni:

- Durante la configurazione della soglia di conformità in [UC_19], 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.

3.75 UC_20 - Visualizzazione della soglia di conformità del requisito

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 viene salvata e utilizzata dal plug-in per determinare la conformità semantica di ciascun requisito: se la valutazione (requirement score) restituita dall'analisi è maggiore o uguale alla soglia impostata, il requisito viene considerato "passed" (conforme); altrimenti viene considerato "not passed" (non conforme).

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 "Requirement 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 semantica dei requisiti

Estensioni:

- **UC_19.1 - 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.

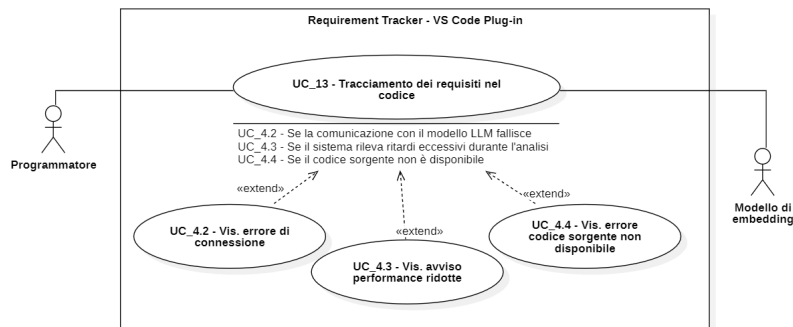
3.76 UC_21 - Visualizzazione dello stato di progressione

Figure 31: UC_21 - Visualizzazione dello stato di progressione

Attore primario: Programmatore.

Precondizioni:

- Il plug-in “Requirement Tracker Plug-in” è installato e attivo in Visual Studio Code.
- L’analisi è in corso ([UC_4] o [UC_11]) o il tracciamento è in corso ([UC_13] o [UC_14]).

Postcondizioni:

- Il sistema visualizza una barra di stato che mostra il progresso dell’analisi e del tracciamento dei requisiti.

Scenario principale:

1. L’utente avvia l’analisi o il tracciamento dei requisiti ([UC_4] o [UC_11] o [UC_13] o [UC_14]).
2. Il sistema inizia a elaborare i requisiti e mostra una barra di stato che indica il progresso dell’operazione.
3. La barra di stato si aggiorna in tempo reale per riflettere il progresso dell’analisi o del tracciamento.
4. Una volta completata l’operazione, il sistema mostra un messaggio di completamento e aggiorna l’elenco dei requisiti con i risultati dell’analisi o del tracciamento.

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_4
RFO002	Obbligatorio	Il sistema deve visualizzare graficamente i risultati delle analisi integrandoli nella lista dei requisiti	UC_9, UC_9.1, UC_9.1.1, UC_9.1.2, UC_9.2, UC_9.3, UC_9.4, UC_9.4.1, UC_9.5, UC_9.5.1
RFO003	Obbligatorio	Il sistema deve informare l'utente nel caso, a seguito di un'analisi, non ci siano risultati	UC_6.1
RFF004	<i>Facoltativo</i> ^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, UC_17
RFD008	<i>Desiderabile</i> ^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	Proponente

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, UC_3
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, UC_3
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_1.1, UC_1.3, UC_2, UC_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_3, UC_7, UC_7.1, UC_7.1.1, UC_8, UC_9
RFO015	Obbligatorio	Per ogni requisito presente nella lista caricata sarà presente un'icona predisposta ad avviare l'analisi del	UC_6.1.4

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_4
RFO017	Obbligatorio	Il sistema deve fornire una valutazione generica (per esempio, "Result: passed/not passed") per indicare se il requisito è stato rispettato	UC_4, UC_9.1, UC_9.1.2, UC_9.1.1
RFO018	Obbligatorio	Il sistema deve fornire una valutazione da 0 a 100 sull'aderenza della porzione di codice analizzata al rispettivo requisito	UC_4, UC_9.2
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_9.4, UC_9, UC_4, UC_9.4.1, UC_9.5, UC_9.5.1
RFO020	Obbligatorio	Il sistema deve essere in grado di aggiornare la vista dei requisiti, aggiungendo i risultati delle analisi	UC_9
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_4.1

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_4.2
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_4.3
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_4.4
RFO025	Obbligatorio	Il sistema, una volta eseguita l'analisi (o il tracciamento) dei requisiti, deve essere in grado di esportare i risultati ottenuti in formato CSV	UC_5
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_5.1
RFO027	Obbligatorio	Il sistema, una volta eseguita l'analisi, deve essere in grado di esportare sia il	UC_5

Codice	Classificazione	Descrizione	Fonti
		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_4.1
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_8, UC_8.1, UC_8.2
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_8.2.1
RFO031	Obbligatorio	Il sistema deve permettere all'utente di filtrare i requisiti in base ai campi ID, descrizione e file sorgente	UC_10
RFO032	Obbligatorio	Il sistema, in presenza di più requisiti caricati, deve essere in grado di eseguire l'analisi di un requisito specifico	UC_11
RFO033	Obbligatorio	Il sistema deve essere in grado di associare ai requisiti non mappati il relativo codice sorgente che lo implementi attraverso una richiesta di	UC_13, UC_14

Codice	Classificazione	Descrizione	Fonti
		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_16
RFO035	Obbligatorio	Il sistema deve permettere la configurazione del modello o dei modelli LLM che saranno utilizzati per l'analisi dei requisiti	UC_17, UC_17.1, UC_17.2, UC_17.3, Proponente
RFO036	Obbligatorio	Il sistema deve permettere di configurare l'endpoint del server Ollama	UC_18, Proponente
RFO037	Obbligatorio	Il sistema deve essere in grado di ripetere l'analisi di uno o più requisiti	UC_11
RFO038	Obbligatorio	Il sistema deve permettere all'utente di modificare la soglia di accettazione di un requisito, la quale permette	UC_19, Proponente

Codice	Classificazione	Descrizione	Fonti
		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_19, UC_19.1

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 $\geq 0.6.4$	Proponente
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
UC_4	RFO001, RFO017, RFO018, RFO019, RFO022, RFO023, RFO024

Fonte	Requisiti
UC_4.1	RFO021
UC_4.2	RFO022
UC_4.3	RFO023
UC_4.4	RFO024
UC_5	RFO025
UC_5.1	RFO026
UC_5.2	RFO021
UC_6	RFO027, RFO028
UC_6.1	RFO028
UC_7	RFO014
UC_7.1	RFO014, RFO015
UC_7.1.1	RFO014
UC_7.1.2	RFO015
UC_8	RFO029
UC_8.1	RFO029
UC_8.2	RFO029
UC_8.2.1	RFO030
UC_9	RFO002, RFO020
UC_9.1	RFO017
UC_9.2	RFO018
UC_9.4	RFO019
UC_9.5	RFO019
UC_10	RFO031
UC_11	RFO032, RFO037
UC_12	RFO033
UC_13	RFO034
UC_13.1	RFO034
UC_14	RFO035
UC_15	RFO036
UC_16	RFO038
UC_16.1	RFO039

Fonte	Requisiti
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	9
Figure 2	UC_2 - Importazione dei requisiti e del tracciamento da file CSV	11
Figure 3	UC_3 - Importazione dei requisiti, del tracciamento e dei risultati da file CSV ..	12
Figure 4	UC_4 - Analisi dei requisiti e della loro implementazione	13
Figure 5	UC_5 - Esportazione del tracciamento su file CSV	16
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 - Ordinamento dell'elenco dei requisiti	25
Figure 10	UC_8 - Visualizzazione dettaglio di un requisito	27
Figure 11	UC_8 - Sottocaso d'uso della visualizzazione in dettaglio di un requisito	27
Figure 12	UC_8.2 - Visualizzazione del tracciamento di un requisito	29
Figure 13	UC_9 - Visualizzazione dei risultati di un requisito	32
Figure 14	UC_9 - Dettaglio della visualizzazione dei risultati di un requisito	32
Figure 15	UC_9.4 - Visualizzazione lista dei suggerimenti	35
Figure 16	UC_9.4.1 - Dettaglio sulla visualizzazione del singolo suggerimento	36
Figure 17	UC_9.5 - Visualizzazione lista dei problemi	38
Figure 18	UC_9.5.1 - Dettaglio sulla visualizzazione del singolo problema	39
Figure 19	UC9 - Filtraggio dei requisiti	40
Figure 20	UC_11 - Analisi di un singolo requisito	42
Figure 21	UC_12 - Gestione dello stato di un requisito	43
Figure 22	UC_12 - Dettaglio sulla gestione dello stato di un requisito	44
Figure 23	UC_13 - Funzione di tracciamento dei requisiti	46
Figure 24	UC_14 - Ricerca del tracciamento nel codice di un singolo requisito	48
Figure 25	UC_15 - Configurazione manuale delle informazioni di tracciamento	49
Figure 26	UC_15 - Dettaglio sulle informazioni di tracciamento modificabili	49
Figure 27	UC_16 - Configurazione dei path da ignorare	53
Figure 28	UC_17 - Configurazione del modello per l'analisi e tracciamento dei requisiti ..	54
Figure 29	UC_18 - Configurazione dell'endpoint di Ollama	57
Figure 30	UC_19 - Configurazione della soglia del quality score	58
Figure 31	UC_21 - Visualizzazione dello stato di progressione	60

Elenco delle tabelle

Table 1	Requisiti Funzionali	61
Table 2	Requisiti di Qualità	67
Table 3	Requisiti di Vincolo	68

Table 4	Requisiti Prestazionali	68
Table 5	Tracciamento dei Requisiti	68
Table 6	Riepilogo	70