

# BugBoard26

Progetto di Ingegneria del Software A.A. 2025/2026

## Autori:

Simone Parente Martone - N86004297

Mario Penna - N86003308

Michela Pollio - N86003607

---

Dipartimento di Ingegneria Elettrica e delle Tecnologie  
dell'Informazione (DIETI)

Università degli Studi di Napoli Federico II

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Specifica dei Requisiti Software</b>	<b>3</b>
2.1	Attori del Sistema . . . . .	3
2.2	Requisiti Non Funzionali . . . . .	4
2.2.1	Requisiti del Prodotto . . . . .	4
2.2.2	Requisiti Organizzativi . . . . .	6
2.3	Modellazione dei Casi d'uso . . . . .	7
2.4	Casi d'uso Significativi . . . . .	8
2.4.1	UC1 - Creazione di una issue . . . . .	8
2.4.2	UC2 - Associazione di etichette personalizzate . . . . .	13
2.4.3	UC3 Ordinamento delle Issue . . . . .	16
<b>3</b>	<b>Glossario</b>	<b>20</b>

# 1 Introduzione

Il presente documento descrive l'analisi e la specifica dei requisiti per il progetto *BugBoard26*, una piattaforma software per la gestione collaborativa di *issue* e *bug* durante il ciclo di sviluppo di un prodotto software.

L'obiettivo principale del sistema è fornire un ambiente:

- **semplice**, per facilitare la segnalazione e la gestione dei problemi;
- **sicuro**, per garantire l'integrità e la riservatezza dei dati;
- **efficiente**, per supportare le attività di classificazione, assegnazione e monitoraggio delle problematiche software.

Il sistema integra funzionalità avanzate come:

- filtri dinamici di ricerca,
- allegati di immagini,
- esportazione dei dati in diversi formati,
- generazione di report mensili sull'attività del team.

Il documento include:

- la specifica dettagliata dei requisiti software;
- la modellazione dei casi d'uso secondo il formalismo di *Alistair Cockburn* [1];
- la descrizione dell'interazione tra utenti e sistema, supportata da diagrammi e mock-up.

Questo testo costituisce una guida completa per **sviluppatori**, **project manager** e **stakeholder**, fornendo una visione chiara e strutturata delle funzionalità e delle logiche alla base del progetto *BugBoard26*.

## 2 Specifica dei Requisiti Software

### 2.1 Attori del Sistema

Considerando il tipo di sistema in sviluppo — una piattaforma per la gestione collaborativa di progetti software — il background tecnico degli utenti sarà relativamente omogeneo. La maggior parte saranno sviluppatori, ad eccezione dei team leader, che potrebbero avere competenze tecniche meno avanzate.

Possiamo quindi definire due classi di utenti:

- **Utente standard:** sviluppatori software, tester e figure tecniche operative.
- **Utente amministratore:** team leader, manager, project manager e altri stakeholders con responsabilità di coordinamento e supervisione del progetto.

Tabella 2.1: Caratterizzazione degli attori

Attore	Utente Standard (Sviluppatore, Tester)	Utente Amministratore (Team Leader, Project Manager)
<b>Ruoli e Responsabilità</b>	Segnalazione issues, sviluppo codice, esecuzione e supervisione dei test, documentazione tecnica, collaborazione su task assegnati	Coordinamento team, assegnazione task, monitoraggio avanzamento, gestione priorità, reporting, supervisione generale del progetto
<b>Competenze Tecniche</b>	Conoscenza avanzata di linguaggi di programmazione, testing, version control, metodologie di sviluppo	Conoscenze tecniche di base/intermedie, competenze gestionali, pianificazione progetti, gestione risorse
<b>Frequenza e Modalità d'Uso</b>	Utilizzo quotidiano, accesso continuo durante orario lavorativo, interazione con task e repository	Utilizzo regolare (giornaliero/settimanale), focus su dashboard, reportistica e gestione team

Attore	Utente Standard (Sviluppatore, Tester)	Utente Amministratore (Team Leader, Project Manager)
Obiettivi	Completare task assegnati, collaborare con il team, mantenere qualità del codice, rispettare le deadline	Garantire rispetto timeline, allocare risorse efficacemente, monitorare performance del team, gestire rischi
Privilegi di Accesso	Lettura/scrittura su task assegnati, accesso ai repository del progetto, creazione issue e commenti	Accesso completo al progetto, creazione, modifica, eliminazione task, gestione utenti, configurazione progetto, accesso a statistiche e reportistica avanzata

## 2.2 Requisiti Non Funzionali

Il sistema sarà composto da due moduli principali: un **frontend** e un **backend**. Il frontend sarà una *Single Page Application (SPA)*, sviluppata presumibilmente tramite framework come **Angular** o **React**, mentre il backend sarà implementato in **Java**, utilizzando tecnologie come **Spring**, **Hibernate** o **Jakarta EE**.

### 2.2.1 Requisiti del Prodotto

#### Usabilità

- Il sistema dovrà essere facile da usare ed intuitivo.
- Il sistema dovrà essere **responsive**, garantendo un'esperienza ottimale sia su dispositivi desktop che mobili.
- Un utente deve poter completare un'attività primaria senza necessità di assistenza.
- Un amministratore con formazione adeguata deve poter registrare un nuovo utente in meno di 2 minuti.
- Gli elementi dell'interfaccia e i comportamenti (es. animazioni) devono essere coerenti nelle varie pagine per colori, dimensioni e tempi.
- I nuovi utenti devono poter completare almeno il 90% dei task basilari (creazione, aggiornamento, ricerca, filtraggio, ordinamento delle issues) dopo 30 minuti di addestramento.

- I messaggi di errore devono essere chiari e fornire indicazioni sulle azioni correttive.
- Le funzioni principali devono essere individuabili senza istruzioni.
- I titoli devono rispettare un limite massimo di caratteri per garantire la leggibilità completa delle issues.
- Ogni funzionalità principale deve essere raggiungibile con un massimo di 5 click dalla homepage.

## Efficienza

- Il sistema deve essere performante, con un'esperienza coerente tra versione desktop e mobile.
- Deve essere possibile caricare immagini fino a 5 MB senza compromettere le prestazioni.
- Il tempo di risposta per le operazioni basilari non deve superare 1 secondo.

## Affidabilità

- Il sistema deve garantire un **uptime del 99.5%** su base mensile durante gli orari lavorativi.

## Sicurezza

- Gli utenti devono cambiare la password temporanea fornita al primo login.
- Le password devono essere sottoposte ad algoritmi di hashing prima del salvataggio.
- Le sessioni devono essere gestite tramite **JSON Web Tokens (JWT)** con validazione.
- Il sistema deve essere protetto da attacchi **CSRF**, **XSS** e **SQL Injection**.
- Le password devono rispettare criteri minimi di complessità (minimo 8 caratteri, almeno una maiuscola, un numero e un carattere speciale).

## Manutenibilità

- L'architettura deve essere modulare per facilitare aggiornamenti e manutenzione.
- Deve essere implementato un sistema di **log strutturato** con livelli di severità.

- La documentazione del codice deve seguire gli standard **JavaDoc** (backend) e **JSDoc** (frontend).
- Le dipendenze di terze parti devono poter essere aggiornate senza modifiche strutturali al codice.

## 2.2.2 Requisiti Organizzativi

### Ambientali

- Il sistema dovrà essere ospitato su piattaforme di **cloud computing**, garantendo accesso continuo e universale.
- Il sistema deve essere **containerizzato** tramite **Docker**.
- Deve poter gestire almeno 200 utenti concorrenti senza degrado percepibile delle prestazioni.

### Operativi

- Browser supportati: ultime due versioni stabili di **Chrome**, **Firefox**, **Edge** e **Safari**.

### Di Sviluppo

- Tutto il codice deve essere versionato con **Git**.
- Ogni *pull request* deve essere approvata da almeno un altro sviluppatore.
- La qualità del codice deve essere verificata automaticamente tramite **SonarQube**.
- Gli sviluppatori devono utilizzare strumenti di analisi statica e *linter* integrati nel proprio IDE.

## 2.3 Modellazione dei Casi d'uso

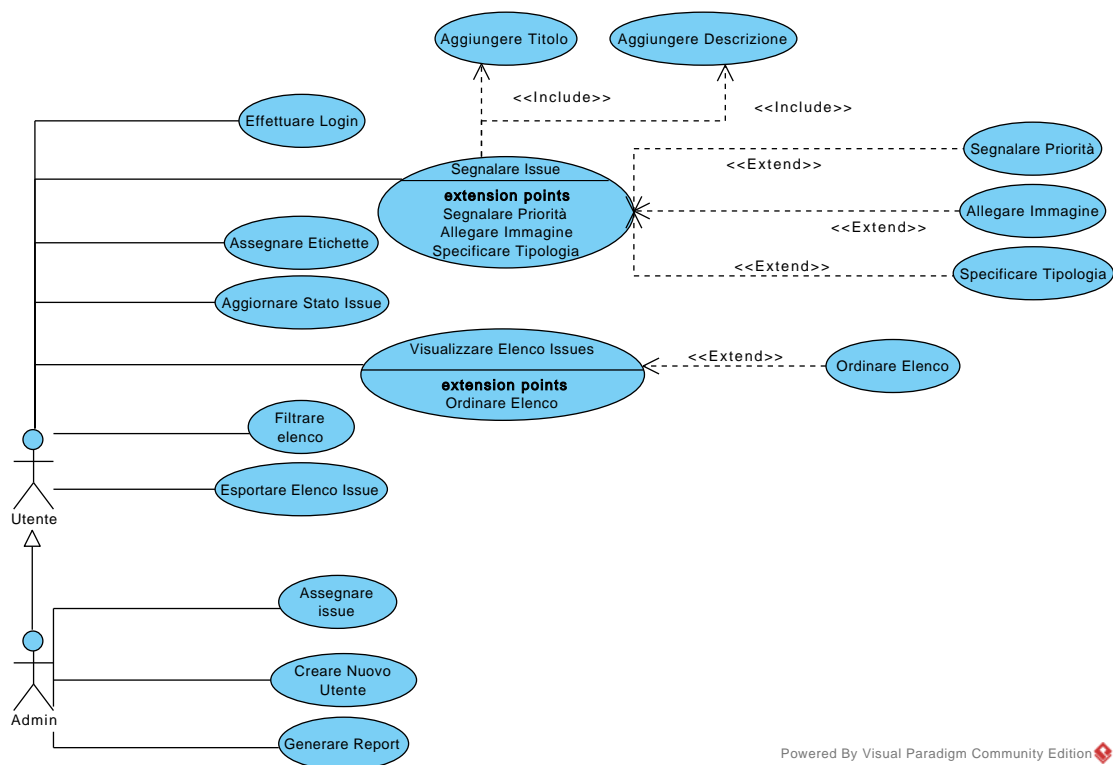


Figura 2.1: Diagramma dei casi d'uso principali di BugBoard26.



## 2.4 Casi d'uso Significativi

I casi d'uso presentati in questa sezione rappresentano alcune delle interazioni principali che caratterizzano il funzionamento del sistema BugBoard26.

La metodologia di documentazione adottata segue le linee guida contenute in *A Practical Style Guide and Templates Repository for Writing Effective Use Cases* [2], che fornisce, citando gli autori, *An opinionated style guide for use case writing* [3] attraverso il formalismo di Alistair Cockburn [1]. Questo approccio ci consente di descrivere in modo sistematico e coerente:

- L'obiettivo primario di ciascun caso d'uso dal punto di vista dell'attore
- Il flusso principale degli eventi e le relative condizioni di attivazione
- Gli eventuali flussi alternativi
- Le precondizioni necessarie e le postcondizioni attese

Per ogni caso d'uso significativo, forniamo una descrizione strutturata secondo il modello di Cockburn e una tabella delle proprietà rilevanti. Questo livello di dettaglio è necessario per garantire che le funzionalità chiave del sistema siano comprese in modo inequivocabile da tutte le parti interessate, dagli stakeholders agli sviluppatori.

Di seguito riportiamo i casi d'uso selezionati, che costituiscono alcune delle funzionalità *core* di BugBoard26 e rappresentano i principali flussi di interazione con il sistema.

### 2.4.1 UC1 - Creazione di una issue

<b>ID - Nome dello UC</b>	1 - Creazione di una issue [4]
<b>Data di creazione</b>	11/10/2025
<b>Attore primario</b>	Utente
<b>Trigger</b>	L'utente indica di voler creare una nuova issue
<b>Descrizione</b>	L'utente ha trovato un problema e intende aprire una nuova issue per tenere traccia di esso.

<b>Precondizioni</b>	<b>PRE-1.</b> L'utente è stato registrato da un amministratore ed ha effettuato correttamente l'autenticazione <b>PRE-2.</b> Il sistema ha caricato la pagina di riepilogo delle issues
<b>Postcondizioni</b>	<b>POST-1.</b> La nuova issue è salvata nel sistema <b>POST-2.</b> La nuova issue è visibile nella vista di riepilogo delle issue
<b>Scenario di successo</b>	<ol style="list-style-type: none"><li>1. L'utente indica di voler segnalare una nuova issue (M1)</li><li>2. Il sistema apre la modale di creazione di una nuova issue (M2)</li><li>3. L'utente inserisce il titolo della issue (M3)</li><li>4. L'utente indica i dettagli della issue nella descrizione (M4)<ol style="list-style-type: none"><li>4a. [Opzionale] L'utente sceglie una priorità per la issue</li><li>4b. [Opzionale] L'utente può allegare un'immagine</li><li>4c. [Opzionale] L'utente sceglie un tipo per la issue</li></ol></li><li>5. L'utente segnala di aver terminato l'inserimento dei dati richiesti (M5)</li><li>6. Il sistema valida i dati inseriti dall'utente</li><li>7. Il sistema salva la nuova issue</li><li>8. Il sistema mostra un messaggio di successo (M6)</li></ol>

Continua nella pagina successiva

**Estensioni****4a. Scelta della priorità per la issue:**

- 4a1. L'utente clicca sul menù priorità (M9)
- 4a2. Il sistema apre un menù con le possibili priorità (M10)
- 4a3. L'utente sceglie una tra le alternative (M11)
- 4a4. Il sistema mostra l'alternativa scelta (M12)
- 4a5. Il flusso ritorna al termine del punto 4 dello scenario di successo

**4b. Caricamento di un'immagine:**

- 4b1. L'utente clicca sul bottone per caricare un'immagine (M13)
- 4b2. Il sistema apre il file explorer (M14)
- 4b3. L'utente seleziona un'immagine dal proprio dispositivo (M15, M16)
- 4b4. Il sistema valida il formato e la dimensione dell'immagine
- 4b5. Il sistema sostituisce l'icona di allega con un messaggio di successo (✓) e aggiunge un bottone di eliminazione accanto alla label immagine(M17)
- 4b7. Il flusso torna al termine del punto 4a dello scenario di successo

**4b3a. L'utente annulla il caricamento dell'immagine:**

- 4b3a1. L'utente clicca sul bottone annulla
- 4b3a2. Il flusso ritorna al termine del punto 4 dello scenario di successo

**4b4a. Errore di validazione dell'immagine:**

- 4b4a1. Il sistema mostra un messaggio di errore e invita l'utente a riprovare (M18)
- 4b4a2. Il flusso ritorna al termine del punto 4 dello scenario di successo

**4b6a. Eliminazione dell'immagine allegata:**

- 4b6a1. L'utente clicca sul bottone di eliminazione
- 4b6a2. Il sistema elimina l'immagine allegata
- 4b6a3. Il sistema sostituisce l'icona di successo (✓) con il bottone allega e rimuove il bottone di eliminazione
- 4b6a4. Il flusso ritorna al termine del punto 4 dello scenario di successo

Continua nella pagina successiva

<b>Estensioni</b>	<p><b>4c. Scelta del tipo di issue:</b></p> <p>4c1. L'utente clicca sul menù tipo (M19)</p> <p>4c2. Il sistema apre il menù con i vari tipi di issues (M20)</p> <p>4c3. L'utente sceglie un tipo di issue (M21)</p> <p>4c4. Il sistema mostra il tipo scelto (M22)</p> <p>4c5. Il flusso ritorna al termine del punto 4 dello scenario di successo</p> <p><b>6a. Errore nella validazione dell'input (titolo):</b></p> <p>6a1. L'utente inserisce un titolo non valido</p> <p>6a2. Il sistema evidenzia il campo del titolo e mostra un messaggio all'utente, segnalando la non validità del titolo scelto (M7)</p> <p>6a3. L'utente modifica il titolo, inserendone uno corretto (M3)</p> <p>6a4. Il flusso ritorna al punto 4 dello scenario di successo</p> <p><b>6b. Errore nella validazione dell'input (descrizione):</b></p> <p>6b1. L'utente inserisce una descrizione non valida</p> <p>6b2. Il sistema evidenzia il campo della descrizione e mostra un messaggio all'utente, segnalando la non validità della descrizione scelta (M8)</p> <p>6b3. L'utente inserisce una descrizione valida (M4)</p> <p>6b4. Il flusso ritorna al punto 5 nello scenario di successo</p> <p><b>7a. Errore nel salvataggio della issue:</b></p> <p>7a1. Il sistema riscontra un errore durante il salvataggio della issue</p> <p>7a2. Il sistema mostra uno specifico messaggio di errore, invitando l'utente a riprovare (M23)</p> <p>7a3. Il flusso ritorna al termine del punto 4 nello scenario di successo</p>
<b>Priorità</b>	Alta
<b>Frequenza d'uso</b>	Presumibilmente tutti gli utenti, tra 1 e 5 volte al giorno per utente
<b>Informazioni associate</b>	Vedi tabella <a href="#">2.4</a>

Nome della proprietà	Tipo	Regole di validazione
Titolo	String	Obbligatorio, compreso tra 10 e 100 caratteri
Descrizione	String	Obbligatorio, compreso tra 30 e 30000 caratteri
Priorità	String	Facoltativo, valori possibili: <ul style="list-style-type: none"><li>- Lowest</li><li>- Low</li><li>- Medium</li><li>- High</li><li>- Highest</li></ul>
Immagine	File (image)	Facoltativo, dimensione massima: 5Mb
Tipo	String	Facoltativo, valori possibili: <ul style="list-style-type: none"><li>- Question</li><li>- Bug</li><li>- Documentation</li><li>- Feature</li></ul>

Tabella 2.4: Regole di validazione delle proprietà

## AC1 - Creazione di una issue

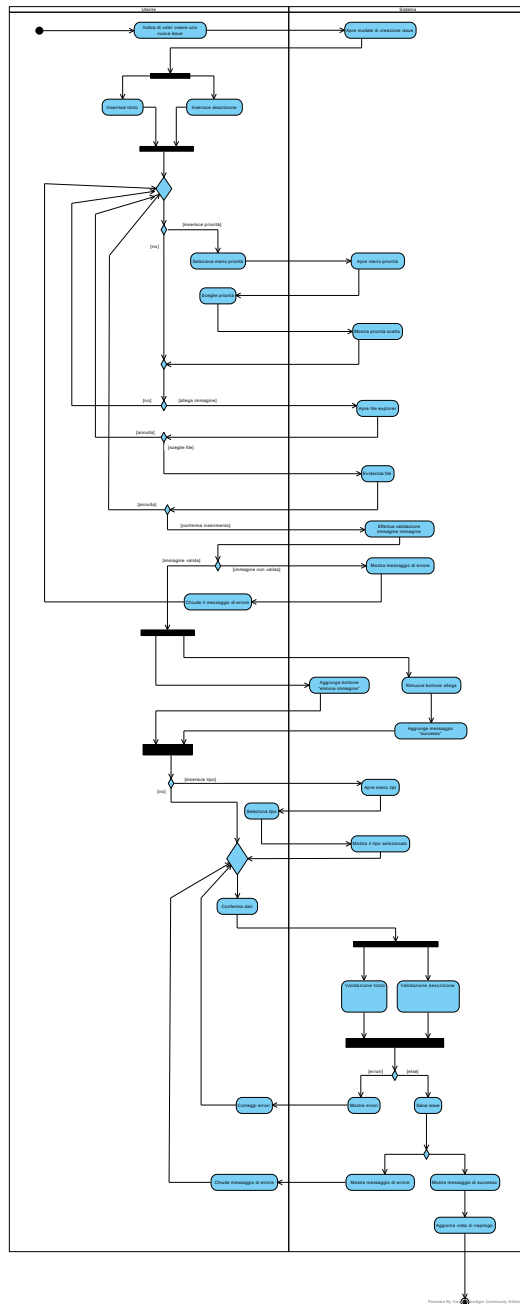


Figura 2.2: Diagramma d'attività della creazione di una nuova issue

## 2.4.2 UC2 - Associazione di etichette personalizzate

Descrizione Strutturata (Formalismo di A. Cockburn)

Tabella 2.5: UC-2 Associazione Etichette Personalizzate  
– Template di Cockburn

Campo	Valore
<b>ID - Nome dello UC</b>	2 - Associazione etichette personalizzate per i Bug [5]
<b>Data di creazione</b>	14/10/2025
<b>Attore primario</b>	Utente
<b>Trigger</b>	L'Utente modifica un bug e indica di voler associare una o più etichette personalizzate
<b>Descrizione</b>	L'Utente desidera associare un numero variabile di etichette personalizzate ad un bug per facilitarne la classificazione.
<b>Precondizioni</b>	<ul style="list-style-type: none"> <li>• PRE-1. L'Utente è registrato nel sistema.</li> <li>• PRE-2. Il bug selezionato esiste nel sistema.</li> </ul>
<b>Postcondizioni</b>	<ul style="list-style-type: none"> <li>• POST-1. Una o più etichette saranno assegnate al bug scelto dall'utente. Le nuove etichette saranno visibili nella schermata riepilogativa di tutte le issues.</li> </ul>
<b>Scenario di successo</b>	<ol style="list-style-type: none"> <li>1. L'Utente seleziona un bug esistente</li> <li>2. Il Sistema visualizza i dettagli del bug, comprese eventuali etichette già associate.</li> <li>3. L'Utente indica di voler aggiungere una nuova etichetta al bug.</li> <li>4. Il Sistema mostra pannello per la creazione di un nuovo tag.</li> <li>5. L'Utente inserisce il nome della nuova etichetta.</li> <li>6. Il Sistema verifica che il nome inserito rispetti i requisiti</li> <li>7. Il Sistema verifica che l'etichetta non sia già presente nel sistema</li> <li>8. L'Utente conferma la creazione dell'etichetta</li> <li>9. Il Sistema salva le etichette associate e aggiorna la visualizzazione del bug</li> <li>10. Il sistema notifica l'Utente che le etichette sono state aggiunte con successo</li> </ol>

Continua nella pagina successiva

Campo	Valore
<b>Estensioni</b>	<p><b>6a. Violazione delle regole di validazione</b></p> <p>6a1. Il sistema avvisa l'Utente che il nuovo nome inserito non rispetta i requisiti</p> <p>6a2. L'Utente corregge l'errore, il flusso prosegue normalmente dal passo 5 dello scenario di successo</p> <p><b>7a. Etichetta duplicata</b></p> <p>7a1. Il sistema rileva che l'etichetta è già presente.</p> <p>7a2. Il sistema mostra l'etichetta.</p> <p>7a3. L'Utente seleziona l'etichetta e il flusso prosegue normalmente dal punto numero 8 dello scenario di successo.</p> <p><b>7b. Modifica del colore della nuova etichetta</b></p> <p>7b1. L'utente seleziona il riquadro per la modifica del colore.</p> <p>7b2. il sistema mostra il pannello per la scelta del colore.</p> <p>7b3. L'Utente seleziona il nuovo colore.</p> <p>7b4. Il sistema salva la modifica, mostrando il nuovo colore scelto e l'esecuzione procede normalmente dal passo numero 8 dello scenario di successo</p> <p><b>9a. Errore nel salvataggio delle nuove etichette</b></p> <p>9a1. Il sistema riscontra un errore durante il salvataggio delle nuove etichette associate.</p> <p>9a2. Il sistema mostra uno specifico messaggio di errore.</p> <p>9a3. Il flusso ritorna al punto 1 nello scenario di successo.</p>
<b>Priorità</b>	alta
<b>Frequenza d'uso</b>	Alta tipicamente ogni volta che viene creato o aggiornato un bug.
<b>Informazioni associate</b>	Mostrato nella Tabella 2.6



Tabella 2.6: UC-2 Associazione Etichette Personalizzate  
– Proprietà e Regole di Validazione

Nome della Proprietà	Tipo	Regole di validazione	Note
<b>Nome</b>	String	Obbligatorio, univoco, lunghezza da 1 a 30 caratteri.	L'aggiunta o la modifica aggiornano la descrizione della issue.
<b>Colore</b>	String (HEX)	Facoltativo, formato HEX (RRGGBB), default:808080.	Permette di personalizzare il colore della nuova etichetta, scegliendo tra un set di colori disponibili
<b>Creatore</b>	String	Generata automaticamente	identificativo dell'Utente autenticato che la crea.

### 2.4.3 UC3 Ordinamento delle Issue

#### Descrizione Strutturata (Formalismo di A. Cockburn)

Tabella 2.7: UC-3 Ordinamenti – Template di Cockburn

Campo	Valore
<b>ID - Nome dello UC</b>	3 - Ordinamento [6]
<b>Data di creazione</b>	14/10/2025
<b>Attore primario</b>	Utente
<b>Trigger</b>	L'utente indica di voler ordinare l'elenco delle issues.
<b>Descrizione</b>	L'utente vuole ordinare le issues presenti nell'elenco tramite un criterio specifico.
<b>Precondizioni</b>	<ul style="list-style-type: none"> <li>• PRE-1. L'utente è stato registrato da un amministratore ed ha effettuato correttamente l'autenticazione.</li> <li>• PRE-2. Il sistema ha caricato la pagina di riepilogo delle issues.</li> </ul>

Continua nella pagina successiva

Campo	Valore
Postcondizioni	L'elenco delle issues è riordinato.
Scenario di successo	<ol style="list-style-type: none"><li>1. L'utente visualizza la tabella contenente le issues con le relative colonne.</li><li>2. L'utente clicca sull'intestazione di una colonna (es. stato, priorità, data creazione) per applicare un ordinamento.</li><li>3. Il sistema riordina la tabella in base ai valori della colonna selezionata e mostra un indicatore visivo della direzione dell'ordinamento (es. o ).</li><li>4. <i>[Opzionale]</i> L'utente clicca su ulteriori intestazioni di colonna per definire un ordinamento secondario o terziario.</li><li>5. <i>[Opzionale]</i> Se l'utente clicca nuovamente su una colonna già ordinata, il sistema inverte la direzione dell'ordinamento o ripristina l'ordine predefinito.</li><li>6. Il sistema mostra la tabella aggiornata e, se previsto, memorizza i criteri di ordinamento nella sessione o nell'URL.</li></ol>

Continua nella pagina successiva

Campo	Valore
<b>Estensioni</b>	<p><b>3a. Ordinamento crescente/decrescente</b></p> <p>3a1. L'utente clicca una seconda volta sulla stessa intestazione di colonna.</p> <p>3a2. Il sistema inverte la direzione dell'ordinamento (da crescente a decrescente o viceversa).</p> <p>3a3. Il sistema aggiorna l'indicatore visivo (freccia /) e ricarica la tabella.</p> <p>3a4. Il flusso ritorna al termine del punto 3 dello scenario di successo.</p> <p><b>4a. Ordinamento secondario o multiplo</b></p> <p>4a1. L'utente clicca su un'altra intestazione di colonna diversa da quella attiva.</p> <p>4a2. Il sistema imposta un secondo livello di ordinamento, mantenendo la gerarchia (es. priority stato).</p> <p>4a3. Il sistema aggiorna la tabella e mostra indicatori visivi multipli per le colonne coinvolte.</p> <p>4a4. Il flusso ritorna al termine del punto 4 dello scenario di successo.</p> <p><b>4b. Rimozione di un ordinamento attivo</b></p> <p>4b1. L'utente clicca nuovamente su una colonna ordinata fino a tornare alla condizione nessun ordine.</p> <p>4b2. Il sistema rimuove l'indicatore visivo e ripristina l'ordinamento predefinito.</p> <p>4b3. Il flusso ritorna al termine del punto 3 dello scenario di successo.</p> <p><b>6a. Persistenza dell'ordinamento</b></p> <p>6a1. Il sistema salva automaticamente le informazioni sull'ordinamento corrente nella sessione o nell'URL.</p> <p>6a2. Quando la pagina viene ricaricata o condivisa, l'ordinamento precedente viene ripristinato.</p> <p>6a3. Il flusso prosegue dal punto 6 dello scenario di successo.</p> <p><b>6b. Ordinamento su colonna non ordinabile</b></p> <p>6b1. L'utente clicca su una colonna che non supporta l'ordinamento (es. colonna tag).</p> <p>6b2. Il sistema ignora il click senza modificare la tabella.</p> <p>6b3. Il flusso ritorna al punto 2 dello scenario di successo.</p>
<b>Priorità</b>	Media

Continua nella pagina successiva

Campo	Valore
Frequenza d'uso	Tutti gli utenti, tra 1 e 3 volte al giorno per utente
Informazioni associate	Mostrato nella Tabella 2.8

Tabella 2.8: UC-3 Ordinamenti – Proprietà e Regole di Validazione

Nome della Proprietà	Tipo	Regole di validazione	Note
Colonne ordinabili	String	Obbligatorio, click sulla colonna per ordinarlo	Definisce su quali campi è permesso applicare l'ordinamento (nome, <i>stato</i> , <i>priorità</i> , <i>data creazione</i> )
Direzione ordinamento	Boolean (/)	Invertibile dall'utente	Indica se l'ordinamento è crescente o decrescente
Indicatore visivo	Icona UI	Aggiornato automaticamente	Mostra la direzione e il livello dell'ordinamento accanto all'intestazione
Persistenza ordinamento	URL param	Facoltativa	Se attiva, il sistema salva i criteri nella sessione o nell'URL
Colonne non ordinabili	String	Click ignorato	Colonna Tag

## 3 Glossario

### A

**Admin** utente con privilegi di gestione, in grado di creare e assegnare utenti e issue.

**Affidabilità** misura della capacità del sistema di funzionare in modo corretto e continuativo nel tempo.

**Angular** framework JavaScript per lo sviluppo di applicazioni web single-page.

**Architettura** struttura logica e fisica che definisce l'organizzazione dei componenti software.

### B

**Backend** parte dell'applicazione che gestisce la logica, il database e la comunicazione tra client e server.

**Background tecnico** insieme delle conoscenze, competenze e tecnologie pregresse che costituiscono la base teorica e pratica di un team di sviluppo.

**Boolean** tipo di dato che può assumere solo due valori: vero o falso.

**Bug** errore o comportamento anomalo nel software che causa un malfunzionamento.

### C

**Casi d'uso** rappresentazione delle interazioni tra un attore e il sistema per raggiungere un obiettivo specifico.

**Chrome** browser web sviluppato da Google.

**Click** azione dell'utente che interagisce con un elemento grafico.

**Cloud computing** modello di erogazione di servizi informatici tramite Internet.

**Code quality (Qualità del codice)** grado di leggibilità, modularità e manutenibilità del codice sorgente.

**Commenti** annotazioni nel codice che descrivono funzionalità o scelte implementative.

**Concorrenti** utenti o processi che operano simultaneamente sul sistema.

**Containerizzato** modalità di distribuzione del software all'interno di contenitori isolati, ad esempio tramite Docker.

**CSRF** vulnerabilità che consente ad un utente malintenzionato di eseguire azioni non autorizzate su un sito autenticato.

## D

**Deadline** scadenza entro la quale una determinata attività o task deve essere completata.

**Degrado** peggioramento delle prestazioni del sistema dovuto a carichi elevati.

**Dipendenze** librerie o componenti esterni necessari al corretto funzionamento del software.

**Docker** piattaforma per creare, distribuire ed eseguire container software.

## E

**Edge** browser web sviluppato da Microsoft.

**Efficienza** misura delle risorse necessarie per eseguire operazioni (tempo, memoria, CPU).

**Esportare** operazione per convertire o salvare dati in formati condivisibili (CSV, PDF, ecc.).

**Etichette** parole chiave che classificano o raggruppano issue e bug.

## F

**File** entità che rappresenta un'informazione salvata in memoria.

**File explorer** finestra di sistema usata per cercare e selezionare file locali.

**Firefox** browser open source sviluppato da Mozilla.

**Flusso** sequenza ordinata di azioni o eventi.

**Formalismo** insieme delle regole che definiscono il modo corretto di rappresentare i casi d'uso.

**Framework** struttura software riutilizzabile che facilita lo sviluppo applicativo.

**Frontend** parte dell'applicazione visibile e interattiva con l'utente.

**Funzionalità core** insieme delle funzioni principali del sistema.

## G

**Gerarchia** relazione di subordinazione o dipendenza tra elementi o ruoli.

**Git** sistema di controllo versione distribuito.

## H

**Hashing** tecnica per trasformare dati (come password) in una stringa di lunghezza fissa non invertibile.

**HEX / RRGGBB** codifica esadecimale utilizzata per rappresentare i colori.

**Hibernate** framework Java per la gestione della persistenza dei dati.

## I

**Icona** elemento grafico che rappresenta un'azione o funzione.

**IDE** ambiente di sviluppo integrato che facilita la scrittura e il debug del codice.

**Identificativo** codice univoco che distingue un elemento nel sistema.

**Image** file grafico o rappresentazione visiva associata a un'issue.

**Interfaccia** insieme degli elementi grafici e funzionali con cui l'utente interagisce.

**Interazioni** comunicazioni o scambi di informazioni tra attori e sistema.

**Issue** registrazione di un problema, richiesta o attività da gestire nel sistema.

## J

**Jakarta EE** piattaforma Java per lo sviluppo di applicazioni enterprise.

**Java** linguaggio di programmazione orientato agli oggetti.

**JavaDoc** sistema di documentazione automatica per codice Java.

**JSON Web Token (JWT)** standard aperto per la trasmissione sicura di informazioni tra client e server.

**JS Doc** strumento per generare documentazione dal codice JavaScript.

## L

**Lint** strumento che analizza il codice per rilevare errori sintattici o stilistici.

**Linguaggio di programmazione** mezzo formale per scrivere istruzioni comprensibili da un computer.

**Login** procedura di autenticazione che consente l'accesso al sistema.

**Log strutturato** registro di eventi con formato standard che facilita l'analisi automatizzata.

## M

**Manutenibilità** facilità con cui il software può essere corretto o migliorato nel tempo.

**MB** unità di misura della quantità di dati (megabyte).

**Menù** elenco di opzioni tra cui l'utente può scegliere.

**Messaggio di errore** testo mostrato quando si verifica un problema nell'esecuzione.

**Metodologia** insieme di processi e strumenti adottati nello sviluppo software.

**Metodologie di sviluppo** approcci sistematici come Agile, Waterfall.

**Modale** finestra sovrapposta che richiede l'interazione dell'utente.

**Modello di Cockburn** formalismo per la descrizione strutturata dei casi d'uso.

**Modifiche strutturali** cambiamenti significativi nell'architettura o nella base di codice.



## O

**Requisiti operativi** requisiti che definiscono le condizioni di utilizzo e manutenzione del sistema.

## U

**UI-(User Interface)** interfaccia grafica con cui l'utente interagisce.

**URL** indirizzo che identifica una risorsa in rete.

**URL param** parametro aggiunto a un URL per trasmettere informazioni.

**Usabilità** facilità con cui l'utente può utilizzare un sistema.

**Up time** tempo in cui il sistema risulta operativo e accessibile.

## V

**Validazione** processo con cui si verifica che un sistema, prodotto o software soddisfi realmente i bisogni e le aspettative dell'utente finale.

**Version Control** sistema che gestisce e traccia le modifiche apportate ai file di un progetto nel tempo.

## X

**XSS** Vulnerabilità nelle applicazioni web che permette a un attaccante di inserire ed eseguire script non autorizzati nel browser di un altro utente

# Riferimenti Bibliografici

- [1] A. Cockburn, *Writing Effective Use Cases*. Addison-Wesley Professional, 2001. DOI: 10.1145/505894.505918. indirizzo: <https://doi.org/10.1145/505894.505918>.
- [2] Y. W. Bingyang Wei Lin Deng, “A Practical Style Guide and Templates Repository for Writing Effective Use Cases,” *Software Engineering Research, Management and Applications (SERA 2022)*, pp. 67–80, 2022. DOI: 10.1007/978-3-031-09145-2\_5.
- [3] B. Wei. “Use Case Style Guide and Templates Repository.” Repository GitHub, GitHub. indirizzo: <https://github.com/Washingtonwei/use-case-style-guide>.
- [4] S. Parente Martone. “UC1- Creazione issue, prototipo mock-up. ”indirizzo: <https://www.figma.com/proto/w40cAPfZoJtGVIUNTzSeXF/BugBoard26?node-id=39-36&t=mPQpbtb5aucC9n5a-1>.
- [5] M. Pollio. “UC2- Creazione etichette, prototipo mock-up. ”indirizzo: <https://www.figma.com/proto/w40cAPfZoJtGVIUNTzSeXF/BugBoard26?node-id=108-609&t=n60xW5kdKw7LNRI7-1>.
- [6] M. Penna. “UC3 - Ordinamento, prototipo mock-up. ”indirizzo: <https://www.figma.com/proto/w40cAPfZoJtGVIUNTzSeXF/BugBoard26?node-id=24-13&t=mPQpbtb5aucC9n5a-1>.