

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

BugBoard26

Documentazione del progetto di Ingegneria del Software

Mario Lombardo

Matricola N. N86004029

Giuseppe Pollio

Matricola N. N86004953

Antonio Paudice

Matricola N. N86004566

Professori:

Porfiro Tramontana

Breve Bernardo

Università degli Studi di Napoli Federico II

Classe di Laurea in Scienze e Tecnologie Informatiche L-31

Anno Accademico 2025/2026

Pagina vuota

Indice

<i>Lista delle Figure</i>	ii
<i>Lista delle Tabelle</i>	iii
<i>Glossario</i>	iv
1 Introduzione	1
2 Descrizione Requisiti	2
2.1 Requisiti di sistema	2
2.2 Requisiti Non-Funzionali	3
3 Utenti del sistema	4
4 Casi d'uso	6
4.1 Use-Case diagram completo	6
4.2 Tabella completa Use-Case	6
5 Prototipazione via Mockup	9
5.1 Descrizione dettagliata dei Mockup	9
5.1.1 Schermata di Login	9
5.1.2 Area progetti	10
5.1.3 Pagina issues del Progetto	10
5.1.4 Modulo creazione Issue	12
5.1.5 Aggiornamento lista Issue	12
5.1.6 Errore durante la compilazione del modulo (creazione issue)	12
5.1.7 Errore interno, per qualche motivo i dati non vengono salvati nel database	12

Lista delle Figure

4.1	Use-Case Completo	6
5.1	Schermata di login	9
5.2	Area progetti	10
5.3	Pagina Issue	11
5.4	Creazione Issue	11
5.5	Aggiornamento lista	12
5.6	Errore 1	13
5.7	Errore 2	13

Lista delle Tabelle

4.1	Tabella completa Use-Case	7
4.2	Cockburn template Use-Case N.02, Segnalazione di una nuova issue	8

Glossario

API

Application Programming Interface - Interfaccia di programmazione delle applicazioni

UML

Unified Modeling Language - Linguaggio di modellazione unificato utilizzato per la specifica, visualizzazione e documentazione dei sistemi software

Framework

Struttura di supporto su cui un software può essere organizzato e progettato

Repository

Archivio centralizzato dove vengono memorizzati e gestiti i dati del progetto

Pagina vuota

Introduzione

Il progetto **BugBoard26** ha come obiettivo la realizzazione di una piattaforma per la **gestione collaborativa di issue** in progetti software. Il sistema consente a team di sviluppo di **segnalare, assegnare, monitorare e risolvere problematiche** legate al ciclo di vita di un progetto.

Il documento illustra l'analisi e la specifica dei requisiti funzionali e non funzionali, la modellazione dei casi d'uso e la prototipazione delle interfacce.

Descrizione Requisiti

I requisiti del sistema definiscono ciò che l'applicazione deve fare e come deve operare.

I **Requisiti Di Sistema** descrivono le funzionalità principali e le caratteristiche tecniche necessarie al corretto funzionamento del software, quali autenticazione e gestione dei progetti/issues.

I **Requisiti Non-Funzionali** specificano le qualità del sistema, concentrandosi su come deve operare. Include aspetti come sicurezza, usabilità e performance, garantendone efficienza, stabilità e affidabilità.

2.1 Requisiti di sistema

- **Gestione Autenticazione:** Il sistema deve consentire il login a utenti tramite email e password, distinguendoli tra utenti *Normali* e *Admin*.
- **Gestione Progetti:** Schermata iniziale dov'è possibile visualizzare e cercare tutti i progetti a cui l'utente lavora, con informazioni annesse come il numero di *issues* aperte per progetto e il numero di membri che ci partecipano.
Nel caso dell'*admin*, dev'essere possibile anche gestire progetti già esistenti o crearne di nuovi.
- **Creazione Issues:** Gli *admin* devono poter creare nuove *issues*, specificando tipo, priorità e una breve descrizione, mentre gli utenti normali possono segnalare nuove *issues* agli *admin*; opzionalmente si possono anche inserire un'immagine e uno stato.
- **Assegnazione Issues:** Gli *admin* devono poter assegnare *issues* a membri del progetto, notificando gli utenti interessati.
- **Gestione stati:** Il sistema deve consentire il cambiamento degli stati delle *issues*, sia dagli *Admin* sia dagli utenti a cui è stata assegnata un *issue*.
- **Filtraggio Issues:** L'utente deve poter filtrare le *issue* per nome, tipo (Bug, Feature, Question, Documentation), stato (Aperta, In Progress, Risolta, Chiusa), priorità (Bassa, Media, Alta, Critica) o in base al tempo di creazione.
- **Archiviazione Issues:** L'*admin* dev'essere in grado di archiviare *issues* risolte, in modo tale che non siano più visibili nella lista principale ma rimangano consultabili.
- **Commenti e aggiornamenti:** Ogni *Issue* deve poter includere una sezione commenti/aggiornamenti, visibile a tutti i membri del progetto .

- **Esportazione dati:** Il sistema dev'essere in grado di consentire l'esportazione dati in vari formati, quali CVS, PDF o Excel.
- **Modalità Read-Only:** Il sistema deve permettere la creazione di utenti in sola lettura per esterni al team di sviluppo, che consente di visualizzare *issues*, progetti e commenti senza però permetterne la modifica; La creazione e gestione di questi account è a carico solo degli *Admin* del sistema

2.2 Requisiti Non-Funzionali

- **Sicurezza:** Tutti i dati di utenti (progetti, issues) sono accessibili solo previa autenticazione, utilizzando email e password memorizzate in forma cifrata.
- **Affidabilità:** Il sistema deve garantire registrazione e conservazione di tutte le modifiche apportate alle *issues* senza perdere informazioni
- **Performance:** Il caricamento dei progetti o delle *issues* non deve richiedere tempo eccessivo, con database di dimensioni standard
- **Manutenibilità:** Il sistema dev'essere sviluppata in modo modulare per semplificare eventuali manutenzioni e aggiornamenti futuri
- **Portabilità:** Dev'essere garantita l'esecuzione del sistema su qualsiasi browser moderno
- **Backup e recupero dati:** Devono essere garantiti meccanismi di backup automatico e di esportazione manuale dei dati in vari formati per prevenire perdite accidentali

3

Utenti del sistema

Per il corretto sviluppo di **BugBoard26**, è necessario definire quali sono gli utenti che andranno a **utilizzare il sistema**, e di quali **competenze** o **esigenze** hanno bisogno

Il sistema è destinato a team di **sviluppo software** e, più in generale, a tutte le organizzazioni che necessitano di uno strumento collaborativo per la **gestione di issues, bug e richieste di miglioramento** nei propri progetti informatici.

All'interno del sistema sono stati individuati quattro *attori* principali, tutti con accesso a funzionalità diverse:

1. Utente non autenticato:

- Ha accesso solo alla schermata di login
- Non può accedere a dati a meno che non entra nel proprio account.

2. Utente normale (*User*):

- Rappresenta il profilo standard utilizzato dalla maggior parte del team di sviluppo (sviluppatori, tester...)
- Può visualizzare i progetti a cui partecipa e le *issues* a lui assegnate.
- Può creare nuove *issues* secondo il procedimento già indicato.
- Può filtrare e ordinare *issues* in base a vari criteri
- Può aggiornare lo stato delle *issues* a lui assegnate

3. Utente Admin:

- Dispone di privilegi avanzati per gestire al meglio l'intero sistema.
- Ha come obiettivo di monitorare e coordinare il lavoro del team.
- Ha completa gestione dei progetti, può crearli, eliminarli o modificarli.
- Può creare account di qualsiasi tipo, specificandone email, password, e tipo di utenza (Normale/Admin/Read-Only).
- Può assegnare *issues* a membri del team e contrassegnare eventuali duplicati.

4. Utente in sola lettura (*Read-Only*):

- Si tratta di un utente esterno al team di sviluppo, come uno stakeholder o un committente.
- Gli account di questa categoria possono essere gestiti e creati solo dagli *admin*
- Ha lo scopo di monitorare l'andamento del lavoro senza interferire con il lavoro del team
- Può visualizzare progetti, *issues* e commenti, senza però modificare o creare nulla

4

Casi d'uso

I **casi d'uso** descrivono **scenari concreti** di **interazione attori-sistema**, illustrando come il software risponde alle azioni dell'utente per soddisfare un obiettivo specifico.

Hanno il ruolo di far comprendere il **flusso operativo** dell'intero sistema.

4.1 Use-Case diagram completo

L'immagine comprende un **diagramma Use-Case** dell'intero sistema, comprendente di tutti gli *Attori* del sistema e degli 8 casi d'uso assegnati.

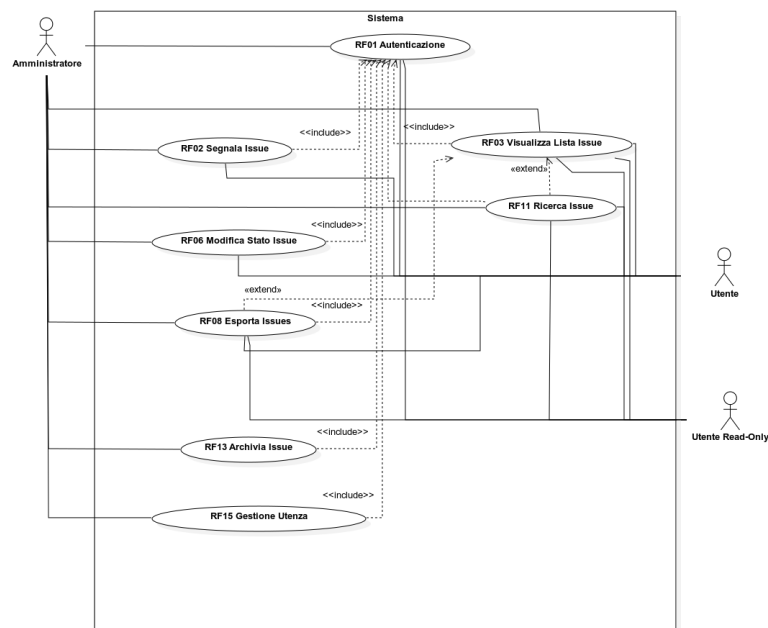


Figura 4.1: Use-Case Completo

4.2 Tabella completa Use-Case

La seguente tabella descrive nel dettaglio i casi d'uso presenti nell'immagine sopra

Tabella 4.1: Tabella completa Use-Case

ID	Titolo	Attori Coinvolti	Scopo	Relazioni
UC01	Autenticazione (RF01)	User Admin Read-only	Accedere al sistema con email e password	-
UC02	Segnala Issue (RF02)	User Admin	Creazione di una nuova issue specificando vari campi	«include»RF01
UC03	Visualizza lista Issue (RF03)	User Admin Read-Only	Visualizza l'elenco di issues presenti relative a un progetto specifico	«include»RF01
UC04	Modifica stato Issue (RF06)	User Admin	Consente di aggiornare lo stato di una issue, cambiandone i dettagli	«include»RF01
UC05	Esporta Issue (RF08)	User Admin Read-Only	Permette di esportare l'elenco di issues in vari formati	«include»RF01 «extend»RF03
UC06	Ricerca Issue (RF11)	User Admin Read-Only	L'utente può cercare una issue utilizzando parole chiave	«include»RF01 «extend»RF03
UC07	Archivia Issue (RF13)	Admin	Consente di archiviare le issues, rimuovendola dalla lista principale	«include»RF01
UC12	Gestione Utenza (RF15)	Admin	Permette di creare, modificare, eliminare altri account	«include»RF01

Tabella 4.2: Cockburn template Use-Case N.02, Segnalazione di una nuova issue

	A	B	C	D
1	Use Case UC02	Segnalazione (Aggiunta) di una nuova issue.		
2	Goal in Context	Si vuole segnalare una issue ad un progetto.		
3	Preconditions	L'utente deve essere autenticato e deve trovarsi nella vista dedicata ad un progetto. Il progetto di riferimento deve esistere nel sistema, L'utente se non e' un admin deve essere stato assegnato al progetto di riferimento.		
4	Success End Condition	La Issue è stata creata e salvata nel sistema con successo.		
5	Failed End Condition	La Issue non viene creata, viene mostrato un messaggio di errore e il sistema torna allo stato precedente.		
6	Primary Actor	Utente o Admin (L'utente non è read-only).		
7	Trigger Action	Click sul bottone "Segnala Issue".		
8	Main Scenario	Step n.	Actor	System
9		1	(Trigger Action) Viene premuto il bottone Segnala Issue.	
10		2		Viene mostrato il form per segnalare una nuova issue.
11		3	Vengono compilati almeno tutti i campi obbligatori	
12		4	Viene cliccato il bottone "Crea"	
13		5		Vengono verificati la validità dei dati inseriti nella form.
14		6		Viene salvata la issue all'interno del database del sistema.
15		7		(Final Action) Il form si chiude e viene lanciata una notifica di Successo.
16	Extension #1	3A	L'utente non compila tutti i campi obbligatori.	
17		3A1		Il sistema mostra un errore, i campi non compilati appaiono rossi.
18		3A2		Ritorna al passo 3.
19	Extension #2	6A		Errore interno, i dati non vengono salvati nel database del sistema.
20		6A1		Il form si chiude e viene lanciata una notifica di Errore
21		6A2		Final Action, Il caso d'uso è fallito

5

Prototipazione via Mockup

I seguenti *Mockup* rappresentano una simulazione grafica delle principali **interfacce utente** presenti nel sistema.

Hanno lo scopo di mostrare in modo chiaro il **flusso di navigazione** e le **funzionalità previste** per gli utenti. Le schermate realizzate permettono di visualizzare il **comportamento del sistema** e l'**esperienza d'uso** prima di passare all'implementazione effettiva.

5.1 Descrizione dettagliata dei Mockup

5.1.1 Schermata di Login

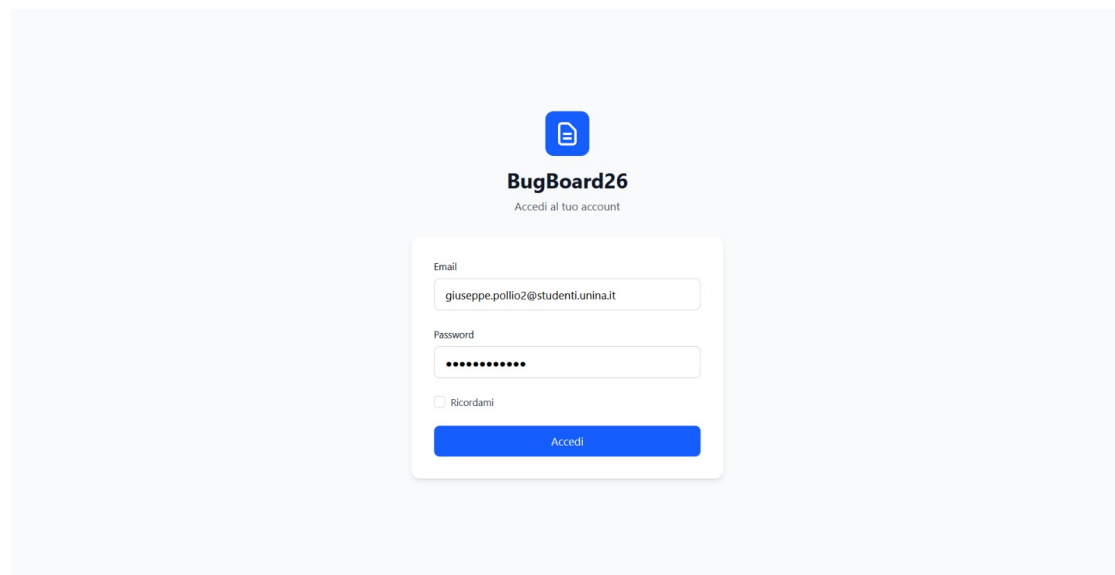


Figura 5.1: Schermata di login

La schermata raffigurata rappresenta la pagina di login al sistema. L'utente da qui inserisce le proprie credenziali per accedere al sistema, scegliendo inoltre se mantenere la sessione attiva tramite la spunta "Ricordami". Una volta autenticato, viene reindirizzato all'*Area progetti*.

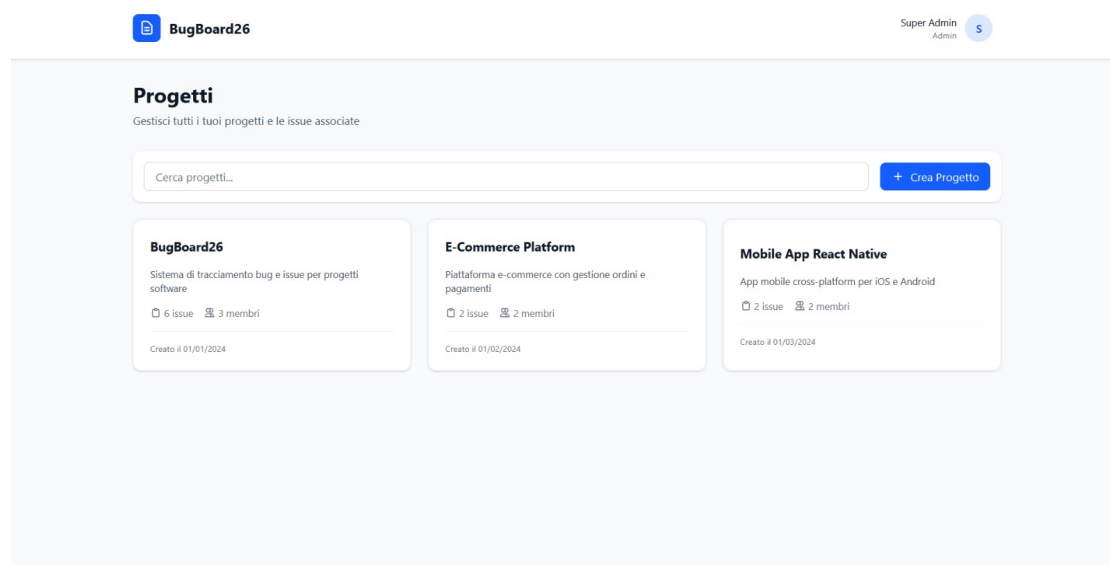


Figura 5.2: Area progetti

5.1.2 Area progetti

Una volta effettuato il login, l'utente accede all'area dei progetti, dove può visualizzare i progetti a cui partecipa.

In alto è presente una barra di ricerca per cercare i progetti e un pulsante per crearne di nuovi (visibile solo con account Admin)

Ogni progetto è caratterizzato da:

- Nome del progetto
- Breve descrizione
- Numero di *Issue* aperte
- Numero di membri del team
- Data di creazione

5.1.3 Pagina issues del Progetto

Una volta selezionato un progetto, l'utente accede alla pagina delle issues associate.

In questa pagina sono visibili tutte le segnalazioni relative al progetto selezionato, con titolo, descrizione, tipo, priorità, stato e autore.

Sopra la lista l'utente può filtrare le issues, cercarle nell'apposita barra di ricerca o ordinarle in base alla data di creazione. Inoltre è presente una spunta per visualizzare le issues archiviate.

Inoltre è presente il tasto "Crea Issue", che apre il modulo di creazione.

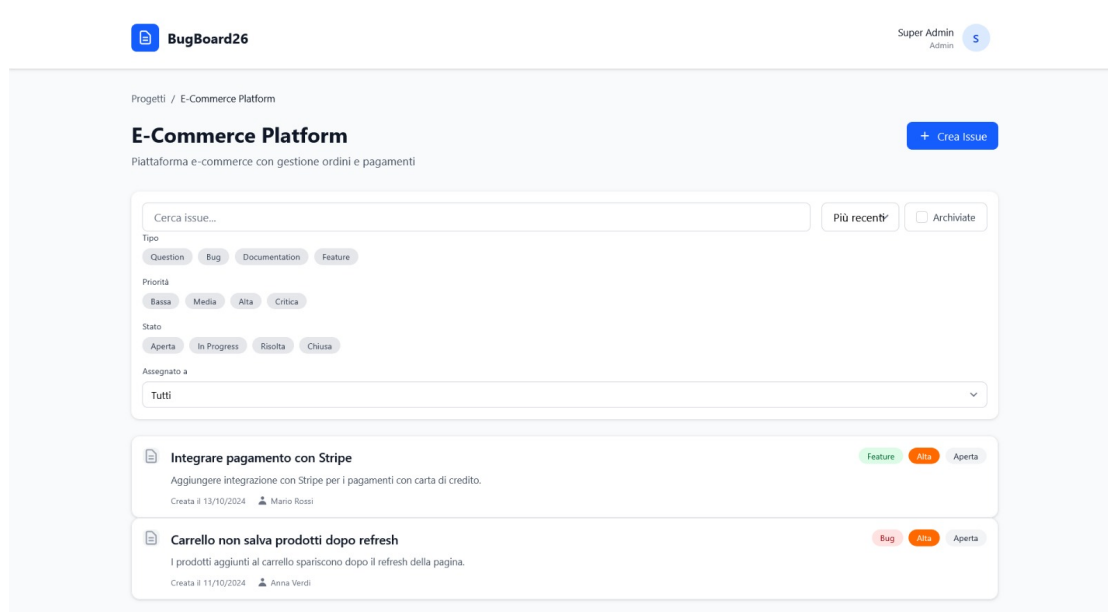


Figura 5.3: Pagina Issue
S

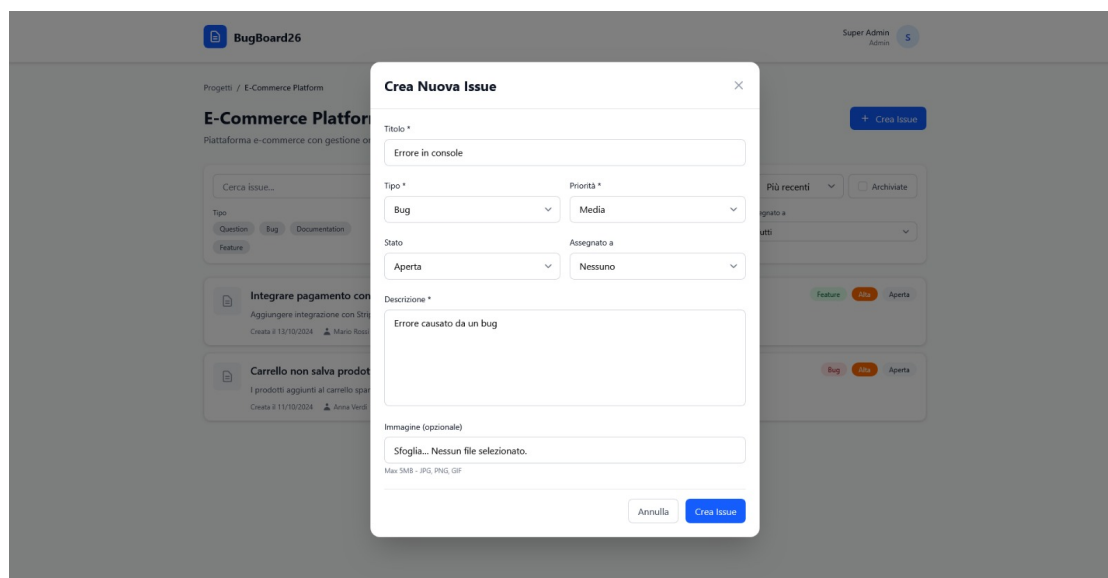


Figura 5.4: Creazione Issue

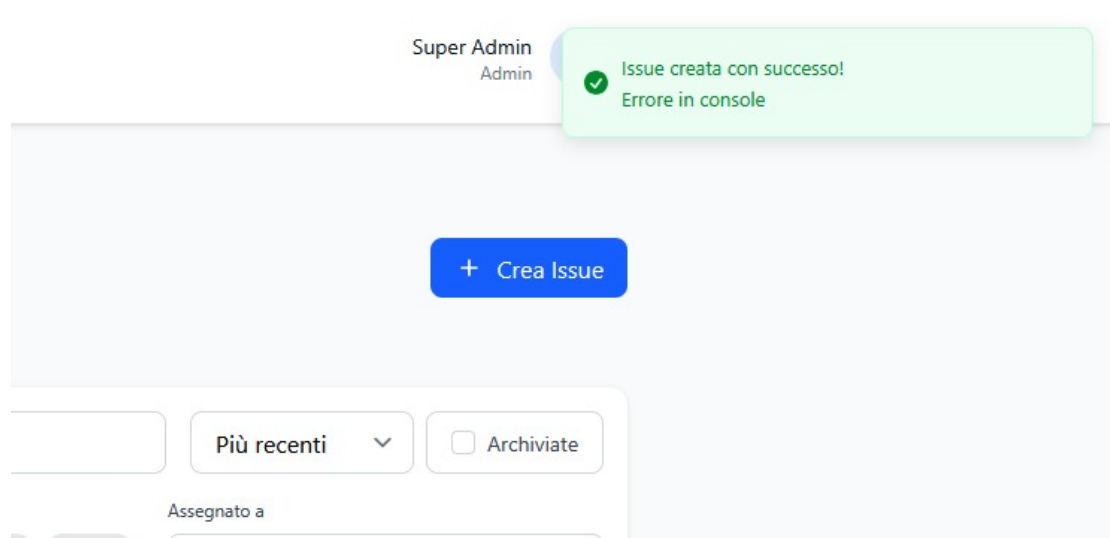


Figura 5.5: Aggiornamento lista

5.1.4 Modulo creazione Issue

Il *mockup* sopra mostra il form di inserimento di una nuova *issue*.

L'utente deve compilare alcuni campi obbligatori (titolo, tipo, priorità, descrizione), ma può anche aggiungerne di facoltativi (immagine, stato, assegnatario).

In fondo al modulo, sono presenti i pulsanti "Annulla" e "Crea Issue", che permettono rispettivamente di chiudere la schermata o di confermare la segnalazione.

5.1.5 Aggiornamento lista Issue

Il *mockup* raffigura la schermata aggiornata dopo la creazione di una nuova *issue*. La lista *issues* viene aggiornata automaticamente con la nuova voce in elenco.

Il sistema mostra inoltre una notifica di *feedback* in alto sullo schermo che conferma l'operazione di aggiunta (o segnala un eventuale errore se non è stato possibile aggiungere l'*issue*)

5.1.6 Errore durante la compilazione del modulo (creazione issue)

Durante la compilazione del modulo, se ci si dimentica di inserire tutti i campi obbligatori, si riscontra il seguente errore. Vengono mostrate i vari messaggi di errore e il sistema aspetta che l'utente corregga l'inserimento dei campi per continuare.

5.1.7 Errore interno, per qualche motivo i dati non vengono salvati nel database

Una volta finita la compilazione del modulo, dopo aver premuto il tasto "Crea", se il sistema cade in un qualsiasi errore interno allora viene lanciato il seguente messaggio di errore nella forma di una notifica.

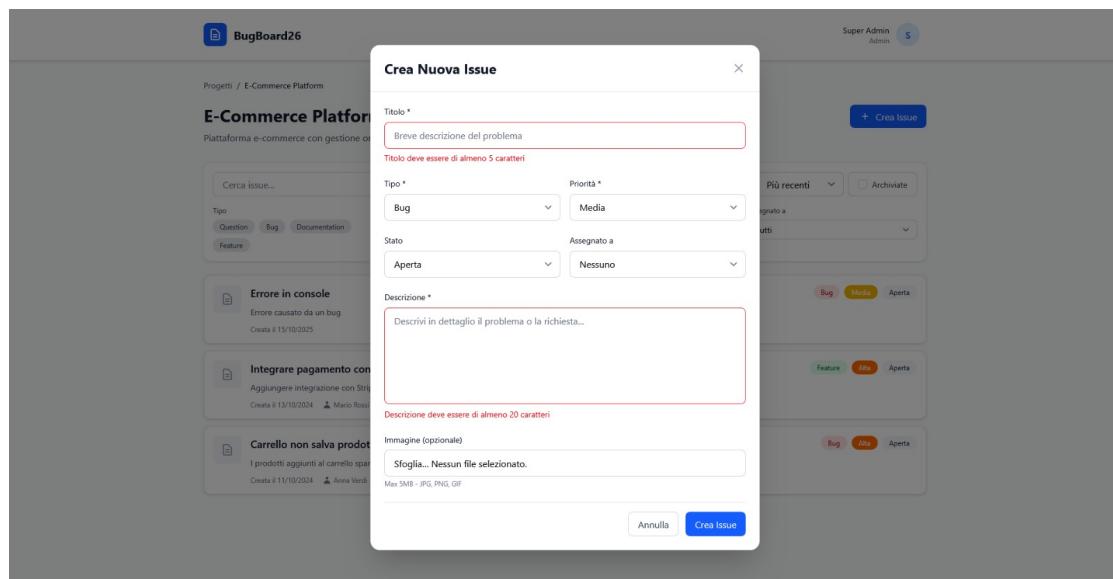


Figura 5.6: Errore 1

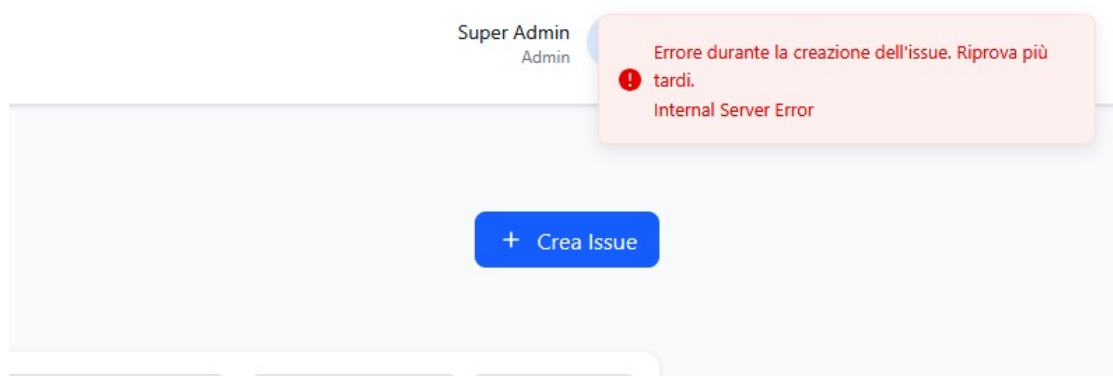


Figura 5.7: Errore 2