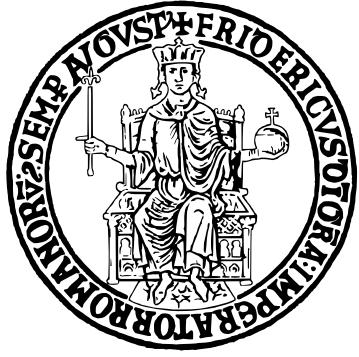


INGEGNERIA DEL SOFTWARE

BugBoard26

Documentazione Progetto



Davide Gargiulo **Francesco Donnarumma**
Matricola: N86004689 *Matricola: N86004658*

Anno Accademico 2024/2025
November 17, 2025

Contents

1 Modellazione di tutti i casi d'uso richiesti tramite Use Case Diagrams	2
2 Individuazione e caratterizzazione del target degli utenti, tramite Personas	3
2.1 Personas	3
2.1.1 Alexey Kutepov	3
2.1.2 Hideo Kojima	4
2.1.3 Michele Deserto	5
2.1.4 Yan Chernikov	6
3 Descrizione dei requisiti non-funzionali e di dominio	8
3.1 Requisiti non-funzionali	8
3.2 Requisiti di dominio	9
4 Dettagli dei casi d'uso specifici	10
4.1 Utente: Creazione di una nuova Issue	10
4.2 Amministratore: Aggiunzione di un nuovo utente	13
4.3 Utente: Aggiunzione di un commento a una Issue esistente	16

Modellazione di tutti i casi d'uso richiesti tramite Use Case Diagrams

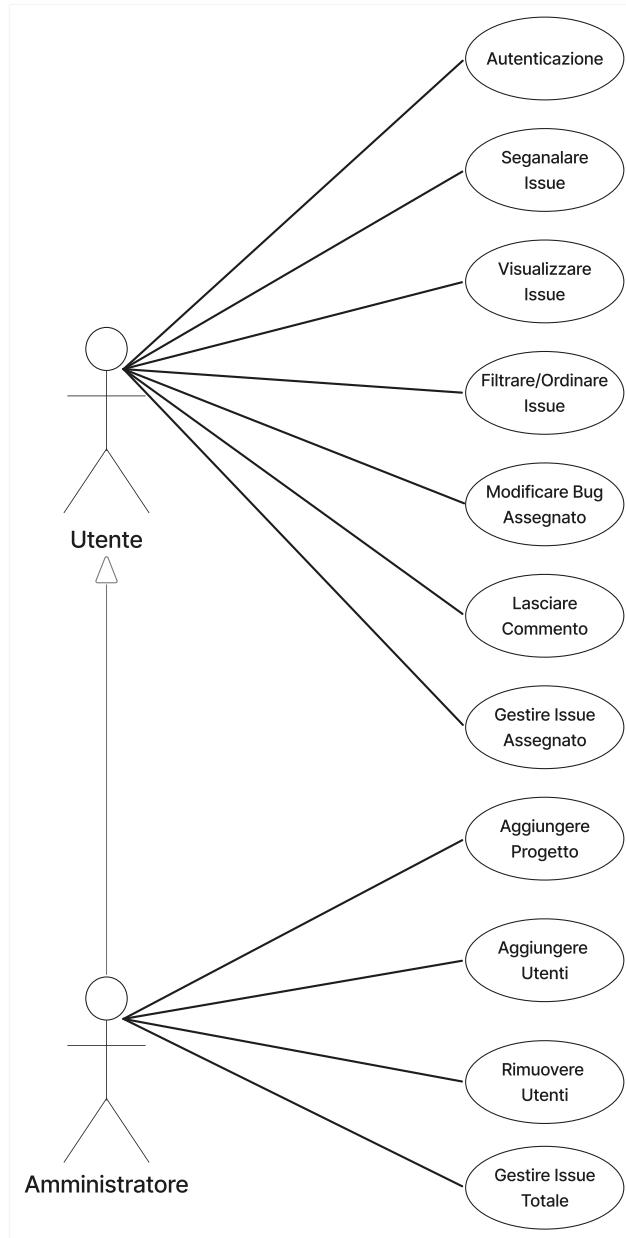


Figure 1.1: Use Case Diagram che rappresenta i casi d'uso principali dell'applicazione BugBoard26.

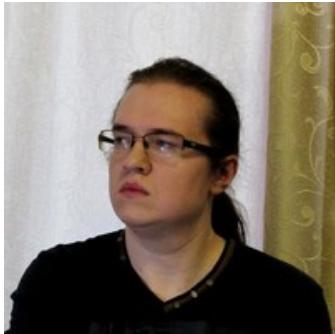
II

Individuazione e caratterizzazione del target degli utenti, tramite Personas

2.1

Personas

Alexey Kutepov (Tsoding)



Età: 35 anni, nato in Russia, attualmente nomade digitale

Residenza: Nomade digitale - Attualmente in Europa dell'Est

Istruzione: Laurea in Computer Science conseguita in Russia nel 2012, con focus su sistemi operativi e programmazione a basso livello

Lavoro: Software Engineer e Content Creator dal 2016. Streamer di programmazione su Twitch e YouTube, specializzato in C, Rust, e sviluppo di linguaggi di programmazione. Creatore di progetti open source e tool di sviluppo. Conosciuto per il suo approccio diretto e senza filtri alla programmazione

Tratti personali: Pragmatico e diretto, con approccio no-nonsense al codice. Perfezionista tecnico che valorizza semplicità ed efficienza. Ironico e sarcastico, ma genuinamente appassionato di condivisione della conoscenza tecnica

Interessi: Appassionato di matematica e teoria dei linguaggi di programmazione. Interessato a minimalismo nel design software e filosofia Unix. Nel tempo libero esplora nuove tecnologie, contribuisce a progetti open source e sperimenta con hardware vintage

Biografia

Alexey Kutepov, 35 anni, conosciuto online come Tsoding, è un software engineer e content creator nomade specializzato in programmazione di sistema e sviluppo di linguaggi. Dal 2016 condivide il suo processo creativo attraverso live coding su Twitch, dove programma in C e Rust senza script né editing.

Con un approccio pragmatico e diretto, Tsoding è diventato una figura di riferimento per developer che apprezzano la programmazione a basso livello e la filosofia del codice minimalista ed efficiente.

Scopi e obiettivi in riferimento al sistema

- **Tracciare bug nei progetti open source personali:** Gestire efficacemente gli issue dei suoi numerosi repository GitHub, mantenendo organizzazione chiara per la community di contributor.
- **Riprodurre e documentare bug complessi:** Creare report tecnici dettagliati con stack trace, memory dumps e condizioni di riproduzione per problemi di basso livello.
- **Prioritizzare fix in base a impatto tecnico:** Categorizzare issue distinguendo tra crash critici, memory leaks, problemi di performance e feature requests.
- **Integrare bug tracking nel workflow di live coding:** Utilizzare l'applicazione durante le sessioni di streaming per mostrare alla community come affrontare sistematicamente i problemi tecnici.

Hideo Kojima



Età: 61 anni, nato a Tokyo

Residenza: Tokyo, Giappone - Quartiere Shibuya

Istruzione: Laurea in Economia conseguita presso l'Università di Tokyo nel 1986, con studi complementari in cinema e letteratura

Lavoro: Game Director, Producer e Autore dal 1986. Fondatore di Kojima Productions nel 2015. Creatore di franchise iconici come Metal Gear e Death Stranding. Visionario dell'industria videoludica con approccio cinematografico

Tratti personali: Visionario e meticoloso, con attenzione maniacale ai dettagli narrativi e tecnici. Perfezionista, esigente ma collaborativo. Comunicatore carismatico con forte presenza mediatica

Interessi: Cinefilo appassionato che colleziona film e oggetti da collezione. Lettore vorace di letteratura e manga. Interessato a tecnologia, intelligenza artificiale e futuro dell'intrattenimento digitale. Attivo sui social media dove condivide le sue passioni quotidiane

Biografia

Hideo Kojima, 61 anni, è uno dei game director più influenti e riconosciuti a livello mondiale. Con una formazione in economia e una passione viscerale per il cinema, ha rivoluzionato l'industria videoludica dal 1986, creando opere che fondono gameplay innovativo e narrativa cinematografica.

Fondatore di Kojima Productions, è celebre per la sua attenzione maniacale ai dettagli e per la sua visione autoriale che ha ridefinito i confini tra videogiochi, cinema e arte interattiva.

Scopi e obiettivi in riferimento al sistema

- **Mantenere la visione artistica del progetto:** Assicurarsi che ogni bug risolto non comprometta l'esperienza narrativa e l'immersione cinematografica prevista per il gioco.
- **Coordinare team multidisciplinari internazionali:** Utilizzare l'applicazione per sincronizzare il lavoro tra programmati, designer, artisti e tester distribuiti in diverse sedi globali.
- **Prioritizzare issue che impattano l'esperienza emotiva:** Categorizzare i bug in base al loro effetto sulla narrazione, l'atmosfera e il coinvolgimento emotivo del giocatore.
- **Documentare dettagliatamente ogni problema:** Creare report completi con context narrativo, screenshot, video e note precise per guidare il team verso soluzioni che rispettino la visione creativa.
- **Monitorare milestone critiche pre-release:** Tenere traccia dello stato di risoluzione degli issue in vista di demo, eventi stampa e lancio finale, garantendo standard qualitativi eccellenti.

Michele Deserto



Età: 35 anni, nato a Bari, residente a Milano da 8 anni

Residenza: Milano, Italia - Zona Porta Nuova

Istruzione: Diploma in Architettura Moderna conseguito a Bari nel 2009, con specializzazione in design sostenibile

Lavoro: Content Creator dal 2015, specializzato in recensioni videoludiche e tecnologia su YouTube e Twitch. Appassionato di giochi indie, RPG e hardware. Assunto da poco come bug tester dalla FromSoftware

Tratti personali: Flemmatico e riflessivo, con approccio creativo ai problemi. Calmo, empatico e buon ascoltatore con la sua community

Interessi: Oltre ai contenuti digitali, ama viaggiare scoprendo nuove culture e tradizioni. Appassionato di architettura e fotografia urbana. Nel tempo libero cucina e suona la chitarra

Biografia

Michele Deserto, 35 anni, è un content creator milanese specializzato in videogiochi e tecnologia. Con un diploma in architettura moderna e un approccio flemmatico e creativo, dal 2015 produce contenuti su YouTube e Twitch dedicati a giochi indie, RPG e innovazioni tech.

Appassionato viaggiatore, ama scoprire nuove culture e tradizioni, che spesso ispirano i suoi contenuti digitali.

Scopi e obiettivi in riferimento al sistema

- **Diminuire la presenza di bug nei giochi recensiti:** Garantire che i giochi presentino meno bug tecnici, migliorando l'esperienza di gioco per i suoi spettatori.
- **Tracciare efficacemente i bug critici:** Utilizzare l'applicazione per categorizzare e prioritizzare gli issue in base alla loro gravità e impatto sul gameplay.
- **Collaborare con il team di sviluppo:** Condividere report dettagliati e screenshot attraverso l'applicazione per facilitare la comunicazione con i programmati.
- **Monitorare lo stato di risoluzione:** Tenere traccia dei bug segnalati e verificare quali vengono risolti nelle patch successive.
- **Organizzare i test per piattaforme diverse:** Gestire gli issue separando i bug per console, PC e altre piattaforme su cui vengono testati i giochi.

Yan Chernikov (TheCherno)



Età: 29 anni, nato in Australia, residente negli Stati Uniti

Residenza: Los Angeles, California - USA

Istruzione: Laurea in Computer Science conseguita in Australia nel 2017, con specializzazione in computer graphics e game engine development

Lavoro: Software Engineer e Content Creator dal 2012. Creatore della popolare serie YouTube su C++ e game engine development. Lead Engine Programmer con esperienza in AAA studios. Fondatore di Hazel Engine, un game engine educativo open source

Tratti personali: Metodico e didattico, con approccio sistematico all'insegnamento della programmazione. Paziente e chiaro nelle spiegazioni, ma esigente sulla qualità del codice. Entusiasta e motivante con la sua community

Interessi: Appassionato di architettura software e design patterns. Interessato a rendering graphics, fisica e ottimizzazione performance. Nel tempo libero sperimenta con nuove tecnologie grafiche, contribuisce alla community open source e gioca a giochi competitivi

Biografia

Yan Chernikov, 29 anni, conosciuto come TheCherno, è un software engineer e educator specializzato in C++ e game engine development. Dal 2012 produce contenuti educativi di alta qualità su YouTube, rendendo accessibili concetti complessi di programmazione grafica e architettura di game engine.

Con esperienza in studi AAA e una passione genuina per l'insegnamento, TheCherno ha costruito una delle community più rispettate per developer che vogliono comprendere profondamente come funzionano i motori grafici moderni.

Scopi e obiettivi in riferimento al sistema

- **Gestire issue del progetto Hazel Engine:** Organizzare e prioritizzare bug report e feature requests della community per il suo game engine educativo open source.
- **Documentare problemi di rendering e graphics:** Creare report dettagliati per bug complessi relativi a shader, pipeline grafiche e ottimizzazioni rendering con screenshot e frame analysis.
- **Coordinare sviluppo tra tutorial series:** Utilizzare l'applicazione per pianificare quali issue affrontare durante le prossime puntate della serie YouTube, mantenendo coerenza didattica.
- **Categorizzare per subsystem dell'engine:** Organizzare issue per moduli specifici (renderer, physics, ECS, scripting) facilitando la navigazione per contributor e studenti.
- **Tracciare problemi cross-platform:** Monitorare bug specifici per Windows, macOS e Linux, assicurando che Hazel Engine funzioni correttamente su tutte le piattaforme.

III

Descrizione dei requisiti non-funzionali e di dominio

3.1

Requisiti non-funzionali

Codice	Requisito
SR01	Il Sistema deve essere realizzato in modo distribuito prevedendo almeno due macro-componenti indipendenti (back-end/front-end).
SR02	Il back-end deve esporre interfacce di programmazione accessibili via rete.
SR03	Il back-end dovrebbe essere distribuito utilizzando tecnologie di containerizzazione.
SR04	Il front-end deve essere un'interfaccia utente che si appoggia ai servizi offerti dal back-end esclusivamente attraverso la rete.
SR05	La parte front-end deve essere realizzata come applicazione web app spa.
SR06	La logica applicativa e la persistenza dei dati non devono essere gestite esclusivamente tramite servizi esterni.
SR07	Il Sistema deve essere scritto in un linguaggio di programmazione che supporta il paradigma orientato agli oggetti.
SR08	Il Sistema deve essere conforme al GDPR per la protezione dei dati personali degli utenti.
SR09	Nel caso in cui il Sistema venga distribuito attraverso store di terze parti, deve essere conforme alle policy sui contenuti per sviluppatori di questi ultimi.

Requisiti di dominio

Codice	Requisito
DR01	Una Issue deve necessariamente contenere un titolo e una descrizione. Opzionalmente anche una priorità, un'immagine e una tipologia.
DR02	Un utente normale può agire solo su Issue a lui assegnate, mentre un admin può agire su tutte le Issue disponibili.
DR03	Ogni Issue ha un ciclo di vita definito da stati e transizioni irreversibili. to-do →in progress →done.
DR04	Le password degli utenti devono essere memorizzate in formato hash utilizzando algoritmi crittografici sicuri.
DR05	La priorità di un'Issue deve assumere valori da un insieme predefinito.
DR06	Solo gli amministratori possono creare nuovi account utente nel sistema.
DR06	Un'Issue può essere assegnata a un solo utente alla volta.
DR07	L'account amministratore di default deve essere configurato con credenziali sicure.

IV

Dettagli dei casi d'uso specifici

4.1

Utente: Creazione di una nuova Issue

Mockup

Figure 4.1: UC02_MC01

Figure 4.2: UC02_MC02



BugBoard26



MR
Mario Rossi
Amministratore

-  Dashboard
-  Utenti
-  Impostazioni
-  Logout

 Progetti

Project 1

Project 2

Project 3

Project 4

Project 5

Project 6

Project 7

+ Nuovo Progetto

Nuova Issue

Titolo

Descrizione

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tincidunt lobortis lorem quis cursus. Morbi ut mattis lacus. Duis ut orci vel diam elementum laclis. In bibendum purus id mauris mollis, ac venenatis eros tristique. Morbi tortor dolor, rhoncus sit amet dui ut, sodales porttitor metus. Nunc finibus, elit id elementum bibendum, ligula metus cursus tellus, non fermentum arcu nisl et odio. Suspendisse et nulla consequat, congue mauris eu, varius purus. Etiam nec leo non nisl tristique. Mauris ac lectus viverra, tellellus, ac auctor en

Confermare Issue?

La Issue verrà caricata nel sistema.

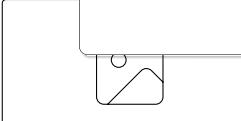




Figure 4.3: UC02_MC03



BugBoard26



Mario Rossi
Amministratore

- [Dashboard](#)
- [Utenti](#)
- [Impostazioni](#)
- [Logout](#)

[Progetti](#)

Project 1

Project 2

Project 3

Project 4

Project 5

Project 6

Project 7

[+ Nuovo Progetto](#)

Nuova Issue

Titolo

Descrizione

Quis ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tincidunt lobortis lorem quis cursus. Morbi ut orci vel diam elementum laouisse. In blandulum purus id mauris mollis, ac venenatis eros tristique. Morbi tortor dolor, rhoncus sit amet dui ut, sodales portitor metus. Nunc finibus, eit id elementum bibendum, ligula metus cursus tellus, non fermentum arcu nisi et odio. Suspendisse et nulla consequat, congue mauris eu, varius purus. Etiam nec leo non nisl tristique.

Mauris ac lectus viverris tellus, ac auctor enim ornare. Cras laoreet, mi nec consequat pharetra, ex quam sollicitudin

Errore! 

Almeno uno tra Titolo, Descrizione o Tipo non è compilato

[Conferma](#)

B I U E E ⌂ ⌂ ⌂ ⌂

[Annulla](#) [Conferma](#)

Figure 4.4: UC02 MC04

Use Case #02 Crea Nuova Issue			
	<i>Scopo</i>		
Precondizioni	L'utente è autenticato nel sistema.		
Condizione finale di successo	La nuova Issue è stata creata con successo nel sistema.		
Condizione finale di insuccesso	La creazione della nuova Issue non è riuscita a causa di un errore del sistema o di dati non validi forniti dall'utente.		
Attore Principale	Utente generale		
Trigger	L'utente clicca sul bottone "Nuova Issue" in UC02_MC01.		
Main Scenario	Step n.	Utente generale	Sistema
	01	L'utente clicca sul bottone "Nuova Issue" in UC02_MC01	
	02		Il sistema mostra UC02_MC02
	03	L'utente compila correttamente il form e clicca "Conferma"	
	04		Mostra UC02_MC03
	05	L'utente clicca sul bottone "Conferma"	
	06		Mostra UC02_MC01 e termina UC
Extension #01	Step n.	Utente generale	Sistema
	3.01	L'utente non compila titolo o descrizione e clicca "Conferma"	
	4.01		Mostra UC02_MC04
	5.01	L'utente clicca sul bottone "Conferma"	
	6.01		Riparte dal punto 2
Extension #02	Step n.	Utente generale	Sistema
	3.02	L'utente clicca sul bottone "Annulla"	
	4.02		Mostra UC02_MC01 e termina UC

4.2

Amministratore: Aggiunzione di un nuovo utente

Figure 4.5: UC08_MC01

Figure 4.6: UC08_MC02

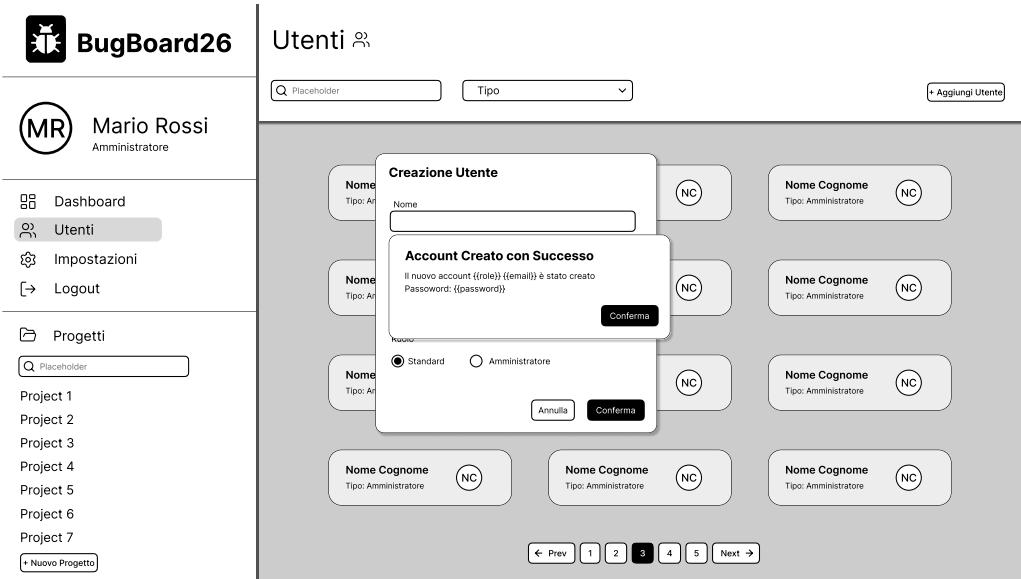


Figure 4.7: UC08_MC03

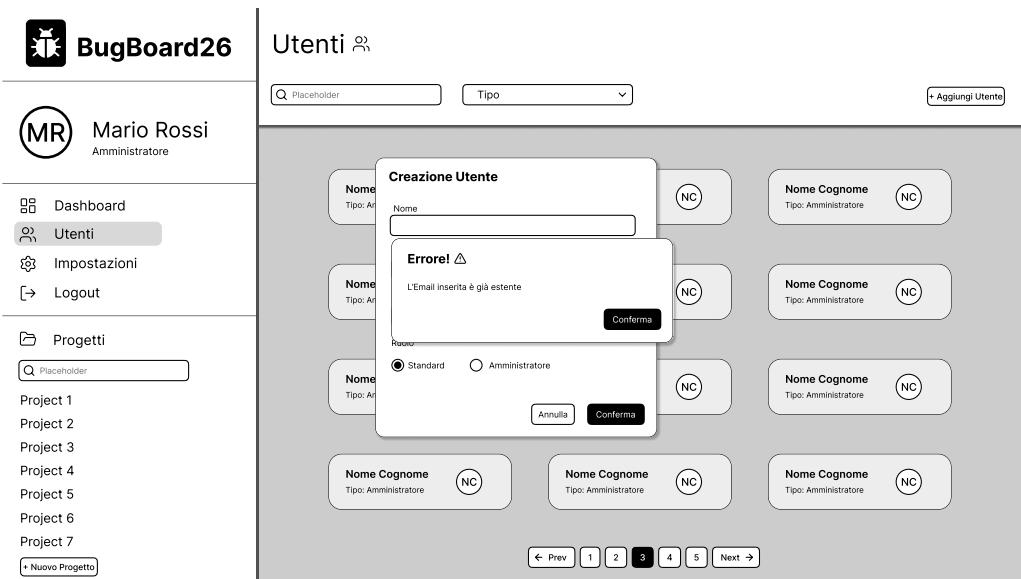


Figure 4.8: UC08_MC04

Use Case #08	Aggiungi Nuovo Utente		
Scopo	L'amministratore vuole aggiungere un nuovo utente.		
Precondizioni	L'amministratore è autenticato nel sistema.		
Condizione finale di successo	Il nuovo utente è stato aggiunto con successo nel sistema.		
Condizione finale di insuccesso	L'aggiunta del nuovo utente non è riuscita a causa di un errore del sistema o di dati non validi forniti dall'amministratore.		
Attore Principale	Amministratore.		
Trigger	L'amministratore clicca sul bottone "Nuovo Utente" in UC08_MC01.		
Main Scenario	Step n.	Amministratore	Sistema
	01	L'amministratore clicca sul bottone "Aggiungi Utente" in UC08_MC01	
	02		Il sistema mostra UC08_MC02
	03	L'amministratore inserisce nome, cognome, ruolo, email e clicca "Conferma"	
	04		Mostra UC08_MC03
	05	L'amministratore clicca sul bottone "Continua"	
	06		Mostra UC08_MC01 e termina UC
Extension #01	Step n.	Amministratore	Sistema
	3.01	L'amministratore clicca sul bottone "Annulla"	
	4.01		Mostra UC08_MC01 e termina UC
Extension #02	Step n.	Amministratore	Sistema
	3.02	L'amministratore compila il form con un'email già esistente e clicca "Conferma"	
	4.02		Mostra UC08_MC04
	5.02	L'amministratore clicca sul bottone "Conferma"	
	6.02		Riparte dal punto 2

4.3

Utente: Aggiunzione di un commento a una Issue esistente

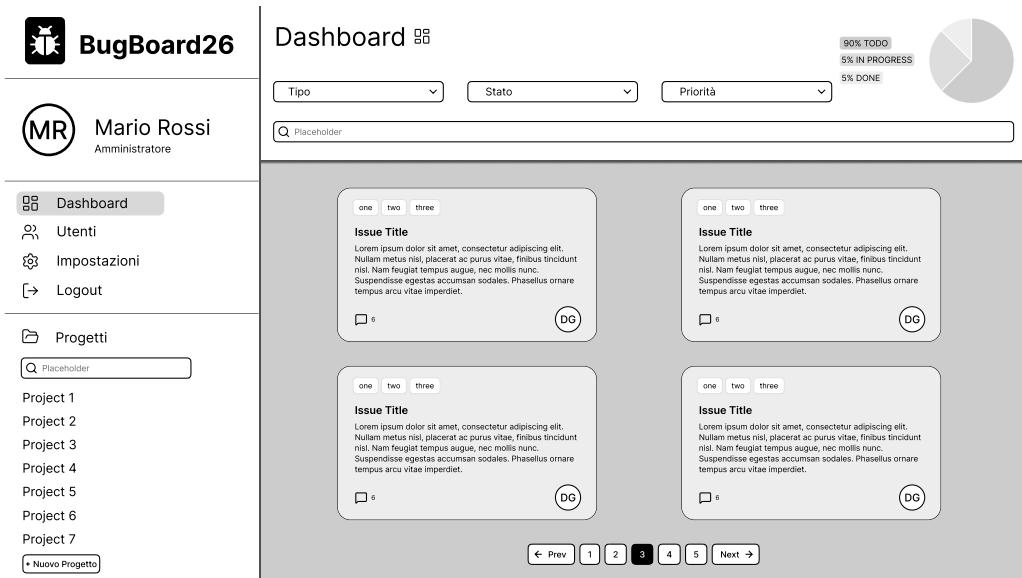


Figure 4.9: UC05_MC01

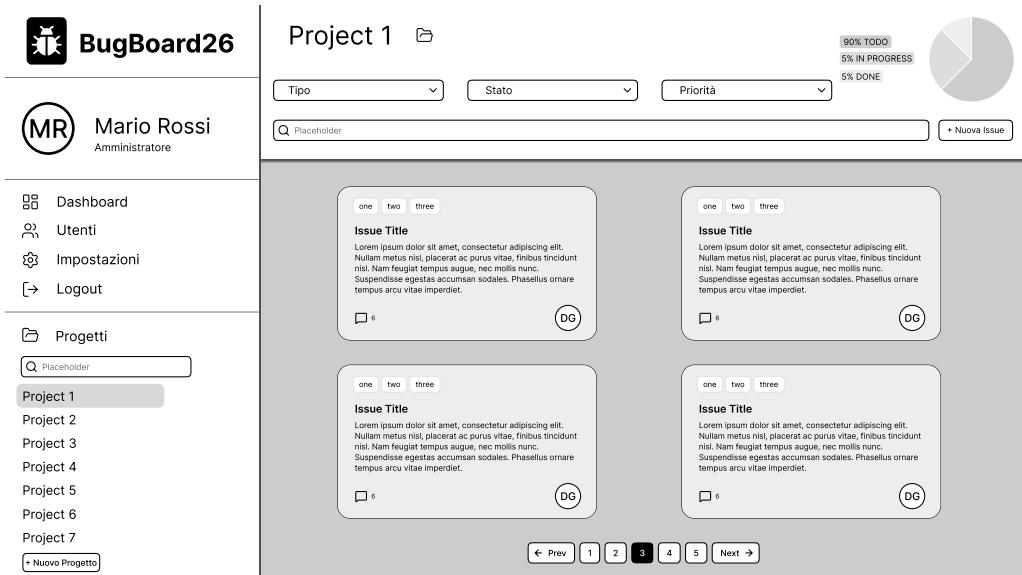


Figure 4.10: UC05_MC02

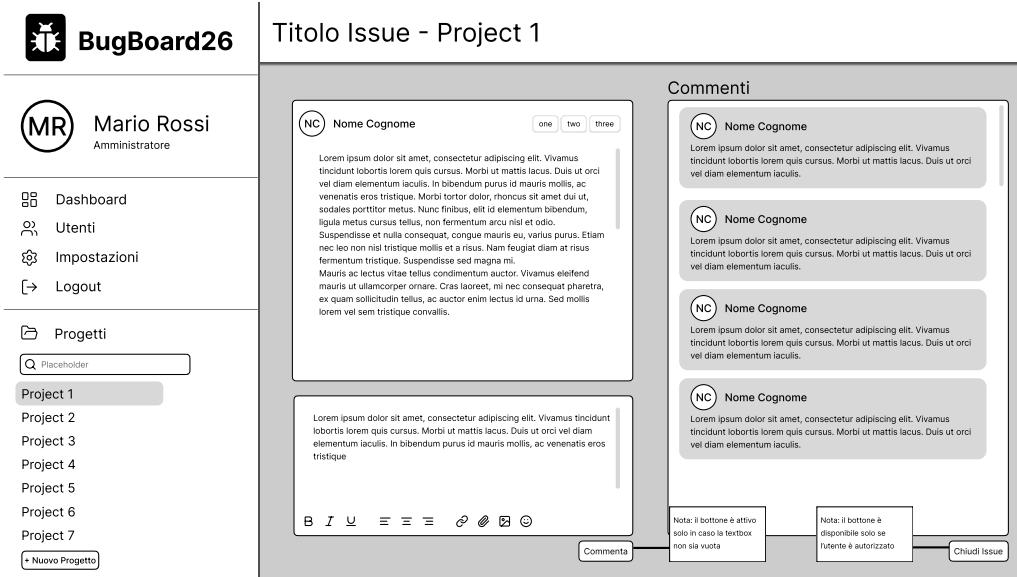


Figure 4.11: UC05_MC03

Use Case #05	Aggiungi Commento a Issue		
Scopo	L'utente vuole aggiungere un commento a una issue esistente.		
Precondizioni	L'utente è autenticato nel sistema.		
Condizione finale di successo	Il commento è stato aggiunto con successo alla issue.		
Condizione finale di insuccesso	L'aggiunta del commento non è riuscita a causa di un errore del sistema o di dati non validi forniti dall'utente.		
Attore Principale	Utente.		
Trigger	L'utente si trova in UC05_MC01 oppure in UC05_MC02 e visualizza un'issue.		
Main Scenario	Step n.	Utente	Sistema
	01	L'utente clicca su una issue per visualizzarla in UC05_MC01 o UC05_MC02	
	02		Il sistema mostra UC05_MC03
	03	L'utente compila la textbox dedicata e clicca "Commenta"	
	04		Mostra il commento in UC05_MC03 e termina UC