

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
CORSO DI LAUREA TRIENNALE IN INFORMATICA

BugBoard26

Documento di Specifica dei Requisiti Software

Ingegneria del Software

Mario Majorano	N86005035
Luca Sanselmo	N86005147

Anno Accademico 2025/2026

Indice

Introduzione	3
1 Scopo	3
2 Termini, definizioni e acronimi	4
2.1 Definizioni	4
2.2 Acronimi	5
2.3 Elementi Informativi	5
3 Descrizione generale	6
3.1 Prospettiva del prodotto	6
3.2 Fondamenti dei requisiti	6
3.2.1 Funzionalità del prodotto	6
3.2.2 Stakeholder	7
3.2.3 Vincoli	7
3.3 Trasformazione dei bisogni in requisiti	8
3.4 Attori del sistema	9
4 Requisiti Specifici	10
4.1 Requisiti Funzionali	10
4.1.1 RF01 - Autenticazione e gestione utenti	10
4.1.2 RF02 - Creazione e personalizzazione issue	11
4.1.3 RF03 - Visualizzazione e ricerca issue	12
4.1.4 RF04 - Dashboard e reporting	13
4.2 Requisiti Non Funzionali	14
4.2.1 RNF01 - Prestazioni	14
4.2.2 RNF02 - Affidabilità	14
4.2.3 RNF03 - Usabilità	15
4.2.4 RNF04 - Sicurezza	15
4.2.5 RNF05 - Manutenibilità	16
4.2.6 RNF06 - Portabilità	17
5 Modellazione dei casi d'uso	18
5.1 Descrizione casi d'uso principali	18
5.1.1 UC1 - Accedi alla piattaforma	18
5.1.2 UC2 - Crea nuova utenza	18
5.1.3 UC3 - Segnala un issue	19
5.1.4 UC4 - Specifica una priorità	19
5.1.5 UC5 - Aggiungi un'immagine	19
5.1.6 UC6 - Visualizza riepilogo	20
5.1.7 UC7 - Filtra risultati	20
5.1.8 UC8 - Ordina risultati	20
5.1.9 UC9 - Visualizza dashboard	21
5.1.10 UC10 - Aggiungi etichetta	21
5.1.11 UC11 - Svolgi attività	22
5.1.12 UC12 - Visualizza report mensile	22

6	Modellazione di un caso d'uso specifico	24
6.1	Formalismo di Cockburn	24
6.2	Mock-up	26
7	Tracciamento dei requisiti	27
7.1	Tracciamento fonte-requisito	27
7.2	Tracciamento requisiti-casi d'uso	27
8	Repository GitHub	27

Indice delle figure

1	Use Case Diagram #1 - [UC1,UC2]	18
2	Use Case Diagram #2 - [UC3,UC4,UC5]	19
3	Use Case Diagram #3 - [UC6,UC7,UC8]	20
4	Use Case Diagram #4 - [UC9]	21
5	Use Case Diagram #5 - [UC10]	21
6	Use Case Diagram #6 - [UC11, UC12]	22
7	Use Case Diagram #7 - Complessivo	23
8	Mock-up derivante dalla modellazione del caso d'uso UC3,UC4,UC5	26

Introduzione

Questo documento rappresenta la **Specifica dei Requisiti Software (SRS)** per il sistema **BugBoard26**, sviluppato nell'ambito del corso di Ingegneria del Software A.A. 2025/2026. La specifica è stata redatta in conformità con lo standard ISO/IEC/IEEE 29148:2018(E) per garantire una definizione completa, coerente e verificabile dei requisiti. BugBoard26 nasce come risposta all'esigenza di gestire in modo efficiente e collaborativo le issue all'interno di progetti software.

In contesti di sviluppo moderni, caratterizzati da team distribuiti e cicli di release sempre più rapidi, la capacità di tracciare, assegnare e monitorare problemi e richieste diventa fondamentale per il successo del progetto.

La piattaforma si propone di integrare funzionalità avanzate di issue tracking con un'interfaccia intuitiva e strumenti analitici, ponendo particolare attenzione all'esperienza utente e alla scalabilità del sistema.

L'architettura distribuita, che separa chiaramente back-end e front-end, garantisce flessibilità nell'evoluzione del prodotto e nelle scelte tecnologiche.

Il documento che segue organizza i requisiti in modo strutturato, facilitando la tracciabilità durante l'intero ciclo di vita del sviluppo software e fornendo una base solida per le decisioni progettuali che verranno assunte nelle fasi successive del progetto.

1 Scopo

Il presente documento ha lo scopo di definire in modo formale, completo e verificabile i requisiti del sistema **BugBoard26**.

Questo documento di Specifica dei Requisiti Software (SRS) rappresenta il fondamento per:

- **Definire le basi contrattuali** tra il fornitore e il committente, chiarendo aspettative e obblighi riguardo alle funzionalità da realizzare.
- **Guidare le attività di progettazione e implementazione**, fornendo ai progettisti e agli sviluppatori una descrizione dettagliata e non ambigua del comportamento atteso del sistema.
- **Supportare le attività di testing e validazione**, permettendo la creazione di piani di test e casi di test che verifichino la soddisfazione di ogni requisito.
- **Fornire un riferimento per la gestione dei cambiamenti**, facilitando la tracciabilità dei requisiti attraverso tutto il ciclo di vita del sviluppo software.
- **Servire come linea guida per le future manutenzioni ed evoluzioni** del sistema, documentando le decisioni prese durante la fase di analisi.

2 Termini, definizioni e acronimi

Introduzione

La presente sezione del documento SRS ha l'obiettivo di elencare i termini, le definizioni e gli acronimi utilizzati per fare riferimento a concetti, istituzioni e ad altri elementi rilevanti trattati nel documento.

2.1 Definizioni

Issue

Segnalazione di un problema, richiesta o domanda relativa al progetto software

Bug

Tipo di issue che rappresenta un malfunzionamento del software

Feature

Tipo di issue che rappresenta una richiesta di nuova funzionalità

Question

Tipo di issue che rappresenta una richiesta di chiarimento

Documentation

Tipo di issue relativa a problemi nella documentazione

Priorità

Livello di importanza assegnato a un issue

Stato

Condizione corrente di un issue (todo, in progress, done, etc.)

Amministratore

Utente con privilegi elevati che può gestire utenti e tutte le issue

Utente standard

Utente che può creare e modificare le issue a lui assegnate

Dashboard

Pannello di controllo che mostra informazioni aggregate

Tag

Marcatore personalizzabile associato ai bug

Stakeholder

Gli stakeholder sono tutti i soggetti che, a vario titolo, interagiscono con il sistema o ne influenzano lo sviluppo e l'evoluzione.

Richiesta attività fornitori esterni

Documento che formalizza l'affidamento di attività, servizi o forniture a fornitori esterni, descrivendone oggetto, requisiti, tempi e modalità di approvazione.

2.2 Acronimi

SRS

Software Requirements Specification

RF Requisiti Funzionali

RNF

Requisiti Non Funzionali

UC Use Case - Caso d'uso

UCD

Use Case Diagram - Diagramma dei Casi d'uso

UML

Unified Modeling Language - Linguaggio di modellazione standard

ISO/IEC/IEEE

International Organization for Standardization / International Electrotechnical Commission / Institute of Electrical and Electronics Engineers

2.3 Elementi Informativi

1. **ISO/IEC/IEEE 29148:2018(E)** - Systems and software engineering - Life cycle processes - Requirements engineering
2. **Richiesta di attività fornitori esterni** - "Progetto di Ingegneria del Software - A.A. 2025/2026", SoftEngUniNA
3. **UML 2.5 Specification**

3 Descrizione generale

Questa sezione del documento fornisce una visione d'insieme del prodotto e del suo contesto.

3.1 Prospettiva del prodotto

BugBoard26 rappresenta una soluzione completa per la gestione delle segnalazioni tecniche nell'ambito dello sviluppo software. Il sistema è progettato per facilitare la collaborazione tra i membri del team attraverso un'interfaccia intuitiva e funzionalità mirate all'ottimizzazione dei processi lavorativi.

Posizionandosi come strumento di supporto alle attività di sviluppo, il prodotto si caratterizza per la sua capacità di integrare tracciamento delle issue, comunicazione team e reporting in un'unica piattaforma coerente. L'approccio modulare garantisce scalabilità e adattabilità a diverse esigenze organizzative.

Destinato principalmente a team di sviluppo software e project manager, BugBoard26 si propone di migliorare l'efficienza operativa riducendo la complessità gestionale e promuovendo una collaborazione più strutturata tra i partecipanti ai progetti.

3.2 Fondamenti dei requisiti

3.2.1 Funzionalità del prodotto

Le funzionalità principali del sistema includono:

- **Gestione utenti:** Autenticazione, autorizzazione e gestione dei profili
- **Gestione issue:** Creazione, modifica e tracciamento delle issue
- **Reporting:** Dashboard amministrative e report analitici
- **Ricerca e filtri:** Ricerca avanzata e filtraggio delle issue

3.2.2 Stakeholder

In base a ciò che si evince nel documento di **richiesta attività fornitori esterni**, sono state individuate le seguenti categorie principali di stakeholder:

- **Utenti Standard**

Sono i principali utilizzatori della piattaforma. Utilizzano BugBoard26 per gestire le issue, creare nuove segnalazioni di bug, monitorarne lo stato e aggiungere allegati. Hanno un ruolo operativo e sono interessati alla semplicità d'uso, alla rapidità di consultazione e alla chiarezza dell'interfaccia.

- **Utenti Amministratori**

Hanno il compito di supervisionare l'avanzamento dei lavori, verificare lo stato delle issue e generare report di sintesi tramite la dashboard. Sono interessati a funzionalità di analisi e controllo, in particolare alla tracciabilità delle attività e all'affidabilità dei dati visualizzati. Gestiscono la piattaforma da un punto di vista tecnico: configurazione, creazione e gestione degli account, controllo dei permessi.

- **Committenza**

È l'organizzazione o il gruppo che ha commissionato lo sviluppo del sistema BugBoard26: SoftEngUniNA. Definisce i requisiti iniziali, approva le specifiche e valuta il rispetto degli obiettivi funzionali e non funzionali stabiliti nel documento.

- **Team di manutenzione**

È il gruppo di esperti che manutiene il sistema. Garantisce che la piattaforma sia correttamente funzionante durante il suo ciclo di vita.

3.2.3 Vincoli

Vincoli architetturali

- Back-end e front-end devono essere componenti indipendenti
- Il front-end deve poter essere sostituito senza modifiche al back-end

Vincoli tecnologici

- Utilizzo obbligatorio di linguaggi Object-Oriented
- Utilizzo di tool CASE per la progettazione (es. draw.io)
- Persistenza dati gestita esclusivamente dal back-end

Vincoli di sicurezza

- Autenticazione obbligatoria per accedere al sistema
- Gestione sicura delle credenziali
- Controllo degli accessi basato sui ruoli
- Preservazione dell'integrità e segretezza dei dati

3.3 Trasformazione dei bisogni in requisiti

In seguito alla precedente sezione in cui vengono definiti i **fondamenti dei requisiti** è possibile trasformare i bisogni descritti nella traccia di richiesta attività in requisiti. Quindi, la seguente tabella mostra come i principali bisogni degli stakeholder siano stati tradotti in requisiti del sistema **BugBoard26**:

Bisogno	Requisito
Gli utenti devono poter segnalare bug e richieste in modo semplice	RF-004 (Creazione issue), RF-005 (Allegati e priorità), RF-006 (Tipologie issue)
L'amministratore deve avere una panoramica delle attività e generare report	RF-011 (Dashboard amministratore), RF-012 (Metriche dashboard), RF-013 (Report mensili), RNF-001 / RNF-002 (Preselezioni)
Un amministratore deve creare e gestire i profili utenti	RF-001 (Login utente), RF-002 (Account amministratore predefinito), RF-003 (Creazione utenti)
Deve essere possibile cercare, filtrare e ordinare le issue in base a criteri	RF-008 (Vista riepilogativa), RF-009 (Filtraggio issue), RF-010 (Ordinamento issue)
Personalizzazione accessibile e semplice degli issue	RF-007 (Etichette personalizzabili), RNF-005 / RNF-006 (Usabilità)
Il sistema deve essere affidabile, sicuro e manutenibile nel tempo	RNF-003 (Disponibilità sistema), RNF-004 (Integrità dati), RNF-005 / RNF-006 (Usabilità), RNF-007 (Sicurezza), RNF-008 / RNF-009 (Manutenibilità), RNF-010 / RNF-011 (Portabilità)
Gli utenti autenticati devono poter accedere al sistema in modo sicuro	RF-001 (Login utente), RNF-007 (Hashing password)
Le funzionalità principali devono essere facilmente accessibili e intuitive	RNF-005 (Interfaccia intuitiva), RNF-006 (Accessibilità funzionalità principali)
Il sistema deve operare con tempi di risposta accettabili	RNF-001 (Tempo di risposta operazioni comuni), RNF-002 (Ricerca in tempo reale)
Il sistema deve mantenere l'indipendenza tra componenti front-end e back-end	RNF-010 (Indipendenza back-end/front-end), RNF-011 (Containerizzazione)
Il codice deve essere ben documentato e manutenibile	RNF-008 (Documentazione codice), RNF-009 (Architettura estensibile)

Tabella 1: Mappatura bisogni - requisiti del sistema

3.4 Attori del sistema

Di seguito si descrivono i ruoli degli attori del sistema e le loro responsabilità, in conformità alle richieste del committente.

Utente

Descrizione: Membro del team di sviluppo

Responsabilità: Gestione delle issue

Privilegi:

- Creazione di nuove issue
- Modifica delle issue assegnate
- Visualizzazione di tutte le issue
- Ricerca e filtraggio issue

Scenario: Segnalazione problemi, aggiornamento stato issue assegnate, collaborazione su issue di team

Amministratore

Descrizione: *Specializzazione* dell'Utente con privilegi di gestione del sistema

Responsabilità: Gestione utenti, configurazione sistema, accesso a tutte le funzionalità

Privilegi:

- Creazione e gestione di tutti gli utenti
- Accesso alla dashboard amministrativa

Scenario: Monitoraggio attività del team, generazione report

Team

Descrizione: Team di sviluppo composto da Utenti della piattaforma

Responsabilità: Sviluppo software collaborativo facilitato dal sistema stesso

Scenario: Gestione e collaborazione collettiva degli issue

Persona non registrata

Descrizione: Persona selezionata da un Amministratore per cui si verifica successivamente la creazione di un proprio account personale nel sistema

Responsabilità: Futuro Utente della piattaforma

Scenario: Nuovo membro del Team di sviluppo

4 Requisiti Specifici

Questo capitolo è presente per descrivere in dettaglio i requisiti funzionali e non funzionali del sistema.

4.1 Requisiti Funzionali

4.1.1 RF01 - Autenticazione e gestione utenti

Identificatore: RF-001

Titolo: Login utente

Descrizione: Il sistema deve permettere il login tramite email e password

Priorità: Alta

Fonte: Traccia progetto, punto 3.1.1

Criteri di accettazione: L'utente può accedere con credenziali valide, accesso negato con credenziali errate

Identificatore: RF-002

Titolo: Account amministratore predefinito

Descrizione: Il sistema deve essere fornito con un account amministratore già attivo con credenziali di default

Priorità: Alta

Fonte: Traccia progetto, punto 3.1.1

Criteri di accettazione: Possibile effettuare login con credenziali amministrative predefinite

Identificatore: RF-003

Titolo: Creazione nuovi utenti

Descrizione: L'amministratore deve poter creare nuovi utenti specificando email, password e tipo (normale/amministrazione)

Priorità: Alta

Fonte: Traccia progetto, punto 3.1.1

Criteri di accettazione: Nuovo utente creato può effettuare il login con le credenziali specificate

4.1.2 RF02 - Creazione e personalizzazione issue

Identificatore: RF-004

Titolo: Creazione issue

Descrizione: Gli utenti autenticati devono poter creare issue indicando almeno titolo e descrizione

Priorità: Alta

Fonte: Traccia progetto, punto 3.1.2

Criteri di accettazione: Issue creata appare nella lista con stato "todo"

Identificatore: RF-005

Titolo: Specifica opzionale priorità e allegati

Descrizione: Il sistema deve permettere di specificare opzionalmente priorità e allegare immagini alle issue

Priorità: Media

Fonte: Traccia progetto, punto 3.1.2

Criteri di accettazione: Issue creata con priorità e/o immagine allegata mostra correttamente questi dati

Identificatore: RF-006

Titolo: Tipologie di issue

Descrizione: Le issue devono essere classificabili in: question, bug, documentation, feature

Priorità: Alta

Fonte: Traccia progetto, punto 3.1.2

Criteri di accettazione: Ogni issue creata ha una tipologia assegnata correttamente

Identificatore: RF-007

Titolo: Etichette personalizzabili

Descrizione: Il sistema deve permettere di associare un numero variabile di etichette personalizzabili alle issue di tipo "bug"

Priorità: Media

Fonte: Traccia progetto, punto 3.1.10

Criteri di accettazione: Bug può avere multiple etichette assegnate

4.1.3 RF03 - Visualizzazione e ricerca issue

Identificatore: RF-008

Titolo: Vista riepilogativa issue

Descrizione: Il sistema deve offrire una vista riepilogativa di tutte le issue

Priorità: Alta

Fonte: Traccia progetto, punto 3.1.3

Criteri di accettazione: Tutte le issue sono visualizzabili in una vista organizzata

Identificatore: RF-009

Titolo: Filtraggio issue

Descrizione: Deve essere possibile filtrare le issue per tipologia, stato, priorità e altri parametri

Priorità: Media

Fonte: Traccia progetto, punto 3.1.3

Criteri di accettazione: Le issue possono essere filtrate secondo i criteri specificati

Identificatore: RF-010

Titolo: Ordinamento issue

Descrizione: Deve essere possibile ordinare le issue per tipologia, stato, priorità e altri parametri

Priorità: Media

Fonte: Traccia progetto, punto 3.1.3

Criteri di accettazione: Le issue possono essere ordinate secondo i criteri specificati

4.1.4 RF04 - Dashboard e reporting

Identificatore: RF-011

Titolo: Dashboard amministratore

Descrizione: Gli amministratori devono avere accesso a una dashboard che mostri informazioni aggregate

Priorità: Alta

Fonte: Traccia progetto, punto 3.1.7

Criteri di accettazione: Dashboard accessibile e mostra dati aggregati

Identificatore: RF-012

Titolo: Metriche dashboard

Descrizione: La dashboard deve mostrare: numero di issue aperte, tempo medio di risoluzione

Priorità: Media

Fonte: Traccia progetto, punto 3.1.7

Criteri di accettazione: Tutte le metriche specificate sono visualizzate nella dashboard

Identificatore: RF-013

Titolo: Report mensili

Descrizione: Gli amministratori devono poter visualizzare report mensili sull'attività del team con metriche aggregate e dettagliate per utente

Priorità: Bassa

Fonte: Traccia progetto, punto 3.1.17

Criteri di accettazione: Report mensili generabili e mostrano metriche richieste

4.2 Requisiti Non Funzionali

4.2.1 RNF01 - Prestazioni

Identificatore: RNF-001

Titolo: Tempo di risposta operazioni comuni

Descrizione: Le operazioni comuni (login, visualizzazione liste, creazione issue) devono completarsi in tempi di risposta accettabili

Priorità: Alta

Tipo: Prestazione

Criteri di accettazione: Test di performance confermano tempi di risposta

Identificatore: RNF-002

Titolo: Ricerca in tempo reale

Descrizione: La ricerca delle issue deve restituire risultati in tempo reale

Priorità: Media

Tipo: Prestazione

Criteri di accettazione: Ricerche completano entro 1 secondo

4.2.2 RNF02 - Affidabilità

Identificatore: RNF-003

Titolo: Disponibilità sistema

Descrizione: Il sistema deve garantire un uptime del 99% durante gli orari lavorativi

Priorità: Alta

Tipo: Affidabilità

Metrica: Uptime $\geq 99\%$

Criteri di accettazione: Monitoraggio conferma disponibilità

Identificatore: RNF-004

Titolo: Integrità dati

Descrizione: Il sistema deve garantire l'integrità dei dati in caso di failure

Priorità: Alta

Tipo: Affidabilità

Metrica: Nessuna perdita dati in condizioni normali

Criteri di accettazione: Test di recovery confermano integrità dati

4.2.3 RNF03 - Usabilità

Identificatore: RNF-005

Titolo: Interfaccia intuitiva

Descrizione: L'interfaccia deve essere utilizzabile senza training formale per utenti con esperienza in sistemi di issue tracking

Priorità: Alta

Tipo: Usabilità

Metrica: Task completion rate elevata per tutti i tipi di utilizzatori

Criteri di accettazione: Test utilizzatore confermano facilità d'uso

Identificatore: RNF-006

Titolo: Accessibilità funzionalità principali

Descrizione: Le funzionalità principali devono essere accessibili facilmente

Priorità: Media

Tipo: Usabilità

Criteri di accettazione: Analisi navigazione conferma accessibilità

4.2.4 RNF04 - Sicurezza

Identificatore: RNF-007

Titolo: Hashing password

Descrizione: Le password devono essere memorizzate in forma hashata utilizzando algoritmi sicuri.

Priorità: Alta

Tipo: Sicurezza

Metrica: Password non memorizzate in chiaro

Criteri di accettazione: Analisi di sicurezza conferma assenza leakage

4.2.5 RNF05 - Manutenibilità

Identificatore: RNF-008

Titolo: Documentazione software

Descrizione: Il software deve essere ben documentato in tutte le sue fasi, per garantire chiarezza e trasparenza

Priorità: Media

Tipo: Manutenibilità

Metrica: Presenza documento per ogni artefatto prodotto

Criteri di accettazione: Documenti prodotti adempiono al loro scopo di descrizione correttamente

Identificatore: RNF-009

Titolo: Architettura estendibile

Descrizione: L'architettura deve permettere l'aggiunta di nuove funzionalità senza modifiche massive alla stessa

Priorità: Media

Tipo: Manutenibilità

Metrica: Modularità e basso accoppiamento

Criteri di accettazione: Design review conferma estendibilità

4.2.6 RNF06 - Portabilità

Identificatore: RNF-010

Titolo: Indipendenza back-end/front-end

Descrizione: Il back-end deve essere completamente indipendente dal front-end e viceversa

Priorità: Alta

Tipo: Portabilità

Metrica: Front-end sostituibile senza modifiche back-end

Criteri di accettazione: Test di sostituzione componenti confermano indipendenza

Identificatore: RNF-011

Titolo: Containerizzazione

Descrizione: Utilizzo di tecnologie di containerizzazione per il deployment

Priorità: Bassa

Tipo: Portabilità

Criteri di accettazione: Software containerizzato disponibile all'uso

5 Modellazione dei casi d'uso

In questa sezione del documento vengono presentati i casi d'uso principali della piattaforma.

5.1 Descrizione casi d'uso principali

5.1.1 UC1 - Accedi alla piattaforma

Attori: Utente

Precondizioni: L'utente possiede credenziali valide

Flusso principale:

1. L'utente accede alla pagina di login
2. L'utente inserisce email e password
3. Il sistema verifica le credenziali
4. Il sistema autentica l'utente e reindirizza alla home page

Postcondizioni: L'utente è autenticato e può accedere alle funzionalità

5.1.2 UC2 - Crea nuova utenza

Attori: Amministratore, persona non registrata

Precondizioni: Nessuna

Flusso principale:

1. L'amministratore definisce le credenziali del nuovo utente e le inserisce nella piattaforma
2. Il sistema crea la nuova utenza
3. L'utente riceve le credenziali appena convalidate

Postcondizioni: L'utente può accedere alla piattaforma correttamente

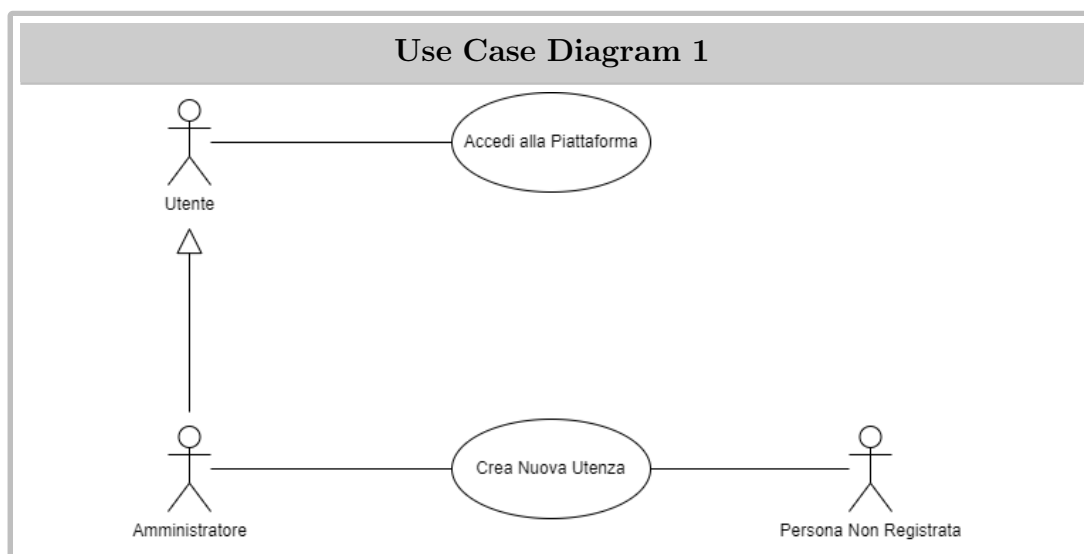


Figura 1: Use Case Diagram #1 - [UC1,UC2]

5.1.3 UC3 - Segnala un issue

Attori: Utente

Precondizioni: L'utente è autenticato

Flusso principale:

1. L'utente accede alla funzione di creazione issue
2. L'utente inserisce titolo e descrizione
3. Opzionalmente, l'utente specifica tipo, priorità e allega un'immagine
4. L'utente conferma la creazione
5. Il sistema crea l'issue con stato "todo"

Postcondizioni: L'issue è creata e visibile nel sistema

5.1.4 UC4 - Specifica una priorità

Attori: Utente

Precondizioni: L'utente sta svolgendo il caso d'uso Segnala un issue

Flusso principale:

1. L'utente sceglie un livello di priorità dell'issue consono
2. La piattaforma assegna il livello di priorità scelto all'issue

Postcondizioni: L'issue presenta la priorità correttamente

5.1.5 UC5 - Aggiungi un'immagine

Attori: Utente

Precondizioni: L'utente sta svolgendo il caso d'uso Segnala un issue

Flusso principale:

1. L'utente sceglie l'immagine da allegare all'issue
2. La piattaforma associa correttamente l'immagine all'issue

Postcondizioni: La piattaforma consente la visualizzazione dell'immagine

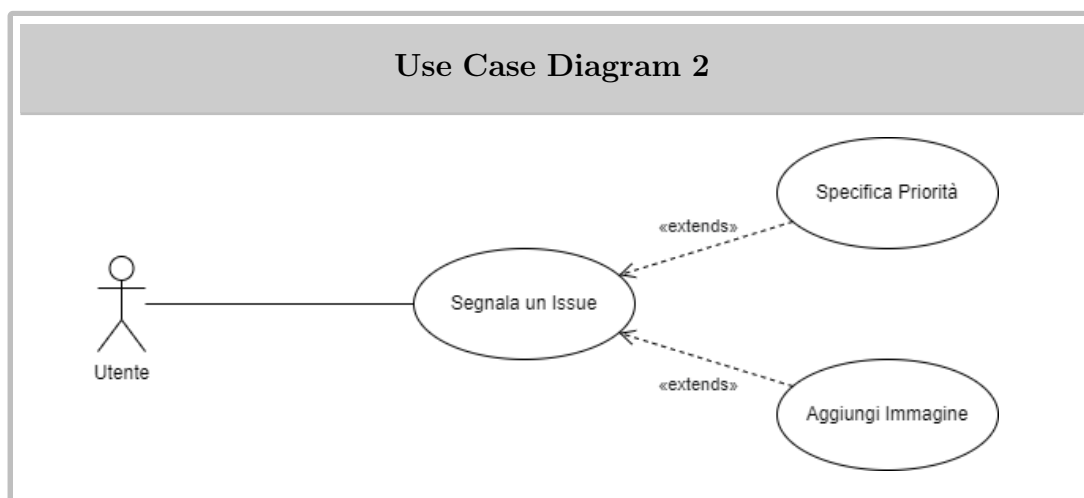


Figura 2: Use Case Diagram #2 - [UC3,UC4,UC5]

5.1.6 UC6 - Visualizza riepilogo

Attori: Utente

Precondizioni: L'utente è autenticato

Flusso principale:

1. L'utente accede alla sezione di riepilogo issue
2. Il sistema seleziona gli issue disponibili per essere mostrati
3. L'utente visualizza i dati correttamente

5.1.7 UC7 - Filtra risultati

Attori: Utente

Precondizioni: L'utente sta svolgendo il caso d'uso Visualizza riepilogo

Flusso principale:

1. L'utente definisce i parametri del filtro
2. La piattaforma estrae correttamente i dati da mostrare

5.1.8 UC8 - Ordina risultati

Attori: Utente

Precondizioni: L'utente sta svolgendo il caso d'uso Visualizza riepilogo

Flusso principale:

1. L'utente definisce i parametri dell'ordinamento
2. La piattaforma riordina correttamente i dati da mostrare

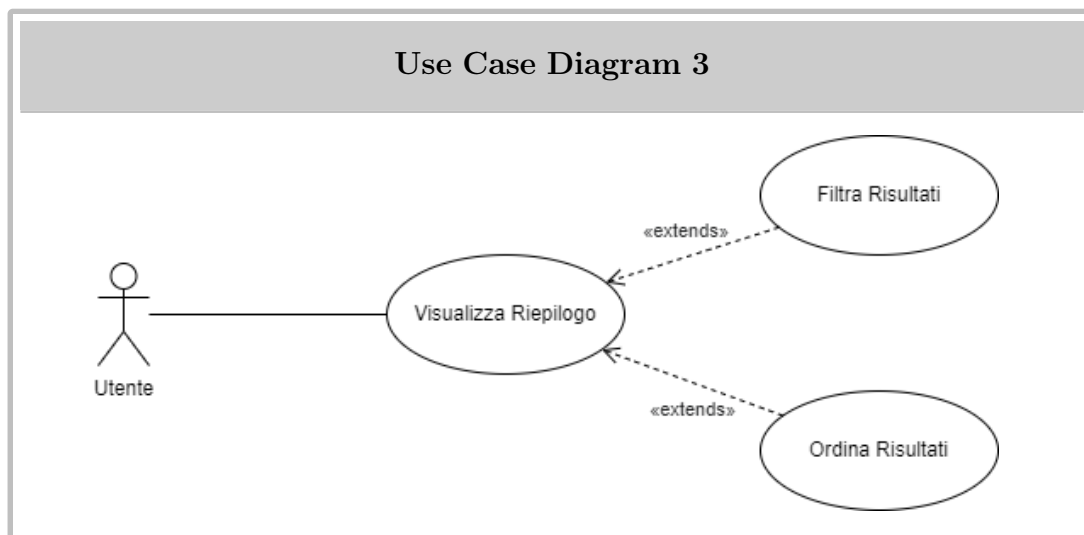


Figura 3: Use Case Diagram #3 - [UC6,UC7,UC8]

5.1.9 UC9 - Visualizza dashboard

Attori: Amministratore

Precondizioni: L'amministrazione è autenticato

Flusso principale:

1. L'amministratore accede alla pagina di resoconto
2. La piattaforma restituisce i dati operativi richiesti in modo coerente

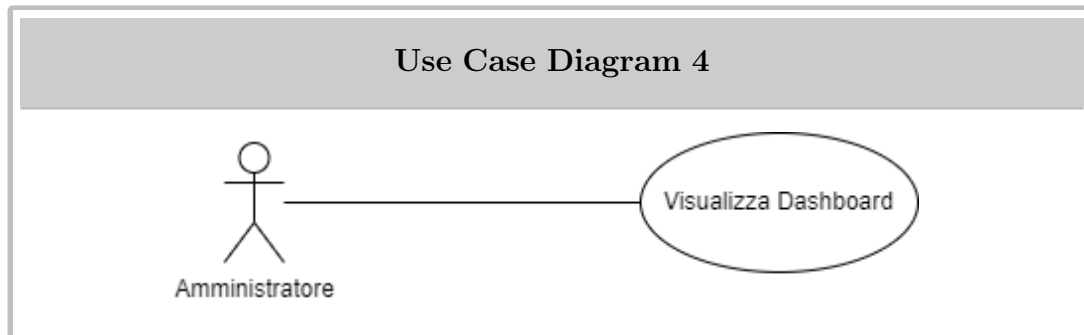


Figura 4: Use Case Diagram #4 - [UC9]

5.1.10 UC10 - Aggiungi etichetta

Attori: Utente

Precondizioni: L'utente è autenticato e ha selezionato un issue

Flusso principale:

1. L'utente determina una o più etichette appropriate e le appone all'issue scelto
2. Il sistema memorizza la combinazione di tag e la relativa assegnazione

Postcondizioni: La piattaforma salva con successo l'aggiunta apportata dall'utente

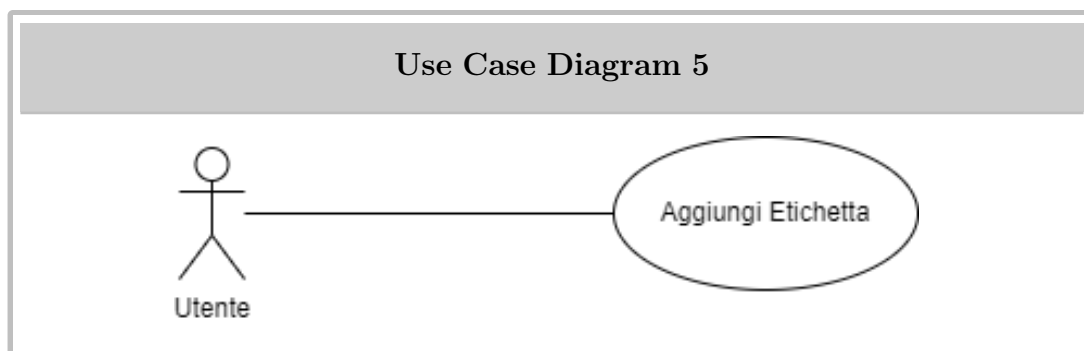


Figura 5: Use Case Diagram #5 - [UC10]

5.1.11 UC11 - Svolgi attività

Attori: Team

Precondizioni: Nessuna

Flusso principale:

1. Il team di sviluppo svolge diverse attività di risoluzione issue
2. La piattaforma tiene traccia correttamente delle evoluzioni del software

5.1.12 UC12 - Visualizza report mensile

Attori: Amministratore

Precondizioni: Almeno un Team ha svolto il caso d'uso Svolgi attività

Flusso principale:

1. L'amministratore seleziona il team di cui visualizzare il report
2. L'amministratore accede alla funzione di visualizzazione report
3. Il sistema mostra il report mensile delle attività del team

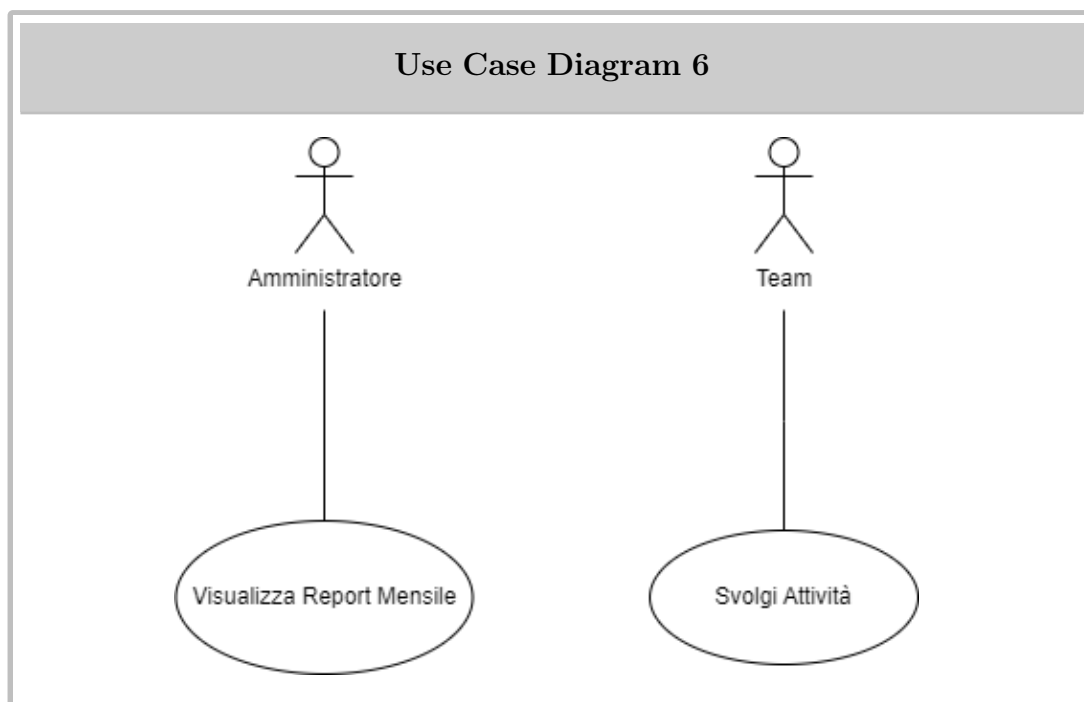


Figura 6: Use Case Diagram #6 - [UC11, UC12]

Use Case Diagram Complessivo

Il presente *Use Case Diagram* (UCD) rappresenta una visione d'insieme delle feature richieste dal committente, modellate secondo i formalismi descritti dall'Unified Modeling Language

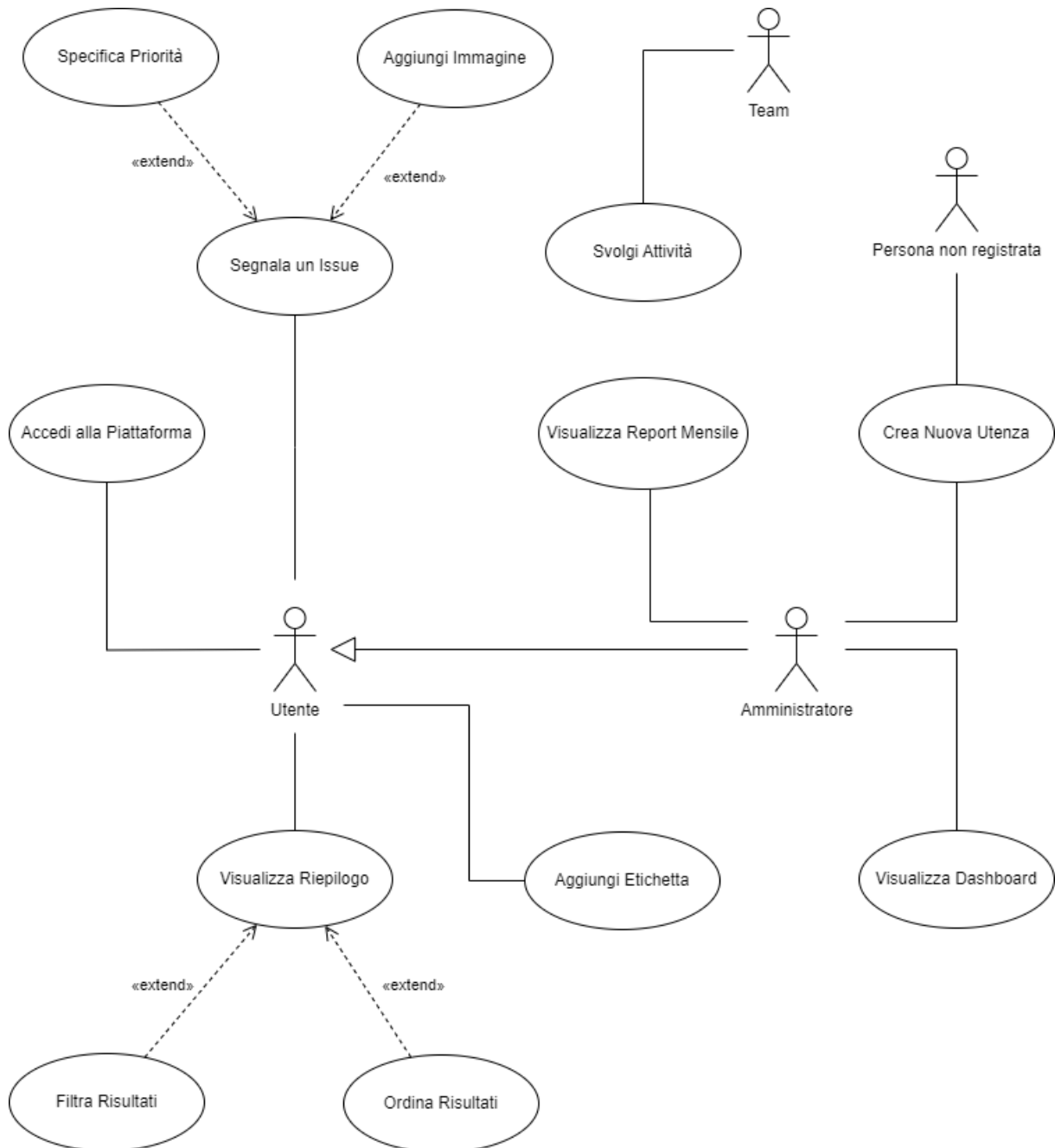


Figura 7: Use Case Diagram #7 - Complessivo

6 Modellazione di un caso d'uso specifico

Di seguito la modellazione estesa dello Use Case **Segnala un Issue** con relative estensioni **Specifica Priorità** e **Aggiungi Immagine** (UC3, UC4, UC5), tramite **formalismo di Cockburn** e **mock-up**.

6.1 Formalismo di Cockburn

USE CASE #3 - Segnala un issue		
Goal in Context	Un utente vuole segnalare un issue sulla piattaforma	
Preconditions	L'utente è autenticato correttamente	
Success End Condition	L'issue è creato correttamente, memorizzato dalla piattaforma e disponibile per la risoluzione	
Failed End Condition	L'issue non viene pubblicato sulla piattaforma	
Primary Actor	Utente	
Trigger	L'utente accede alla schermata per la segnalazione di un issue e clicca sul bottone di “Segnala un issue” in M0	
MAIN SCENARIO		
Step n.	Utente	Sistema
1	Clicca “Segnala un issue” in M0	
2		Mostra M1 - Creazione dell'issue
3	Compila i campi	
4	Clicca “Segnala”	
5		Mostra M2 – L'issue è stato segnalato
6	Clicca “Ok”	
7		Chiude la finestra
EXTENSION #1: Specifica Priorità		
4a	Clicca “Priorità”	
5a		Mostra M3 – Selezione della priorità
6a	Seleziona un'opzione e clicca “Ok”	
7a		Mostra M1 – Ritorna allo step 3 del Main Scenario

EXTENSION #2: Aggiungi immagine		
4b	Clicca “Immagine”	
5b		Mostra M4 – Scelta dell’allegato
6b	Clicca “Sfoglia”	
7b		Mostra selettore file di sistema
8b	Seleziona file	
9b		Torna a M4
10b	Clicca “Ok”	
11b		Mostra M1 – Ritorna allo step 3 del Main Scenario
EXTENSION #3: Errore di mancata compilazione dei campi		
3c	Clicca “Segnala” con uno o più campi vuoti	
4c		Mostra M5 – Errore sulla completezza dei campi
5c	Clicca “Ok”	
6c		Mostra M1 – Ritorna allo step 3 del Main Scenario
EXTENSION #1.1: Errore di mancata scelta della priorità		
6aa	Clicca “Ok” con scelta nulla	
7aa		Mostra M6 – Opzione nulla
8aa	Clicca “Ok”	
9aa		Mostra M3 – Ritorna allo step 6a dell’Extension #1
EXTENSION #2.1: Errore di mancata selezione d’immagine		
6ba	Clicca “Ok” senza aver selezionato un file	
7ba		Mostra M7 – Immagine nulla
8ba	Clicca “Ok”	
9ba		Mostra M4 – Ritorna allo step 6b dell’Extension #2
EXTENSION #2.2: Errore relativo la dimensione dell’immagine		
11bb		Mostra M8 – Immagine di dimensioni troppo grandi
12bb	Clicca “Ok”	
13bb		Mostra M4 – Ritorna allo step 6b dell’Extension #2
EXTENSION #2.3: Errore formato dell’immagine non supportato		
11bc		Mostra M9 – Formato dell’immagine non supportato
12bc	Clicca “Ok”	
13bc		Mostra M4 – Ritorna allo step 6b dell’Extension #2

6.2 Mock-up

M0

Segnala un Issue

M1

Segnala un issue

Titolo:

Descrizione:

Priorità

Immagine

Segnala

M2

L'issue è stato segnalato correttamente!

Ok

M3

Alta

Media

Bassa

Ok

M4

Scegli allegato:

Sfoggia

Ok

M5

Compila tutti i campi!

Ok

M6

Seleziona un'opzione o annulla!

Ok

M7

Seleziona un'immagine o annulla!

Ok

M8

La dimensione è superiore al limite! (100Mb)

Riprova

Ok

M9

Questo formato non è supportato! (.png, .jpg, .jpeg)

Riprova

Ok

Figura 8: Mock-up derivante dalla modellazione del caso d'uso UC3,UC4,UC5

7 Tracciamento dei requisiti

7.1 Tracciamento fonte-requisito

Fonte	Descrizione	Requisiti
Traccia 3.1.1	Sistema autenticazione	RF-001, RF-002, RF-003
Traccia 3.1.2	Creazione issue	RF-004, RF-005, RF-006
Traccia 3.1.3	Vista riepilogativa e ricerca	RF-008, RF-009, RF-010
Traccia 3.1.7	Dashboard amministratore	RF-011, RF-012
Traccia 3.1.10	Etichette personalizzabili	RF-007
Traccia 3.1.17	Report mensili	RF-013

Tabella 3: Tracciamento requisiti dalla traccia di progetto

7.2 Tracciamento requisiti-casi d'uso

Requisito	Casi d'uso	Descrizione
RF-001	UC1	Login utente
RF-002	UC1	Account amministratore predefinito
RF-003	UC2	Creazione nuovi utenti
RF-004	UC3	Creazione issue
RF-005	UC4, UC5	Specifica priorità e allegati
RF-006	UC3	Tipologie di issue
RF-007	UC10	Etichette personalizzabili
RF-008	UC6	Vista riepilogativa issue
RF-009	UC7	Filtraggio issue
RF-010	UC8	Ordinamento issue
RF-011	UC9	Dashboard amministratore
RF-012	UC9	Metriche dashboard
RF-013	UC11, UC12	Report mensili

Tabella 4: Tracciamento requisiti funzionali - casi d'uso

8 Repository GitHub

Il presente artefatto è disponibile pubblicamente su GitHub al seguente link:

github.com/TrialShock26/BugBoard26

La repository contiene:

- Questo documento di specifica dei requisiti software
- I file sorgente e PNG dei diagrammi esposti

Fine del Documento

DOCUMENTO DI SPECIFICA DEI REQUISITI SOFTWARE INGEGNERIA DEL SOFTWARE

Si ringrazia il lettore per l'attenzione

Mario Majorano	N86005035
Luca Sanselmo	N86005147

Anno Accademico 2025/2026



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II