

Indice

1	Introduzione	2
1.1	Scopo	2
1.2	Obiettivi	2
1.3	Definizioni, acronimi e abbreviazioni	2
1.3.1	Definizioni	2
2	Modellazione tramite Use Case Diagram	5
2.1	Sistema di Autenticazione	5
2.2	Segnalazione Issue	5
2.3	Vista Riepilogativa Delle Issue	6
2.4	Dashboard Amministratore	6
2.5	Archiviazione Bug	7
2.6	Contrassegna Bug Duplicato	7
3	Caratterizzazione degli Utenti	8
4	Requisiti Specifici	9
4.1	Requisiti Funzionali	9
4.1.1	Autenticazione	9
4.1.2	Crea Utente	9
4.1.3	Segnala Issue	9
4.1.4	Vista Riepilogativa Delle Issue	9
4.1.5	Visualizza Dashboard	9
4.1.6	Gestisci Bug	9
4.1.7	Visualizza Bug Archiviati	9
4.1.8	Contrassegna Bug come duplicato	9
4.2	Requisiti Non Funzionali	10
4.2.1	Prestazioni	10
4.2.2	Affidabilità	10
4.2.3	Sicurezza	10
4.2.4	Usabilità	10
4.2.5	Scalabilità	10
4.2.6	Manutenibilità	10
4.2.7	Portabilità	10
4.2.8	Compatibilità	10
4.2.9	Testabilità	10
4.2.10	Disponibilità	10
5	Use Case Formalizzati	11
5.1	Contrassegna Bug Duplicato	11
5.1.1	Cockburn's Table	11
5.1.2	Visuale Mock-Up	12

1 Introduzione

1.1 Scopo

Questo documento si rivolge agli utenti che lavorano in team ed è volto allo scopo di illustrare il funzionamento del software “BugBoard26”, un sistema che consente di collaborare attraverso una piattaforma per la gestione collaborativa di issue in progetti software.

1.2 Obiettivi

Il prodotto software che si intende realizzare prende il nome di “BugBoard26”. Questo deve svolgere le seguenti funzionalità:

- 1) Prevede un sistema di autenticazione, consentendo di accedere alle sole funzionalità che gli competono.
- 2) Tutti gli utenti autenticati possono segnalare una issue indicando almeno un titolo e una descrizione.
- 3) Vista riepilogativa delle issue, con possibilità di filtrare o ordinare i risultati.
- 4) Dashboard per amministratori che mostri informazioni aggregate sui bug.
- 5) Gestione di un bug segnalato con relativa cronologia di modifiche.
- 6) Modalità “read-only” per utenti esterni.
- 7) Report mensile sull’attività dei team.

BugBoard26 non dovrà gestire:

- 1) Chat tra utenti di team.
- 2) Personalizzazione grafica dell’interfaccia da parte dell’utente.
- 3) Allegati multimediali di grandi dimensioni
- 4) Gestione di ruoli o permessi personalizzabili, oltre a quelli previsti.

BugBoard26 ha come obiettivo principale quello di affiancare team di sviluppo, fungendo da piattaforma di segnalazione di problemi relativi a un progetto e di monitorarne lo stato, agevolando dunque il lavoro dei team.

1.3 Definizioni, acronimi e abbreviazioni

1.3.1 Definizioni

- Issue – Indica un elemento che rappresenta qualcosa da risolvere o da gestire all’interno di un progetto.
- Amministratore – Indica un ruolo “speciale” di utente, che ha uno scopo di gestione, come creare nuove utenze ecc...
- Utente – Indica un’utenza “normale”, che può usufruire di tutte le funzionalità che gli competono, come la segnalazione di issue ecc...
- Todo – Indica un’attività ancora da completare o un promemoria, in questo caso, la gestione di una issue.
- Bug – È una issue per segnalare un malfunzionamento.
- Question – È una issue usata per richieste di chiarimenti.
- Feature – È una issue per indicare la richiesta o il suggerimento di nuove funzionalità.
- Documentation - È una issue per segnalare problemi relativi alla documentazione
- Team – È un gruppo di persone che collaborano insieme per raggiungere un obiettivo comune. Ci possono essere in un team dei ruoli specifici, come sviluppatori, designer, tester o project manager, lavorano tutti in modo coordinato per portare a termine attività o progetti.

Documento Di Specifica Dei Requisiti (BugBoard26)

- Dashboard – È un'interfaccia che mostra in modo chiaro e riassuntivo le principali informazioni sui bug del sistema.
- CSV – È un formato di file di testo in cui i dati sono organizzati in righe e colonne, separati da virgole o altri simboli.
- Excel – Si riferisce ai file creati con Microsoft Excel, un programma per fogli di calcolo; questi file contengono tabelle di dati, formule, grafici e altre funzionalità per analizzare e visualizzare informazioni.
- PDF - È un formato di file universale che mantiene intatto l'aspetto del documento (testo, immagini, impaginazione) su qualsiasi dispositivo.
- Modalità read-only – È una modalità di accesso che permette solo la visualizzazione dei contenuti per utenti esterni, in questo caso bug e commenti, senza possibilità di modificarli.
- Utenti esterni – Persone che non fanno parte del team interno di sviluppo o gestione del progetto, ma che hanno interesse a seguirne l'andamento.
- Report mensili – I Report mensili sull'attività del team sono documenti che riassumono, con cadenza mensile, le prestazioni e le attività svolte dal gruppo di lavoro. Contengono metriche come:
 - o il numero di bug aperti, chiusi o ancora in gestione,
 - o il tempo medio di risoluzione,
 - o altri dati utili, sia in forma aggregata (cioè complessiva per tutto il team) sia dettagliata per singolo utente.

Servono per monitorare l'efficienza, individuare eventuali criticità e valutare le performance del team nel tempo.

- Allegato – È un documento aggiuntivo che viene unito o collegato a un testo principale, nel nostro caso ad una segnalazione, per fornire informazioni di supporto o approfondimenti. Può contenere dati, tabelle, immagini, grafici o documenti tecnici che completano o chiariscono il contenuto principale, senza appesantirlo.
- BugBoard26 – È il nome usato per identificare il software.
- Software – Insieme di programmi e istruzioni che permettono di svolgere determinate funzioni o compiti.
- E-mail – Strumento informatico per l'invio della posta elettronica.
- Password – Insieme di caratteri alfanumerici che abilitano l'accesso ad aree riservate a determinati utenti.
- Chat – È un sistema di comunicazione in tempo reale che permette a due o più persone di scambiarsi messaggi istantanei via Internet.
- Interfaccia utente – È la parte di software con cui l'utente interagisce direttamente. Include tutti gli elementi visivi e di controllo, come menù, pulsanti, finestre, icone e moduli, che permettono di usare il programma in modo semplice ed efficace.
- File – È un contenitore digitale che memorizza dati o informazioni su un computer o altro dispositivo. Può contenere testo, immagini, video, audio, programmi o altri tipi di dati.
- Autenticazione - È il processo tramite il quale un sistema verifica l'identità di un utente prima di permettergli l'accesso. Serve a garantire che solo persone autorizzate possano entrare in un'applicazione, un sito o un servizio.

- Use Case Diagram - È un tipo di diagramma UML che serve a rappresentare graficamente le funzionalità principali di un sistema viste dal punto di vista dell'utente. Mostra chi (gli attori) interagisce con il sistema e quali azioni (i casi d'uso) può compiere.
- UML - (Unified Modeling Language) è un linguaggio di modellazione standard usato per rappresentare graficamente la struttura, il comportamento e le interazioni di un sistema software.

2 Modellazione tramite Use Case Diagram

2.1 Sistema di Autenticazione

Deve essere implementato un sistema di autenticazione semplice e sicuro, basato su email e password. Le informazioni gestite dall'applicazione sono critiche per l'azienda, ed è fondamentale preservarne l'integrità e la segretezza. Il sistema viene fornito con un account da amministratore già attivo, con credenziali di default. Un amministratore può creare ulteriori utenze, specificando una email, una password, e indicando se quell'utenza sarà "normale" oppure "di amministrazione".

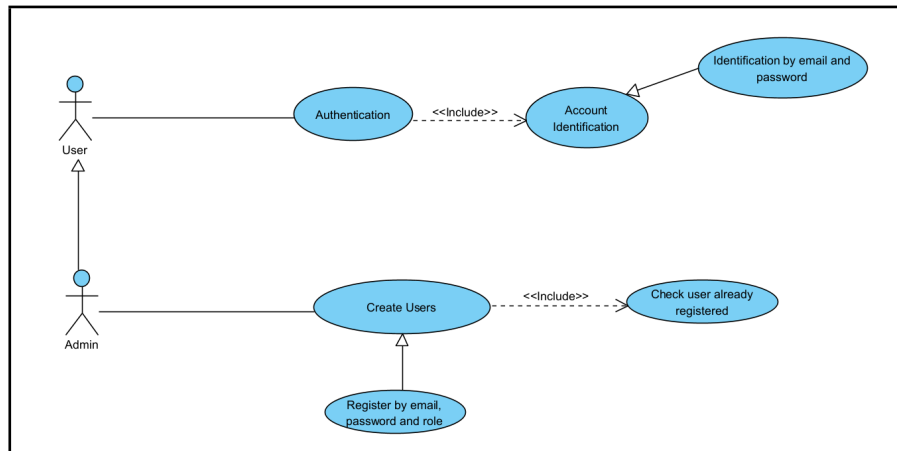


Figura 1: Sistema di Autenticazione (Utenti della funzionalità: Utente, Amministratore)

2.2 Segnalazione Issue

Tutti gli utenti autenticati possono segnalare una issue indicando almeno un titolo e una descrizione. Alcuni utenti potrebbero voler specificare anche una priorità e sarebbe gradita la possibilità di allegare un'immagine. Le issue possono essere di diverso tipo: Question (per richieste di chiarimenti), bug (per segnalare malfunzionamenti), Documentation (per segnalare problemi relativi alla documentazione), e Feature (per indicare la richiesta o il suggerimento di nuove funzionalità). Le issue create sono inizialmente nello stato "Todo".

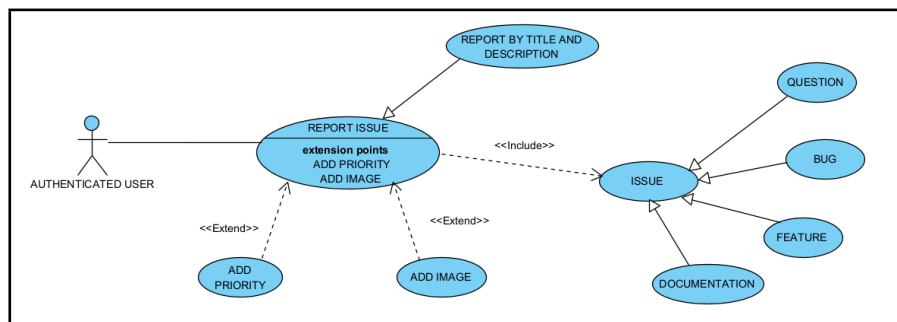


Figura 2: Segnalazione Issue (Utenti della funzionalità: Utente)

2.3 Vista Riepilogativa Delle Issue

Il sistema deve offrire una vista riepilogativa delle issue, con la possibilità di filtrare o ordinare i risultati in base a criteri come tipologia, stato, priorità o altri parametri rilevanti.

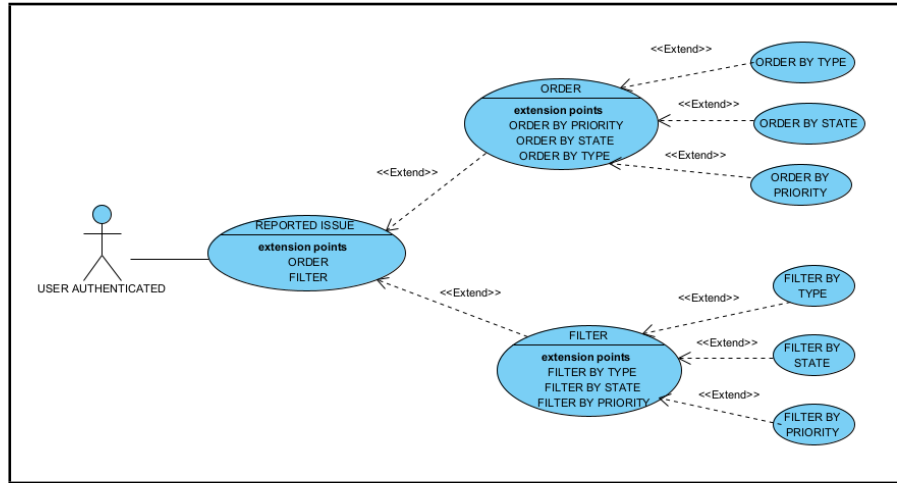


Figura 3: Vista Riepilogativa Delle Issue (Utenti della funzionalità: Utente)

2.4 Dashboard Amministratore

È richiesta una Dashboard per gli amministratori che mostri informazioni aggregate sui bug, come il numero di bug aperti, quelli assegnati per utente, il tempo medio di risoluzione (aggregato e per utente) e altri dati utili.

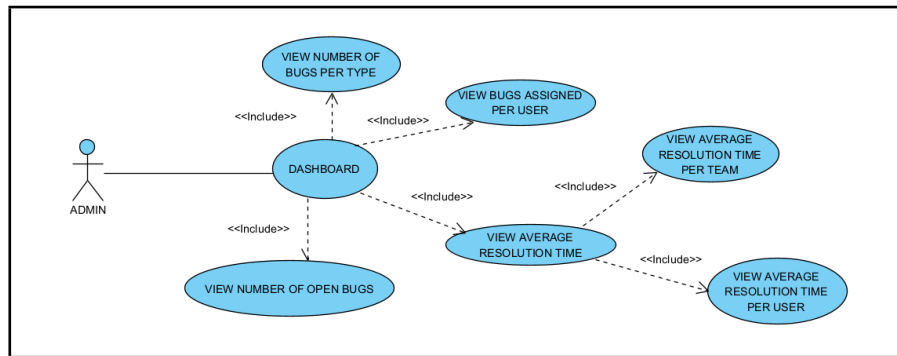


Figura 4: Dashboard Admin (Utenti della funzionalità: Amministratore)

2.5 Archiviazione Bug

Deve essere possibile archiviare un bug, in modo che non appaia più nelle liste principali ma resti comunque accessibile da una pagina dedicata. Soltanto gli amministratori possono archiviare un bug.

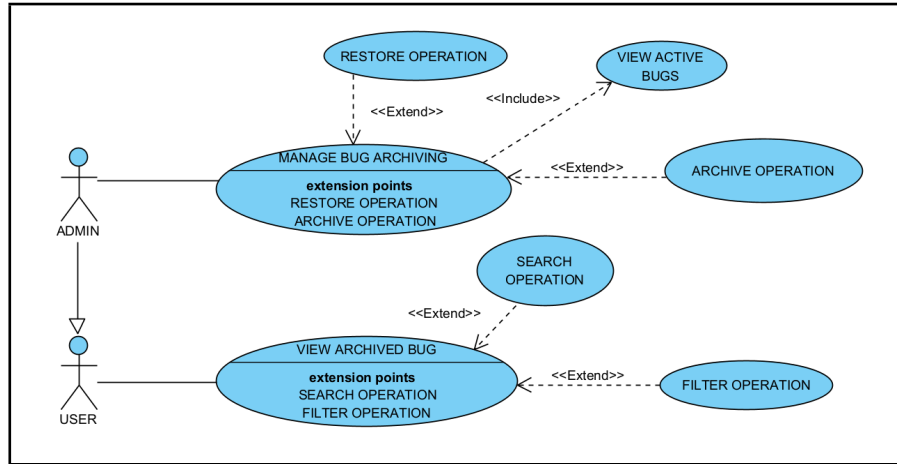


Figura 5: Archiviazione Bug (Utenti della funzionalità: Utente, Amministratore)

2.6 Contrassegna Bug Duplicato

Un amministratore può contrassegnare un bug come duplicato di un altro. Questo comporta la chiusura immediata del bug duplicato.

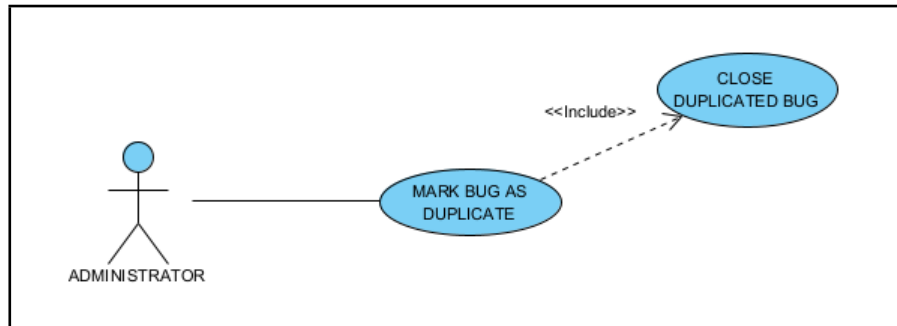


Figura 6: Contrassegna Bug Duplicato (Utenti della funzionalità: Amministratore)

3 Caratterizzazione degli Utenti

Gli utenti di BugBoard26 devono avere una conoscenza base dell'uso del computer. In base al ruolo, ogni utente può accedere esclusivamente a determinate funzionalità ed ai dati di loro competenza. Gli utenti di BugBoard26 sono:

- Amministratore - È un utente speciale che ha accesso completo a tutte le funzionalità del sistema e può visualizzare dati aggregati e statistici attraverso una Dashboard di controllo. Ha dunque in particolare modo un ruolo di gestione.
- Utente - È un classico utente che rappresenta un membro del team di sviluppo che accede al sistema tramite credenziali personali (email e password). Non dispone di permessi amministrativi e può operare solo sulle issue di sua competenza.
- Utente esterno - È un soggetto non appartenente al team interno di sviluppo, ma interessato a consultare lo stato dei bug o dei commenti (ad esempio stakeholder o clienti). Account di questo tipo sono creati e gestiti dagli amministratori e servono per garantire trasparenza e monitoraggio verso soggetti esterni all'organizzazione.

4 Requisiti Specifici

4.1 Requisiti Funzionali

BugBoard26 prevede le seguenti funzionalità:

4.1.1 Autenticazione

All'avvio di BugBoard26, l'utente, che sia un semplice utente o un amministratore, deve autenticarsi, tramite una coppia di e-mail e una password, sarà poi il sistema a verificare che questa coppia sia valida, dando all'utente/amministratore l'accesso alle funzionalità che gli competono.

4.1.2 Crea Utente

Un amministratore può scegliere di creare una nuova utenza, andando a specificare una e-mail, una password e un ruolo (utente normale o amministratore), il sistema poi verificherà che quella e-mail non sia già registrata.

4.1.3 Segnala Issue

Un utente può segnalare una nuova issue andando a specificare almeno un titolo ed una descrizione; quest'ultimo può anche voler specificare una priorità ed allegare un'immagine. Le issue possono essere di diverso tipo: Question, bug, Documentation, e Feature.

4.1.4 Vista Riepilogativa Delle Issue

Il sistema deve dare la possibilità ad un utente o amministratore di poter visualizzare una vista riepilogativa delle issue, con la possibilità di filtrare o ordinare i risultati.

4.1.5 Visualizza Dashboard

Il sistema deve dare la possibilità ad un amministratore di visualizzare una dashboard, che mostri informazioni aggregate sui bug, come il numero di bug aperti, quelli assegnati per utente, il tempo medio di risoluzione ed altri dati utili.

4.1.6 Gestisci Bug

Un amministratore può visualizzare la lista dei bug presenti, scegliere se archiviare un bug, in modo che non appaia più nelle liste principali ma resti comunque accessibile da una pagina dedicata; inoltre oltre che archiviare un bug, può anche decidere di recuperarne uno precedentemente archiviato.

4.1.7 Visualizza Bug Archiviati

Un utente visualizzare la lista dei bug archiviati, con la possibilità di cercare o filtrare in maniera specifica.

4.1.8 Contrassegna Bug come duplicato

Un amministratore può contrassegnare un bug come duplicato di un altro, in modo tale da non riempire la lista con dei duplicati; la contrassegnazione comporta la chiusura immediata del bug duplicato.

4.2 Requisiti Non Funzionali

BugBoard26 rispetta le seguenti caratteristiche qualitative e vincoli:

4.2.1 Prestazioni

Il sistema deve garantire tempi di risposta immediati per le operazioni principali (login, creazione issue, aggiornamento stato). Il caricamento della dashboard non deve rallentare eccessivamente anche in presenza di molteplici bug registrati.

4.2.2 Affidabilità

Il sistema deve garantire integrità e consistenza dei dati anche in caso di crash o interruzioni. Devono essere previsti meccanismi di rollback e gestione eccezioni a livello di rete e applicazione.

4.2.3 Sicurezza

Tutte le comunicazioni devono avvenire tramite connessioni sicure (HTTPS). Le password degli utenti devono essere hashate con un algoritmo sicuro e mai archiviate in chiaro. Gli utenti possono accedere e modificare solo le informazioni coerenti con il proprio ruolo (utente normale, amministratore, stakeholder in sola lettura).

4.2.4 Usabilità

L'interfaccia utente deve essere intuitiva e coerente, accessibile anche da dispositivi con risoluzione più bassa. Le funzionalità principali devono essere raggiungibili con meno interazioni possibili. L'applicazione deve essere facilmente utilizzabile anche da utenti esterni in modalità "readonly".

4.2.5 Scalabilità

Il sistema deve poter gestire centinaia o migliaia di utenti simultanei senza degradare le prestazioni. L'architettura modulare deve permettere l'aggiunta di nuove funzionalità senza richiedere modifiche al back-end.

4.2.6 Manutenibilità

Il codice sorgente deve essere commentato, modulare e aderente ai principi SOLID. Devono essere utilizzati strumenti di versioning (es. GitHub) per il tracciamento delle modifiche e per la collaborazione. È richiesto l'impiego di un CASE tool (es. Visual Paradigm) per favorire la documentazione e l'evoluzione del progetto.

4.2.7 Portabilità

L'interfaccia front-end deve essere compatibile con i principali browser moderni (Google Chrome, Mozilla Firefox, Microsoft Edge). Il back-end deve poter essere eseguito su diversi ambienti server (Windows, Linux) senza modifiche sostanziali.

4.2.8 Compatibilità

Il front-end deve interagire con il back-end esclusivamente tramite API REST, così da poter essere sostituito o aggiornato indipendentemente dal server.

4.2.9 Testabilità

Il sistema deve consentire l'esecuzione di test automatici di unità e deve supportare l'analisi della qualità del codice tramite strumenti come ad esempio SonarQube.

4.2.10 Disponibilità

Il sistema deve poter essere distribuito su infrastrutture cloud pubbliche (es. AWS o Azure) e accessibile via Internet in modalità continua.

5 Use Case Formalizzati

5.1 Contrassegna Bug Duplicato

5.1.1 Cockburn's Table

USE CASE #X	DUPLICATED BUG		
Goal in Context	L'amministratore vuole segnalare che un bug è un duplicato di un altro già esistente nel sistema, in modo da evitare ridondanza e chiudere automaticamente il duplicato.		
Preconditions	<ul style="list-style-type: none"> - L'amministratore è autenticato nel sistema. - Esiste almeno un bug aperto nel database. - Il bug da contrassegnare non è già chiuso o marcato come duplicato. 		
Success End Condition	Il bug selezionato viene contrassegnato come duplicato di un altro e viene automaticamente chiuso.		
Failed End Condition	Il sistema segnala un errore (es. bug inesistente o bug già chiuso) e nessuna modifica viene applicata.		
Primary Actor	Amministratore		
Trigger	L'amministratore seleziona un bug e sceglie l'opzione "Contrassegna come duplicato".		
Main Scenario	Step n.	Actor 1 (Admin)	System
	1	Accede al sistema di gestione dei bug.	Verifica credenziali e mostra la dashboard dei bug.
	2	Seleziona un bug aperto dall'elenco.	Mostra i dettagli del bug selezionato.
	3	Clicca su "Contrassegna come duplicato".	Mostra una finestra di input per il bug originale.
	4	Inserisce l'ID del bug originale e conferma.	Verifica che il bug esista e che il bug non sia già chiuso.
	5		Aggiorna lo stato del bug selezionato a "Duplicato".
	6		Chiude automaticamente il bug duplicato.
	7		Mostra un messaggio di conferma "Bug contrassegnato come duplicato e chiuso con successo".
Extension A: (ID inesistente)	4a	Inserisce un ID di un bug inesistente.	Mostra un messaggio di errore, es. "L'ID del bug inserito risulta inesistente"
	5a		Riporta l'amministratore alla dashboard dei bug.
Open Issues	<ul style="list-style-type: none"> - Definire se il sistema deve inviare una notifica automatica agli utenti coinvolti nei bug duplicati. - Specificare se l'amministratore può annullare l'operazione di contrassegno. - Chiarire se la chiusura automatica deve aggiornare anche i log o lo storico dei commenti. 		

Figura 7: Tabella Cockburn: Contrassegna Bug Duplicato

5.1.2 Visuale Mock-Up

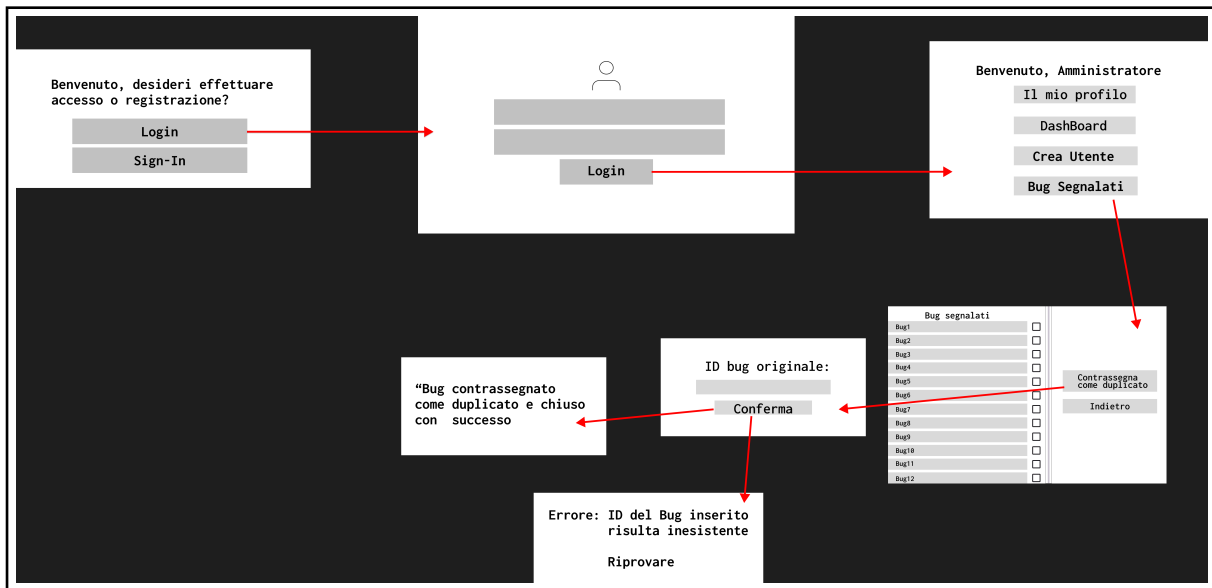


Figura 8: Mock-Up: Contrassegna Bug Duplicato