

Integrated Bug Controlling System

¹G.Radhika Deepthi,²D.Sreeja, ³G .Sushma,⁴J.P.Kashish,
⁵A.Nithyasritha

¹Assistant professor , Department of Computer Science and Engineering, Narayana Engineering College Nellore, Andhra Pradesh, India

²Student, Department of Computer Science and Engineering, Narayana Engineering College Nellore, Andhra Pradesh, India²

³Student, Department of Computer Science and Engineering, Narayana Engineering College Nellore, Andhra Pradesh, India³

⁴Student, Department of Computer Science and Engineering, Narayana Engineering College Nellore, Andhra Pradesh, India⁴

⁵Student, Department of Computer Science and Engineering, Narayana Engineering College Nellore, Andhra Pradesh, India⁵

Abstract: A bug tracking system is a critical component in the software development lifecycle, designed to systematically manage and resolve software defects. It centralizes the process of reporting bugs, providing a structured platform for users and testers to document detailed information about each issue, including reproduction steps, severity, and environmental specifics. allowing for effective prioritization and management based on the bug's impact and severity. The system significantly enhances communication and collaboration among developers, testers, and project managers. It facilitates ongoing discussions and status updates, ensuring all team members are aligned and informed. This improved communication streamlines the resolution process, reducing misunderstandings and delays. Moreover, the system's robust analytics and reporting tools provide insightful data on bug trends and team performance, enabling informed decision-making and strategic planning to prevent future issues.

Keywords : Bug tracking, Bug Assignment , Centralized repository, Severity management, Qualitative Project , Communication and collaboration, Analytics and reporting

I. INTRODUCTION

In the fast-paced and ever-evolving realm of software development, maintaining high standards of quality and reliability is paramount. A critical component of achieving these standards is the efficient management and resolution of software bugs. A Bug Tracking System (BTS) serves as an indispensable tool in this Endeavour, providing a comprehensive framework for recording, monitoring, and addressing software defects throughout the development lifecycle.

The BTS project is conceived to empower development teams with a centralized platform that facilitates meticulous bug management. By systematically logging reported issues, assigning them to appropriate developers, and tracking their progression towards resolution, the system ensures that no bug goes unnoticed or unresolved, ultimately contributing to the delivery of robust and reliable software products. It will incorporate features that allow for the categorization and prioritization of bugs, enabling teams to focus on critical issues that could significantly impact the software's performance or user experience.

Prioritization is another key feature of the BTS, helping teams focus on the most critical issues first to minimize impact on end users and maintain project timelines. we aim to streamline the debugging process, improve productivity, and uphold the integrity of our software projects. The Bug Tracking System (BTS) project is designed to provide a comprehensive solution for managing the entire lifecycle of software bugs, from initial reporting through to final resolution. The system will offer a user-friendly interface for submitting detailed bug reports, capturing essential information such as descriptions, severity levels, and priorities. This ensures that all relevant details are available from the outset, facilitating efficient tracking and resolution.

II . EXISTING SYSTEM

In many software development environments, the existing system for bug tracking often relies on a patchwork of manual processes and disconnected tools. Bugs are typically reported through informal channels such as emails, chat messages, or shared spreadsheets. This approach can result in inconsistent and incomplete documentation of bugs, as different team members may use varying formats and levels of detail. The lack of a standardized method for recording bugs can lead to miscommunication and make it difficult to track the history and status of each issue accurately. The decentralized nature of the existing system means that information about bugs is scattered across multiple platforms. This fragmentation not only consumes valuable time but also increases the risk of important details being overlooked or lost. Without a central repository for bug information, maintaining a comprehensive view of all reported issues becomes challenging, impeding effective management and resolution.

Assigning bugs to developers is often done manually, typically by project managers or team leads, who must manually review and distribute tasks based on their understanding of each team member's workload and expertise. This process can be time-consuming and prone to errors, leading to suboptimal task distribution. Critical bugs might not be prioritized appropriately, and some issues may fall through the cracks, delaying their resolution and potentially impacting the quality of the software product. The scalability of these fragmented systems also becomes a significant concern as projects grow, underscoring the need for a more integrated and efficient bug tracking solution.

Disadvantages

- Manual Logging :

Manual logging involves reporting bugs through informal methods such as emails, chat messages, or shared spreadsheets, lacking a standardized format. This approach leads to inconsistent and incomplete documentation as team members describe bugs in varying levels of detail and formats.

- Decentralized Information :

Decentralized information means that bug details are scattered across multiple platforms such as emails, chat logs, and spreadsheets. This fragmentation makes it difficult to maintain a comprehensive and accessible record of all reported bugs.

- Inefficient Assignment :

Inefficient assignment occurs when bugs are manually distributed to developers by project managers or team leads, without an automated system. This manual process can be time-consuming and prone to human error, resulting in suboptimal task distribution.

- Lack of Prioritization :

The lack of prioritization means there is no systematic way to rank bugs based on their severity and impact on the project. Without proper prioritization, critical bugs that significantly affect functionality or user experience might not be addressed promptly.

- Scalability Issues :

Scalability issues arise as projects grow in size and complexity, overwhelming manual and fragmented bug tracking systems. These systems struggle to handle an increasing volume of bugs, leading to inefficiencies and potential oversight of critical

- Communication gaps:

Communication gaps occur when there is no centralized and standardized system for sharing updates and collaborating on bug resolutions. Without a unified platform, team members rely on disparate communication channels like emails and chats, leading to fragmented and inconsistent information. occur when there is no centralized and standardized system for sharing updates and collaborating on bug resolutions.

III . PROPOSED SYSTEM

The proposed Bug Tracking System aims to address the limitations of existing systems by providing a centralized, integrated, and automated platform for managing and resolving bugs. This system will standardize the process of bug reporting, ensuring that all bugs are documented with essential details such as descriptions, severity levels, steps to reproduce, and relevant attachments. By consolidating all bug information into a single, unified repository, the proposed system enhances accessibility and ensures that team members have a comprehensive view of all reported issues. This centralization facilitates better tracking, reduces the risk of information loss, and ensures consistency in bug documentation. Automated assignment and prioritization are key features of the proposed system.

Bugs will be automatically assigned to developers based on their expertise, current workload, and the severity of the issues. This ensures that critical bugs are addressed promptly while less severe ones are managed accordingly. The system will also provide real-time updates and notifications, enhancing visibility and accountability throughout the bug resolution process. By automating these tasks, the proposed system reduces the manual effort involved in bug management, leading to more efficient task distribution and faster resolution times. Additionally, the proposed system's robust reporting and analytics capabilities provide valuable insights into bug trends, resolution efficiency, and team performance. These data-driven insights enable teams to identify recurring issues, measure the effectiveness of their bug resolution processes, and implement continuous improvements. Enhanced visibility and real-time updates improve accountability and coordination among team members, leading to more timely and effective bug resolutions.

3.1 Advantages

- **Centralized Bug Repository:**

The proposed Bug Tracking System provides a centralized bug repository that consolidates all bug-related information into a single, accessible location. This repository ensures that team members can easily find and reference comprehensive bug details without needing to search through multiple platforms.

- **Standardized Data Format :**

Implementing a standardized data format for bug reporting ensures consistency and completeness in documenting bugs. The proposed system mandates specific fields and structures for bug reports, such as descriptions, severity levels, steps to reproduce, and relevant attachments.

- **Improve Communication and Collaboration**

The proposed Bug Tracking System significantly enhances communication and collaboration among team members by providing a unified platform for sharing updates and discussing issues. With all bug-related information accessible in one place, team members can easily collaborate on resolutions, track the progress of bug fixes.

- **Time and Cost Savings :**

By automating many aspects of bug tracking and resolution, the proposed system significantly reduces the time and effort required to manage bugs. Automation of tasks such as bug assignment, prioritization, and status updates minimizes manual intervention, freeing up developers to concentrate on coding.

- **Historical Data**

The proposed Bug Tracking System maintains a comprehensive history of all bug reports and resolutions, enabling teams to analyze past data and identify patterns. Historical data allows teams to track the evolution of specific bugs, understand their root causes, and implement preventive measures to avoid recurring issues.

- **Assignment and Prioritization**

The proposed system features automated assignment and prioritization, ensuring that bugs are allocated to developers based on their expertise, current workload, and the severity of the issues. This automation streamlines the task distribution process, reducing the manual effort and potential for human error. • Critical bugs are prioritized and assigned promptly, ensuring that high-impact issues are addressed quickly, while less severe bugs are managed accordingly.

System Architecture

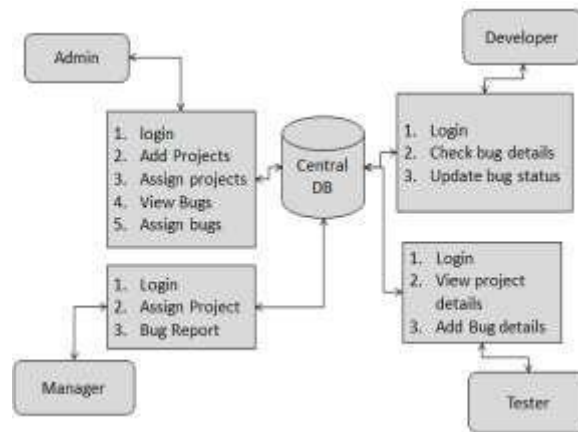


Fig 1 : Architecture

Admin: Login, Add Projects, Assign Projects, View Bugs, Assign Bugs

Developer: Login, Check Bug Details, Update Bug Status

Manager: Login, Assign Project, Bug Report

Tester: Login ,View Project Details, Add Bug Details

Central DB :

Admin, Developer, Manager, and Tester interact with the central database to perform their respective tasks. It also stores all the information in database and facilitates its interaction among the four users.

Interaction Flow:

Administrator: Admin used to add the employee details such as manager, Technical person and tester along with their allotting username and passwords.

Tester: Here at first the tester raises the bug in the certain project, along with its priorities and severity of the bug.

Manager : When ever a bug is raised it goes on the manager module and then he is able to assigning the bug to a technical person to resolve the bug of a concern project lead.

Technical Person: After the assignment of bug details concern tech person can be able to upload the solution status aftermath it goes to tester to verify its resolution on it .

Modules

Admin Module : The Admin module is the backbone of the Bug Tracking System, with overarching control and access to all other modules. Key functionalities include Project Creation and Management, The admin creates projects and assigns them to managers. They can update project details as needed.

Manager Module : The Manager module provides project-specific control and Project Access: Managers have full access to projects assigned by the admin, allowing them to oversee the entire project lifecycle. Managers can assign bugs to developers and testers, ensuring that issues are addressed by the appropriate team members.

Technical person Module : The Tech person module focuses on the execution and resolution of assigned tasks. They can access tasks or bugs assigned to them by the manager. This module provides a list of all assigned tasks for easy tracking and work on resolving the assigned bugs, updating the status and details of each bug as they progress.

Tester Module : Testers are responsible for identifying and reporting bugs. Testers can access projects or bugs assigned to them by the manager. They have a list of all assigned projects for easy navigation. Testers can add new bugs to the system, providing detailed descriptions and reproduction steps. They can then send these bugs back to the manager for further action.

IV. EXPERIMENTAL RESULTS

The process involves meticulous planning, requirements gathering, system design, and deployment, followed by continuous monitoring and improvement. Thoroughly understanding the objectives and scope of the project, engaging with stakeholders, and selecting the appropriate bug tracking tool are fundamental in the planning phase.



Fig 2 : Home page

- The home page of a bug tracking system typically features an intuitive dashboard providing a comprehensive overview of active and resolved issues.
- It often includes search and filter options to help users locate specific bugs or view issues by category, severity, or assigned personnel.

| PROJECT_ID | PROJECT_NAME | VERSION | VENDOR_NAME | MANAGER_ID |
|------------|----------------|---------|-----------------|------------|
| 1 | AI MOBILING | 8 | XYZ | 2 |
| 2 | ML DETECTION | 6 | XYZ | 1 |
| 3 | ATM | 5 | PQR | 3 |
| 4 | COLLEGE FEE | 6 | SS INSTITUTIONS | 2 |
| 5 | MARKETING | 9 | ID | 2 |
| 6 | EVENT ANALYSER | 9 | XYZ | 2 |
| 7 | ATM | 4 | XYZ | 1 |
| 8 | LEAF DISEASE | 5 | SS INSTITUTIONS | 5 |

Fig 3: Projects list

- As of we seen the admin modules contain the reports options that shows the list of projects that are dealing with a organisation.
- It was assigned to a manager along with its ID during its creation.



Fig 4 Post New Bug

- Posting a new bug in a bug tracking system involves filling out a detailed report form. Users must provide a descriptive title and a thorough description of the issue, including steps to reproduce the bug.
- The form typically requires specifying the bug's severity, priority, and the environment where it was encountered. Screenshots or other attachments can be added to provide visual context. Once submitted, the bug is assigned a unique identifier and enters the tracking system for triage and resolution.

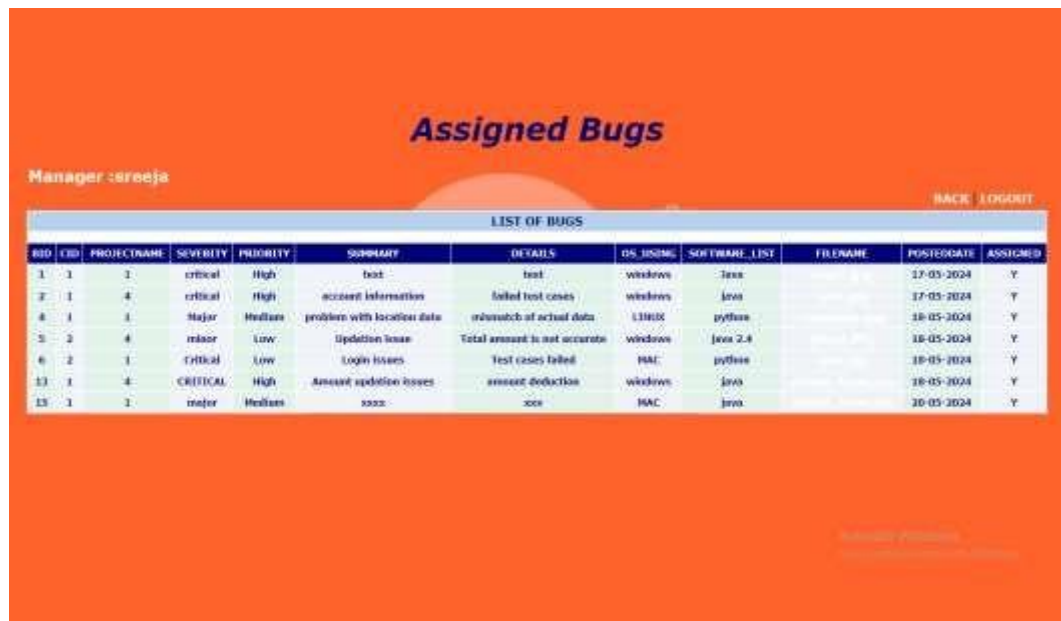


Fig 5 : Assigning Bugs

- Assigning bugs in a bug tracking system involves designating a specific team member responsible for investigating and resolving the issue.
- This process typically takes into account the team member's expertise, workload, and the bug's severity and priority. The assigned individual receives notifications and updates about the bug, ensuring they are aware of the new task.
- Clear assignment helps streamline the resolution process, preventing confusion and overlap in efforts.



Fig 6 : Assigned Bug

- After a technical person logs into the bug tracking system, they typically access a dashboard or a dedicated section where they can view all the bugs assigned to them for resolution.
- This section provides a comprehensive overview of the assigned bugs, allowing the technical person to prioritize their tasks and focus on resolving critical issues efficiently.
- The list of assigned bugs usually includes essential details such as bug ID, description, severity, status, priority, reporter, and deadline.
- This centralized view enables the technical person to track the progress of each assigned bug, update their status as they work on them, communicate with stakeholders, and collaborate with other team members if necessary.



Fig 7 : Tester Verification

- Verification includes reproducing the bug, conducting regression testing to check for new issues, performing functional testing to validate the fixed functionality, and possibly involving user acceptance testing by stakeholders .
- Once verified, documentation is updated, and the bug is marked as "Resolved" or "Closed" in the bug tracking system, signifying successful resolution and closure of the issue.
- This systematic approach ensures the timely and effective resolution of bugs, contributing to improved software quality and user satisfaction.
- After the developer implements the fix, it undergoes verification by quality assurance engineers to ensure that the bug is genuinely resolved and that the fix meets specified requirements.

V. CONCLUSION

Implementing a bug tracking system is a critical step for any software development team aiming to maintain high standards of quality and efficiency. Such a system centralizes the process of identifying, documenting, and resolving bugs, ensuring that every issue is tracked from discovery to resolution. This centralized repository not only helps in maintaining an organized workflow but also ensures that no bug is overlooked. By providing a systematic approach to bug management, teams can efficiently prioritize and address issues based on their severity and impact on the end-user. A key advantage of a bug tracking system is the enhanced communication and collaboration it facilitates among developers, testers, and project managers. Clear documentation and real-time status updates reduce the risk of misunderstandings and ensure that everyone involved in the project has access to the latest information.

This improved communication leads to more effective problem-solving and quicker resolution times. Additionally, by assigning specific bugs to team members, the system fosters accountability and ensures that every issue has a designated owner responsible for its resolution. Beyond immediate bug resolution, a bug tracking system also provides valuable historical data and analytics. This data helps in identifying recurring issues, assessing the performance of the development team, and making informed decisions to improve the software development process. By analyzing trends and patterns in bug reports, teams can implement preventive measures to avoid similar issues in the future, leading to a more robust and reliable product. Ultimately, the systematic approach facilitated by a bug tracking system enhances the overall quality of the software, leading to higher customer satisfaction and trust in the product.

VI. REFERENCES

- [1]. J. Anvik, L. Hiew, and G. C. Murphy. Who should fix this bug? In ICSE'06: Proceedings of the 28th International Conference on Software Engineering, pages 361–370, 2006.
- [2]. J. Aranda and G. Venolia. The secret life of bugs: Going past the errors and omissions in software repositories. In ICSE'09: Proceedings of the 31st International Conference on Software Engineering (to appear), 2009
- [3]. S. Artzi, S. Kim, and M. D. Ernst. Recrash: Making software failures reproducible by preserving object states. In ECOOP'08: Proceedings of the 22nd European Object-Oriented Programming Conference, pages 542–565, 2008.
- [4]. N. Bettenburg, What makes a good bug report? In FSE'08: Proceedings of the 16th International Symposium on Foundations of Software Engineering, pages 308–318, November 2008.
- [5]. N. Bettenburg, S. Kim. Duplicate bug reports considered harmful ... really? In ICSM'08: Proceedings of the 24th International Conference on Software Maintenance, pages 337–345, 2008.
- [6]. R. Premraj, T. Zimmermann, on failures of projects..based on the bug information of decentralized system international conference 2012.
- [7]. S. Just, A. Schröter, C. Weiss had a conference on the Roles of communication and collaboration in the “tracking of bugs” for the production of qualitative projects in software industry..2008
- [8]. Bug <https://myprojectideas.com/bug-tracking-system-in-java-javascript-project/> by cisco raman November 2012
- [9]. s.Cameron, k.Vikas Yadav on august 2012 <https://www.elsner.com/how-to-implement-a-successful-bug-reporting-and-tracking-process/>