International Journal of Engineering Science and Advanced Technology (IJESAT)

IJESAT (Enriching the Research)

Open Access Research Article

Volume: 23 Issue: 07

July, 2023

BUGTRACKINGFORIMPROVINGSOFTWARE RELIABILITY

Mrs. T. Padmaja [1], G.SriLekha [2], B.Chandini [3], Ch.Manasa [4], G.Santhi [5]

[1] Assistant Professor, Department of CSE, Ramachandra College of Engineering, Eluru, A.P
[2,3,4,5] UG Students, Department of CSE, Ramachandra College of Engineering, Eluru, A.P

ABSTRACT

Bug Tracking for Improving Software Reliability (BTS) is an automated system that can be useful to employees and the managers in any functional organization. Bug Tracking System gives the facility to define the tasks in the organization and also allows the managers to track the bugs spent by the employee for that particular task. A report generation facility is supported in BTS that allows the managers to analyze which are those skills by employee are utilized and those which are not utilized. This tool helps employees to document their Bugs and analyze. This project aims at creation of a Bug Tracking System. This project will be accessible to all developers and its facility allows developers to focus on creating the database schema. The objectives of this system are it can be used to map bug reports to relevant files for bug fixing. To keep track of employees kills and based on the skills assigning of the task is done to an employee. Employee does bugs capturing. It can be done on daily basis. Various Reports are generated by this System for an employee and as well as to a manager.

1. INTRODUCTION

Bug tracking is an essential process in software development that involves identifying, documenting, and managing defects or errors that occur in software applications. It is an essentialtoolforimprovingsoftwarereliabilitybyensuringthatissuesareidentified,tracked, and resolvedin a timelyand effectivemanner.

The process of bug tracking involves capturing detailed information about the bug, including the steps to reproduce it, the environment in which it occurred, and any relevant error messages or logs. This information is then used to assign the bug to a developer or development team, who will work to resolve the issue and provide a fix or patch for the software.

Bugtrackingsoftwareprovidesacentralizedlocationfortrackingbugsand managingthe entirebugresolutionprocess, frominitialreportingto finalresolution. Itenablesdevelopers toprioritizebugsbasedonseverityandimpact, assignthemtotheappropriateteammembers, and track progress throughout the resolution process.

By using bug tracking software, software development teams can ensure that bugs are identifiedandresolvedbeforetheyimpactusers,improvingtheoverallreliabilityandquality ofthesoftware.Thedataandinsightsgatheredthroughbugtrackingcanalsobeusedtoinform future development efforts, identify areas for improvement, and enhance the overall development process.

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

Vision

The purpose of Bug Tracking for improving software reliability is to provide better service to the administratoror useful for applications developedinanorganization.

Scope

TheBugTracking for ImprovingSoftwareReliabilityisawebbasedapplicationthat can be accessed throughout the organization. This systemcan be used for logging bugs against anapplication/module, assigning bugstoteammembersandtrackingthe bugsto resolution. There are features like user maintenance, user access control, report generators etc in this system.

Definition

Bug - A software bug (or just "bug") is an error, flaw, mistake, failure, or fault in a computer program that prevents it from behaving as intended (e.g., producing an incorrect result).Mostbugsarisefrommistakesanderrorsmadebypeopleineitheraprogram'ssource codeorits design, anda few are caused bycompilers producingincorrect code.

Overview

Bugtracking istheprocessofreportingandtrackingtheprogressofbugs fromdiscovery through to resolution, where a bug is defined as a deviation from requirements. Other terminology frequentlyused to describe this process include

- problemtracking
- changemanagement
- faultmanagement

Thegoalofbugtrackingisto improvesoftwarereliabilitybyensuringthat alldefectsare identified, prioritized, and resolved beforetheycan causesignificantproblems for users.

Bug tracking typically begins with the identification of a bug, which can be reported by variousstakeholders,includingend-users,testers,anddevelopers.Onceabug isreported,it isassigned to ateammember whois responsiblefor verifying the issue.

Thebugtrackingprocessinvolvesseveralkeysteps:

Identification: Bugs are identified through various means, including user feedback, automated testing, andmanual testing.

Reporting: Bugs are reported through a bug tracking system that allows stakeholders to submitinformationabouttheissue,includingthestepstoreproduceit,theexpectedbehavior, and the actual behavior.

Verification:Once a bugis reported, itis assigned toateam memberwhois responsible for verifyingthe issueanddetermining itsseverityandpriority.Theteammemberwillattempt to reproducethe issue and confirm its validity.

Prioritization: Once a bug isverified, it isassigned aprioritylevelbased onitsseverityand impactonthe application. Bugs with higher prioritylevels are addressed first.

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

Assigning: The bug isassigned tothe appropriateteammember orteamfor resolution. The teammember willanalyze the code, debug the issue, and make the necessarychangesto fix the bug.

Closing: Finally, the bug is marked as resolved and closed in the bug tracking system. The teammembermayalsoprovide a summaryofthe issue and the stepstakento resolveit.

The use of a bug tracking system can help improve software reliability by ensuring that bugs are identified, tracked, and resolved inatimelymanner. Byprioritizing bugs based on theirseverityandimpact,teamscanfocustheireffortsonfixingthemost criticalissuesfirst. Additionally, the bug tracking system can help inform future development efforts and improve overall software quality.

2. LITERATURE SURVEY

Bugtrackingisanessentialpartofsoftwaredevelopment thathelpstotrack, andmanage defectsorissuesthatariseduringthesoftwaredevelopmentprocess.Thefollowing literature surveyprovides a comprehensive overview ofstudies related to bug tracking for improving software reliability.

"Whoshouldfixthisbug?[1]"isaresearchpaperbyJ.Anvik,L.Hiew,andG.C.Murphy that discusses a studyconductedto determine whichdevelopers are most suitable for fixing different types of bugs. The authors collected data from several open-source projects and analyzedtherelationshipsbetweenthetypeofbug,thedevelopers'experienceandexpertise, andthebug fixer'sabilitytofixthe bugcorrectlyandquickly. Thestudyfoundthat thetype of bug and the expertise of the developer are key factors in determining who should fix a particularbug.Theauthorsidentified five categoriesofbugs,eachrequiringdifferent levels of expertise and experience to fix. They also found that assigning the right developer to a specific bug can significantly reduce the time to fix the bug. The authors recommend that software teams adopt a bug triage process to identify and classify bugs according to their complexityand required expertise. Thiscan help ensure that the right developer isassigned to eachbug, leading to faster and more effective bug fixes.

"Thesecret lifeofbugs:Going pasttheerrorsandomissions insoftwarerepositories[2]" is a research paper by J. Aranda and G. Venolia published in the Proceedings of the 2011 IEEE33rdInternationalConferenceonSoftwareEngineering.Thepaperpresentsastudyof the bug reports and fixes in four large open-source software projects, namely Eclipse, Firefox,Gnome,andOpenOffice.Theauthorsanalyzedthebugrepositoriesoftheseprojects tounderstandthenatureand lifecycleofsoftwarebugs.Thestudyfoundthat softwarebugs have a complex life cycle that involves multiple stages, including discovery, reporting, triaging, fixing, and verification. The studyalso uncovered several interesting insights into the bug reporting and fixing process. For example, the authors found that mostbugs are discovered byusers, but onlya smallpercentage ofthemare reported. Theyalso found that bugtriage isacriticalstep inthebug fixingprocess, as it helpstoprioritizeandassignbugs to the appropriate developers. Overall, the paper provides a valuable insight into the world ofsoftwarebugsandhighlightstheimportanceofbugtrackingandmanagementinsoftware development. The authors also suggest severalrecommendations
for improving the bug reporting and fixing process, such as providing better feedback to users, improving bugtriage, andencouragingdeveloperstо contributeto bugfixingefforts.

The paper "Making software failures reproducible by preserving object states[3]" by S. Artzi, S. Kim,

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

and M. D. Ernst proposes a technique called "Recrash" that aims to improve the reproducibilityof software failures. The authors notethat reproducing software failures can be challenging because they often depend on the state of the program's objects at the time of the failure, and that state is often lost when the program crashes. To address this challenge, Recrashusesdynamicanalysis to capturethestateoftheprogram'sobjectsat the timeofacrashandthensavesthatstatetodisk.Whentheprogramisrunagain, Recrashloads thesavedobject stateandusesitto recreatetheconditionsthat ledtotheoriginalcrash.The authors evaluate Recrash using a set of real-world software crashes and show that it can reproduceahighpercentageofthosecrashes.TheyalsocompareRecrashtoothertechniques for improving crash reproduction and find that it outperforms them in terms of both reproducibility and performance. Overall, the paper presents Recrash as a promising techniqueforimprovingthereproducibilityofsoftwarefailures,whichcouldultimatelylead to better software quality and more efficientdebugging.

"Whatmakesagoodbugreport[4]?"isaresearchpaperbyBettenburgetal.thatexplores the characteristics of effective bug reports in software development. The study was conducted through a survey of 171 developers from nine different open-source software projects. The paper highlights the followingfindings:

Clarity: Good bug reports should be clear and unambiguous, so that the developer can quickly understand the problem and reproduce it.

Detail:Thebugreport shouldprovideenoughinformationto helpthedeveloperunderstand the problem and identify its cause. This includes providing steps to reproduce the issue, systemand environmentdetails, and anyrelevantlog files.

Relevance:Thebugreportshouldberelevanttothesoftwareprojectandfocusonissuesthat are important or criticalto the software's functionalityor user experience.

Conciseness: The report should be concise and avoid irrelevant details or information that may distract the developer.

Respectfulness: The bug report should be written in a respectful and professional tone, avoiding accusations or insults.

Reproducibility:Agoodbugreportshouldbereproducible,meaningthatthedevelopercan recreatetheissueintheirownenvironment.

Priority:Thereportshould indicatethepriorityofthe bug, sothatthedeveloper can prioritize their work accordingly.

Overall,thepaperemphasizestheimportanceofclearanddetailedcommunicationbetween developers and bug reportersto ensure effective bug fixing and software development.

"Modeling Bug Report Quality[8]" isa researchpaper byP. Hooimeijer and W. Weimer that was published in 2007. The paper presents a method for modeling the quality of bug reportsinsoftwaresystems,whichisanimportanttaskforsoftwaredevelopmentteams.The authors propose a model that uses various features of bug reports, such as their length, complexity, and the presence of code snippets, to predict their quality. They also suggest usingmachine learningtechniquestotrainthe modelonadataset ofbugreportswithknown quality ratings. The paper describes experiments conducted to evaluate the effectiveness of their model.Theresultsshowthattheir modelisabletoaccuratelypredictthequalityofbug reports in a variety of software projects. Additionally, they demonstrate that the model can beusedtoidentifyfactorsthatcontributetohigh-qualitybugreports,whichcanhelpsoftware development teams improve their bug reporting processes.

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

Overall, the paper provides a usefulframeworkformodelingandpredictingthequalityofbugreportsinsoftwaresystems, which can be used to improve software development practices and ultimately lead to better software products.

3. EXISTING SYSTEM

The existing system consists of entering the details in the Microsoft Excel Sheets for the storing ofthe data. Whena manager needs informationofthe employee he searches for the specified file in the file system. He opens the file and takes the information. Report Generation done manually by copying the content of the different files into another file. TheManually generated report was then printed.

LimitationsinExistedSystem

- Informationretrievalisaverybigprocess.
- Lackoforganizationofthe files mayporntoinformation lossduetoaccidentaldeletion offiles.
- Nosecuritybecausethefilesarevisibletotheusers.

Reportgenerationwillbeabigtask.

4. PROPOSED SYSTEM

TheProposed systemisaa ranking approachto theproblemofmapping sourcefilesto bug reports that enables the seamless integration of a wide diversity of features; exploiting previously fixed bug reports as training examples for the proposed ranking model in conjunction with a learning-to-rank technique andbrowser which is completely related to
onlinesystem, whichprovidesthecentralized database. It storesbugsdataand description of the particular bug data.

AdvantagesOverProposedsystem

- Theperformanceisincreasedduetowelldesigneddatabase.
- Itcanlocatetherelevantfileswithinthetop10recommendations forover70percentof the bug reports in Eclipse Platformand Tomcat.
- Securityisincreased.
- Timesavinginreportgeneration.
- Easytoupdatethedetails.

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

5. RESULTS

Fig1. Homepageofbugtrackingsystem

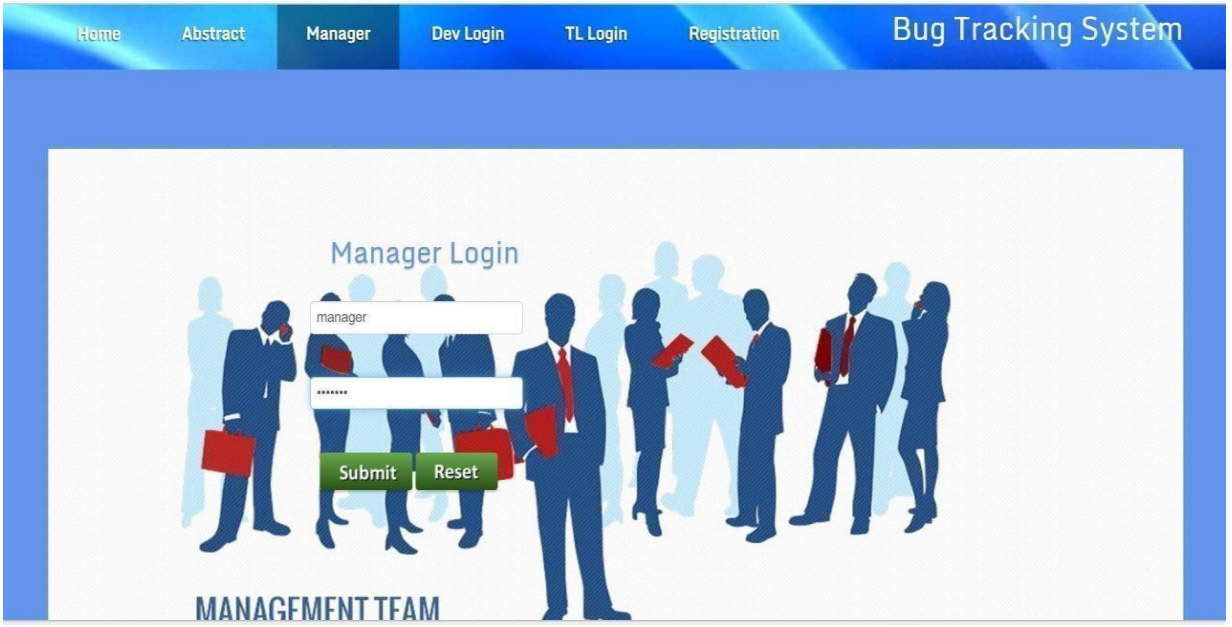


Fig.2Managerloginpage

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

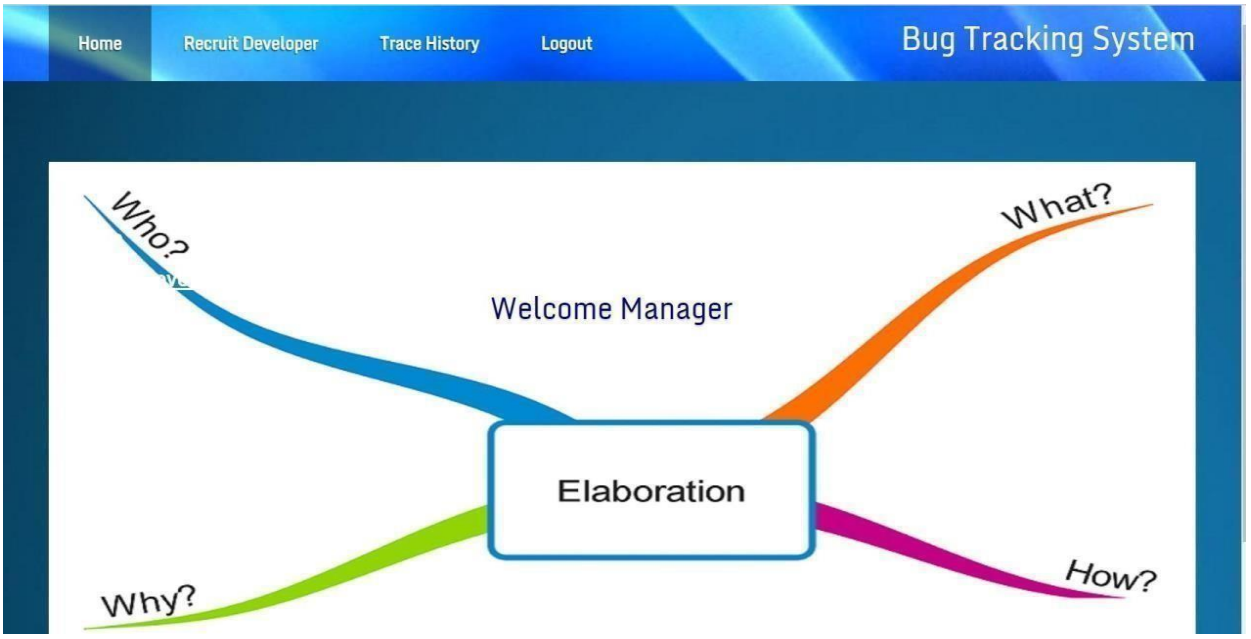
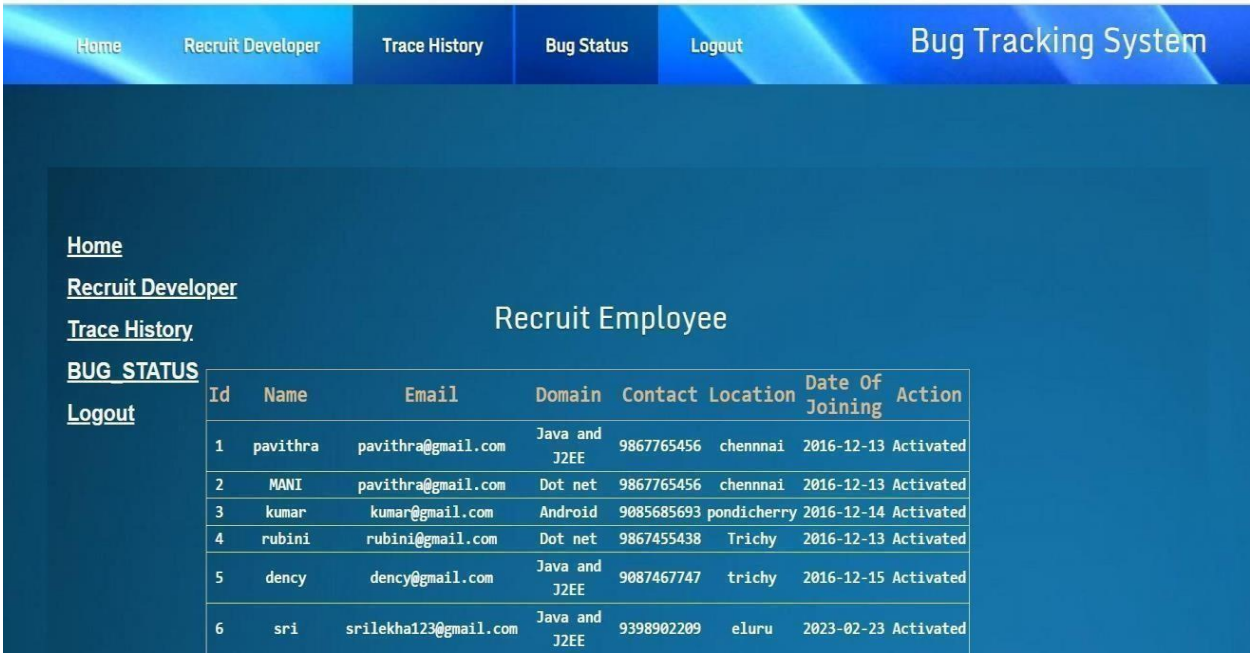


Fig.3Manageraccessandfunctionalities



Id	Name	Email	Domain	Contact	Location	Date Of Joining	Action
1	pavithra	pavithra@gmail.com	Java and J2EE	9867765456	chennnai	2016-12-13	Activated
2	MANI	pavithra@gmail.com	Dot net	9867765456	chennnai	2016-12-13	Activated
3	kumar	kumar@gmail.com	Android	9085685693	pondicherry	2016-12-14	Activated
4	rubini	rubini@gmail.com	Dot net	9867455438	Trichy	2016-12-13	Activated
5	dency	dency@gmail.com	Java and J2EE	9087467747	trichy	2016-12-15	Activated
6	sri	srilekha123@gmail.com	Java and J2EE	9398902209	eluru	2023-02-23	Activated

Fig.4DeveloperRecruitementbythemanager

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

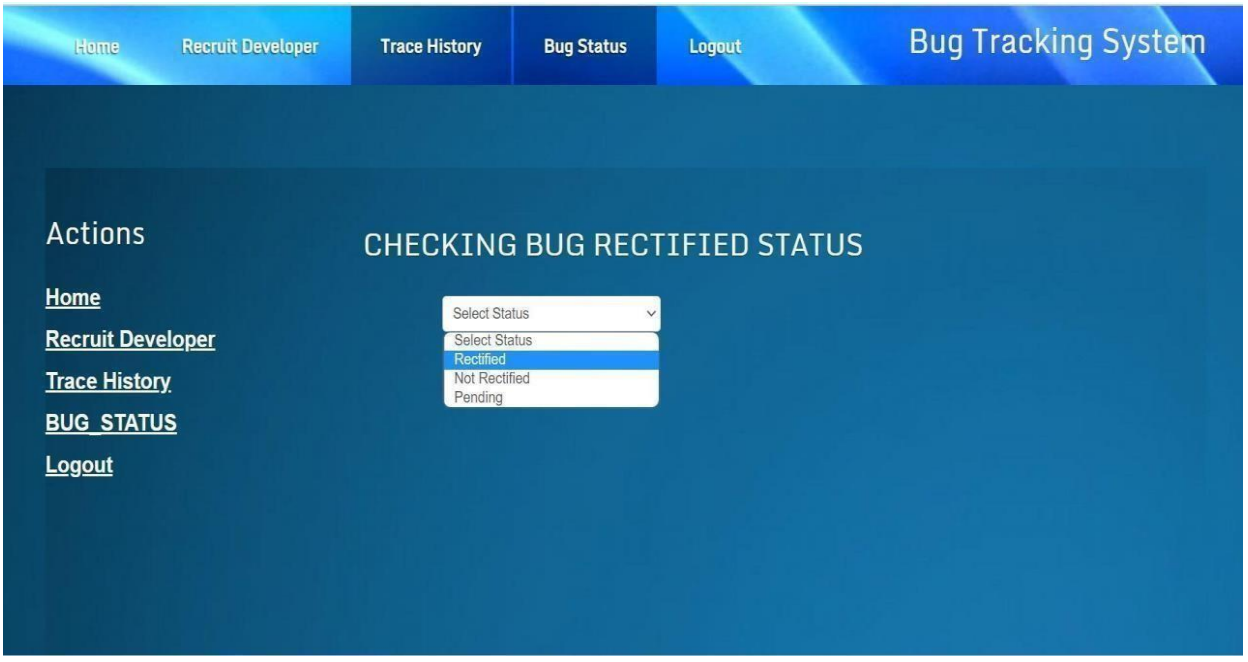


Fig.5Historyofthebugreports

Fig.6Developerloginpage

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

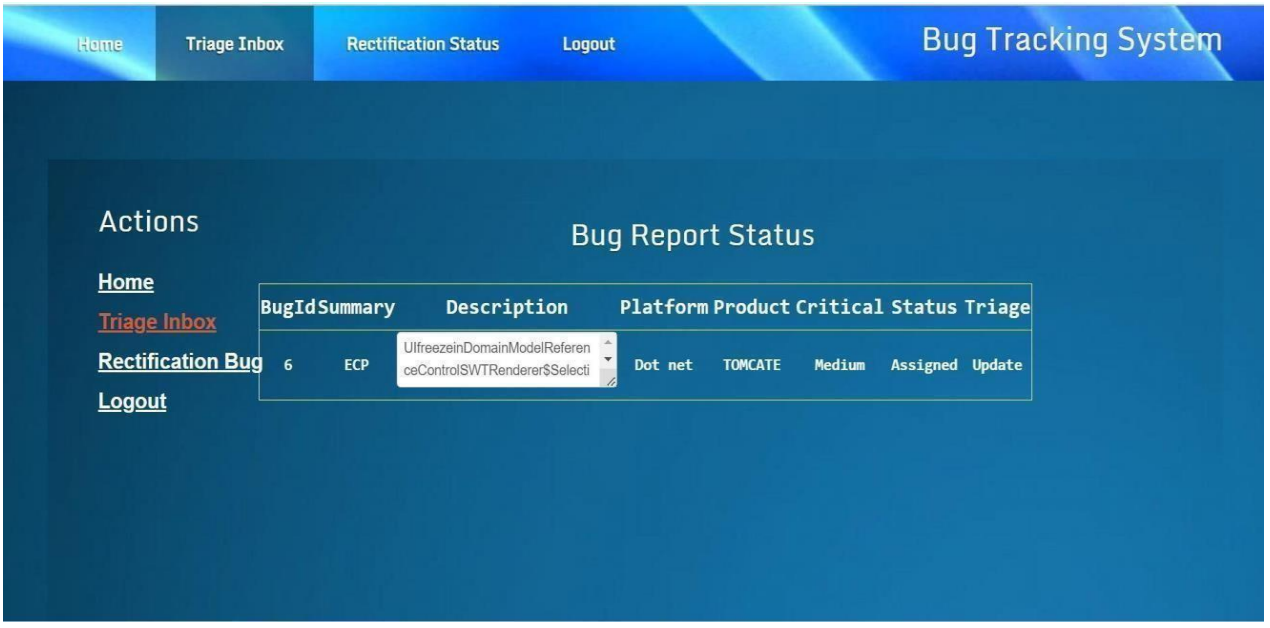


Fig.7Triageinboxoftheparticulardeveloper

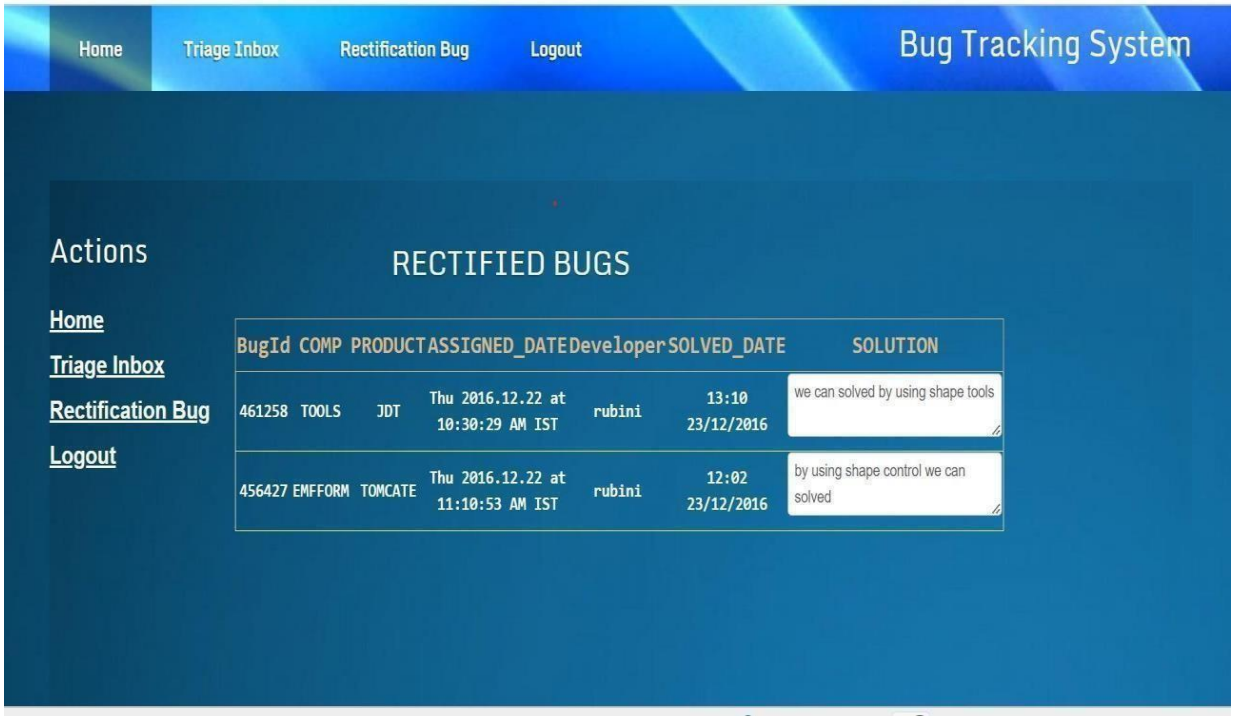


Fig.8Rectificationstatusofthebugsassignedtoaparticulardeveloper

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

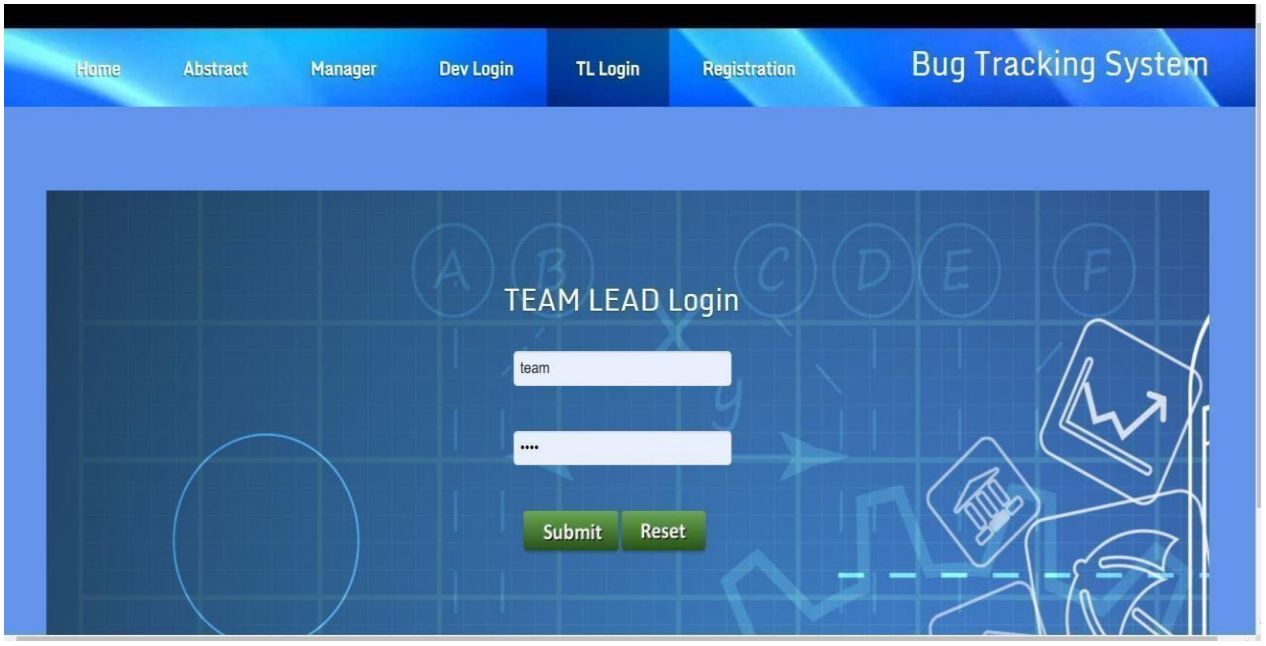


Fig.9Teamleaderloginpage

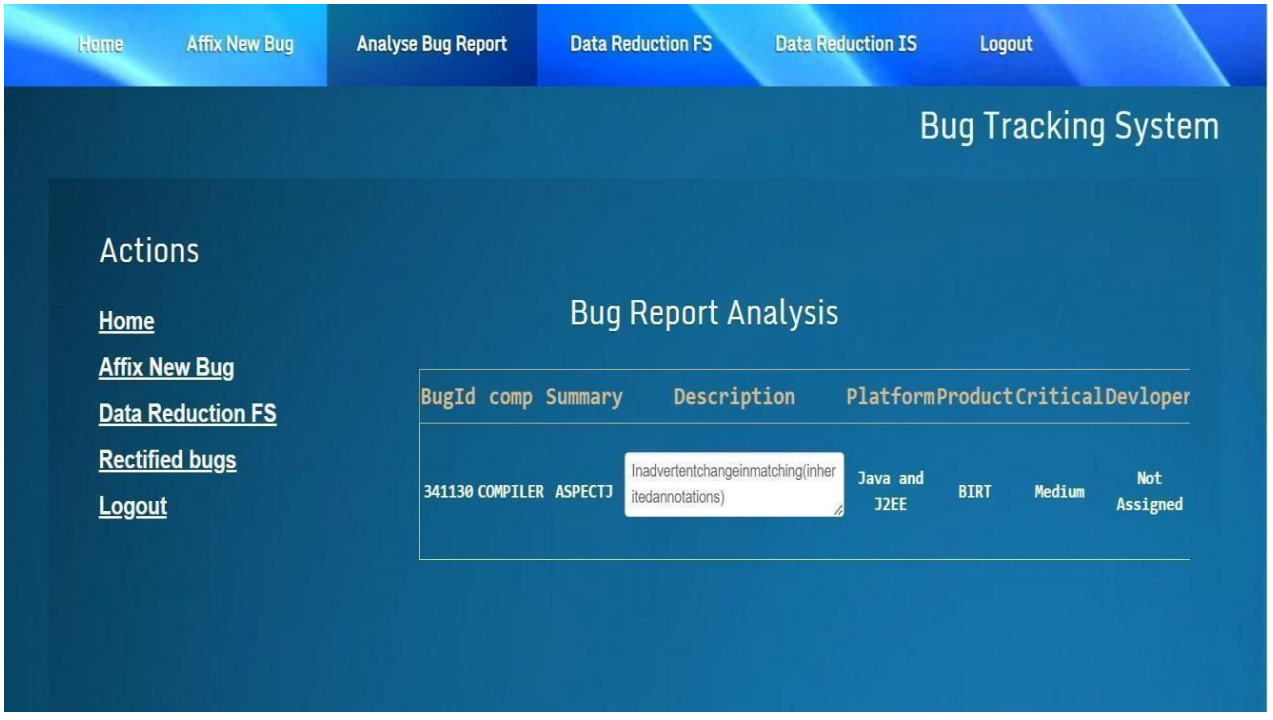


Fig.10Teamleaderacessessibiltyand functionalities

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

Fig.11Developerregistrationapplicationform

6. CONCLUSION

ThisprojectBugTrackingforImprovingSoftwareQualityandReliabilityistokeeptrack of employee skills and based on the skills assigning of the task is done to an employee. Employee doesbugs capturing. It canbe done ondailybasis. Various Reports aregenerated bythis System for anemployee and as wellas toamanager.

Thisprojectwillbeaccessibletoalldevelopersand itsfacilityallowsdeveloperstofocus on creating the database schema and while letting the application server define table based onthe fields in JSP and relationships between them.

Thisapplicationsoftwarehasbeencomputedsuccessfullyand wasalsotested successfully bytaking"testcases".Itisuser friendly,andhasrequiredoptions,whichcanbeutilizedbythe user to performthe desired operations.

The software is developed using Java as front end and MySQLas back end in Windows environment.The goals thatare achieved bythe software are:

- Instantaccess.
- Improvedproductivity.
- Optimumutilizationofresources.
- Efficientmanagementofrecords.
- Simplificationoftheoperations.
- Lessprocessingtimeandgettingrequiredinformation.
- Userfriendly.
- Portableandflexibleforfurther enhancement.

12.1FUTUREENHANCEMENTS:

It is not possible to develop a system that makes all the requirements of the user. User requirementskeep changing as the system is being used. Some of the future enhancements that canbe done to this systemare:

IJESAT (Enriching the Research)
Open Access Research Article
Volume: 23 Issue: 07
July, 2023

- As the technologyemerges,itis possibletoupgrade the systemandcanbe adaptableto desiredenvironment.
- Becauseitisbasedonobject-orienteddesign,anyfurtherchangescanbeeasilyadaptable.
- Basedonthefuturesecurityissues,securitycanbeimprovedusing emergingtechnologies.
- Attendancemodulecanbeadded.
- Adminmodulecanbe added.

REFERENCES

1. J.Anvik,L.Hiew,andG.C.Murphy.Whoshouldfixthisbug?InICSE'06:Proceedings ofthe 28th International Conference onSoftware engineering, pages 361–370, 2006.
2. J. Aranda and G. Venolia. The secret life ofbugs: 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.InECOOP'08: Proceedingsofthe 22nd EuropeanObject-Oriented Programming Conference, pages 542–565, 2008.
4. N.Bettenburg, S. Just,A. Schroter, C.Weiss, R. Premraj, and ¨T. Zimmermann. What makes a good bug report?In FSE'08: Proceedings ofthe 16th InternationalSymposium on Foundations ofSoftware Engineering, pages 308–318, November 2008.
5. N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim. Duplicate bug reports considered harmful ... really? In ICSM'08: Proceedings of the 24th IEEE International Conference on Software Maintenance, pages 337–345, 2008.
6. S. Breu, J. Sillito, R. Premraj, and T. Zimmermann. Frequentlyasked questions in bug reports. Technicalreport, UniversityofCalgary,March 2009.
7. P. Fritzson, T. Gyimothy, M. Kamkar, and N. Shahmehri. Generalized algorithmic debugging and testing. In PLDI'91: Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 317–326, 1991.
8. P.Hooimeijerand W.Weimer.Modeling bugreportquality.InASE'07:Proceedingsof the22ndInternationalConferenceonAutomatedSoftwareEngineering, pages34–43, 2007.
9. S. Just, R. Premraj, and T. Zimmermann. Towardsthe next generation ofbug tracking systems. In VL/HCC'08: Proceedings ofthe 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, pages 82–85, September 2008.
10. [12]A. Kumar, M.Madanu, H.Prakash, L. JonnavithulaandS. R.Aravilli, "Advaita:Bug Duplicity Detection System", *arXiv [cs.SE]*, 2020.
11. [13]W. Aljedaani, Y. Javed and M. Alenezi, "Open source systems bug reports: Meta- analysis", Proceedings of the 2020 The 3rd International Conference on Big Data and Education, *2020*.
12. [14]Y. Yuk and W. Jung, "Comparison of extraction methods for bug tracking system analysis",2013InternationalConferenceonInformationScienceandApplications(ICISA), 2013.
13. [15]KaurandS.Goyal,"Comments-based analysisofabugreport collectionsystemand its applications" inIntelligent Data Analysis,Wiley, pp. 265-288, Jun2020.