Assign bug automatically to the Developer Using Data Reduction Techniques

[1]Pranali Bhujbal, [2]Ashwini Khandagale, [3]Pragati Kokane, [4]Prof. Wavhal.D.N

Department Of Computer Engineering
Jaihind College Of Engineering,Kuran.
Savitribai Phule Pune University,Pune, Maharastra, India

***Abstract*— Software companies spend over 50 percent of cost in dealing with software error. An inevitable step of fixing bugs is bug triage, which goal to correctly assigning a developer to a new bug. To reduce the time cost in manual work, text classification techniques are applied to conduct automatic bug triage. To address the problem of data reduction for bug triage, i.e., how to decrease the scale and improve the quality of bug information. To simultaneously decrease data scale on the bug dimension and the word dimension by combining insatance selection with feature selection. To determine the order of applying instance selection and feature selection, an extract attributes from historical bug data sets and build a predictive model for a new bug data set. This data reduction technique will effectively decrease the data and improve the accuracy of bug triage.**

***Index Terms*— Bug, Bug-triage, Source code, machine learning algorithm, Instance selection, Feature selection, Failure, Software Repositories.**

I. Introduction

Software change requests, such as bug fixes and new function, are an integral part of software evolution and maintenance. Effectively suppor software changes is essential to provide a sustainable large-quality evolution of large-scale software systems. One of the change management problem that has increase a wide attention in the last few years is the automatic support for recommending expert developers to address change requests.Mining software repositories aims to employ data mining to deal with software engineering issue. In modern soft- ware development, software repositories are large-scale databases for storing the output of software development, e.g., source code, bugs, emails, and specifications.

Traditional software analysis is not complete suitable for the large-scale and complex data in software repositories. Data mining has emerged to handle software information. By leveraging data mining techniques, mining software repositories can uncover interesting information in software repositories and solve real- world software problems. A bug repository plays an main role in managing software bugs. Software bugs are unavoidable and fixing bugs is expensive in software development. There are two challenges related to bug information that may affect the effective use of bug repositories, namely the high scale and the low quality. A large number of new bugs are stored in bug repositories, due to the daily-reported bugs. Taking an open source project, Eclipse , as an example, an average of 30 new bugs are reported to bug repositories per day in 2007; from 2001 to 2010, 333,371 bugs have been reported to Eclipse by over 34,917 developers and users. It is a challenge to manually examine such large-scale bug data in soft- ware development. On the other hand, software techniques suffer from the low quality of bug data. Two typical characteristics of low-quality bugs are noise and redundancy. Noisy bugs may mislead related developers while redundant bugs waste the minimum time of bug handling.

.

II. Motivation

A small-scale and large-quality set of bug data can be obtained by removing bug reports and words, which are redundant or non-informative. This is called the Data reduction in Bug triage. Bug triage is an age-old technique that is used to handle the software bugs, whose utilization can often be time consuming. Bug triage aims to assigning a correct developer to fix a new bug. In traditional software development, new bugs are manually triaged by an expert developer, i.e., a human triage. Due to the large number of daily reported bugs and the lack of knowledge of all the bugs, manual bug triage is expensive in time cost and also results in low accuracy. In proposed system avoid expensive cost of manual bug triage and it can be used to assist human triggers rather than replace them. It is helpful to improve the data quality and the cost of prediction is not expensive. This system is helpful for analysts for analysis of results. It is helpful for better decision making and more accurate prediction.

III. Objectives

The data reduction along with bug triage has this objective,

1) Decreasing the data scale

2) Improve the accuracy of bug triage

3) Assure new errors is assigned to expert developer.

4) To decrease the number of instances by removing noisy and redundant instances.

5) A reduced data set by removing non-representative instances.

IV. EXISTING SYSTEM

A time-consuming step of handling software bugs is bug triage, which aim is to assign a correct developer to fix a new bug. In conventional software development, new bugs are manually triaged by an expert developer, i.e., a human triage. Due to the large number of daily bugs and the lack of expertise of all the bugs, manual bug triage is costly in time cost and low in accuracy. To reduce the time cost in manual work, text classification techniques are applied to conduct automatic bug triage in proposed system.

VI. PROPOSED SYSTEM

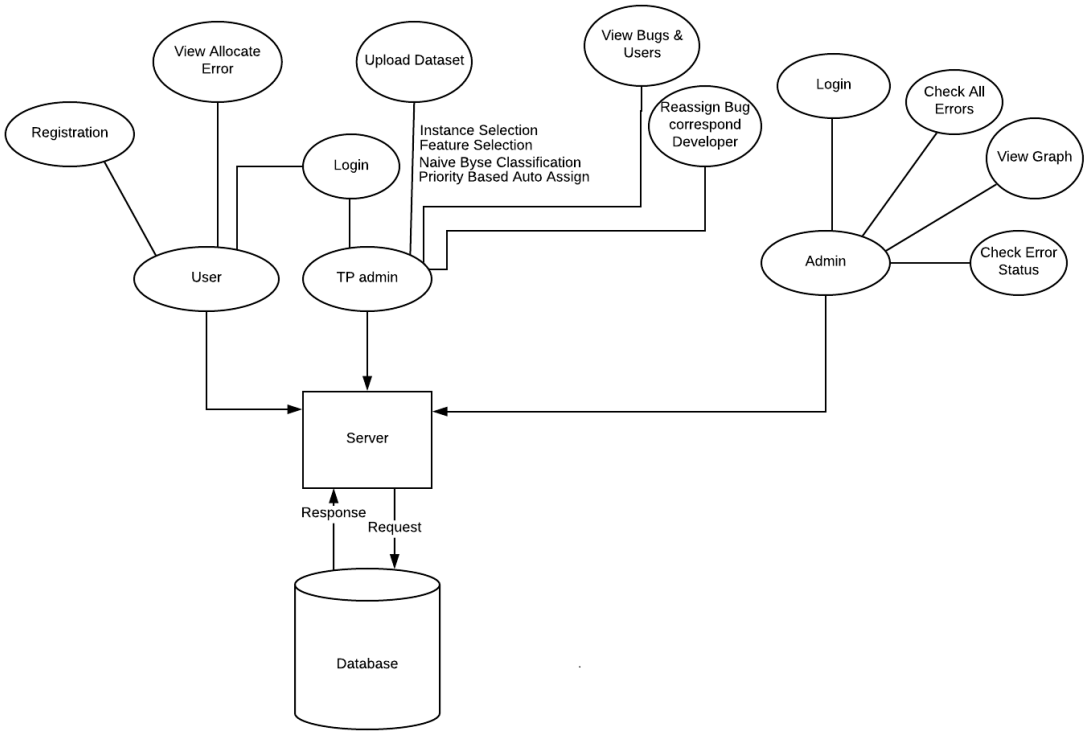


Fig1. System Architecture

In fig1. we propose how to decrease the bug information to save the work cost of developers and improve the quality to facilitate the process of bug triage. Data reduction for bug triage expects to build a small-scale and large-quality set of error data by removing bug reports and words which are not informative and redundant. We use existing techniques of instance selection and feature selection to decrease the bug dimension and the word dimension. The reduced bug data contain less bug reports and fewer words than the original bug data and provides same information over the original bug data. Proposed system is evaluating the reduced bug data, which are size of a data set and the correctness of bug triage.

In the proposed system the input is in the form of bug data set. The bug data set consists of bug report and the details of the developer who have worked on the respective bug. The bug report is mainly separated in two parts:

1. Summary and

2. Description

The proposed system gives predicted results in the form of output. There are two types of users in the proposed system:

1. Developer

2. Tester

In bug dimension we reduce noisy or duplicate bug report to decrease the number of historical bug. In word dimension to reduce noisy and duplicate words. In the system bug triage use to predict the correct developer who can fix the bugs. We follow existing

work of developer to solve the problem of fixing bugs. We predict a new bug to the developer by his/her expertise. In system bug repositories, several developers only fixed very few bugs. Such inactive developer does not provide sufficient information for assignment correct developers. In proposed system we eliminate the developers, who have fixed less than 10 bug.

1. Bug triage:

In bug triage, a bug data set is converted into a text matrix with two dimensions, namely the bug dimension and the word dimension. Each row of the matrix indicates one bug report while each column of the matrix indicates one word.

2. Classification:

We are going to use classification techniques to take the input data set and divide it under different classification label to form the sets. We can do classification using supervised and unsupervised learning.

3. Data reduction

To reduce the scale of bug data sets as well as improve the data quality.

4. Word dimension:

To remove noisy duplicate words in a data set. By removing uninformative words we can improves the accuracy of bug triage.

5. Bug dimension:

To remove uninformative bug reports as soon as the accuracy may be decreased by removing bug reports.

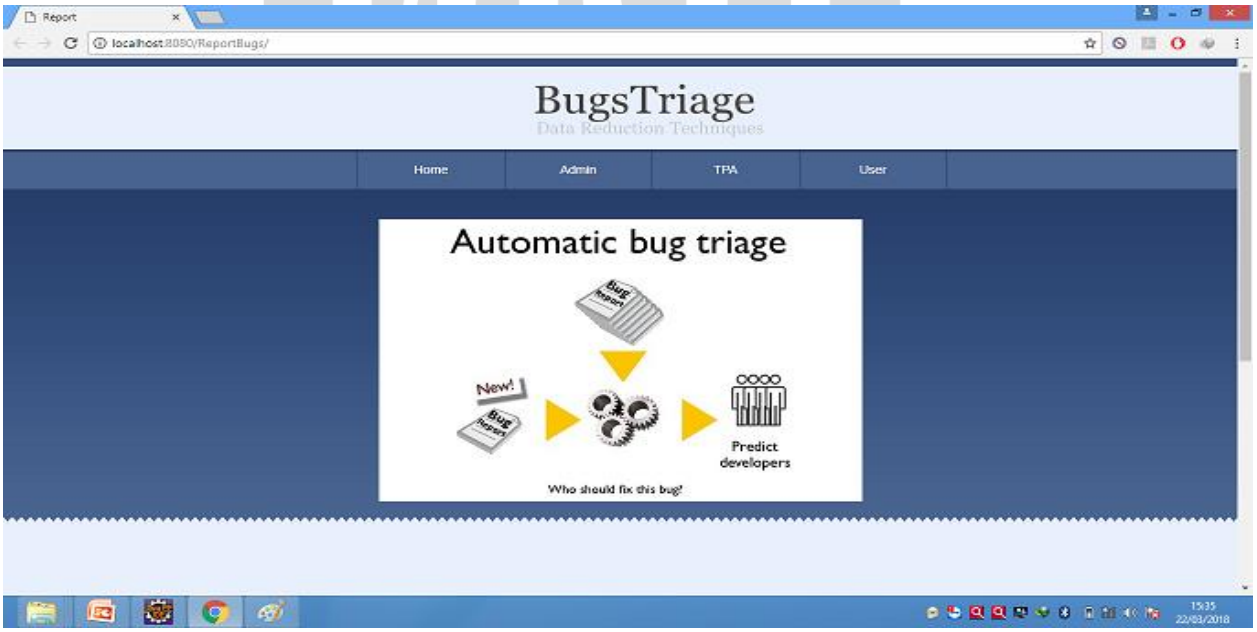
VII. Implementation

For implementation of project, we used following softwares:
• Eclipse Oxygen
• Mysql
• Apache Tomcat
• SqlLog

We use Eclipse Oxygen in client side. Regardless of your operating system, you will need to install some Java virtual machine (JVM). You may either install a Java Runtime Environment (JRE), or a Java Development Kit (JDK), depending on what you want to do with Eclipse. If you intend to use Eclipse for Java development, then you should install a JDK. If you aren't planning to use Eclipse for Java development and want to save some disk space, install a JRE.We use mysql and Apache tomcat in server side.

Screenshots of implemented system:-

1) Front Page



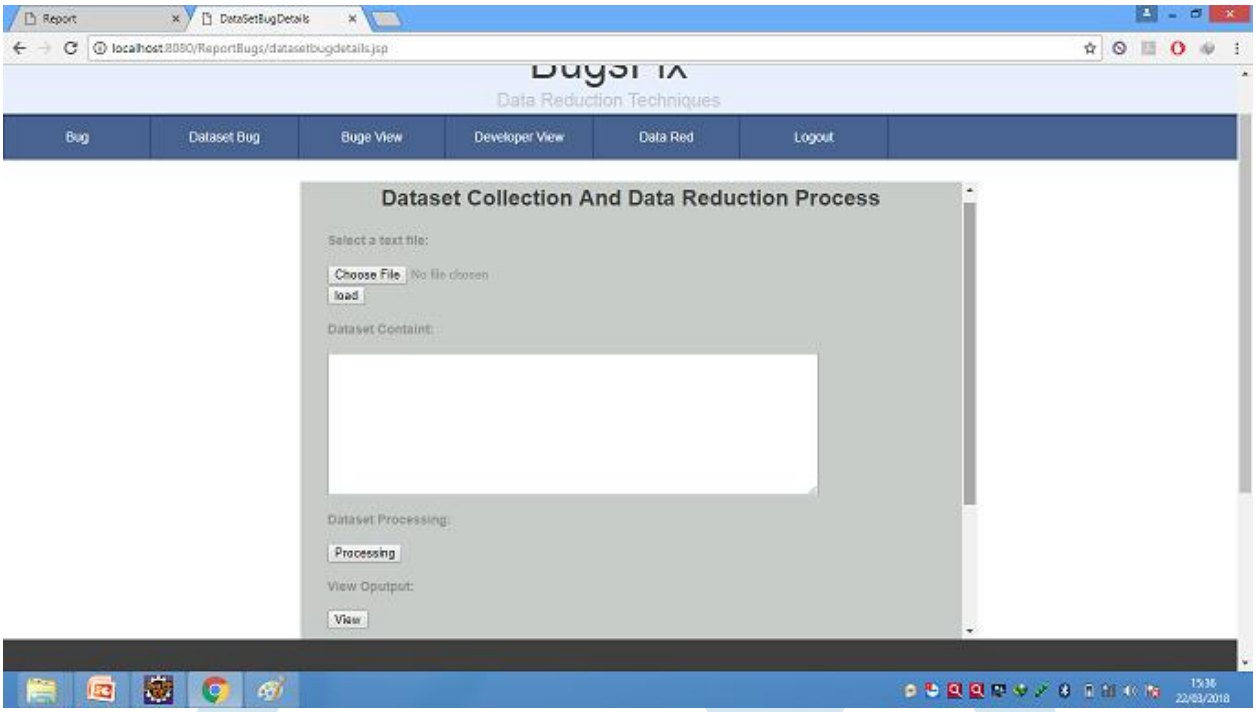
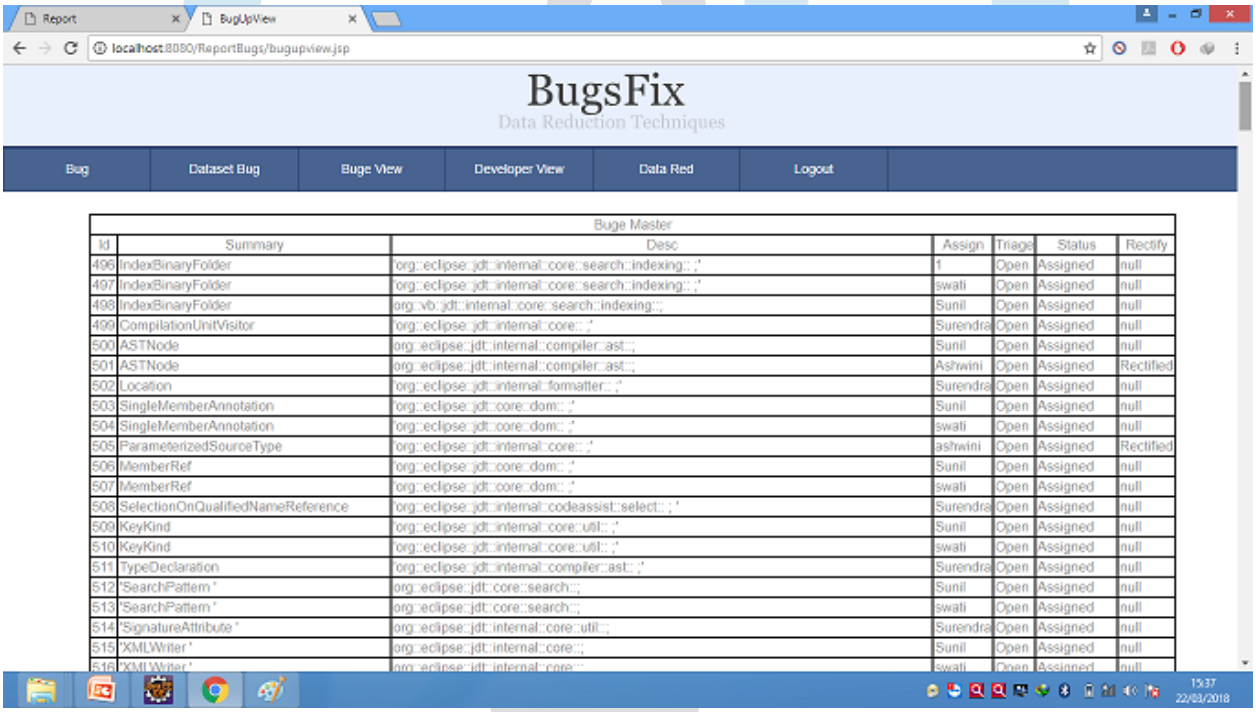


Fig 2.Admin Login

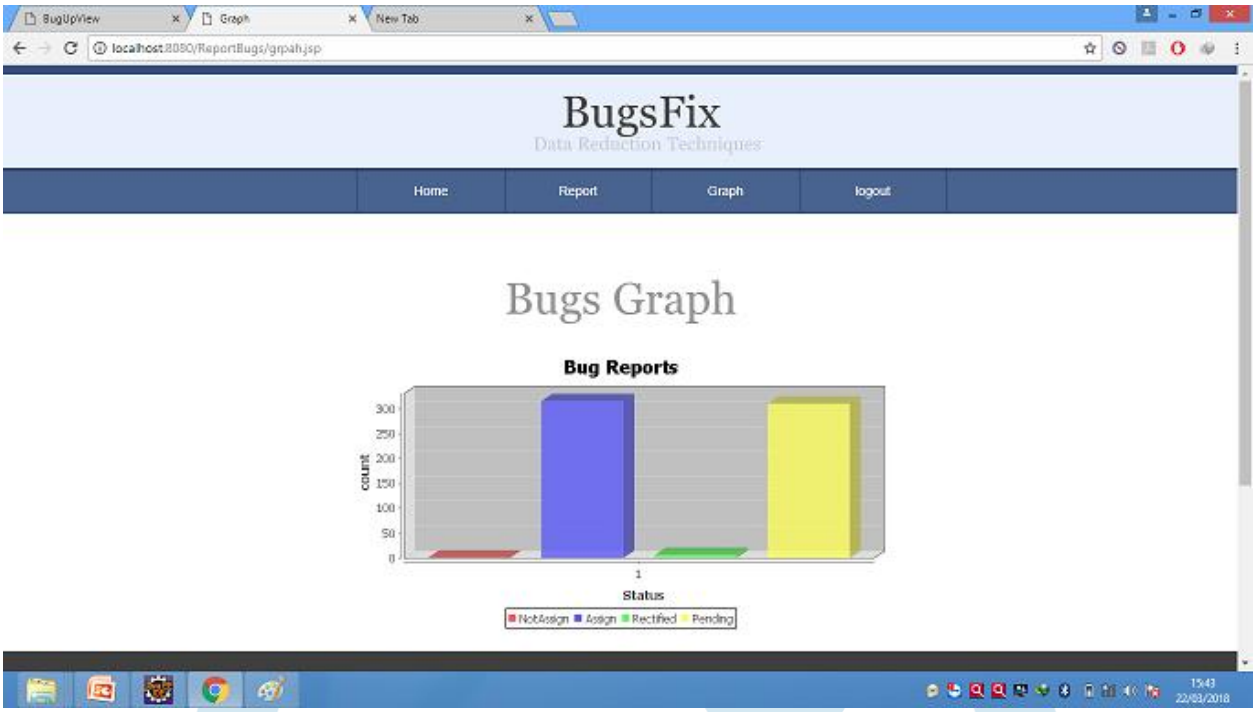


Buge Master

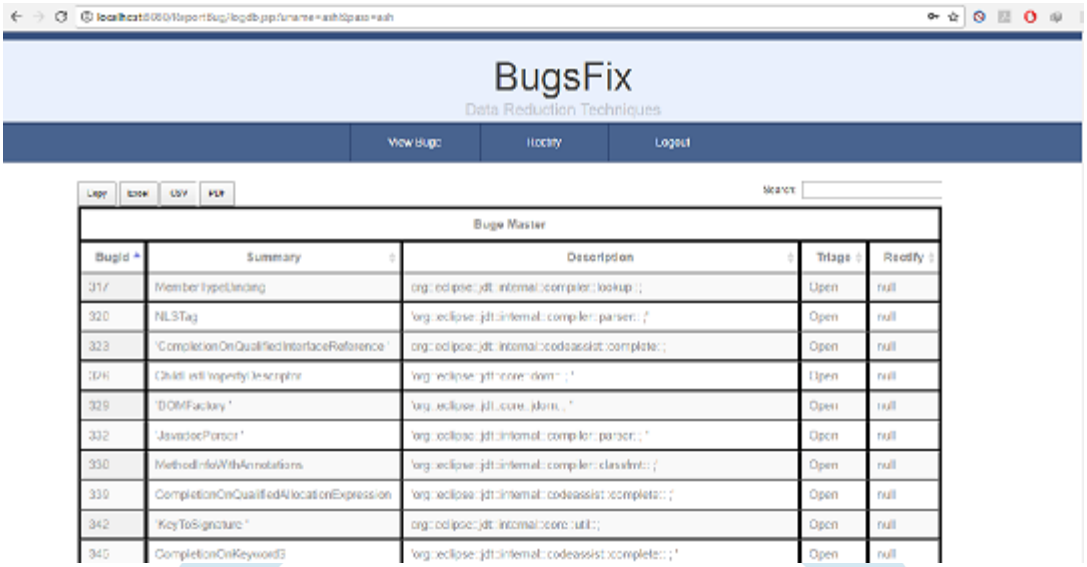
Id	Summary	Desc	Assign	Triage	Status	Rectify
496	IndexBinaryFolder	'org::eclipse::jdt::internal::core::search::indexing:: ;'	1	Open	Assigned	null
497	IndexBinaryFolder	'org::eclipse::jdt::internal::core::search::indexing:: ;'	swati	Open	Assigned	null
498	IndexBinaryFolder	org::vb::jdt::internal::core::search::indexing::	Sunil	Open	Assigned	null
499	CompilationUnitVisitor	'org::eclipse::jdt::internal::core:: ;'	Surendra	Open	Assigned	null
500	ASTNode	org::eclipse::jdt::internal::compiler::ast::	Sunil	Open	Assigned	null
501	ASTNode	org::eclipse::jdt::internal::compiler::ast::	Ashwini	Open	Assigned	Rectified
502	Location	'org::eclipse::jdt::internal::formatter:: ;'	Surendra	Open	Assigned	null
503	SingleMemberAnnotation	'org::eclipse::jdt::core::dom:: ;'	Sunil	Open	Assigned	null
504	SingleMemberAnnotation	'org::eclipse::jdt::core::dom:: ;'	swati	Open	Assigned	null
505	ParameterizedSourceType	'org::eclipse::jdt::internal::core:: ;'	ashwini	Open	Assigned	Rectified
506	MemberRef	'org::eclipse::jdt::core::dom:: ;'	Sunil	Open	Assigned	null
507	MemberRef	'org::eclipse::jdt::core::dom:: ;'	swati	Open	Assigned	null
508	SelectionOnQualifiedNameReference	'org::eclipse::jdt::internal::codeassist::select:: ;'	Surendra	Open	Assigned	null
509	KeyKind	'org::eclipse::jdt::internal::core::util:: ;'	Sunil	Open	Assigned	null
510	KeyKind	'org::eclipse::jdt::internal::core::util:: ;'	swati	Open	Assigned	null
511	TypeDeclaration	'org::eclipse::jdt::internal::compiler::ast:: ;'	Surendra	Open	Assigned	null
512	'SearchPattern '	org::eclipse::jdt::core::search::;	Sunil	Open	Assigned	null
513	'SearchPattern '	org::eclipse::jdt::core::search::;	swati	Open	Assigned	null
514	'SignatureAttribute '	org::eclipse::jdt::internal::core::util::;	Surendra	Open	Assigned	null
515	'XMLWriter '	org::eclipse::jdt::internal::core::;	Sunil	Open	Assigned	null
516	'XMLWriter '	org::eclipse::jdt::internal::core::;	swati	Open	Assigned	null


Report
Developer List View
localhost:8080/ReportBugs/developerlistview.jsp
BugsFix
Data Reduction Techniques
Bug
Dataset Bug
Buge View
Developer View
Data Red
Logout
Developer Master
Id UserName Email Mobile Gender City Action
3 Surendra thoratsurendra86@gmail.com 9579147789 male Pune Get Buge Details
4 Sunil manesunil099@gmail.com 9600071577 male Mumbai Get Buge Details
5 priyanka amit@gmail.com 9988776533 male Pune Get Buge Details
6 swati jadhavd@gmail.com 8484048367 female pune Get Buge Details
7 swati jadhavswati3333@gmail.com 8484048367 female parbhani Get Buge Details
8 Ashwini 101996ash4@gmail.com 9860561214 Female narayangoan Get Buge Details
9 ash sakskkLl ankkakak Female na Get Buge Details
Image Gallery
External Links
The First Computer Bug!
Common Weakness Enumeration
BUG type of Jim Gray
Picture of the "first computer bug"
Contact Us
Don't Hesitate to contact with Our Branch


BugUpView
Report
New Tab
localhost:8080/ReportBugs/adminlogdb.jsp?adname=admin&adpass=admin
BugsFix
Data Reduction Techniques
Home
Report
Graph
logout
Bug Reports
Assign
1
Not Assigned
315
Rectified
6
Pending
310
Image Gallery
External Links
Contact Us

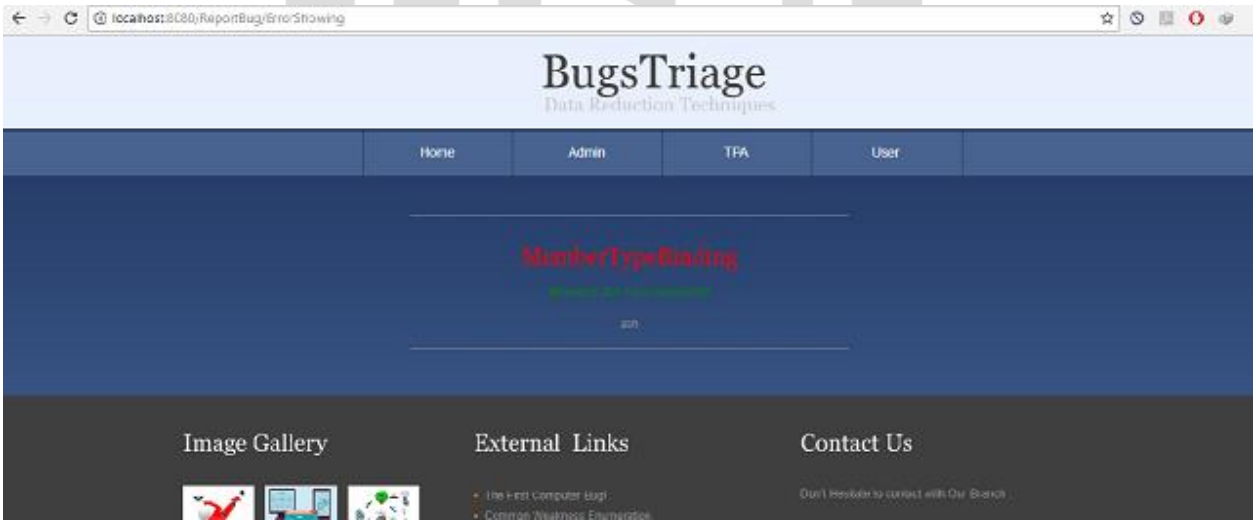

BugsFix
Data Reduction Techniques
Home
Report
Graph
logout
Bugs Graph
Bug Reports
count
Status
NotAssign
Assign
Rectified
Pending

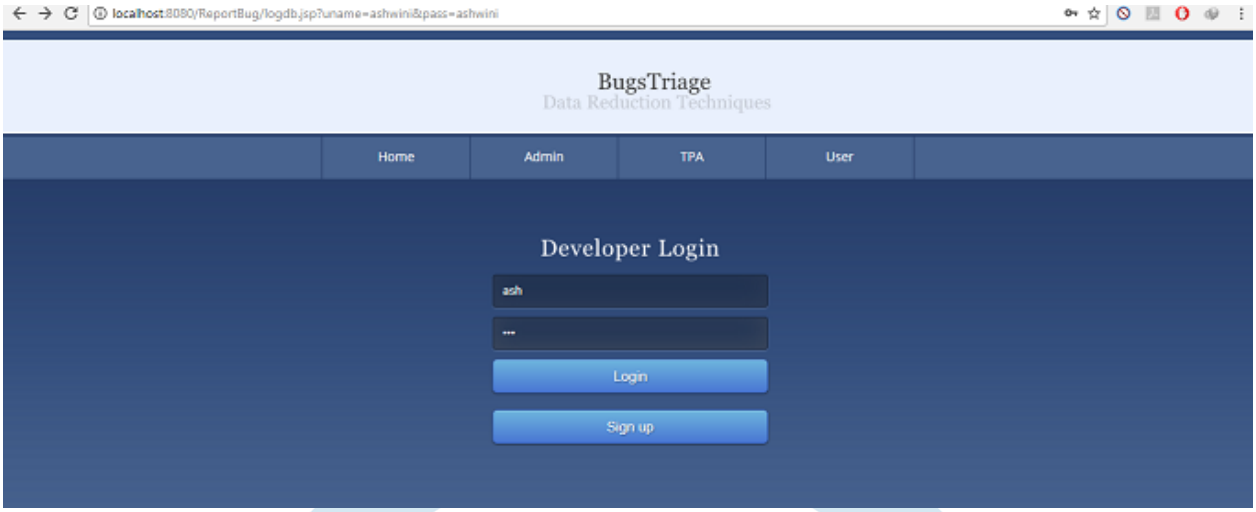

BugsFix
Data Reduction Techniques
Bug
Dataset Bug
Buge View
Developer View
Data Red
Logout
Bug Details
Bug Summary
Description
Assigned
Triage
Reset
Submit
Image Gallery
External Links
Contact Us


BugsFix
Data Reduction Techniques
View Bugs
Rectify
Logout
Search
Bugs Master
Bugid
Summary
Description
Triage
Rectify


localhost:8080/ReportBug/errrectify.jsp
BugsFix
Data Reduction Techniques
View Bugs
Rectify
Logout


Correction Status
Bug Id
317
Bug Error Description
Moving to JDT Core compon
Correction Stat
Rectified
Reset
Submit


localhost:8080/ReportBug/ErrorShowing
BugsTriage
Data Reduction Techniques
Home
Admin
TFA
User
Image Gallery
External Links
Contact Us



VIII. Conclusion

The goal was to achieve the system which will decrease the human effort in software development to assigning the bugs to the respective developers. Our proposed Bug Triage System approximately try to achieve the same one. Proposed system focused on reducing bug data set in order to have less scale of data and storage. For that we have used feature selection and instance selection techniques of data mining as well as we have used Naïve bayes classifier for classification. Our experimental results showed that this data reduction technique will give correct data as well as it will reduce the data scale.

References

[1] S. Artzi, A. Kie_zun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D. Ernst, "Finding bugs in web applications using dynamic test generation and explicitstate model checking," IEEE Softw., vol. 36, no. 4, pp. 474–494, Jul./Aug. 2010.

[2] J. Xuan, H. Jiang, Z. Ren, and W. Zou, "Developer prioritization in bug repositories," in Proc. 34th Int. Conference Software Engineering,2012,pp:25– 35.

[3] W. Zou, Y. Hu, J. Xuan, and H. Jiang,"Towards training set reduction for bug triage," in Proc. 35th Annu. IEEE Int. Computer Software Application Conference, Jul. 2011, pp. 576–581.

[4] Jian Zhou, Hongyu Zhang and David Lo.,"Where the Bugs Should Be Fixed?" ,ICSE 2012, 978-1-4673-1067-3/12/$31.00, 2012 IEEE.

[5] Chen Liu, Jinqiu Yang and Lin Tan and Munawar Hafiz,"R2Fix:Automatically Generating Bug Fixes from Bug Reports ",2013 IEEE Sixth International Conference on Software Testing, Verification andValidation, 978-0-7695-4968-2/13© 2013 IEEE DOI10.1109/ICST.2013.24.

[6] Yuhua Qi, Xiaoguang Mao and Yan Lei,"Making Automatic Repair for Large-scale Programs More Efficient Using Weak Recompilation", 28th IEEE International Conference on Software Maintenance (ICSM), 2012, 978-1-4673-2312-3/12/$31.00 "c 2012 IEEE.

[7] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schr€oter, and C. Weiss, "What makes a good bug report?" IEEE Transaction Software Engineer, vol. 36, no. 5, pp. 618–643, Oct. 2010.

[8] Jifeng Xuan, He Jiang, "Towards Effective Bug Triage with Software Data Reduction Techniques," IEEE Trans. on Knowledge and Data Engineering, vol. 27, no. 1, Jan. 2015.