BUG TRACKING AND REPORTING SYSTEM (BTS)

Abhishek Karmokar*1, Sanchita Kolambe*2, Saloni Mundergi*3,
Prof. Medha Kulkarni*4

*1,2,3,4Department Of Information Technology, Vasantdada Patil Pratishthan's College Of Engineering And Visual Arts, Mumbai, Maharashtra, India.

ABSTRACT

Bugs are an element of a developer's project since coding was invented. Bugs are logical defects in an exceedingly program where some logic doesn't quite go well with another logic implemented within the module. To beat this, we proposed a tracking system which is able to upload, search and store bug details, authenticity and if its solved or not. We are approaching by making an internet app to store and search and upload project files for the need of bug searching. this may help any applied scientist to travel through a project file while debugging process from anywhere within the world during a place with internet access. Bug Tracking System is a perfect solution to trace and keep records of the bugs in an application. Bug Tracking System allows individual or groups of developers to stay track of outstanding bugs of outstanding bugs in their products effectively. Bug Tracking System the system which enable to detect the bugs. Having a bug tracking system is extremely valuable in software development, and that they are used extensively by companies developing software products. A major component of a bug tracking system could be a database that records facts about known bugs.

Keywords: Bug, Defects, Bug Tracking System, Bug Report, Bug Repository.

I.   INTRODUCTION

Every time a person performs some kind of computer program, there is a chance of them facing a flaw, fault or an error. This flaw or error is what we call a Bug. Bugs basically cause unexpected results and cause our system or crash to crash down. Bugs are logically conflicting lines within a code which needs to be Debug. Debugging these bugs is necessary to move on further with the code. A lot of companies bear an enormous number of bugs. No matter how much the developer tries to Debug the code there is always a loophole found in the system by some users. Getting out of these loopholes can become a tedious task. Thus there is a need of a Bug Tracking System to make sure that these bugs are being looked upon, handled and debugged and also a database where in information about the Bug that has been Debugged is stored securely.

A tracker is something that tracks down or detects a particular object. So here, Bug Tracker is the system which allows one to track or detect a single or multiple bugs. Bug Tracking System is a system which is used to track any type of bugs in any software. It is mostly useful for any software company. A bug tracker is a software application that is designed in order to help programmers to keep track of all the reported software bugs so far in their work. This system can be used to identify bugs against any application/module, assigning it to individuals and tracking the bugs to resolution. Finding Bugs can be a tedious task and hence uploading their project to a platform on the internet gives a chance for even other developers/users to give their piece of knowledge and experience for the project to get a better outcomes with more possibilities in the end result. Bugs have been a part of any developer's project since the time coding came into existence. Thus proposing a system that eases a developer's project and helps them to not waste much time behind a single or multiple bugs is our moto.

Our bug tracking system not only helps developers/users track bugs but also gives them facilities to search their earlier detected bugs from our database, provides them with information like how much time our team took to overcome these bugs, the type of bugs, the seriousness and authenticity of the task and the level at which each bug is prioritized. This information helps users to use our system efficiently. We are further approaching by making a web app for developers/software engineers to store, search and upload project files for the need of bug searching and bug tracking. This will help any our users to go through a project file while debugging process from anywhere in the world in a place with internet access. The rest of the paper is organized as follows. Section II describes the Literature survey on in the field of bug tracking system. Section III

highlights the working of the bug tracking and reporting system followed by the experimental results in Section IV. Section V concludes the paper.

II. LITERATURE SURVEY

Following are the papers which we have used for literature survey:-

BUGVILLA: Calibrating Bug Reports with Correlated Developers, Tracking Bug Reports, and Performance Analysis:-

According to Sreeram Gutha[1],an unexpected output is termed as a bug .When the amount of bugs becomes large, then ambiguity gets created if one spreadsheet is employed. For avoiding this problem, bug tracking systems are developed and used for handling the bugs within the process of handling bugs, plenty of resources are being spent by software companies. Here, both the manager and the developer are able to communicate with each other. The bugs will be allocated to the appropriate developer and then the developer will update it with an appropriate solution. Also at the end point of the project, analyzing how many bugs are allocated and out of them how many are resolved by means of a graph can be done.

Bug Report Collection System(BRCS):-

According to Dr. Arvinder Kaur[2],the Bug reporting collection system(BRCS) is developed that can be probably utilized for the gathering of bugs from the Open supply Jira bug tracking system for Apache projects. The tool developed by Bug reporting collection system (BRCS) gathers all the details of reports. This tool is executed in C# which gets the data from repository of Jira by using REST APLs. Also this tool gets the details of large number of projects of Apache which is kept by Jira repository and then produce data in the form of report. The Bug reporting collection system executes well defined steps for gathering bug reports of several Apache projects. The BRCS use the Atlassian REST API's which gives access to information through URI paths.

Towards the next generation of bug tracking systems:-

According to Thomas Zimmermann[3], the developers usually depend on the data which the end-users send in order to resolve the bugs. Here, a survey has been conducted on problems which are faced while reporting the bugs within hundreds of developers and the users of MOZILLA and APACHE projects. Also the results of a card sort on 175 comments which are sent by the responders of the survey are presented. Based on the results, card slot showed few hurdles which are included in bug reporting and bug resolving process. A card slot is used to arrange these comments in terms of hierarchies to reduce higher level of abstraction and classify themes in the participants feedback which are common.

Has this bug been reported?:-

According to Kaiping Liu[4], bug reporting is basically an uncoordinated procedure. Here, an approach is proposed which is based on more effective ranking technique - Ranking SVM and identifying 83 textual and some categorical characteristics. This proposed approach helps the bug reporter for locating appropriate bug reports more accurately. A novel approach which uses Ranking SVM, a Learning to Rank(LTR) method for searching the potential appropriate bug reports of a bug tracking system is proposed.

Automated Bug Reporting System in Web Applications:-

According to Yashika Sharma[5], bug reporting system is an necessary element of a satisfactory software development infrastructure. The primary contributions of this paper are that it focuses on testing the web application automatically that generates a bug report which consists of all the details and mail functionality is also provided by the system model. The main objective is establishing model of automated bug reporting system that performs automated regression testing and generates a bug report. It also provide facility of logging the report, reusable methods and multiple browser support that helps to reduce the human efforts and time which is required for performing the regression testing.

III. PROPOSED SYSTEM

The Bug tracker allows to us carry out important tasks such as reporting the bugs, fixing them, maintaining the bug information, and more. For the source system we are using project management and team management. In the team management section, the admin will assign various tasks to the team. While under project management, the task assigned to the team will be divided among the team i.e. each team member will be assigned to at least one role so they can log in and perform the tasks within the given deadline.

Next the user will provide (upload) the problem statement. After that, the file is sent for the testing phase for identifying various bugs. When the tester encounters the number of bugs, an identification number for every individual bug is generated. All the bug information along with its identification number or ID is kept in account in the database. The update regarding any changes done with respect to the bug reported are mailed to the user/developer which helps them to keep an account of these bugs that they initially reported. These are stored in the database of the Bug Tracking System. It is useful to the user/developer for further references as they can look back to this database whenever required. This system will thereby keep a track of all the bug information for instance Bug ID, Bug Type, Bug Name, Project Name, at what time the bug was reported, when the bug was fixed, how much time was taken in order to fix a particular bug, etc.

The user will be able to enter details regarding the bug. This data then goes to our database. For every user to have a unique ID number, we have an Automated Bug ID Generation System. There are various ways to upload a project in the database for instance a user can upload their file in the form of a .zip file,.7z file, .rar file etc.

Next we would go to the Search page. As the name suggests, here the details of any required Bug can be found by doing nothing but just entering the unique Bug ID that was generated in the home page. After choosing the file, user/developer has the options and leisure to either go for downloading, updating or even reporting the file selected which will notify our user about the action taken on their file. This notification will be there by sent through an email to the users on their respective Email ID. The system will give a visual representation of the bug database of different statuses of different bugs in the database.

Further, a graphical representation of the amount of bugs reported will be given which will give us an idea of the reported number of bugs. These graphs make it easier to understand and have a basic idea about the details of these bugs. Be it pie charts or bar charts, graphs make it trouble-free to give an over view about a particular information. It also provides us with the information about audits on different level and also the current work status. Number of Solved Bugs, Unsolved Bugs, Total number of Bugs and Bugs updated daily will be displayed through these graphs.

The time and date of when these bugs were added in addition to the priority levels of these bugs will also be displayed through graphs.

The basic idea of our report page is that it collects data from the database and if and when any user wishes to have a copy of the details of the Bug, they can simply get a print of this report page, making it comfortable for the user to have an account of the database.

This report page will consist of details like the Serial number, Bug Data, Bug ID, Bug Type, Bug Priority, File Name, Project Name & Date of Submission.
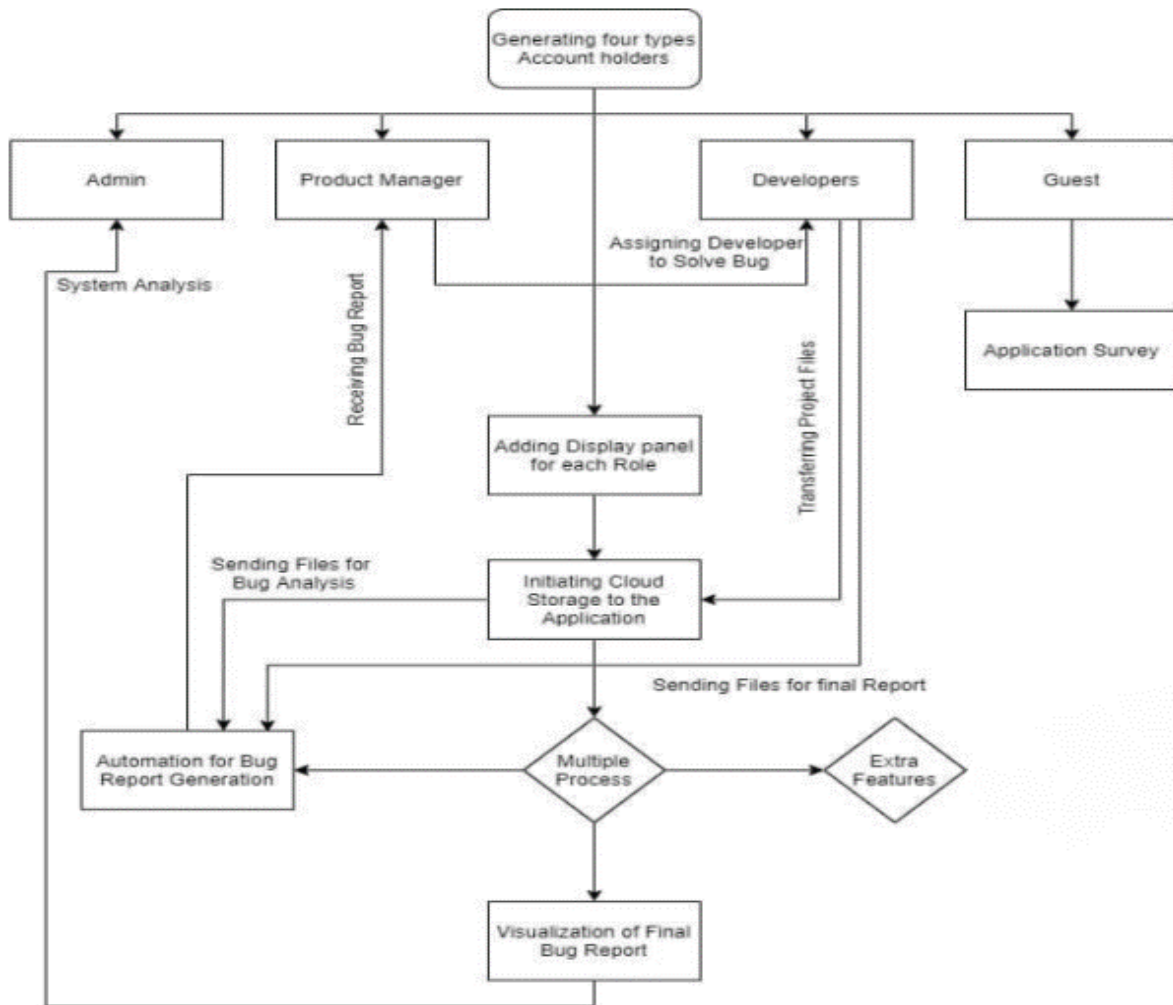
IV.   RESULT

This section presents the demonstration of the Bug tracking system and its performance. Here, various modules output are shown.

Figure 1: Home page

This is the core of the project i.e the initial phase where the user can input the data and generate a unique Bug Id. User can put all the details necessary for entering the bug information in repository/database. He/She can generate the Bug ID automatically(Automated Bug ID generation) for making sure user always gets an unique

Bug ID. Similarly, the user will put other Bug details like Bug Description, Bug Type, Bug Priority level, Project name, Bug Status etc. Here, the user can choose a file and and upload it. Also all types of file format are allowed. For eg. If a Bug project has to be uploaded in the database then it can be done by putting project into a .zip, .7z, rar file etc and upload it.
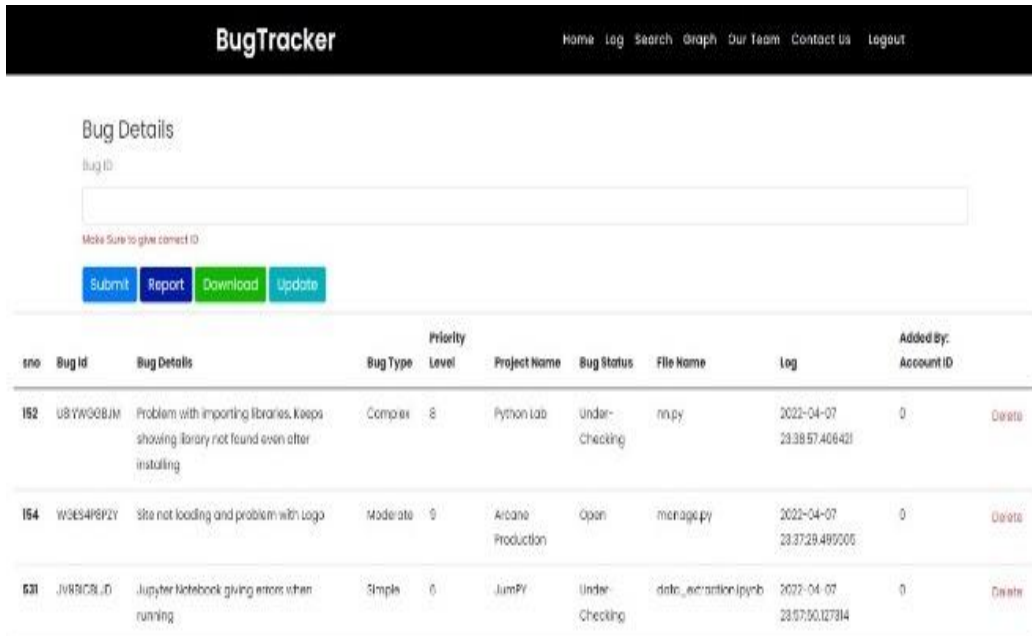

Figure 2: Search Page

This is the search module where you can search the bug details in the database by putting the Bug ID in the search box with the options of reporting, downloading, updating as well as deleting the bug detail if needed.

Here, after updation of the bug details, the user who has reported the bug will get an email notification about the modification being completed on their respective email id.

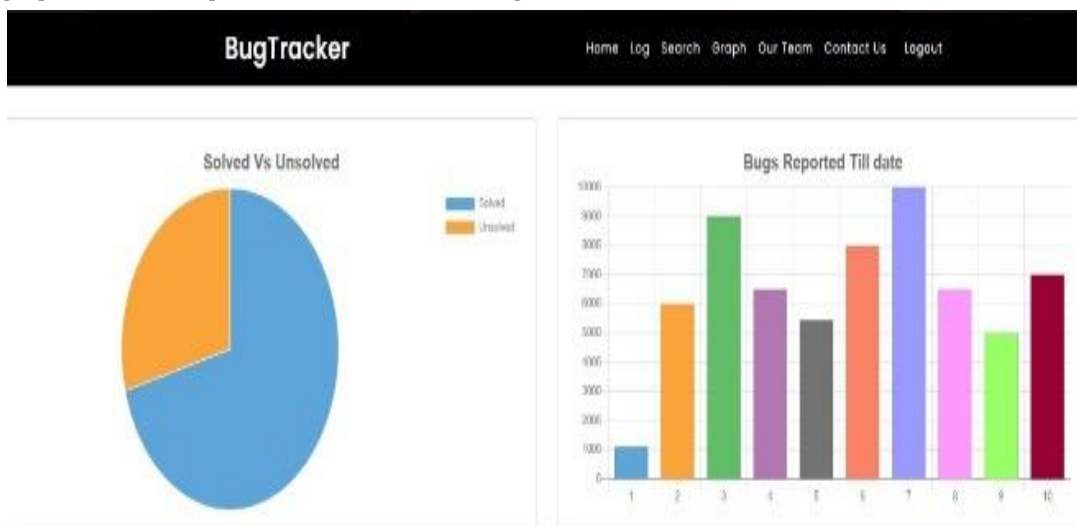For the graph module, output results are shown in Figure 3 below.

Figure 3: Graphical representation of Bug reported based on Priority

Pie chart displays all the bugs entered and also out of all the bugs how many bugs are still opened or under-checking also termed as unsolved. This graph uses calculation of all the solved bugs and subtracts it from total bugs then we get the unsolved bugs present the bug repository.

i.e Unsolved bugs = Total bugs – Solved bugs

Bar graph gives visual representation of total number of bugs reported based on the priority level. It basically displays all the bugs which are reported till date.
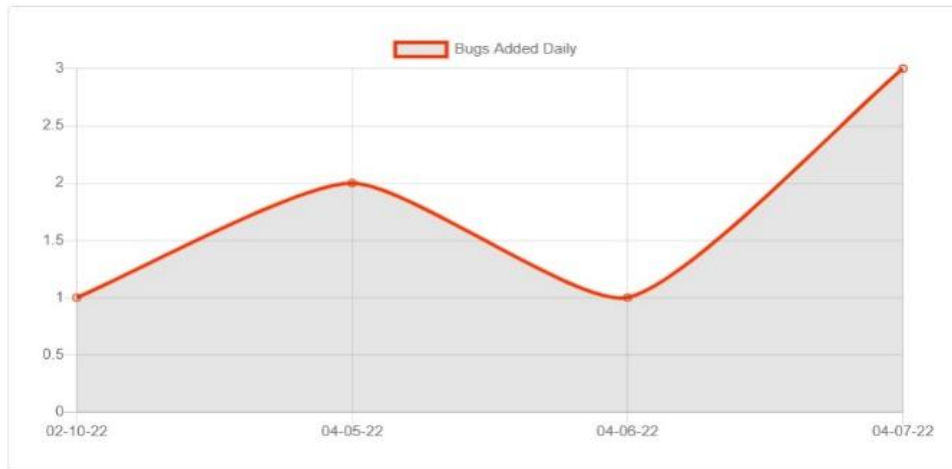
Figure 4: Graphical representation of Bugs reported daily

Above figure is a type of graphical representation of bug details. Here, initially data is gathered from the database about the date and time the bugs were reported and it displays the number of bugs which are added on a particular date.

Figure 5: Report Page

Report page is an automated system which takes the data from the database and displays the details without much hassle. Bug details like Serial number, Bug ID, Bug Data, Bug Type, Bug Priority, Date of Submission etc. You can also take a print of the report page if required/needed for documentation.

V. CONCLUSION

We all know that the world is working towards making everything automated and many projects are under process and there needs to be system to make sure that all the problems and issues faced by the software developer has to be sorted and stored somewhere. Like all data, bug's raw data can also be turned into information to manage the project to see who is conflicting more with logic produced by the team and make sure he/she corrects it. We all have faced going through Logical problems and loopholes while executing a program. This system just makes sure all the mistakes done by a human be recorded for future reference and in all give a report of the bug addressed and solved in a project completed. Any similar bug related to a similar project can be viewed and how it was solved can also be viewed to get a faster approach towards the bug and get it cleared as soon as possible. In future, we can implement automated bug priority prediction as well as the automated bug severity prediction.

VI. REFERENCES

[1] Gudapati Syam Prasad, D. Kiran, and G. Sreeram ," BUGVILLA: Calibrating Bug Reports with Correlated Developers, Tracking Bug Reports, and Performance Analysis"-2020

[2] Dr.Arvinder Kaur and Shubhra Goyal Jindal ,"Bug Report Collection System (BRCS)" 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence 2017

[3] Sascha Just Rahul Premraj , Thomas Zimmermann," Towards the next generation of bug tracking systems" IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) 2008

[4] Kaiping Liu and Hee Beng Kuan Tan," Has this bug been Reported?" 20th Working Conference on Reverse Engineering (WCRE) 2013

[5] Yashika Sharma, Shatakshi, Palvika, Arvind Dagur, Rahul Chaturvedi ,"Automated Bug Reporting System In Web Applications" in Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018)

[6] Tommaso Dal Sasso and Michele Lanza," A Closer Look at Bugs" REVEAL @ Faculty of Informatics — University of Lugano, Switzerland.