**PROJECT REPORT**

**ON**

### Bugs Tracking

Submitted in the partial fulfillment of the requirement for the award of degree of

**Bachelors of Technology**

in

**Computer Science Engineering**

****

**Submitted By:**  Preety Dubey (1237088)

Harwinder kour (1237019)

Department of Computer Science and Information Technology

Baba Banda Singh Bahadur Engineering College

**DECLARATION**

We, the undersigned hereby declare that the project report entitled Online Thesis Management is designed, written and submitted by us in the partial fulfillment of the requirement for the award of degree of Bachelors of Technology in Information Technology is our original work. The empirical findings in this report are based on the data collected by us through discussion with the project guide.

We understand that any such copying is liable to us punishment in way the college authorities deem fit.

Yours faithfully,

Preety Dubey (1237088)

Harwinder kour(1237019)

TABLE OF CONTENTS

1. Introduction 6
   1. Project Overview 6
   2. Purpose 6
   3. Scope 7
   4. Definition, Acronyms, Abbreviations 7

1.5. Existing System 7

1.6. Proposed System 8

1. Functional Or Specific Requirements 9
   1. External Interface Requirements 9
      1. Hardware Interfaces 10
      2. Software Interface 10
   2. Non Functional Requirements 10
      1. Security 11
      2. Reliability 11
      3. Portability 12
      4. Performance 12
   3. Database Requirements 12
   4. Technologies 12
   5. Hardware 13
      1. Server 13
      2. Client 13
2. Analysis And Design 14
   1. Data Flow Diagram 15
      1. DFD Symbols 16
      2. DFD Level 0 16
      3. DFD Level 1 17
3. User Interfaces 18

## 4.1 Home Page 18

4.2 Admin Login 18

4.3 Admin Page 19

4.4 User login 19

## 4.5 User Page 20

## 4.6 Bugs show in user page 20

4.7 Expert Login 21

* 1. Expert resolve bugs 21
  2. Expert enter solution 22
  3. User see solution of bugs 22

4.11 About us 23

1. Test Cases 24
   1. Introduction 26
   2. Time Detected 27
2. Future Plans 28
3. Project Summary 29
4. Bibliography 30

**1.INTRODUCTION**

## Project Overview

Bug tracking is the process of reporting and tracking the progress of bugs from discovery through to resolution, where a bug is defined as a deviation from requirements. Other terminology frequently used to describe this process include

* problem tracking
* change management
* fault management
* trouble tickets

Bug tracking systems are most commonly used in the coding and testing phases of the software development process. However, tracking systems can in fact be used for many other purposes such as general issue tracking, simple task lists, help desk situations or contact management, where the focus is on the tracking aspect rather than what is being tracked. Even in software development, tracking systems are quite often not limited to simply tracking bugs, but extended to track feature requests or enhancements as well as enquiries.

## Purpose

The purpose of Bug Tracking for improving software reliability is to provide better service to the administrator or useful for applications developed in an organization.

* To reduce the time constraint involved.
* To centralize all the data needed.
* To generate various reports as required
* To maintain the user rights.

## Scope

The Bug Tracking for Improving Software Reliability is a web based application that can be accessed throughout the organization. This system can be used for logging bugs against an application/module, assigning bugs to team members and tracking the bugs to resolution. There are features like email notifications, user maintenance, user access control, report generators etc in this system.

**1.4. Definition, Acronyms, Abbreviations**

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). Most bugs arise from mistakes and errors made by people in either a program's source code or its design, and a few are caused by compilers producing incorrect code.

**1.5. Existing System**

The existing system consists of entering the details in the Microsoft Excel Sheets for the storing of the data. When a manager needs information of the 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. The Manually generated report was then printed.

**Limitations in Existing System**

* Information retrieval is a very big process.
* Lack of organization of the files may porn to information loss due to accidental deletion of files.
* No security because the files are visible to the users.
* Report generation will be a big task.

**1.6. Proposed System**

The Proposed system is a browser which is completely related to online system, which provides the centralized database. It stores bugs data and description of the particular bug data. It can also create Excel reports and PDF documents based on the information in its database.

**Advantages over Existing System**

* The performance is increased due to well designed database.
* Security is increased
* Time saving in report generation
* Easy to update the details

# FUNCTIONAL OR SPECIFIC REQUIREMENTS

Required software is for conducting online thesis management and provides upload and download function. The system should satisfy the following requirements:

* **Administrator Aspect**

To upload thesis,documents, presentation by providing the title and description.

To approve users to access the system.

To maintain the details of members and thesis.

To view all the members and thesis.

To Upload thesis, documents, presentations

To change password.

Scheduling the work.

* **Student Aspect**

To view / update their profile.

To update their password.

To download thesis, documents, presentations.

To search thesis by name, title, authors etc.

* **Analysis**

Authenticating users based on username and password.

Keeping session track of user activity.

Only registered candidate can download thesis.

Only administrator can upload thesis.

## External Interface Requirements

**Software Requirements**

Operating System : Windows XP/ Windows 7/8

User Interface : HTML, CSS

Client-side Scripting : JavaScript

Programming Language : Java

Web Applications : JDBC Servlets, JSP

Database : Oracle/Access

**Hardware Requirements**

Processor : Pentium IV

Hard Disk : 40GB

RAM : 256MB

**Database**

Oracle Server

### Non-Functional Requirements

* System should be able handle multiple users
* Database updating should follow transaction processing to avoid data inconsistency.

### Security

* Administrator has the highest authority to edit/delete/create database
* Administrator have the authority to add/expel students
* Students can view and download thesis

### Reliability

Data validation and verification needs to be done at every stage of activity.

* Validating user input
* Use of locking mechanism while updating database like transaction processing
* Recovering the transaction using rollback.

### Portability

* The web application will be built using JSP which has support to run on any platform provided the required compilers are available.
* For database SQL Server would be used, that too has extensive support over many popular architectures and operating systems.

### Performance

The system would be used by multiple users at a time and may grow as time passes; the system would need to implement multithreading to achieve acceptable performance. Further a database connection pool may also be required for assigning faster database connection.

## Database Requirements

The overall objective in the development of the database technology has been to treat data as an organizational resource and as an integrated whole. Database management system allows data to be protected and organize separately from other resources. Database is an integrated collection of data. The most significant of data as seen by the programs and data as stored on the direct storage access storage devices.

## Technologies

This section lists all the technologies for the web based system.

* **Core Java**: Java is a set of computer software and specifications developed by Sun Microsystems, later acquired by Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment.
* **JSP:** JavaServer Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types.To deploy and run JavaServer Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.
* **Servlet**:Java Servlets are programs that run on a Web or Application server and act as a middle layer between requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.
* **NetBeans IDE:NetBeans** is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third development party.
* **HTML**: HTML is a markup language for describing web documents (web pages). HTML stands for Hyper Text Markup Language. A markup language is a set of markup tags. HTML documents are described by HTML tags. Each HTML tag describes different document content.
* **JavaScript:**JavaScript is a high-level, dynamic, untyped, and interpreted programming language.
* **SQL**: SQL is a fast, easy-to-use RDBMS being used for many small and big businesses. SQL is becoming so popular because of many good reasons like SQL is released under an open-source license. So we have nothing to pay to use

## Hardware

The recommended hardware specified by the respective software would suffice the needs. The memory and processing power needed would increase as the number of users increase. The estimated hardware requirements are as specified.

### Server

The minimum hardware as recommended by all of the software required on server side say web server, operating system and development software

* Processing speed of 1.6 GHz
* 2 GB of RAM
* Network interface

### Client

The minimum hardware as recommended by all of the software required on client side say web browser, operating system

* Minimum hardware depending on the operating system used
* True color visual display unit
* User peripherals for better interaction

# ANALYSIS AND DESIGN

**3.1. Data Flow Diagram**

In our DFD, we give names to data flows, processes, and data stores. Although the names are descriptive of the data, they do not give details. So the following the DFD, our interest is to build some structured place to keep details of the contents of data flow, processes, and data store. A data dictionary is a structured repository of data about data. It is a set of rigorous definition of all DFD data element and data structure

**3.1.1. DFD Symbols**

In the DFD, there are five symbols,

1. A Square defines a source (originator) or destination of system data.
2. An Arrow identifies data flow- data in motion .It is pipeline through which information flows.
3. A circle or a bubble (or a oval bubble) represents a process that transforms incoming data flow(s) into outgoing data flow(s)
4. An Open rectangle is a data store-data at rest, or temporary repository of data.

**3.2.2. Data Flow Diagram: Level 0**

**Bug Tracking**

**System**

Programmer

Administrator

Database

**3.2.3. Data Flow Diagram: Level 1**

**..\DFDs\BTS\BTS-TOPLEVEL.wmf**

**3.2.4. Diagram which shows the working of the Site**

User

1.1

User Details

1.2

Validate

Admin User

Programmer

tbl\_Authentication

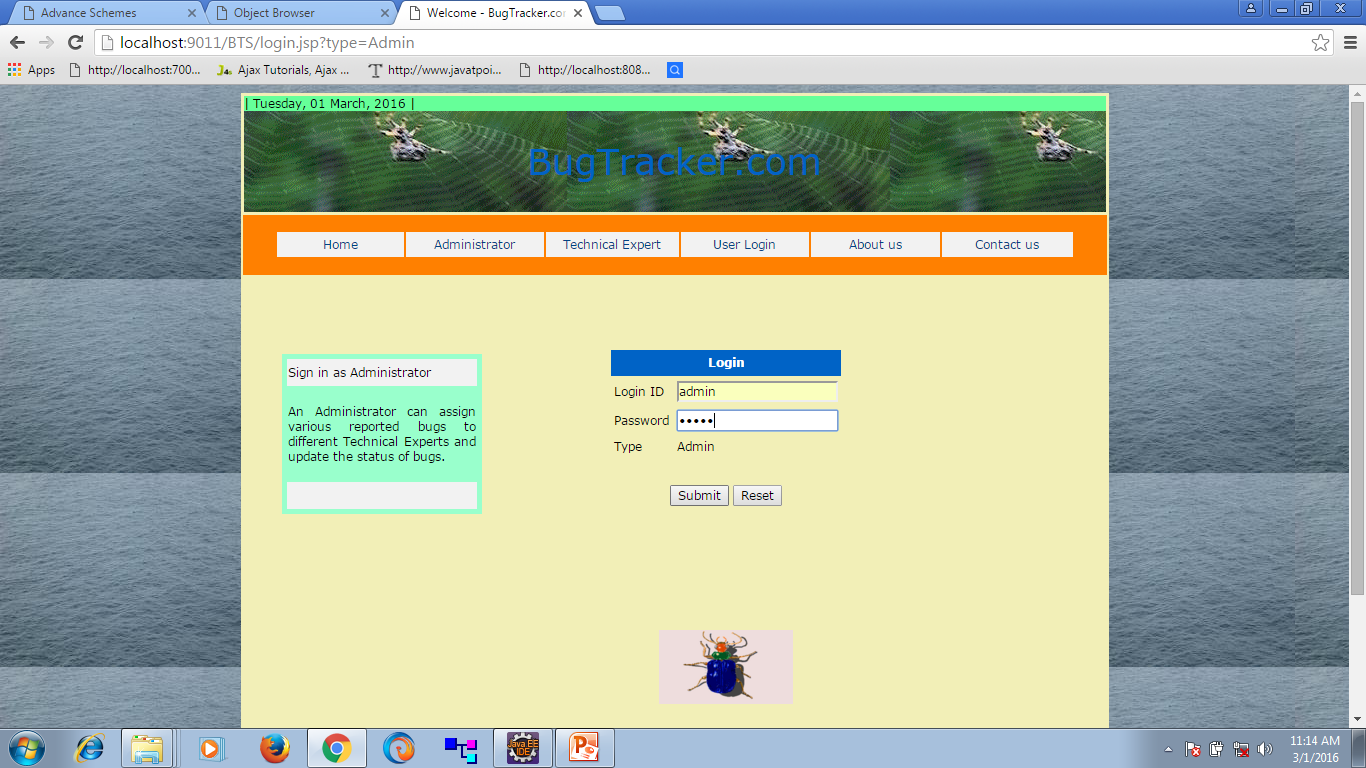
# 4. USER INTERFACES

The interfaces of this system are as below

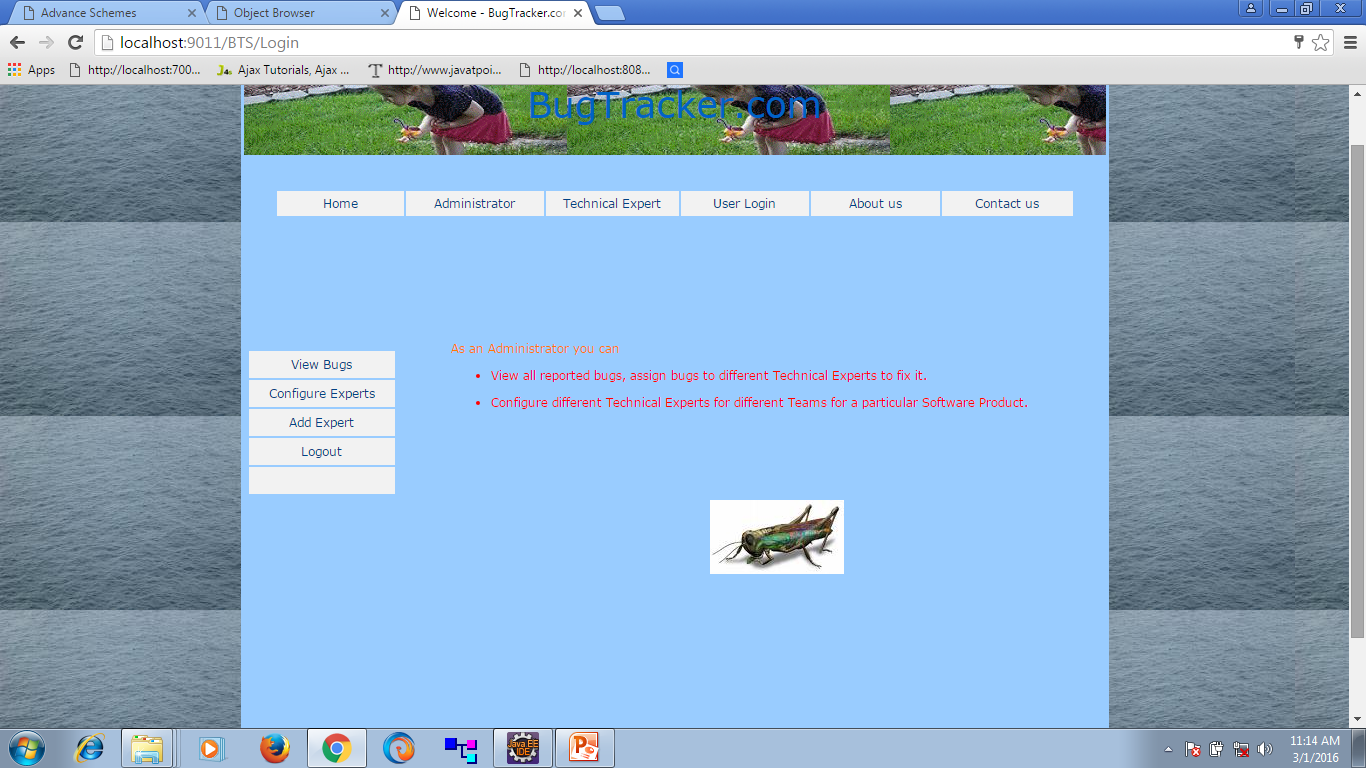
## 4.1Home Page

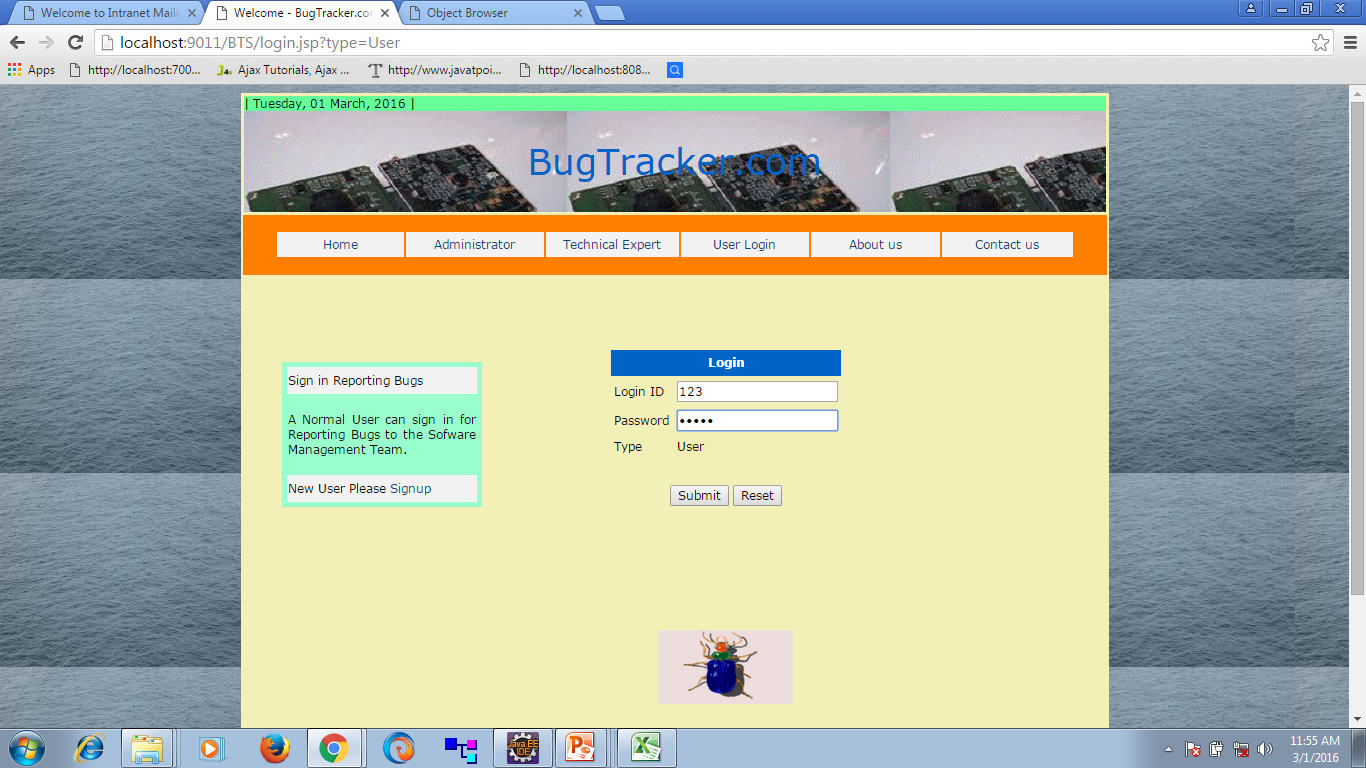


**4.2Admin Login**

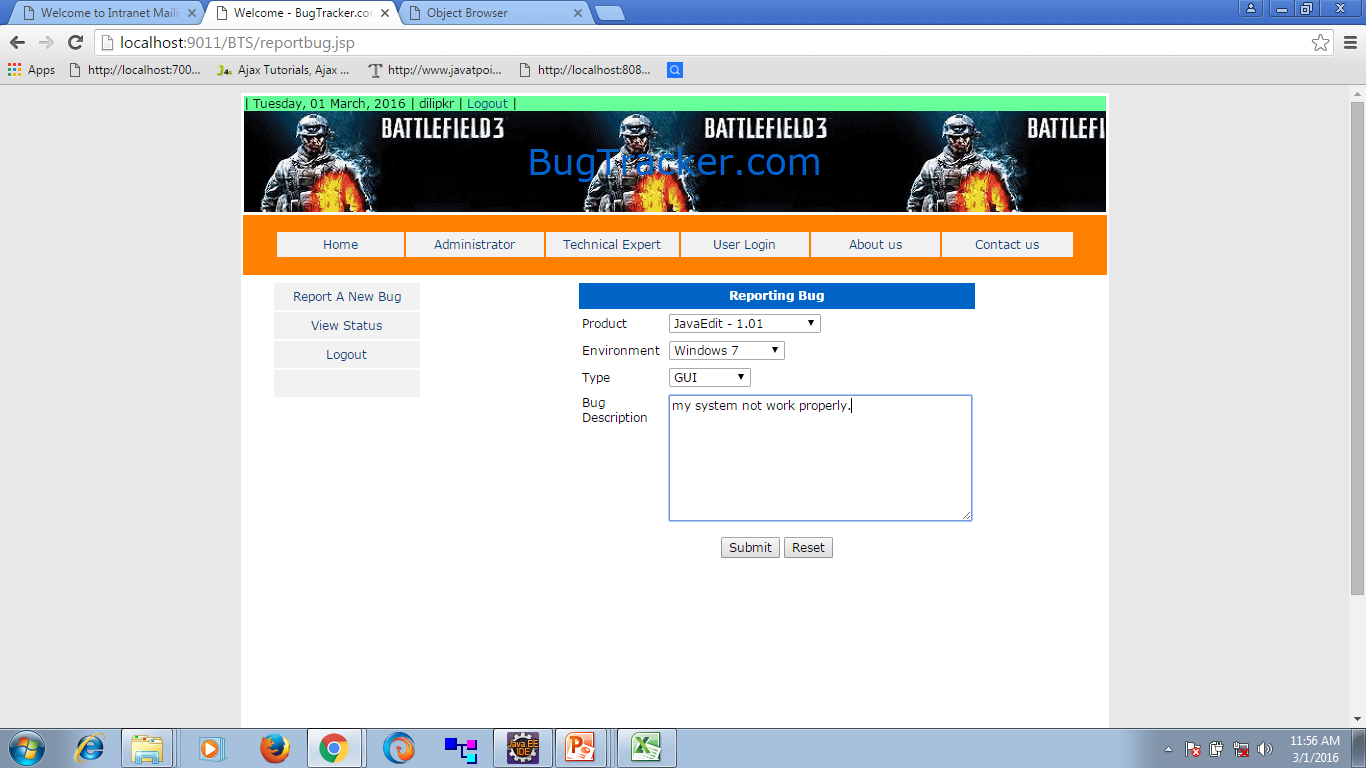


**4.3 Admin Page**

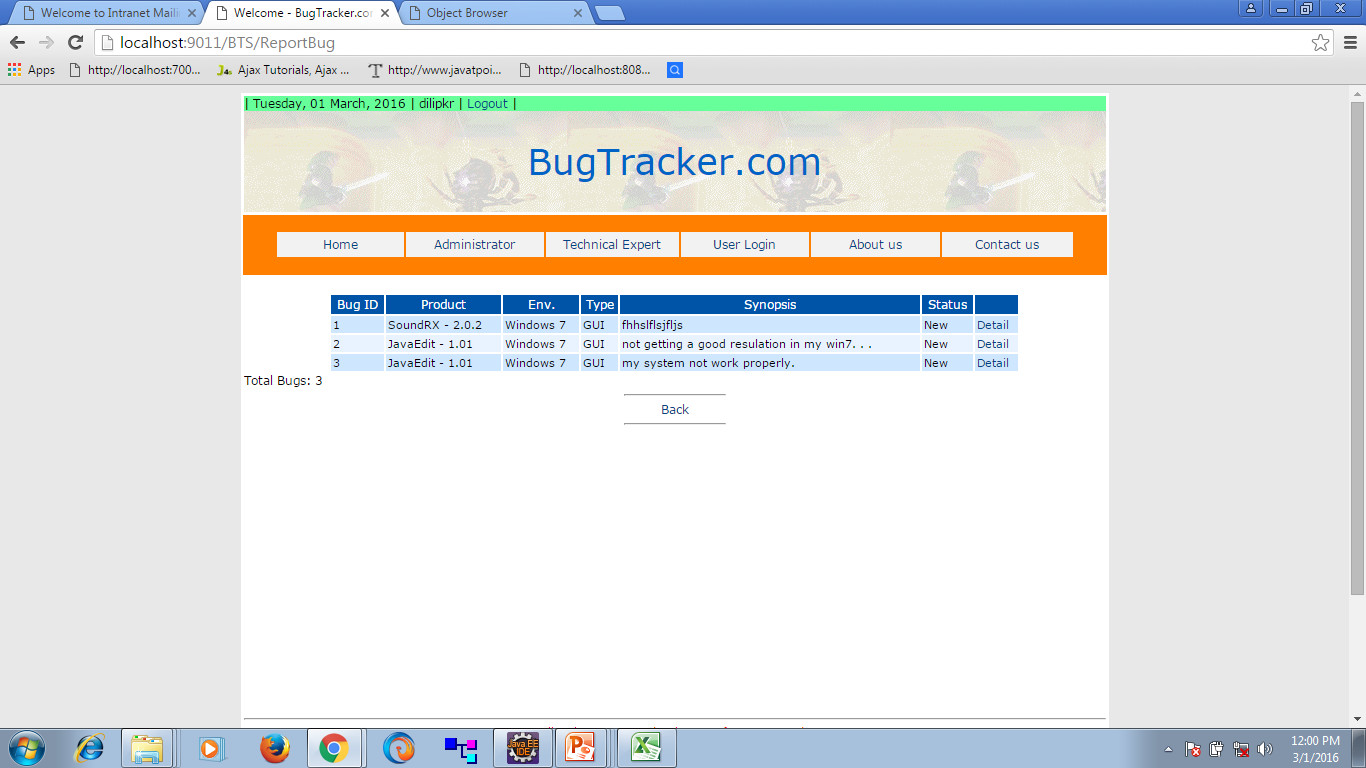


**4.4User login**

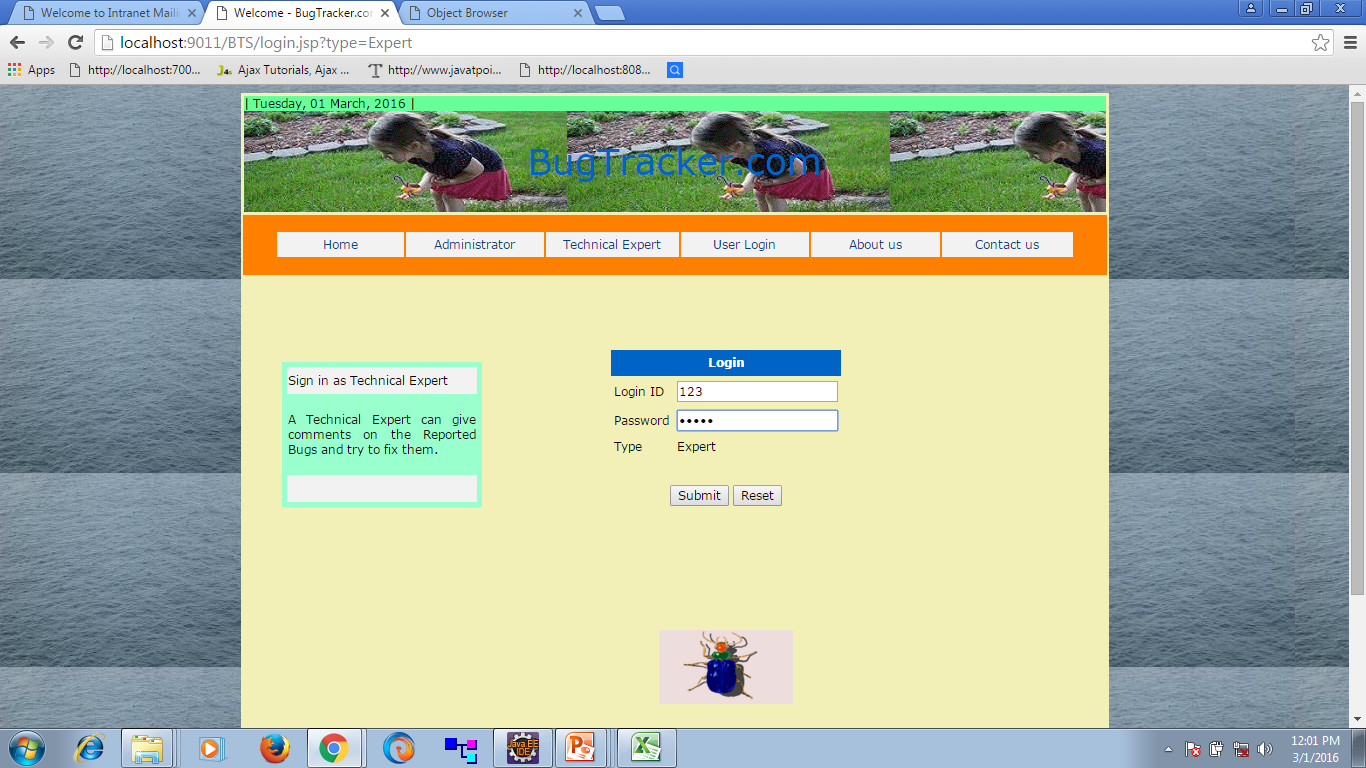
## 4.5 User Page



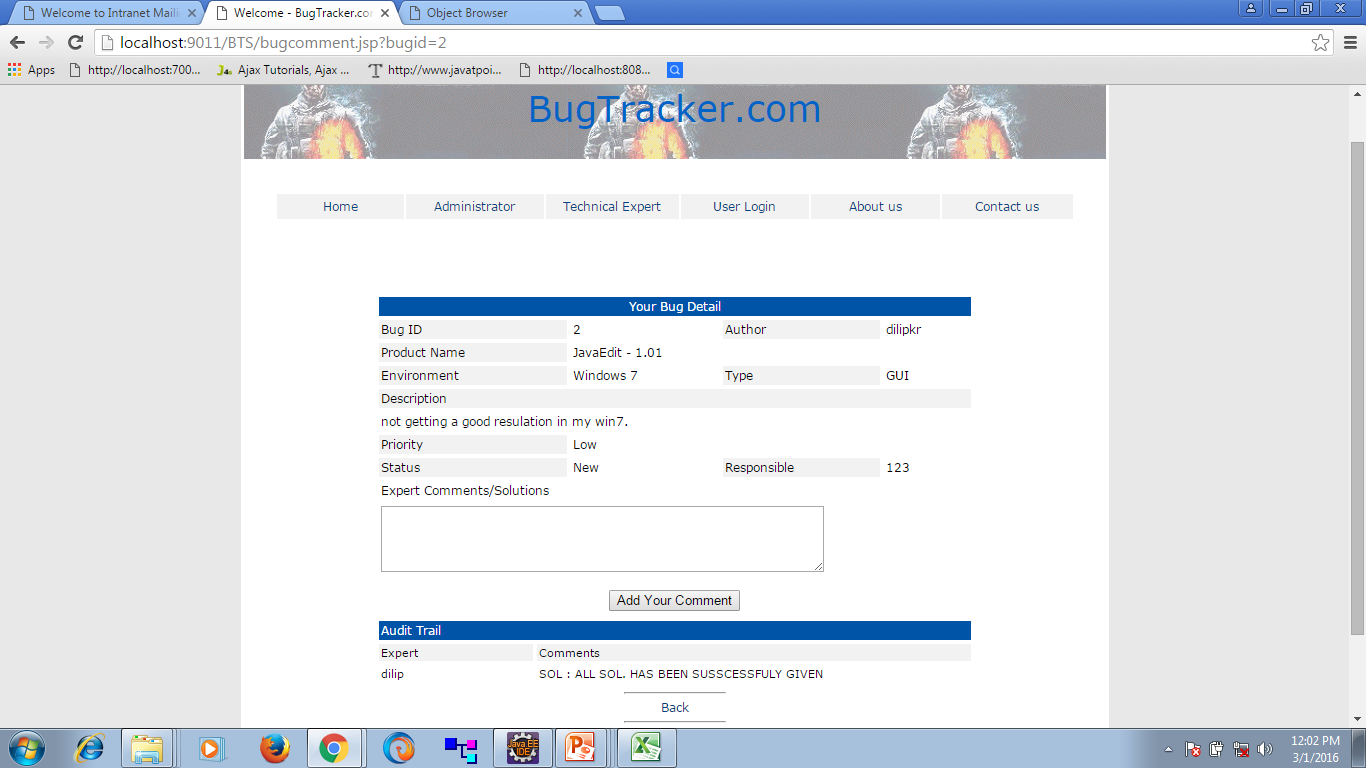
## 4.6 Bugs show in user page



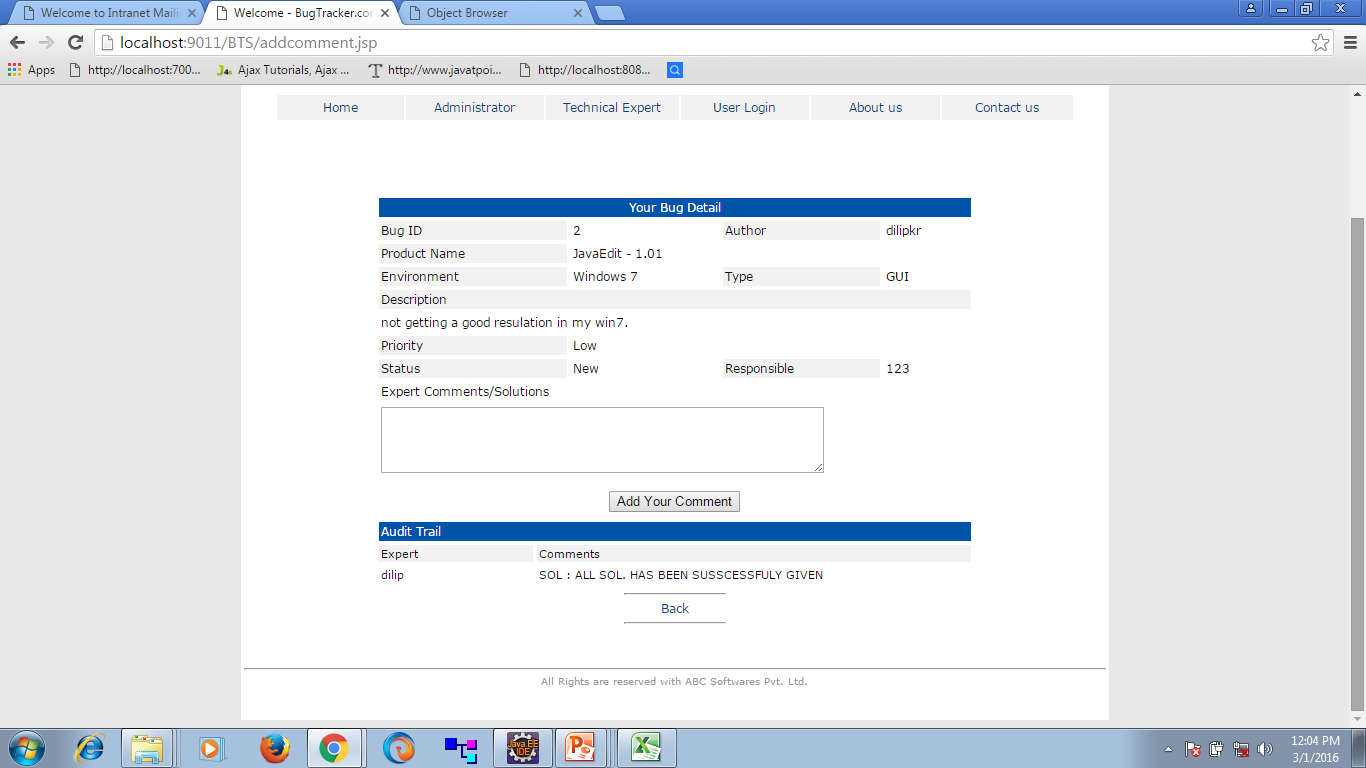
**4.7 Expert Login**



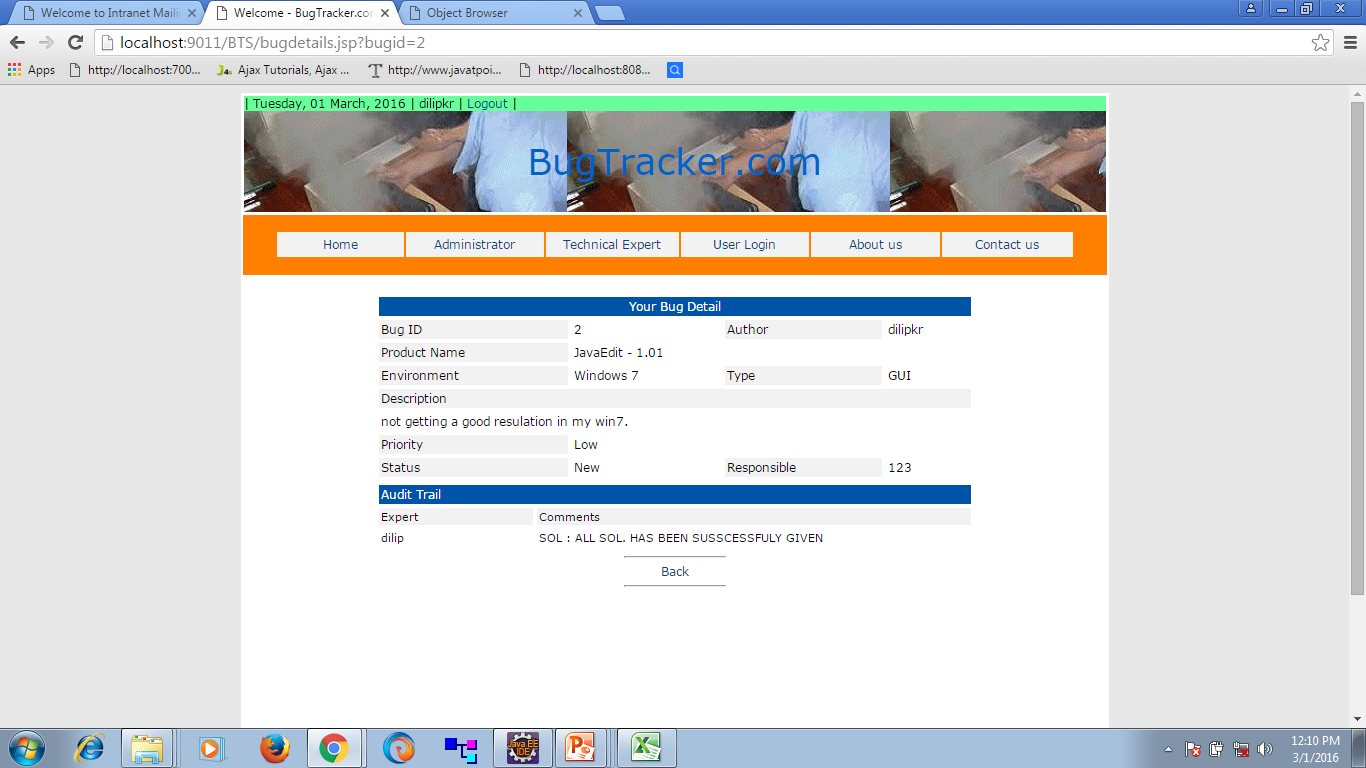
* 1. **Expert resolve bugs**



* 1. **Expert enter solution**



* 1. **User see solution of bugs**



**4.11 About us**



**5. TESTING**

Software Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behavior of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following routine procedure. One definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are operations the tester attempts to execute with the product, and the product answers with its behavior in reaction to the probing of the tester[citation needed]. Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connoted to mean the dynamic analysis of the product—putting the product through its paces. Some of the common quality attributes include capability, reliability, efficiency, portability, maintainability, compatibility and usability. A good test is sometimes described as one which reveals an error; however, more recent thinking suggests that a good test is one which reveals information of interest to someone who matters within the project community.

**Introduction**

In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

Software testing may be viewed as a sub-field of Software Quality Assurance but typically exists independently (and there may be no SQA areas in some companies). In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in the code or deliver faster.

Regardless of the methods used or level of formality involved the desired result of testing is a level of confidence in the software so that the organization is confident that the software has an acceptable defect rate. What constitutes an acceptable defect rate depends on the nature of the software. An arcade video game designed to simulate flying an airplane would presumably have a much higher tolerance for defects than software used to control an actual airliner.

A problem with software testing is that the number of defects in a software product can be very large, and the number of configurations of the product larger still. Bugs that occur infrequently are difficult to find in testing. A rule of thumb is that a system that is expected to function without faults for a certain length of time must have already been tested for at least that length of time. This has severe consequences for projects to write long-lived reliable software.

A common practice of software testing is that it is performed by an independent group of testers after the functionality is developed but before it is shipped to the customer. This practice often results in the testing phase being used as project buffer to compensate for project delays. Another practice is to start software testing at the same moment the project starts and it is a continuous process until the project finishes.

Another common practice is for test suites to be developed during technical support escalation procedures. Such tests are then maintained in regression testing suites to ensure that future updates to the software don't repeat any of the known mistakes.

It is commonly believed that the earlier a defect is found the cheaper it is to fix it.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Time Detected | | | | |
| Time Introduced | Requirements | Architecture | Construction | System Test | Post-Release |
| Requirements | 1 | 3 | 5-10 | 10 | 10-100 |
| Architecture | - | 1 | 10 | 15 | 25-100 |
| Construction | - | - | 1 | 10 | 10-25 |

In counterpoint, some emerging software disciplines such as extreme programming and the agile software development movement, adhere to a "test-driven software development" model. In this process unit tests are written first, by the programmers (often with pair programming in the extreme programming methodology). Of course these tests fail initially; as they are expected to. Then as code is written it passes incrementally larger portions of the test suites. The test suites are continuously updated as new failure conditions and corner cases are discovered, and they are integrated with any regression tests that are developed.

Unit tests are maintained along with the rest of the software source code and generally integrated into the build process (with inherently interactive tests being relegated to a partially manual build acceptance process).

The software, tools, samples of data input and output, and configurations are all referred to collectively as a test harness.

**6 . Limitations and Future Enhancements**

**Limitations of the system:**

* + Only the permanent employees can access the system.
  + System works with windows’98 and its compatible environments.
  + Advanced techniques are not used to check the authorization.
  + Once the employee is registered to a course cannot drop, without completing.

**Future Enhancements:**

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

* As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
* Because it is based on object-oriented design, any further changes can be easily adaptable.
* Based on the future security issues, security can be improved using emerging technologies.
* Attendance module can be added
* sub admin module can be added

**7.Project Summary**

This project Bug Tracking for Improving Software Quality and Reliability is to keep track of employee skills 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.

This project will be accessible to all developers and its facility allows developers to focus on creating the database schema and while letting the application server define table based on the fields in JSP and relationships between them.

This application software has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using Java as front end and Oracle as back end in Windows environment. The goals that are achieved by the software are:.

* Improved productivity.
* Optimum utilization of resources.
* Efficient management of records.
* Simplification of the operations.
* Less processing time and getting required information.
* User friendly.
* Portable and flexible for further enhancement.

**8. BIBLIOGRAPHY**

1) [www.google.com](http://www.google.com)

2) [www.wikipedia.org](http://www.wikipedia.org)

3) Jewei Han and Micheline Kamber(2005) “Elementary implementation of Tomcat”.

4) The complete reference for Java by Helbert Schildt

5) The complete reference for Java Server Pages by Helbert Schildt

6) John M. Pierre(2000) “Practical issues for automated generation of blogs”

7) Google News [www.googlenews.com](http://www.googlenews.com)

8) Microsoft Network [www.msn.com](http://www.msn.com)

9) CDNOW [www.cdnow.com](http://www.cdnow.com)

10) Lycos [www.lycos.com](http://www.lycos.com)

11) 95506am.blogspot.com

12) Eblog.com

13) Bloggers.com

14) Programming in Java by Balaguruswamy.