

Bachelor's Thesis

Information Technology

2011

Sharma Dahal Prabhakar

**SOFTWARE PROJECT TRACKING  
SERVER INSTALLATION AND  
IMPLEMENTATION IN PARTERRE  
PROJECT**



**TURUN AMMATTIKORKEAKOULU**  
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Information Technology

Spring 2011

Ferm Tiina

Sharma Dahal Prabhakar

## SOFTWARE PROJECT TRACKING SERVER INSTALLATION AND IMPLEMENTATION IN PARTERRE PROJECT

This thesis aims to design a server for tracking subproject of project PARTERRE. It aims to install and implement a project tracking and bug tracking tool on the server after selecting a suitable tool for the project. Additionally, thesis defines procedure of securing information stored on the server related to tracking server of PARTERRE project. It includes securing login credentials and user accesses. It also mentions backup process for securing data stored on the server.

The thesis covers project and project tracking theory. It also discusses bugs and recently available bug tracking tools with their features. Including research, it deals with installation and administration of the selected tool. It also discusses benefits and drawbacks of the TracProject tool.

On successful implementation of thesis, the developed server will be deployed for the PARTERRE project for generating and reporting tickets and milestones during the pre-pilot and pilot phase testing of Demos and eTM program of PARTERRE.

KEYWORDS:

PARTERRE, Demos Plan, eTM, Trac, Bug

## **FOREWORD**

This thesis is developed from the practical activities carried during PARTERRE project pre-pilot testing phase; I acknowledge and owe myself to Ferm Tiina and Ojala Olli for accepting me as a member of their team. I am grateful to Turku University of Applied Sciences accepting me as its International student. I would like to thank the Cisco Academy of Turku University of Applied Sciences teachers Vesa Slotte, Ossi Väänänen, Tero Virtanen for their guidance, suggestions and encouragement during the whole course. I express my gratitude to Poppy Skarli and Riikka Kulmala for providing me with work placement opportunities. Lastly, I appreciate and thank my dear parents and lovely brother for their enthusiastic support and encouragement during my studies in this foreign land.

Spring 2011, Turku

Sharma Dahal Prabhakar

## **TABLE OF CONTENTS**

<b>1.</b>	<b>INTRODUCTION</b>	<b>- 6 -</b>
<b>2.</b>	<b>PROJECT TRACKING</b>	<b>- 8 -</b>
2.1.	Project Tracking	- 10 -
<b>3.</b>	<b>BUG</b>	<b>- 11 -</b>
3.1.	Controlling bugs	- 12 -
3.2.	Bug tracking tool	- 13 -
3.3.	Comparison of bug tracking systems	- 25 -
<b>4.</b>	<b>TRACKING THE PARTERRE PROJECT</b>	<b>- 30 -</b>
<b>5.</b>	<b>INSTALLATION AND IMPLEMENTATION OF TRACPROJECT</b>	<b>- 34 -</b>
5.1.	Installation of TracPoject	- 34 -
5.2.	Implementation of TracProject tool	- 38 -
5.3.	Benefits and Drawbacks of TracProject Installation	- 39 -
<b>6.</b>	<b>SECURING SERVER AND INFORMATION</b>	<b>- 40 -</b>
<b>7.</b>	<b>SUMMARY</b>	<b>- 41 -</b>
<b>8.</b>	<b>RECOMMENDATION</b>	<b>- 42 -</b>
<b>9.</b>	<b>REFERENCES</b>	<b>- 43 -</b>

## **FIGURES**

Figure 1. Welcome page of Trac tool for PARTERRE Demos Project viewed on 10/04/2011

Figure 2. GUI showing ticket list on Redmie tracking tool viewed on 10/04/2011

Figure 3. Dashboard for Mantis tracking tool viewed on 10/04/2011

Figure 4. Home page for JIRA tracking tool viewed on 10/04/2011

Figure 5. A view of Web Bug Tracker viewed on 10/04/2011

Figure 6. Bugs reporting page view for Bugzilla viewed on 10/04/2011

Figure 7. A view of chart page for FastBug Track viewed on 10/04/2011

Figure 8. AVS demo project view with the Woodpecker IT tool viewed on 10/04/2011

Figure 9. Ticket page view of the Ozibug tracking tool viewed on 10/04/2011

Figure 10. User access control view of the BugImpact tracking tool viewed on 10/04/2011

Figure 11. Milestone creation page view of Assembla viewed on 10/04/2011

Figure 12. Demos Plan view viewed on 18/04/2011

Figure 13. Flow diagram showing working procedure of Genshi viewed on 20/04/2011

## **TABLES**

Table 1. Comparison of tracking tools on the basis of general characteristics

Table 2. Comparison of available features on different tracking tool

## **Glossary and Terms**

Bug	Bug is an error present on a program that provides output different than the expected one.
Bug Tracking	Process of keeping records errors and problem that are found on the program or any product.
Bug Tracking tool	Different software or programs designed for keeping track of errors present on the program.
PARTERRE	PARTERRE is an EU funded project developing Demos and eTM program for spatial planning and development of society.
Project	A piece of work having a certain life cycle and target.
Project tracking	The process of keeping record of a project activities on the basis of different factors like time, money etc.
TracProject	Project and bug tracking tool developed by Edgewall.

## **1. Introduction**

Software or a programming code crosses different phases during its development. Phases include generating of UML diagram, generating of codes, implementation of codes and testing. None of the programs can be 100% complete in its first phase. It should be tested from different views, like a coder's view, end user's view, customer's view and so on. Different kinds of testing are carried out on different level which include end user testing, code level testing, usability, functionality and feasibility testing of the product. Product testing helps to determine bugs, errors and problems that are present on the programs. Reports are generated while managing and keeping records of these kinds of error and problems which need to be properly managed and well-tracked so that developer can easily find the related problems in a short period of time and fix them.

This thesis aims in installation and implementation of a server for storage of documents and reports using different supporting tools. Supporting tools track project activities, create tickets for reporting bugs and errors present in the Demos and eTM programs of the project PARTERRE. It includes learning about different supporting tools that are available for bug and project tracking, installing and implementing them for the proper tracking of pre-pilot phase testing of project PARTERRE. The developed server will be used by the pre-pilot testing phase members for creating and storing their reports.

The PARTERRE project is an EU-sponsored Electronic Participation Tools for Spatial Planning and Territorial Development Project. The main coordinator for the project is Regione Toscana, Italy along with other seven participating members. Turku University of Applied Sciences is one of the partner organizations of the project and is working on the pre-pilot testing phase of the program. PARTERRE contains of two different products: Demos Plan and eTM. Demos Plan is used for spatial planning of development of certain infrastructures on a certain locality and the eTM program is used for collection of ideas, votes of the different personnel who are directly and indirectly affected by the spatial planning.[1]

The Author has volunteered for the project. He developed a separate server and installed the necessary project tracking and bug tracking tool for the pre-pilot and pilot phase testing of the program. This thesis mainly focuses on installation and implementation of project and the bug tracking tool on the PARTERRE server after selecting the best tool meeting all the requirement of the project. This thesis is composed of four parts. The first part deals with the basics of tracking systems of a project. It introduces the tracking tools, their necessities and uses on a project. The second part deals with different kinds of tracking tools that are currently available. It briefs about the different kinds of tracking tools available, their advantages and disadvantages. It also compares these products from different aspects and chooses the best tool for PARTERRE. The third part deals with installation, implementation and usage of the tracking tool used. It contains detailed information about the installation procedure and the use of installed tool. The final part of this thesis deals with the possible problems that arise during the installation of the product followed by conclusion and suggestions for the project.



## **2. Project tracking**

Project is defined as a piece of work that has a certain life period. It is goal-oriented piece provided with certain resources and listed under certain rules and regulations. Projects are of different kinds depending on the work they relates to. They may be research or development projects. For example: Designing and developing software is a development project whereas studying the impacts of different environmental phenomena during the development of project is related to research and development.[2]

A project with a certain life cycle with a starting and an ending point has different phases in its life. These phases include steps from starting of the project till its end. Different phases that appear during project activities are as follows:

### Conceptual phase

The conceptual phase is the first phase where different concepts about the project are generated. It includes ideas like what the project is about, what will be the end product after the completion of project etc. Different types of concepts from different level of people are accumulated during this phase. These concepts are analyzed and studied for the feasibility of the project.

### Planning phase

The planning phase can be started from the middle of the conceptual phase as a second phase of project. The feasible and best concepts among the different concepts collected will be planned in this phase. This phase mainly deals with planning of the resources, supplement materials required during the project work. The planning phase is the most important phase of the project because it carries almost 50% life of the whole project. Only a well- planned project can only be successfully completed.

### Definition and design phase

In this phase the order of working of the project is defined. It can contain rules and regulations that are to be met by the project. Similarly, designing indicates

diagrammatic representation of the project. It can be indicated as a Unified Modelling Diagram (UML) that shows the procedural picture of the project.

#### Implementation phase

The planning and designing that were built in the previous phases will be developed by the personnel involved in this phase when the real working begins.

#### Acceptance phase

The acceptance phase is last phase of project studies and testing like end user testing, usability testing, feasibility testing etc are carried out in this phase. If the developed product from the project meets the specified requirements in complete order, it is then launched to the market for public use.

#### Example of Project phases on a software development process

Let's consider Software Company that wants to develop a new software for the market. Initially different managerial level people will try to find what kind of new product can be developed which can be useful for local use as well as the development of the company. Different ideas are collected from different persons who are involved in the process. After the collection of all ideas, a base line concept is developed. This crosses the conceptual phase of the project. After having a certain concept, they will start to make plan for the project. They will decide on what kind of interface can be used for the software, what the best programming language for the best performance will be, how many coders and designers are necessary for the completion of task and how long it will take to complete it. Also, they will define the monetary value that will be necessary for the project. Salary, benefits that must be provided for coders and others staff. Also, they will study the procedural failures, risk analysis, and other aspects of the project during the phase. On planning the project in a well organized manner, they will start to define the project rules and design the work flow to reach the project target in a successful manner. Coders will start to write the necessary programs and interface designers will start designing the software on the implementation phase and finally after completion of implementation phase it will be tested for its usability, feasibility condition from the end user point of view in the acceptance phase.

## 2.1. Project Tracking

For the successful achievement of a project, it is necessary to keep records of all activities of a project. Project tracking is a process of managing and keeping records of project activities. Tracking of a project is vital part of a project. It helps to determine the flow of project activities. The status of project, working process, risk and other management activities can be easily read if the project is tracked. It helps to resolve the errors that are arose during the project process in a good order. Project tracking saves time, resources and makes the workload easier so that it can be completed within its life span.

Let us consider the example used in the Project part of this document. If the defined software development is not tracked, it is not possible to develop the software in a properly organized order. For example: If the interface designers do not develop interface modelling in time and provide it to the programmer, programmers may develop a different kind of interface than it needs to be. Also, during the testing phase of the product, if the activities are not tracked, many people may work on the same function which, although beneficial, causes the loss of time that affects the whole life span of the project. Also, it may not be possible to report the bugs and errors on product and that can worsen project status. Thus, it is vital to track each and every step of a project.

Project tracking is highly important. A properly tracked project has various distinguished features. In a tracked project, it is easier to know the status of project. It reveals complete information about the flow of project activities. It also determines the ongoing activities and upcoming activities. When a project is tracked, the comparison of project with respect to time frame becomes easier and that helps to understand the life span of the project showing progress report of project and its future.

### **3. Bug**

A bug is defined as the error present on any software or hardware. A bug is produced on the software level rather than hardware level. A bug mainly is present on the coding level of program. It is the illogical expression of the codes, mathematical expressions on the programming. It is generated if the mathematical expression is miscalculated or presented in wrong order. A bug harms the program in different levels. The level of disaster can be from severe to normal depending on the type of product or error. A bug is responsible for producing negative results than expected. On software level, a bug generally makes software act slower than their normal time. Software may crash or stop to work sometime because of a bug present in it. On hardware level, a bug can crash and destroy the hardware making hardware unusable for any future purpose. Some of the errors created by bugs are discussed below:[3] [4]

#### Bug present on Toyota Prius Car

In the hybrid car, Prius, designed by Toyota, a bug was present on the programming code. The bug caused warning light system illuminate for no reason. It also stalled gasoline engine unexpectedly. This caused the company to withdraw its product from the market losing of millions of dollars.

#### Mariner I space probe

This is another bug present in the rocket. The mathematical formula implemented for the rocket was not properly transcribed to the programming code and this caused rocket to move away from its trajectory path. The developer had to destroy the rocket in the Atlantic Ocean. It caused loss of millions of dollars and could have placed many lives in danger if not destroyed on time.

#### Kerberos Random Number Generator

Due to a bug present on the Kerberos Random Number, it was possible to break the security authentications of computers that used Kerberos Security authentication. The bug was created because the programs random number generator was not properly translated with real random seed.

### AT&T Network outage

The bug present on newly released software controlling AT&T's 4ESS long distance switch crashed the computers when they received a specific message from their neighbour machines. Specifically, 114 switches crashed within every six seconds which affected 60,000 people for nine hours interrupting their long distance services.

### Intel Pentium floating point divide

An error on silicon of Intel's Pentium chip miscalculated dividing floating-point numbers in a specific range. The error defected about 5 million chips causing Intel to lose about \$475 million.

### 3.1. Controlling bugs

Since a bug is the product of human error or negligence, it can be controlled by a human being. Some of the methods of controlling bugs are as follows:

#### Proper calculation of Mathematical formulae and expressions

As the error Mariner I space probe created due to improper transcribing of mathematical expressions, it is important to carry out proper calculations of mathematical formulae before they are implemented on any kind of programming code. All the mathematical formulas that are to be used for the program should be properly organized in a logical order so that no error would occur on the program.

#### Expression of programming codes in a logical format and order

It is very important for a programmer to design the programming codes in a logical and arranged order to avoid any bugs and errors. Designing and expressing codes in an improperly organized order can cause a program to create wrong output. For example: If an ATM machine is programmed to ask withdraw cash information before account information; it will lead to error and the whole program is of no use. Hence it is vital for a program to design the codes in a logical and organized manner. A UML diagram representing the whole program in a step-by-step order can be used to control bug.

#### Testing the product

This is the final effort to control the bugs. Testing helps to determine the bugs and errors that are present on any kinds of program. Different kinds of testing can be done on different levels for controlling the bugs that are present on the program. Some of the testing types are described below:

➤ End-user testing:

End-user testing is the testing carried on the customer level. It helps to determine user-friendliness of the product. End user testing helps to determine if the developed product meets the requirement of the customer.

➤ Usability testing:

This is another kind of testing to determine the bugs present on the software. It determines the usability of the developed program or software. During the usability testing of a product, the usefulness of the product is determined. Similarly, the speed of the programs, stability and other factors are studied during the usability testing of the product.

➤ Feasibility testing:

Feasibility testing helps to determine the feasible condition of the developed product. It studies similar kinds of products available on the market and compares the developed product with the rest.

➤ Code-level testing:

Code level testing is done on the code of the program. This testing helps to determine the errors that are developed on a program during the generation of codes. Coders develop different test cases for code study for removing the errors that are present on the code.

### 3.2. Bug tracking tool

During the testing environment different bugs are to be found. Such bugs need to be tracked and reported to the coders so that they can be removed in the future. Different tools are available to keep records of such bug and error. Such tools are called bug tracking tools. Bug tracking tools are very useful in the software developing industry because they help to organize the reports and track activities that are generated during the testing of the developed software. Bug tracking tools create tickets that indicate the severity level of the bugs present on the developed software. They help in quality assurance of the system. Bug tracking tools are web-based as well as software-based

depending on the vendor. Some of the available bug tracking tools are discussed below. [3]

### The TracProject:

TracProject developed by Edgewall Software is a bug tracking system with an enhanced wiki and issue tracking system which is useful for software development projects. It is a web-based approach providing a subversion for the version control of software development along with wiki and other reporting facilities. PHP and Python were used to develop the TracProject software. The welcome page Trac tool for PARTERRE Demos project is shown in Figure 1. Some of the features of TracProject are as follows:[5]

- web-based software project management
- provision of a subversion along with inbuilt wiki and reporting facilities
- authentication and authorization with different privilege level
- easy installation, management and administration
- support of multiple database platform
- powerful search options for the records, tickets and milestones

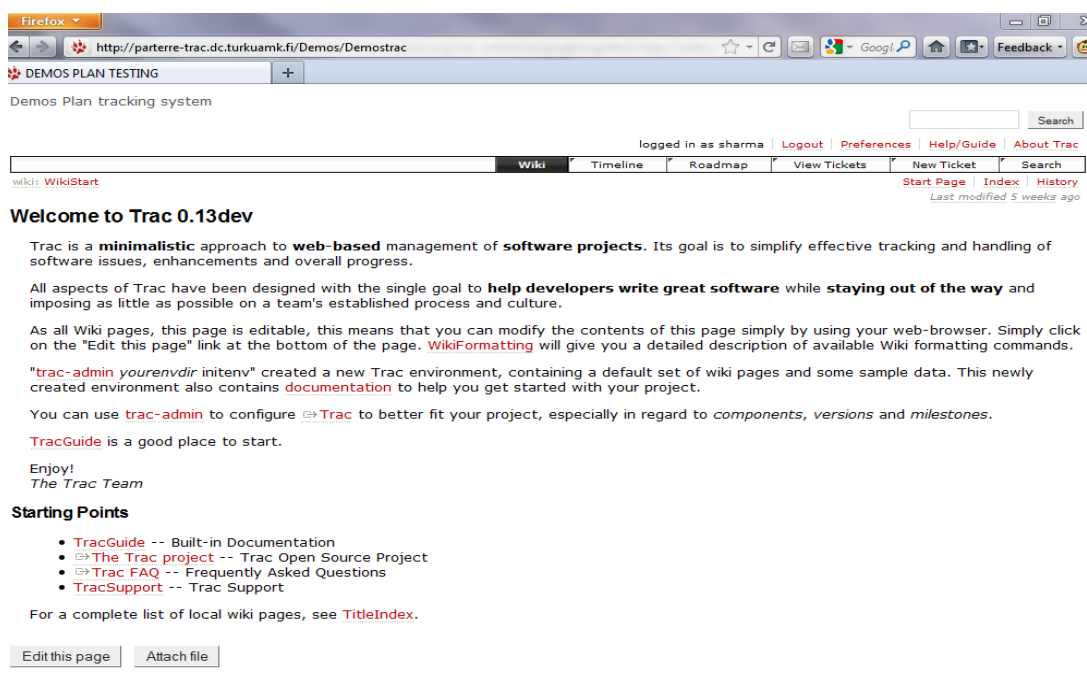


Figure 1. Welcome page of Trac tool for PARTERRE Demos Project viewed on 10/04/2011 <http://PARTERRE-trac.dc.turkuamk.fi/Demos/Demostrac>

### Redmine

Redmine, licensed under GNU General Public License is a web-based flexible project management and bug tracking application developed using Ruby as its programming language. It supports multiple projects with flexible tracking and role-based access control system. Multiple databases system can be associated with the application along with Multilanguage support. Figure 2 shows the created tickets list using the Redmine tracking tool. Some of the features of Redmine are as follows:[6]

- support of multiple projects with a single instance
- user based, authentication-based project can be defined
- flexible role-based access control and issue tracking system
- automatic generation of Gantt and calendar based on issue start and end dates
- time tracking functionality
- custom filed
- authentication, multiple language support
- multiple database support

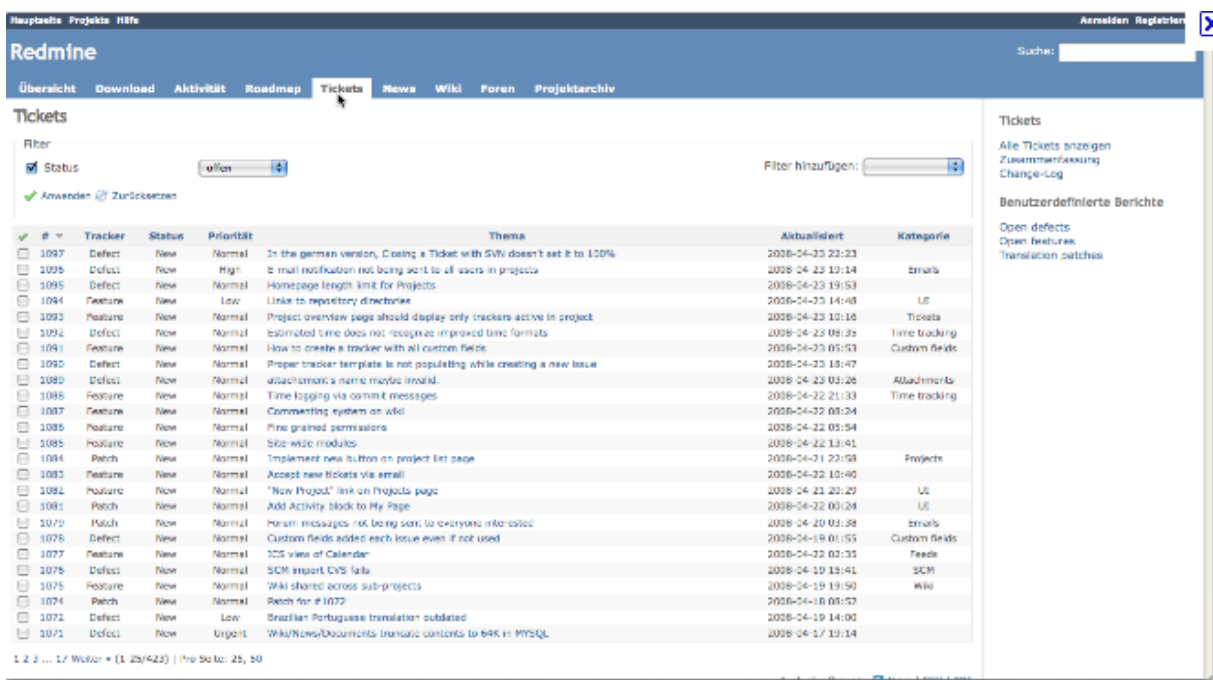


Figure 2. GUI showing ticket list on Redmine tracking tool viewed on 10/04/2011

<http://www.opensourcerails.com/projects/17-redmine>



## Mantis

Mantis, developed using the PHP scripting language, is a web-based bug tracking system that supports multiple database systems. It supports both UNIX and windows based operating systems. Mantis, open source application developed under GPL license, supports multiple projects on the same instance and is branched into sub-projects. The dashboard view for Mantis is shown in Figure 3. Some additional features of Mantis are as follows:[7]

- GPL licensed and free available
- easy installation, usability, simple user interface
- web-based supporting multiple platform, database, and projects
- built-in reporting and graphs for the issues and ticket
- email notification
- user authentication and access privilege level.
- mobile devices support

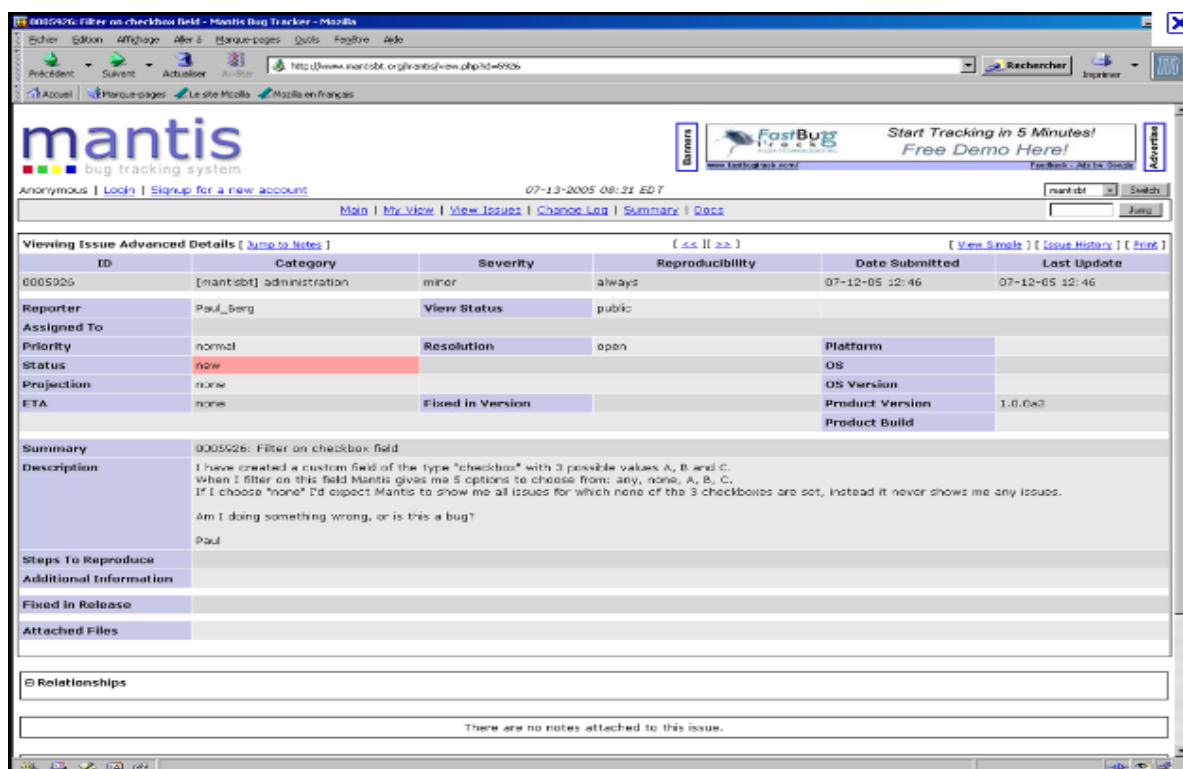


Figure 3. Dashboard for Mantis tracking tool viewed on 10/04/2011

<http://www.eclipse.org/m2m/atl/usecases/SoftwareQualityControlToolsInteroperability/>

## JIRA

JIRA is a leading project and bug tracking tool developed by the Australian company Atlassian. It is a free source for non-commercial projects but needs license for commercial users. JIRA comprises of issue and project tracking software useful for software developing company for generating improved quality of code or program. BMW, HBO, Apache, Boeing etc are using JIRA for controlling code qualities. JIRA provides bug tracking and project tracking tools separately. Bug tracking is a strong tool for issuing views, tickets and yet simple to use. The project tracking tool manages project activities by tracking a single application or several applications at the same time. Figure 4 indicates the home page view for JIRA. Some additional features of JIRA are as follows:[8]

- easily accessible from all browsers and mobile devices
- provision of email and RSS subscription on change on project
- runs as a standalone application
- built-in database and imports and supports multiple database system
- automated backup of the system
- multiple language support

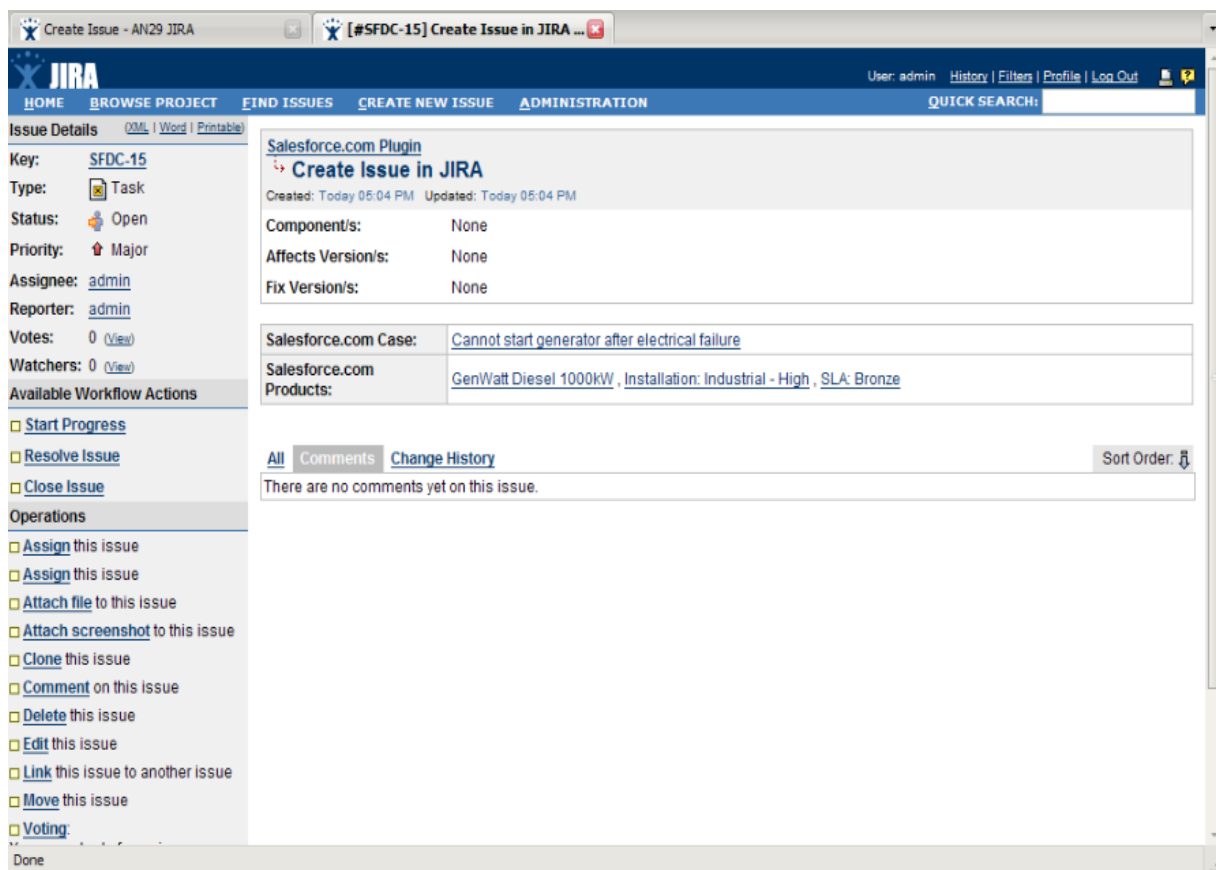


Figure 4. Home page for JIRA tracking tool viewed on 10/04/2011

<http://www.customware.net/repository/download/attachments/5177389/view-jira-case.PNG>

#### Web Bug Tracker:

Web Bug Tracker, developed by Vax Technology, is a platform dependent web-based bug tracking application used for bug and defect tracking during the software development process. It is easy to maintain, has a flexible managerial level, authentication and authorization of users with right control, quick searching options of tickets and issues. Figure 5 listed below is view of the Web Bug Tracker tool. Some of the other features are listed below: [9]

- open source application
- web-based application, easy installation and configuration
- use of SQL server database which is easy in development and management
- multi-user access through the application at the same time

Some of the software requirements for the application are as follows:

- .Net Framework 2.0
- IIS 5,0 or above
- IE 6.0 or above
- Windows XP/NT or above versions
- MS SQL Server

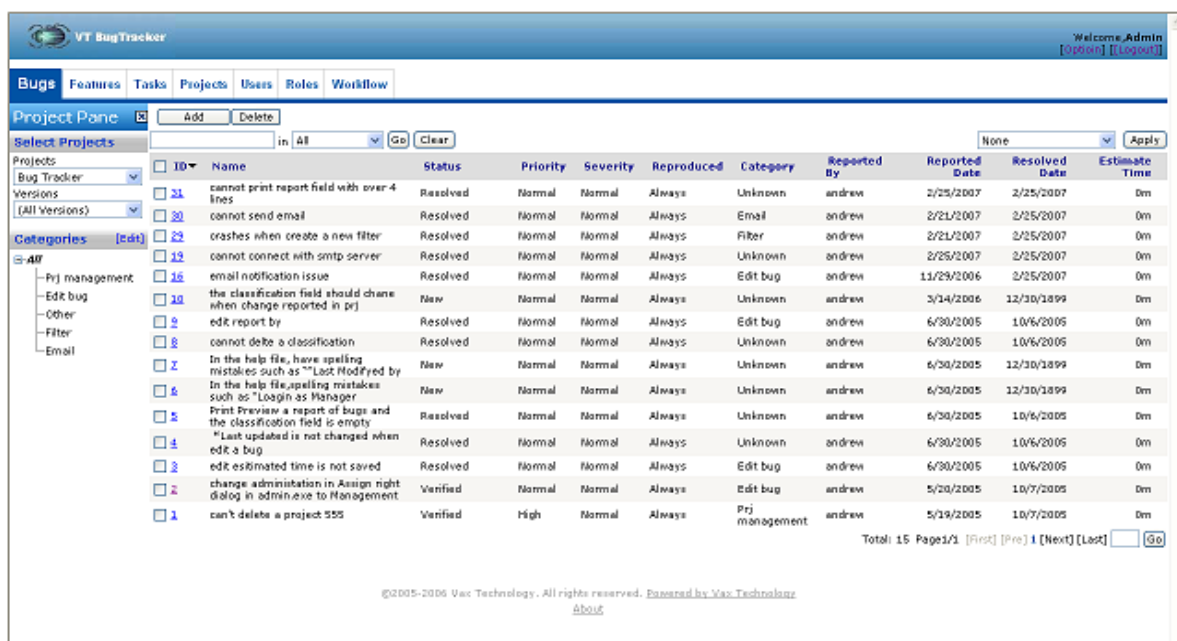


Figure 5 A view of Web Bug Tracker viewed on 10/04/2011

<http://www.vaxtech.com/web-bug-tracker/images/bug-big.png>

## Bugzilla

Bugzilla, a defect tracking system tracks bugs and code changes, manages quality assurances in a software development industry. It is a free product developed with different features from users to administration level. Figure 6 shows bug reporting page view of Bugzilla. Some of the features of this product are as follows:[10]

- email notifications if users prefer
- multiple formats bug listing facilities
- automatic duplicate bug detection
- time tracking/ request system
- higher security features with customizable installing features

- custom workflow and fields
- privilege level, authentication, authorization features
- multiple database supporting system

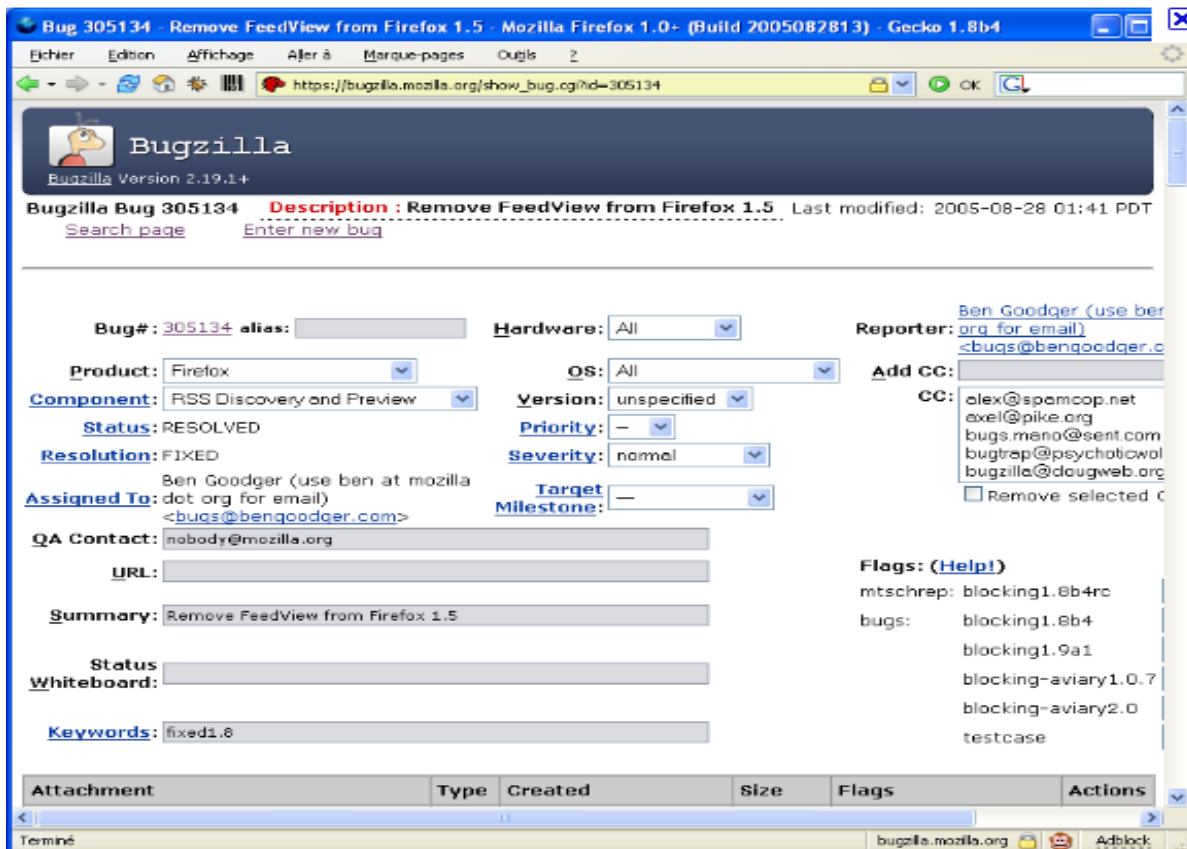


Figure 6. Bugs reporting page view for Bugzilla viewed on 10/04/2011

<http://www.download.ba/bugzilla/>

### Fast Bug Track:

Fast Bug track is a tracking tool developed by the Fast Bug Track team at Alcea Technologies. Fast Bug Track (FBT) maximizes the work efficiency of an organization by providing continuous communication between the users issue generated till it gets resolved. It provides the chart view of the project status as shown in Figure 7. Some of the features of FBT are as follows: [11]

- easy and fast installation
- simple and easy to use
- customizable interface
- hosting services for the users if they do not have servers.

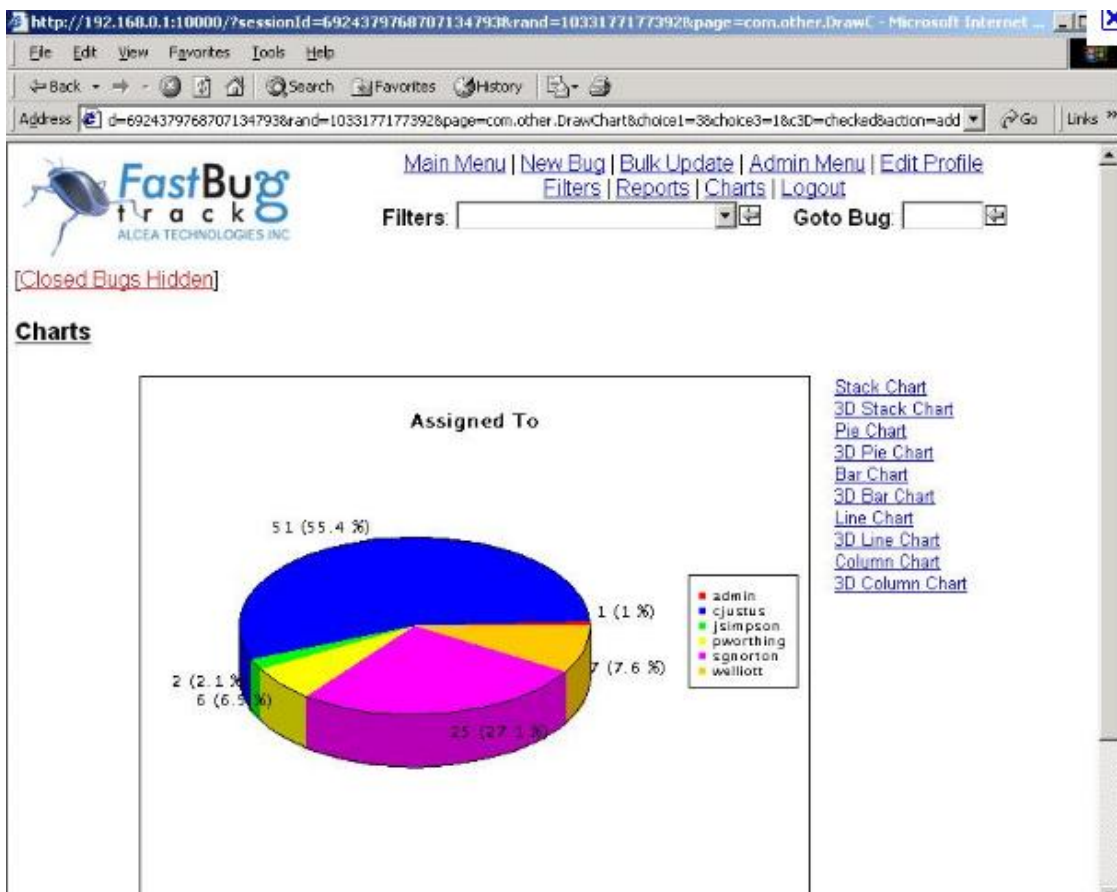


Figure 7. A view of chart page for FastBug Track viewed on 10/04/2011

<http://www.downpanda.com/alcea-fast-bugtrack-free-download.html>

### Agility:

Agility is an issue tracker developed by Agile Edge for the software development industry. Some of the main features of Agility tracker are as follows. [12]

- all the requirements, bugs and issues are located in a single interface
- easy search mechanism for bugs, tickets
- simple and easy to handle user interface
- report generation for the issues
- email notification to the users if any changes occur on the system

### Woodpecker IT

Woodpecker IT is a management software used for workflow management in a project.

It is a web-based flexible software providing different integration options for smooth

communication between the team members of the project. Figure 8 shows Woodpecker IT tracking tool used by AVS demo project to create multiple number of projects. Some of the features of Woodpecker IT are as follows: [13]

- workflow management
- tracking tables
- email notifications and entries
- import export options
- filter and search options
- online help

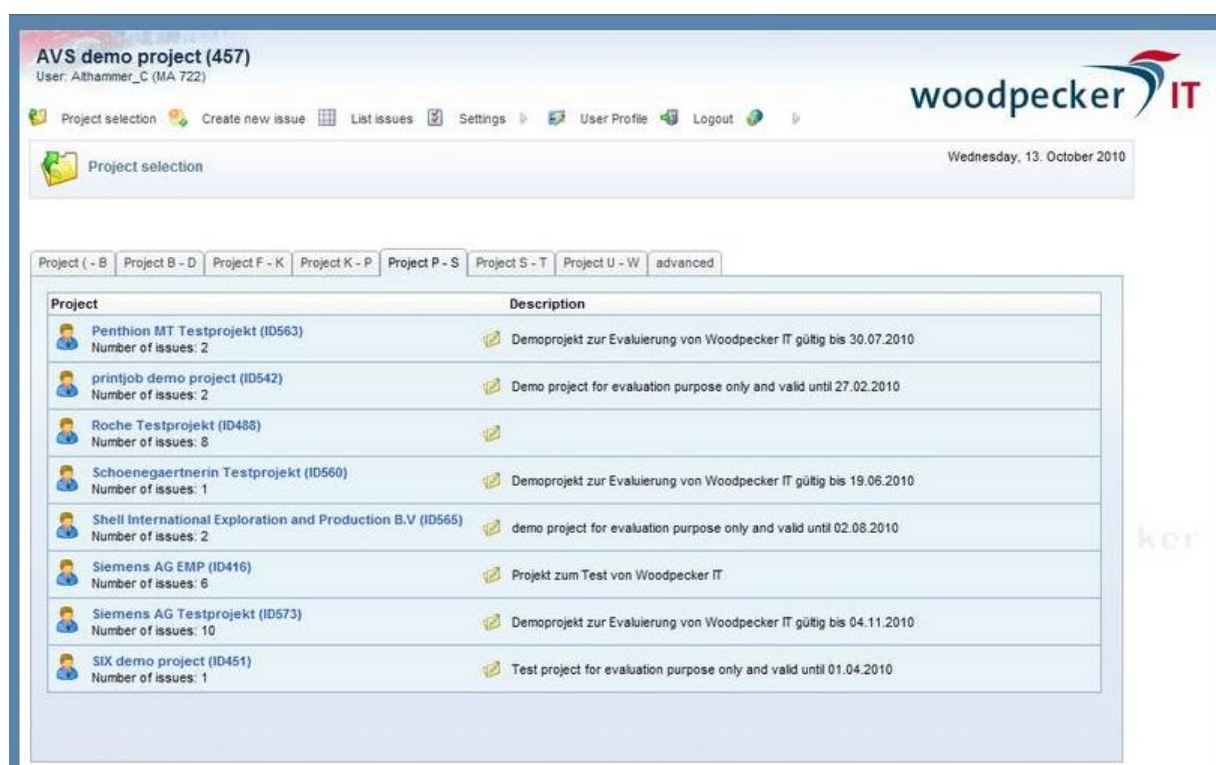


Figure 8. AVS demo project view with the Woodpecker IT tool viewed on 10/04/2011

<http://www.woodpecker-it.com/>

### Ozibug

Ozibug, independent of operating system is web-based bug tracking run on the Java servlet supporting platform. Ozibug manages bug reporting and tracks progress level

from bug initiation to its complete resolution. Figure 9 shows the ticket creation on Ozibug tracking tool. Some of the main features of Ozibug are as follows: [14]

- free licensing
- multiple module support
- file attachments
- email notifications
- pluggable authentication
- internationalization with multiple language support
- access control

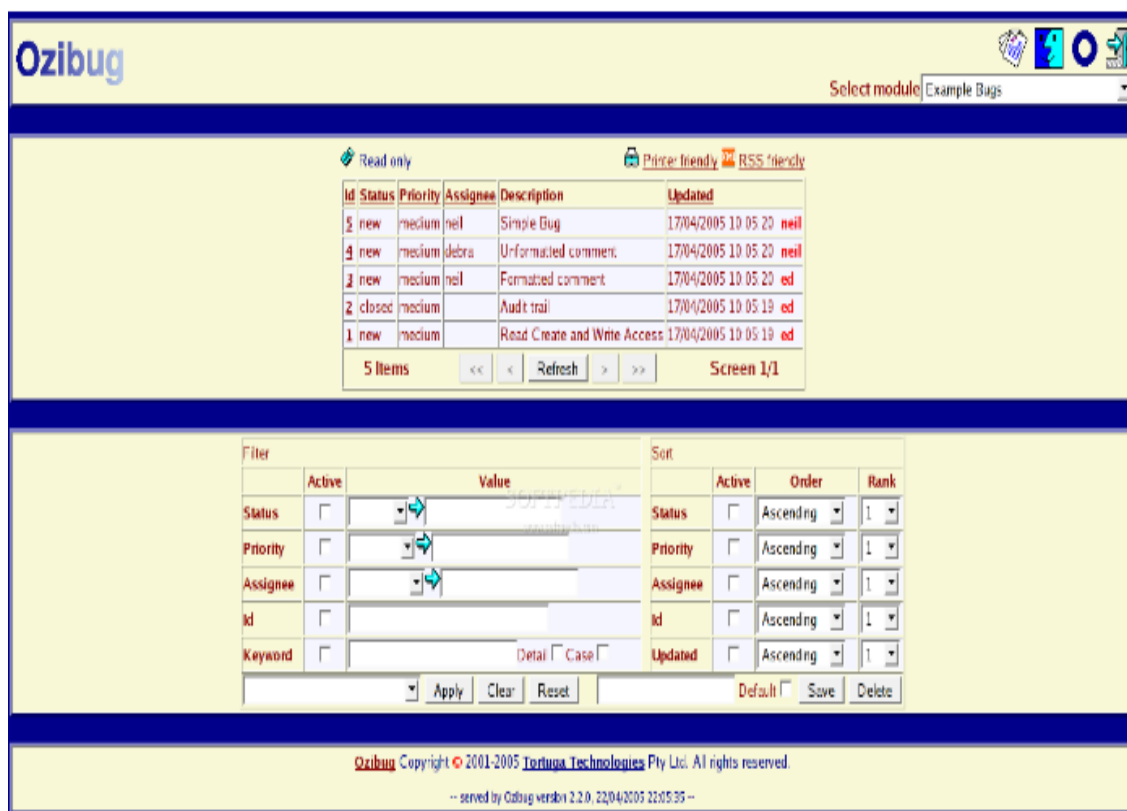


Figure 9. Ticket page view of the Ozibug tracking tool viewed on 10/04/2011

<http://linux.softpedia.com/progScreenshots/Ozibug-Screenshot-946.html>

### BugImpact

BugImpact is a reliable, secure, convenient bug tracking tool developed for bug-tracking and project tracking. It is based on Microsoft .Net technology. Its simple user interface makes it easier and convenient to use. Figure 10 shows the creation of users



with access control. Some of the important features of bug tracking tool are as follows:  
[15]

- It has unlimited data storage capacity.
- It is completely web-based and can be accessed from any standard web browser.
- Necessary files and folders can be attached with related issues.
- Email notification notifies users on any sorts of change on the project.
- It provides colour facilities for the bugs and issues according to their priority level.
- It provides authentication and role-based privilege level management.

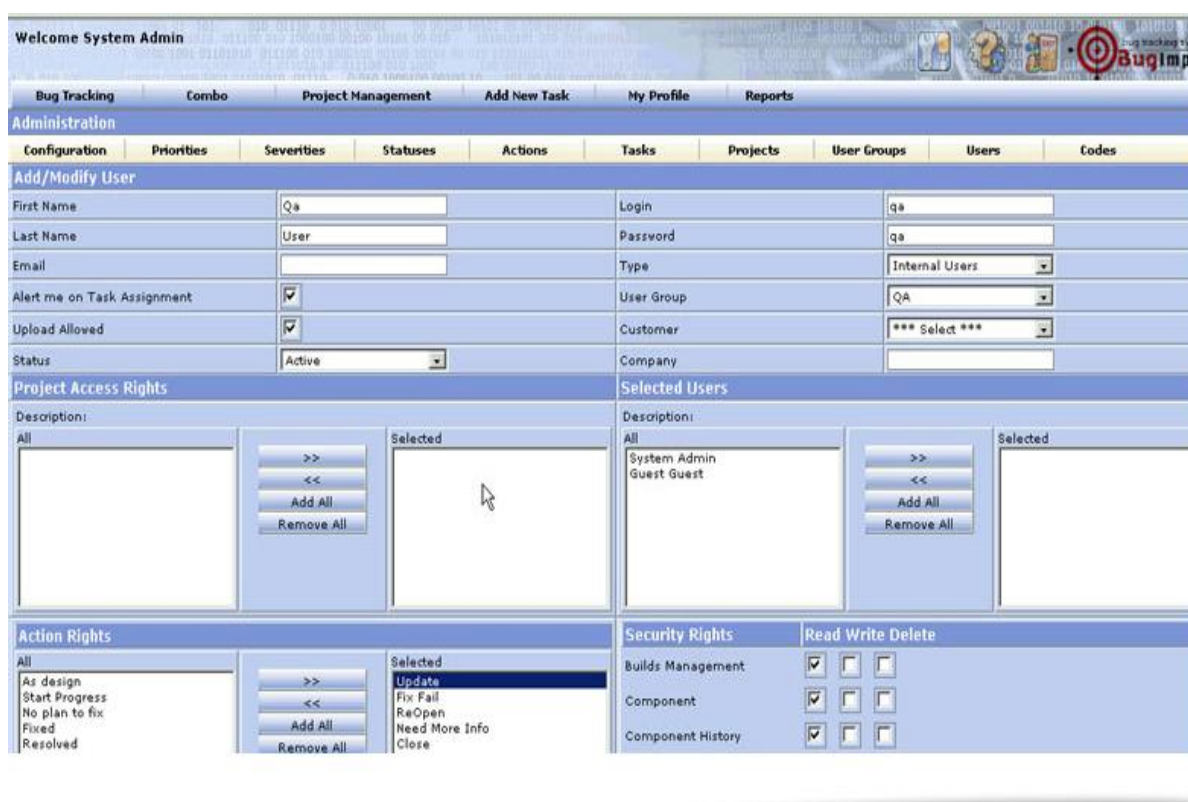


Figure 10. User access control view of the BugImpact tracking tool viewed on 10/04/2011

<http://www.bugimpact.com>

### Assembla Tickets

Assembla Tickets is one of the bug tracking and ticketing tool developed by the Assembla community. Assembla is used for task management system for tracking bugs, issues and other new features according to client requests. Figure 11 shows the

milestone creation page for Assembla Tickets. Some of the features of Assembla ticketing system are as follows: [16]

- easy to use
- flexible configuration for developing clear graph for the ongoing process on the project
- links with codes and requirements
- Milestones created are shown on integrated calendars.

The screenshot shows the Assembla Tickets interface. At the top, there's a navigation bar with 'assembla' logo, 'My Start Page', 'Logout', 'Search this space', and 'Help'. Below that, the page title is 'Breakout Software Development' with 'Pro/Private | Owner' on the right. A secondary navigation bar includes 'Wiki', 'Messages', 'Team', 'Files', 'Stream', 'Milestones', 'Tickets' (highlighted), 'StandUp', 'Source/Git', 'Dashboard', and 'more...'. A third navigation bar includes 'Tickets', 'Filters', 'Search', 'Burndown', 'Metrics', 'Batch Update', 'Agile Planner', and 'Settings'. Below the navigation, there are buttons for 'New Ticket' and 'Post a ticket via email'. A filter dropdown is set to 'Active by Milestone' with a search input and a 'Go' button. Below the filter, there are export options: 'Export: RSS Feed', 'CSV', 'XML', and 'Print'. The main content area shows two tables:

No Milestone			
#	Summary	Assigned To	Status
☆ 6561	[PI] Allow only admin to create spaces	Artiom D	Accepted

3.8 Release			
#	Summary	Assigned To	Status
☆ 6505	Fix sweep_spec failing test	Sergio Romano	Accepted
☆ 6569	[Master] Tweaks to Notification System	Vladimir Zdorovenko	New
☆ 4962	Cardwall	Joachim Larsen	Accepted
☆ 6156	[Master] Portfolio Manager		New
☆ 6263	Fixme: increase in New Relic		New

Figure 11. Milestone creation page view of Assembla viewed on 10/04/2011  
<http://www.assembla.com/features/bug-tracking>

### 3.3. Comparison of bug tracking systems

The following chapters include the comparison of bug tracking system on the basis of different methods from different aspects like in general, features, authentication and others. Tracking tools are compared in two different ways. The first comparison compares the general characteristics of the tracking tools available which includes

developer of system, license type, language used for development of project, type of database system supported and used by the system and the first released date of the product. The second comparison compares availability of different features on the product. It includes what different features are available on different tracking systems. [3] [17]

### General

Table 1 compares different tracking tools on the basis of general characteristics. The comparison is based on five different characteristics. System represents the tracking tool used for comparison. Creator defines the developer that may be company or personnel. License indicates the type of project under which it is developed. It can be GPL or MPL or free open source project. Implementation language means the programming language used for development of product. Database refers to the databases system or back end that is supported by the product and launch date shows the first release of the product.

Table 1. Comparison of tracking tools on the basis of general characteristics

<b>System</b>	<b>Creator</b>	<b>License</b>	<b>Implementati on Language</b>	<b>Database support</b>	<b>Launc h Date</b>
TracProjec t	Edgewall software	New BSD	Python	SQLite,MySQL, PostgreSQL	2006
Redmine	Jean-Philippe Lang	GPLv2	Ruby on Rauls	SQLite,MySQL, PostgreSQL	2006
Mantis	open-source	GPLv2	PHP	SQLite,MySQL, PostgreSQL	2000
JIRA	Atlassian	proprietary, free for non- commercial use	Java	MySQL,PostgreSQ L ,Oracle, SQL server	2003
Web Bug Tracker	Corey Trager	GPL	ASP.NET, C#	SQL Server, SQL server express	2002
Bugzilla	Mozilla Foundation	MPL	Perl	MySQL, Orcale, PostgreSQL	2002
FastBug Track	ALCEA technologies inc	-	Java	MySQL, Oracle	2004
Ozibug	Tortuga Technologie s	-	Java	MySQL, Oracle, Microsoft SQL Server	2001
BugImpac t	Anva International	-	.NET	MySQL	2001
Assembla Tickets	Assembla	Proprietary, available for open source projects.	Ruby	MySQL	2008

## Features

Table 2 compares the features available on different tracking tables. It shows if the features are available on the system or not. The feature table is compared using six different comparing aspects. Systems used indicate the available tracking tool. Integrated documents define availability of wiki pages, reports for the easy use of system. Test planning defines indicates the ticketing plans available on the system. Customizable work flow deals with the defining work timeline on the system. Custom field relates to milestone creation. Input interface indicates the type of accessing method to the system. Multiple project supports defines if the available project supports the multiple project within single instance.

Table 2. Comparison of available features on different tracking tool

<b>System</b>	<b>Integrated Documentation</b>	<b>Test-planning</b>	<b>Customizable Workflow</b>	<b>Custom Field</b>	<b>Input Interface</b>	<b>Multiple Projects Support</b>
TracProject	Integrated wiki	yes	yes	yes	web	yes
Redmine	Integrated wiki	yes	yes	yes	web, email	yes
Mantis	Integrated wiki	no	yes	yes	web	yes
JIRA	yes	yes	yes	yes	web,email, cli, GUI	yes
Web Bug Tracker	integrated charts and reports	no	yes	yes	web, email	yes
Bugzilla	integrated reporting	yes	yes	yes	web,email, cli, GUI	yes
FastBug Track	no	yes	yes	yes	web	yes
Ozibug	yes	no	yes	yes	web	yes
BugImpact	yes	yes	yes	yes	web	yes
Assembla Tickets	yes	yes	yes	yes	web	yes

#### **4. Tracking the PARTERRE Project**

The PARTERRE project is an EU project funded under ‘The Information and Communication Technologies Policy Support Programme.’ The total cost of the project is € 2.33 million out of which EU has contributed €1.17 million. The execution period of the program is from 01/09/2010-31/08/2012. It is ‘Information and Communication Technologies Policy Support Programme PB Pilot type B’ contracting project. Regione Toscana, Italy is the main coordinator of the project including seven other participating partners. Turku University of Applied Sciences is one of the partner organizations of the project and is working on the pre-pilot and pilot phase testing of the program. The member partners of the project are as follows:

- Regione Toscana, Italy
- Community Council of Voroklini, Cyprus
- Turun Ammattikorkeakoulu, Finland
- TUTECH Innovation GMBH, Germany
- Freie Und Hansestadt Hamburg, Germany
- Regione Sicilia, Italy
- Avventura Urbana SRL, Italy
- University of Ulster, United Kingdom

PARTERRE deals with the development of Electronic Participation Tools for Spatial Planning and Territorial Development. The tools can be used for electronic participation platforms for spatial and strategic planning in process. EU states the project as “Among the lessons learnt from the eParticipation Preparatory Actions, a particularly relevant one refers to the long-term sustainability of the undergone trials: permanent adoption of electronic tools for civic participation by public sector organizations is more likely to occur when there is a convincing business model showing financial savings or at least organizational and political advantages in a

clear way. A necessary precondition for this business model to “square” is that (offline) participation has to be formally integrated in the legal or regulatory framework the targeted organization belongs to. This is the case of environmental assessment (at strategic level) and spatial planning (at operational level), provided an EU-wide framework through the ESDP first and now the Territorial Agenda. EU Directives and/or National legislations exist to establish participation as a mandatory requirement for a great deal of policy processes and administrative procedures – mostly involving Regional and City Councils in Europe. With this lesson in mind, the PARTERRE consortium intends to validate the market perspective of a pan-European service based on two electronic tools that have been successfully trialled on previous Preparatory Actions: the Electronic Town Meeting for citizen’s involvement in agenda setting and the DEMOS-Plan application for stakeholder’s online consultation in spatial planning. Six pilots in five EU countries will deploy these tools to enhance direct participation of citizens, stakeholders and civil society in the decision-making processes regarding spatial planning and environmental assessment, both at the local and regional level where specific decisions are taken and applied and at the national and EU level where the policy frameworks for Europe’s territorial development are defined. The business goal is to provide a complete “turnkey” solution (both technologically and methodologically assessed) to European public authorities.” [1] [18]

PARTERRE mainly is focused on development of two different ICT solutions for validating and demonstrating effect of participatory methods and tools for spatial and strategic planning at European Level. The two different ICT solutions are as follows:

The Electronic Town Meeting:

The electronic town meeting is one of the ICT solutions developed for spatial and strategic planning developed by Avventura Urban Srl. It is a method that provides an opportunity for integrating small group discussion from different locations combining different electronic communication methods and toolset. [18]

The DEMOS:



The Demos, developed by TuTech Innovation GmbH, is a discussion platform used for developing Preparatory Action on e-Participation. Fig 12 shows a web interface view of Demos plan available in the German language. [18]

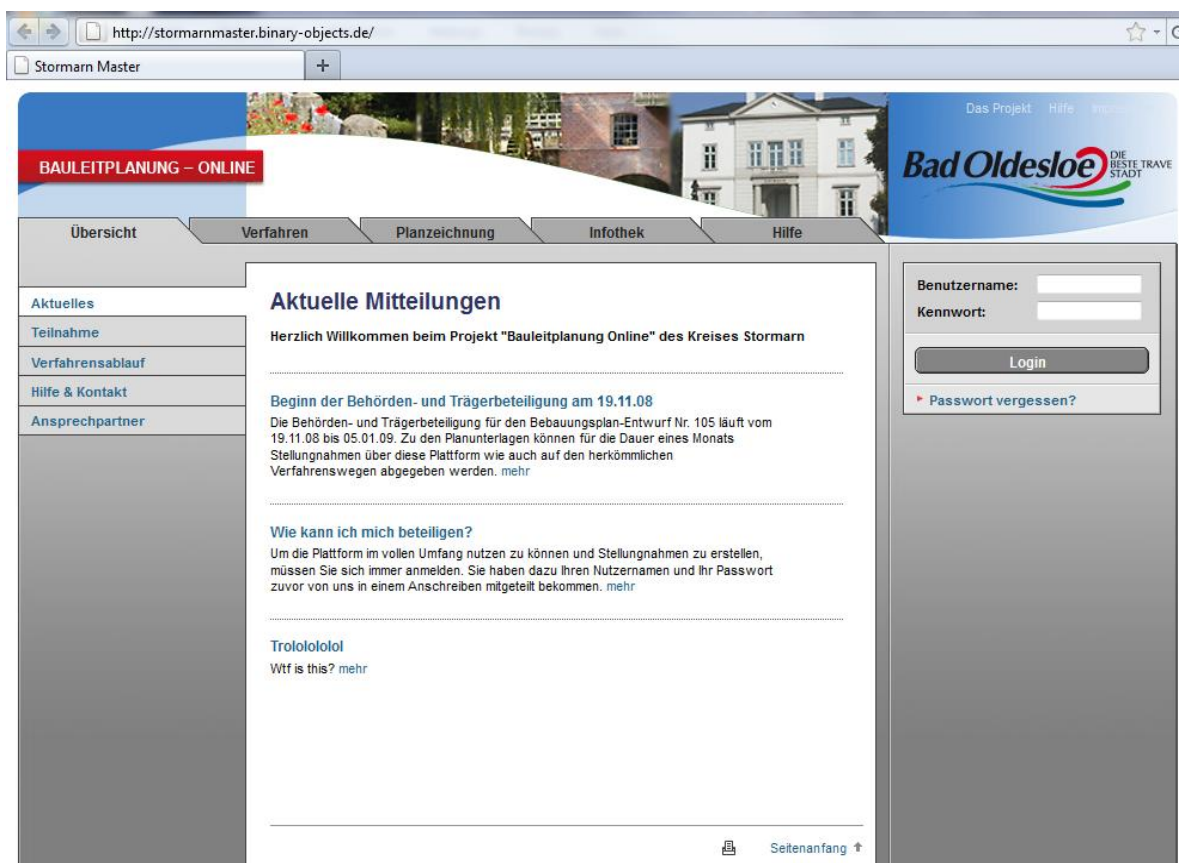


Fig [12]: Demos Plan view viewed on 18/04/2011

<http://stormarnmaster.binary-objects.de/>

Turun Ammattikorkeakoulu, a PARTERRE partner, works on pre-pilot phase of the project. During this phase, the two program Demos and eTM were tested from the aspects of software testing in the Living Lab of Turun Ammattikorkeakoulu. The phase has life cycle of end of August 2011 hence all the activities of project should be carried and tracked on time basis. Similarly, it is vital to keep the records of the bugs and error that are found on eTM and Demos during the testing phase of the project. It is necessary to develop a server with project and bug tracking tool for the proper management and tracking of the pre-pilot phase of PARTERRE project in Turun Ammattikorkeakoulu.

TracProject, was chosen as the tracking tool and installed for tracking subproject of the PARTERRE project in Turun Ammattikorkeakolu. TracProject fulfils the entire requirements necessary for the project and hence was chosen for the whole project. TracProject provides all the features like project tracking, bug tracking, ticket creation, modification and reporting along with milestones necessary for the project. The requirement, installation method and implementation mode for TracProject is discussed briefly in Chapter 5.

## **5. Installation and implementation of TracProject**

### 5.1. Installation of TracProject

TracProject works on a single standalone server and it can be installed on a web server. Many testers testing the Demos and eTM can access the server from several locations at same time. TracProject can be installed directly from the Ubuntu source package. It also can be installed from Edgewall webpage at <http://trac.edgewall.org/wiki/TracDownload> which contains HTTP and FTP source files for TracProject which can be download and installed for the server. The installation procedure and commands are attached on Appendix [A.1]. [5]

TracProject requires different pre-requirements for its installation. It includes operating server, web server and its different dependencies. Pre-requirements and their importance for TracProject and the installation method are discussed later in this chapter. The installation commands of all requisites are attached in Appendix [A.1].

#### Operating system

TracProject works on both UNIX and Microsoft-based operating system. The UNIX-based operating system is open source meaning that it is available for free of cost whereas the Microsoft-based system is priced. Microsoft-based products creates the issue of compatibility with other open source products. For example: Microsoft has its IIS server as a default web server which uses port number 80. When apache server is installed on Microsoft server, it creates conflict in port number 80 between the Apache server and IIS server. Because of the issue of pricing and compatibility, the LTS server 10.04 was chosen as the operating system for the server. Linux, being open source, freely available, and easily manageable, was installed for the server. TracProject supports the Ubuntu LTS server and works fine with it.

#### Database System:

TracProject needs a database system for recording data. TracProject supports the following database systems: MySQL, PostgreSQL and SQLite. SQLite is the default database system used by TracProject. Other database systems need configuration and root level access to work properly with a database system. MySQL was not fully

supported by the system. TracProject could not get enough root level access for the MySQL database system. SQLite was installed for the server because of its default connection to the TracProject system and easier implementation with the server.

#### Python:

TracProject mandatorily requires Python software for its installation. TracProject is developed in the Python programming language and hence Python is the preliminary requisite of TracProject. Python integrates the system effectively and quickly and creates the project environment for TracProject. It has an open source license developed by Python Software Foundation and works with Linux Ubuntu LTS 10.04 in an effective manner. TracProject only supports Python versions equal or higher than 2.4 and lower than 3.0. Other all versions are not supported or dropped by TracProject. [19]

#### Setup tools

TracProject requires setup tools for its proper installation and working on the server. Setuptools enhance Python distribution utilities building a Python package with its dependencies easily. TracProject requires setup\_tools for easy installation of Python packages and its dependencies. [20]

#### Genshi

TracProject needs Genshi as its mandatory dependency. Genshi, as a Python library, contains Python-integrated components necessary for parsing, generating and processing HTML, XML for producing web output. Figure 13 is a flow diagram describing the working procedure of the Genshi package. [21]

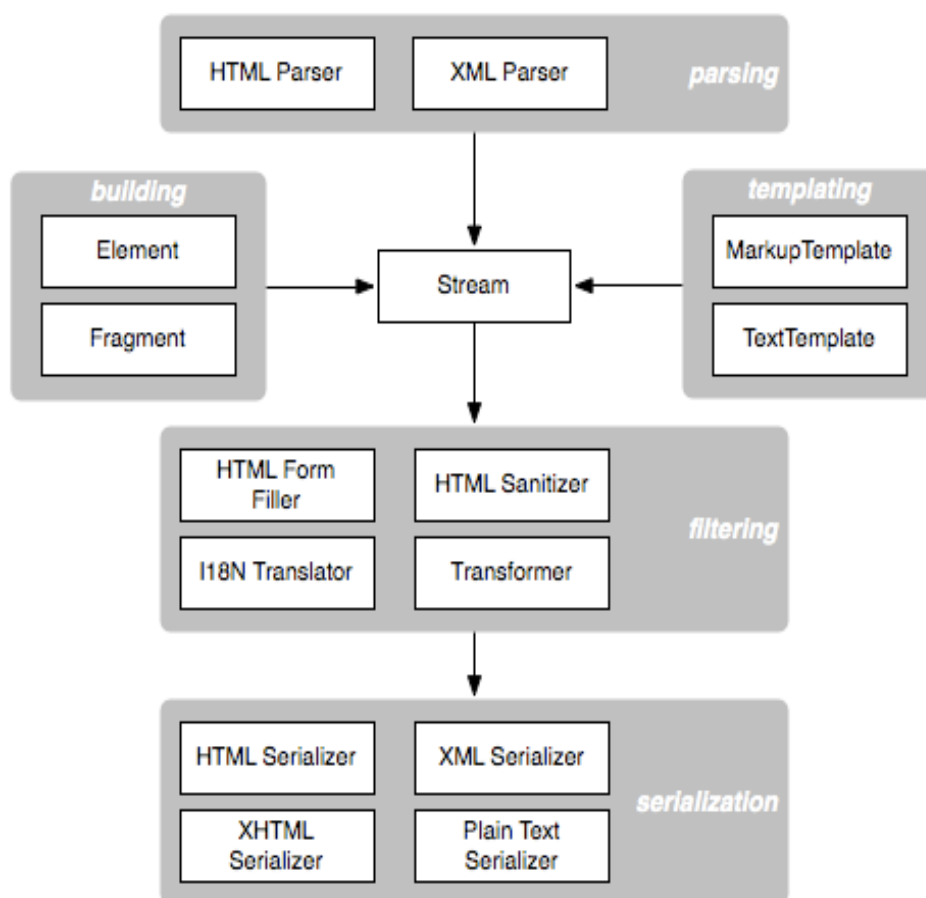


Figure 13. Flow diagram showing working procedure of Genshi

### Web Server

The Trac server for this project needs to be accessed from different locations at the same time. The standalone server for Trac cannot satisfy the necessity of accessing server from different locations. Thus, it has to be a web server. For the better running of TracProject on the server, the Apache web server was chosen and configured. Trac ran on different environments of the Apache server. The different configuration environment is as follows: [5] [22] [23] [24] [25]

#### ➤ mod\_wsgi

Mod\_wsgi, Apache integration runs WSGI-compatible python applications. The environment developed in the C programming language runs applications on top of the web server. It performs better and is recommended for use by the TracProject

developer community. Delay time response to the server is the drawback for this module.

➤ mod\_python

Mod\_python, another Apache-integrated component, runs Trac response faster than other modules of the Apache server. It supports the 'tracd' features which may or may not be supported by other modules. Because of some fixed issues on the server, mod\_python is no more supported by the TracProject community. An administrator should handle the module if used on his own.

➤ FastCGI

FastCGI, an integrated module of the Apache server runs Trac like other modules. It responds more quickly to the server than the external CGI even though CGI module is supported and used widely by different web servers. FastCGI runs on different levels of permission. Incompatibility with the Windows server is a disadvantage for FastCGI.

Even though the TracProject community recommends not to use mod\_python module, the author decided to use mod\_python for the Trac installation on the server because of the following reasons.

- Trac responds faster to the server than other modules of the Apache server.
- The mod\_python module can be configured without changing the Apache configuration file.
- It is easier to create projects.
- It supports the tracd feature of TracProject.

### Subversion

Subversion controls the version of the project. TracProject runs without subversion. Subversion was not used during this project. [25]

## 5.2. Implementation of TracProject tool

On successful installation of Trac and all its mandatory dependencies, TracProject can be used. Projects related to Trac should be created on the server. The created server needs to be available on the web server so that it can be accessed by the users from the external machines. TracProject needs different administrators for its administration. Normal users can create and view tickets and comment on them but they are not authenticated to change milestones on the system.

### Local Administrator

A local administrator administers the system for the creation of projects on the local server. The projects created on the local machine are then linked to the web server making the projects available on the web. The local administrator configures the Apache server with suitable modules along with user authentication and privileges. The local administrator determines the web administrator. The backup and maintenance of projects and web are maintained by the local administrator. Appendix A.2 includes a brief procedure and instructions for creation of projects and the Apache server configuration. It also shows the process for creating a web administrator along with backup and maintenance of a project.

### Web Administrator

The local administrator creates a web administrator on local machine. The web administrator has privilege status for the creation of milestones for the project. The web administrator controls general features like basic settings, logging, permissions, plugins etc., ticket system and version control of the related project. General administration includes basic settings, logging, permissions and plug-in services. Basic settings relate to naming of project and URL defined for the project. The project type description, language and time zone including date format can be maintained on basic settings. Logging settings keep the log level of the project. Permission settings maintain view, authentication level for the project. The web administrator can administer project components, milestones, priorities, resolutions, ticket type, and versions under ticket system. Version control sets repositories if sub-version is used on the project. Screenshots for web administration are attached on Appendix A.3.

### 5.3. Benefits and Drawbacks of TracProject Installation

TracProject includes all the necessary requirements for tracking subproject of the PARTERRE project. Even though it is not a perfect bug tracking tool, it is used for the project. The merits and demerits of the tool are listed below.

#### Benefits

- Easy installation, project development and management.
- Easy ticketing, milestone creation.
- Easy web administration.

#### Drawbacks

- Lack of graphical representation of projects tracking.
- Logging out functionality on the trac web portal does not function.



## 6. Securing Server and Information

The PARTERRE Trac server needs server and information security. Two different security options are used for the security. [23] [24] [25]

### Physical security

The PARTERRE Trac server is physically secured. Restricted access of users to the server machine makes it more secure. Only the local administrator has the requested access for the server for higher security level.

### Logical and Data security

Data and information available on the server is also highly secured. The following measures are used for data security on the server.

- SSH access only to administrator secures unnecessary access to the server.
- Projects are secured by being accessed by authenticated users. Authenticated users and their passwords are hashed using SHA technology and stored securely within the server. Appendix A.2 includes the password file creation methodology.
- Projects are located on different places and accessed via the Apache server which provides attackers with fewer options of attacking the stored project files and folder on the server.
- Regular backup of the server is maintained for the data security.
- All the ports except port 80, 443 and 22 are denied to access to the server.
- Regular updated server and firewall makes the server more secure.

## 7. Summary

This thesis is focused on the design of server and installing tracking tool in it. The author successfully designed the server and installed the defined tracking tool, TracProject, for the project. The aim of this thesis was achieved by the author. The author also successfully used the tracking tool and recommends that the chosen tracking tool can be used easily on any web server. The server is deployed for the PARTERRE project for the testing of the Demos and eTM program.

This thesis provides information about developing a server installed with tracking tools necessary for the project and bug tracking in the software industry. It develops basic knowledge about project methodology and Linux server basic administration with TracProject installed. The readers of this thesis acquire knowledge about project, project phases in Chapter 1. In Chapter 3, they build knowledge about bugs, remarkable errors created due to bugs on system and different bug tracking tools available in the world.

Chapter 4 introduces the PARTERRE project, its objectives and its partner organisations. It also includes the importance of bug and project tracking tool for partner organisation Turun Ammattikorkeakoulu during the testing phase of the PARTERRE program. This chapter also mentions the best selected tool for the project. Readers can select their tool according to the needs of their project.

Readers can develop ideas about necessary dependencies of the selected tools, their installation procedure on chapter five. Chapter 5 also includes the errors/bugs present on bug tracking itself. Readers can learn about all the project generating codes and their working method from the appendix mentioned in this chapter. In conclusion readers can develop an idea about developing a server, installing project and bug tracking tools and their components along with securing the server and information.

## **8. Recommendation**

This thesis developed a server capable of tracking the necessary software development and testing process of the PARTERRE project. Still, some more work can be done for project creation and server security. Following work can be improved on the server.

- Studying the features of the tools for creating multiple projects within single instance. It eases local administrator's workload of creating multiple URLs for different projects. Similarly, users need not remember multiple passwords and URLs during their working hours.
- For higher security Secure Socket Layer (SSL) encryption could be introduced for the URL.
- Using Encryption technology for data transmission so that data transmitted is not affected by middle-man-attacks.

## 9. References

- [1] PARTERRE Project, [www document], Available at [http://ec.europa.eu/information\\_society/apps/projects/factsheet/index.cfm?project\\_ref=256244](http://ec.europa.eu/information_society/apps/projects/factsheet/index.cfm?project_ref=256244) , Accessed on 2<sup>nd</sup> April 2011
- [2] Nicholas J., *Project Management for business and Technology: Principles and Practice*, Prentice Hall, 2<sup>nd</sup> edition, 1<sup>st</sup> November, 2000
- [3] Bugs, [www document], <http://www.aptest.com/bugtrack.html> Accessed on 3<sup>rd</sup> April 2011
- [4] Bug History, [www document], <http://www.wired.com/software/coolapps/news/2005/11/69355> , Accessed on 5<sup>th</sup> April 2011
- [5] TracProject, [www document]. Available at <http://trac.edgewall.org/> Accessed on 10<sup>th</sup> April, 2011
- [6] Redmine, [www document], Available at <http://www.redmine.org/> . Accessed on 10<sup>th</sup> April, 2011
- [7] Mantis, [www document], Available at <http://www.mantisbt.org/> . Accessed on 11<sup>th</sup> April, 2011
- [8] JIRA, [www document], Available at <http://www.atlassian.com/software/jira/> . Accessed on 10<sup>th</sup> April, 2011
- [9] Web Bug Tracker, [www document], Available at <http://www.vaxtech.com/> . Accessed on 11<sup>th</sup> April, 2011
- [10] Bugzilla, [www document], Available at <http://www.bugzilla.org/>. Accessed on 10<sup>th</sup> April, 2011
- [11] Fast Bug Track [www document], Available at <http://www.fastbugtrack.com/> Accessed on 10<sup>th</sup> April, 2011

- [12] Agility, [www document], Available at <http://www.agileedge.com/> Accessed on 10th April, 2011
- [13] Woodpecker IT, [www document], Available at <http://www.woodpecker-it.com/> . Accessed on 10<sup>th</sup> April, 2011
- [14] Ozibug, [www document], Available at <http://www.tortuga.com.au> . Accessed on 10<sup>th</sup> April, 2011
- [15] BugImpact, [www document], Available at <http://www.bugimpact.com> . Accessed on 10<sup>th</sup> April, 2011
- [16] Assembla Tickets, [www document], Available at <http://www.assembla.com> Accessed on 11<sup>th</sup> April, 2011
- [17] Comparison of issue-tracking systems, [www document],  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_issue-tracking\\_systems](http://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems). Accessed on 11th April, 2011
- [18] PARTERRE Project, [www document], Available at <http://PARTERRE.ning.com/> , Accessed on 18<sup>th</sup> April 2011
- [19] Python, [www document], Available at <http://www.python.org/> Accessed on 20<sup>th</sup> April, 2011.
- [20] Setup tools, [www document], Available at <http://peak.telecommunity.com/DevCenter/setuptools> Accessed on 20<sup>th</sup> April, 2011
- [21] Genshi, [www document], Available at <http://genshi.edgewall.org/wiki/Download> Accessed on 20th April, 2011
- [22] Apache, [www document], Available at <http://apache.org/> Accessed on 20<sup>th</sup> April, 2011
- [23] Thomas K. & Sicam J., *Beginning Ubuntu Linux*, Apress, 3rd edition, 26th June 2008.

[24] Hunt C, *Linux Network Servers*, SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501, 2<sup>nd</sup> edition, August 2002

[25] Adelstein T. & Lubanovic B., *Linux System Administration*, O'Reilly Media, 1st edition, March 2007

[26] Subversion, [www document], <http://subversion.apache.org/> Accessed on 20<sup>th</sup> April, 2011

## Appendix A.1 – Installation of TracProject and its dependencies.

This section describes the installation of TracProject tool and its supporting dependencies. Screenshots 1 to 7 below show the steps for installing web server and its dependencies on the server machine.

Apache server installation:

```
parterre@parterre-trac:~$ sudo apt-get install apache2
```

Mod\_python installation:

```
parterre@parterre-trac:~$ sudo apt-get install libapache2-mod-python
```

Python-setuptools installation:

```
parterre@parterre-trac:~$ sudo apt-get install python-setuptools
```

TracProject installation:

TracProject can be installed in different ways.

1) By using easy\_install mode:

```
parterre@parterre-trac:~$ sudo easy_install trac
```

2) Installing from dpkg package manager:

```
parterre@parterre-trac:~$ sudo apt-get install trac
```

3) Downloading and installing package from the developer source:

```
parterre@parterre-trac:~$ sudo easy_install http://ftp.edgewall.com/pub/trac/Trac-0.12.2.tar.gz
```

Database Installation

```
parterre@parterre-trac:~$ sudo apt-get install sqlite
```

Depending on the system, SQLite requires developer headers for removing GCC related errors on the system. The SQLite developer header is installed using the following command.

```
parterre@parterre-trac:~$ sudo apt-get install libsqlite3-dev
```

## **Appendix A.2 Developing projects, project URLs, backup, and user permissions.**

On successful installation of all the dependencies and Trac tools as mentioned above, it is necessary to develop the project for tracking system. This appendix includes the detailed and simplified procedure of developing projects, URLs and permissions along with backing up the data. The author will develop a project called **test** for this server that makes reader easy to understand the procedure. Screenshots available below show the development procedure of projects, URLs for tracking subproject of the PARTERRE project.

### Creating project

It is necessary to develop project environment directory.

```
parterre@parterre-trac:~$ sudo mkdir -p /var/www/test/test1/testtrac
```

The above created directory is used for creating trac project. The path `/var/www/test` is used to deploy the htdocs and common site files chromed from the project. The directory `testtrac` will be the main directory for creating the project environment. On creation of project environment directory, it is necessary to initiate the Trac environment by using the following command.

```
parterre@parterre-trac:~$ sudo trac-admin /var/www/test/test1/testtrac initenv
```



The above command initiates trac-environment on testtrac directory. trac-admin is the administrative command for trac tool. On successful creation of the project, the following files are created on the /var/www/test/test1/testtrac directory.

```
attachments  conf  db  htdocs  log  plugins  README  templates  VERSION
```

The above directory and files are the integrated components of the Trac tool which needs to deploy to Chrome environment in order to create web access URL. In this project the /var/www/test directory will be used for chroming the above mentioned files and directory. The following commands are used for deploying the Chrome.

```
parterre@parterre-trac:~$ sudo trac-admin /var/www/test/test1/tracetest/ deploy /var/www/test/
```

On deploying the files Chrome, it creates htdocs and cgi-bin directory which constitutes the Trac development environment.

```
parterre@parterre-trac:~$ ls /var/www/test/  
cgi-bin  htdocs  test1
```

cgi-bin folder trac.cgi scripts which later is mapped for URL development. The htdocs directory includes 'common' and 'site' for the content of developed URL. It is necessary to override common and sites from Chrome to make the URLs available universally. The project environment created should be deployed on the htdocs/common and htdocs/site files so that necessary scripts gets created for web access. The following is the procedure for creating the deploying project environment on the htdocs directory.

```
parterre@parterre-trac:~$ sudo trac-admin /var/www/test/test1/testtrac deploy /var/www/test/htdocs/common/
```

```
parterre@parterre-trac:~$ sudo trac-admin /var/www/test/test1/testtrac deploy /var/www/test/htdocs/site
```

The project environment /var/www/test/test1/testtrac created previously is chromed to var/www/test/htdocs/common as well as to /var/www/test/htdocs/site.

On successful deployment of these commands, the project is created and is ready for use. Now the Apache server should be configured and linked to the project. Mod\_python module of the Apache server is used for creation of sites (URL) in this

project. The following is the method for configuring the mod\_python-based Apache server.

```
<Location /test>
    SetHandler mod_python
    PythonInterpreter main_interpreter
    PythonHandler trac.web.modpython_frontend
    PythonOption TracEnvParentDir /var/www/test/test1
    PythonOption TracUriRoot /test
    AuthType Basic
    AuthName "Parterre Project"
    AuthUserFile /path to the password file
    Require valid-user
</Location>
```

The above configuration creates a site named test whose configuration files are located at /var/www/test/test1. SetHandler defined the type of the Apache module used for the server. PythonInterpreter wraps Python objects. PythonHandler assumes for built-in publisher. PythonOption TracEnvParentDir leads to the project environment directory. AuthType is used for authentication of users for the created website.

On creation of the website, it is necessary to enable website and restart the Apache server. The following commands enables the website. This project is developing a site called test and hence it is enabled using the following command.

```
parterre@parterre-trac:~$ sudo a2ensite test
```

The Apache server is restarted using the following command. Restart can be used to replace reload.

```
parterre@parterre-trac:~$ sudo /etc/init.d/apache2 reload
```

On restarting the Apache server, the website is available. The project could not access the database and create error if the root owner for the Apache server is not provided to

the created project. The following command is used for changing the ownership of the project.

```
parterre@parterre-trac:~$ sudo chown -R www-data\:www-data /var/www/test/
```

The above mentioned command provides root access to the website for the server. Python may create error on the webpage if it is not enabled during the webpage, thus it should be enabled. The following command is used for enabling python.

```
parterre@parterre-trac:~$ sudo a2enmod python
```

The site is then publicly available and can be accessed from the server. For the authentication of the users, it is necessary to create a password file. The path of the file should be placed on the Apache configuration file so that only specified users get access to the site developed. The author has developed the password file using the SHA hash algorithm for higher security of the passwords. The following is the procedure for creating password file.

```
parterre@parterre-trac:~$ sudo htpasswd -cs /path/to/file/location/.filename add USER
```

The above command htpasswd command creates the password file. c denotes creation and s denotes the SHA hash algorithm. /path/to/file/location is the file path. /.filename which makes the password file invisible. add USER creates user naming USER for that password file. For adding more users to the file, the following command is used. A new user named USER1 is added to the file after using this command.

```
parterre@parterre-trac:~$ sudo htpasswd -s /path/to/file/location/.filename add USER1
```

The local administrator creates the web administrator for the project. In this project, the author is creating the web administrator named Sharma. The following command should be developed for creating web administrator.

```
parterre@parterre-trac:~$ sudo trac-admin /var/www/test/test1/testtrac/ permission add sharma TRAC_ADMIN
```

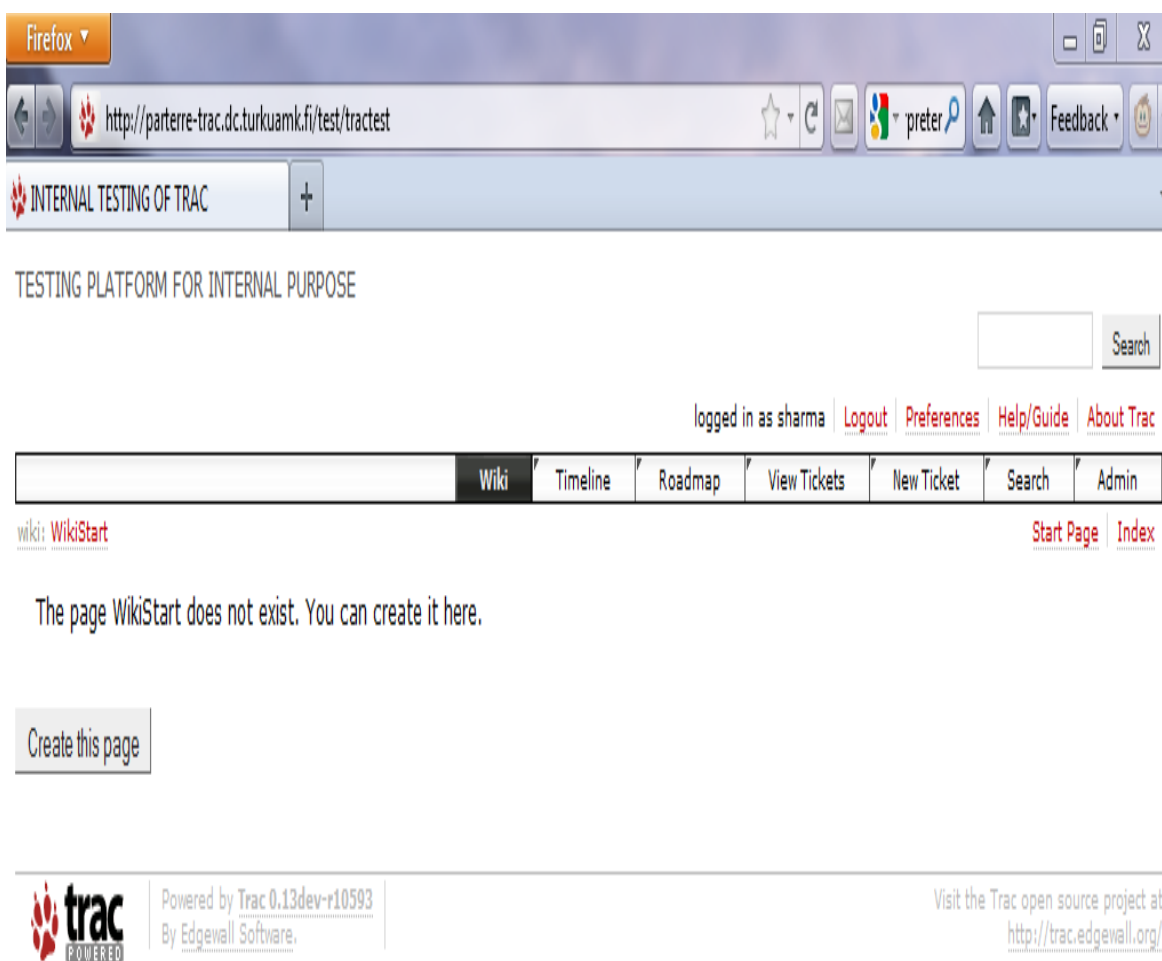
The backing up of the Trac system is made by the local administrator. The backup can be performed using the following command. It is necessary to create a separate directory for the backup of the system.

```
parterre@parterre-trac:~$ sudo mkdir -p /var/www/testbackup
```

```
parterre@parterre-trac:~$ sudo trac-admin /var/www/test/test1/tracetest/ hotcopy /var/www/testbackup/
```

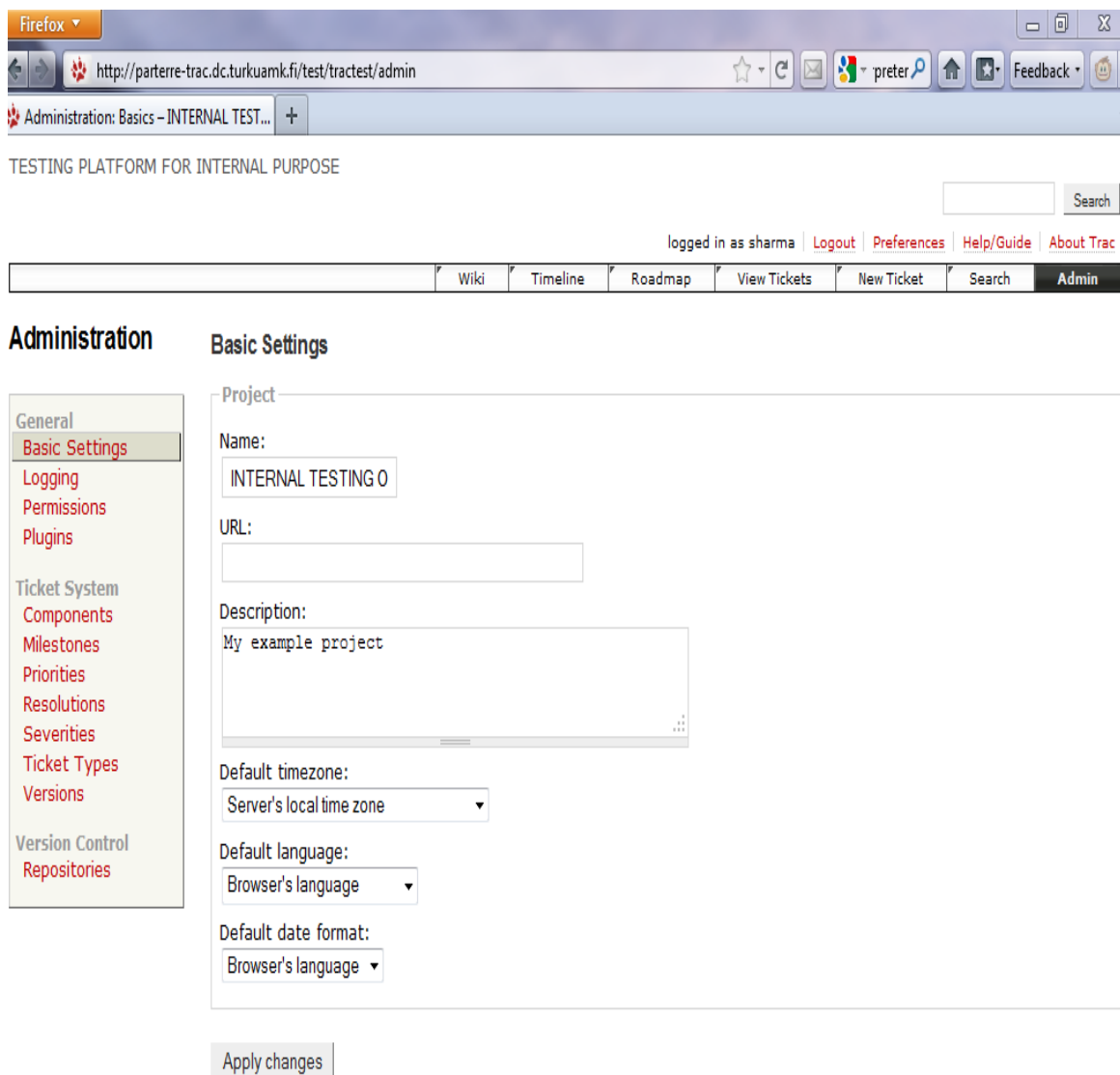
## Appendix A.3 Screenshots showing web administration

If a user has administrative right, the admin functionality appears on the side of the site specified as shown in the screenshot [1].



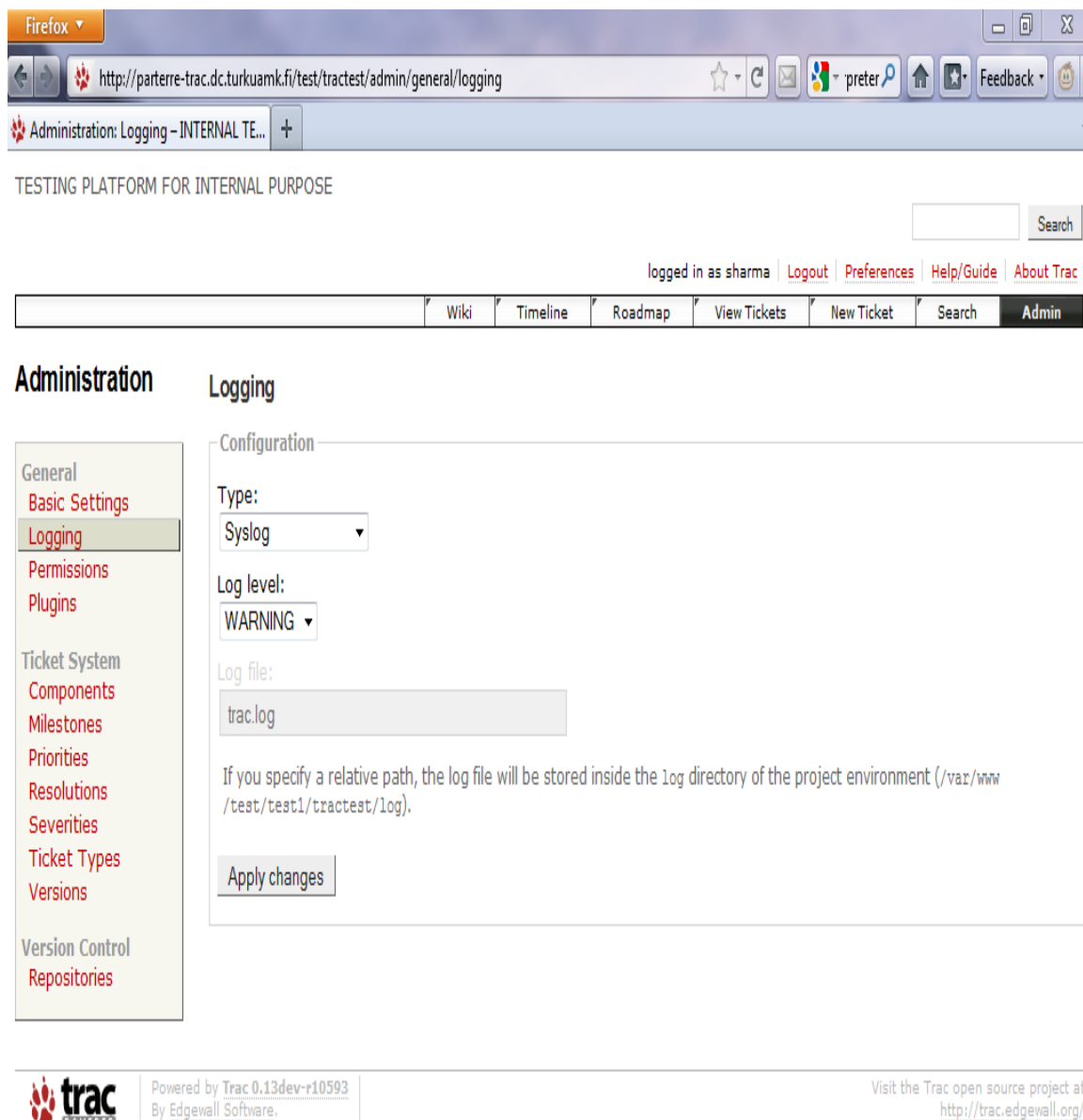
[1] Screenshot for test project with admin access.

Different administrative rights are available for admin of the site. Admin can set General settings, ticket system setting and version control on the available project. Screenshot [2] reveals the basic settings available for the web administrator. The administrator can define the name of project, the URL for the project and can provide a description of the project and set time and date format for the project on this setting.



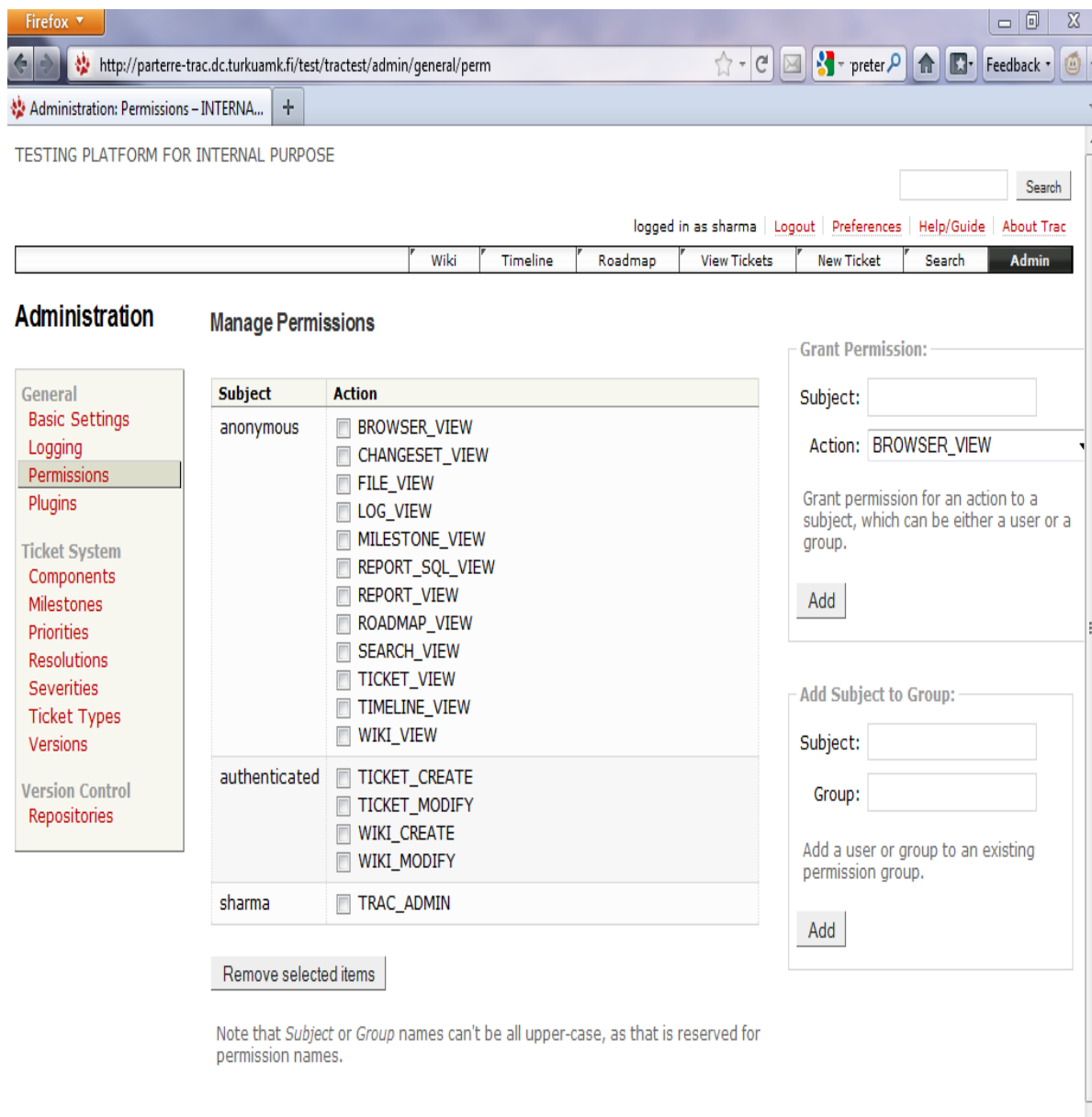
[2] Screenshot with basic settings.

Screenshot [3] shows how the logging settings admin can change for the project. The web administrator has right to choose the type of logging. Three different types of logging are available. They are Syslog, File and Console. The administrator also can set the log level. Five different kinds of logging level are available. They are Warning, Info, Critical, Debug and Error.



[3] Screenshot showing logging admin right.

Screenshot [4] shows permissions administration. The web administrator can provide various permissions to the users for the project from these settings.



[4] Screenshot showing permissions that can be set by web admin of the project.