Software Requirements Document for Team TerraUser

The Web-based User Management Project

Michelle Harr
Naoko Tsunekawa
Daniel Wallace

2 December 2001
Revision 1.0

List of tables and Figures

i) Purpose

This document is an official requirements specification of the TerraUser web-based user management software.

Our primary objective is to design and implement an efficient, secure interface to other USGS TerraWeb applications, along with a stand-alone application used to administer the user management system. The software will allow users to securely and easily access other interactive TerraWeb applications. Both the client and development team must agree with the information provided in this document. The final product should meet the complete, consistent, and correct set of requirements that are addressed in this document.

ii) Intended Audience

The goal of this document is to communicate clearly what the TerraUser software will do. This document was written to provide a clear understanding of the functions/requirements of the system. It will not be assumed that the readers if this document have any prior knowledge about this system. The audience consists of our client, the design team, and our advisor. The design team will use this document when writing test scripts.

iii) Product Scope (High Level Requirements)

The goals of the TerraUser project consist of the following:

- ⇒ Provide secure user interface to the TerraWeb applications.
- ⇒ Allow for user customizations (look and feel of web applications).
- ⇒ Allow easy update of users, priority levels and applications that users have access to.
- ⇒ Develop a system following the Software Engineering Guidelines presented in NAU's CET CSE476-CSE486 Senior Capstone Design Class.

iv) References, Standards and Resources

Team cooperation standards have been gathered in the Team Standards document and available on our website at http://www.cet.nau.edu/~dw2/terrauser/documents.html. This is where all team standards will be posted.

Some resources that our team will be using:

Resource	Address
SuSE Linux	http://www.suse.com/index_us.html
Apache	http://www.apache.org/
Java	http://java.sun.com/
MySQL	http://www.mysql.org/ http://www.mysql.com/
Section 508	http://www.section508.gov/
Tomcat	http://jakarta.apache.org/tomcat/
USGS TerraWeb	http://terraweb.wr.usgs.gov/

Table 1: Resources

1. INTRODUCTION

This project is part of Northern Arizona University (NAU) College of Engineering and Technology's (CET) Senior Capstone Design 2001-2002, with sponsorship provided by Deborah Lee Soltesz from the U.S. Geological Survey and advisement provided by Dr. Eck Doerry, Professor of Computer Science at NAU.

USGS Terrestrial Remote Sensing Group at the Flagstaff Field Center works with many kinds of remote sensing data. It is Deborah's job to support her group members work by finding ways manage, present and analyze this data through the design and implementation of web pages and web applications.

1.1 Problem & Objective

In this project we are to design and implement a system that will:

- Create a secure interface to currently existing web applications
- Centralized user management system
- Be a way for different users to have different access levels
- Set user priority levels
- Have an application to manage all the users
- And allow for basic user customizations

The objective of the project is to design and implement an efficient, secure interface to other USGS TerraWeb applications, along with a stand-alone application used to administer the user management system. The software will allow users to securely and easily access other interactive TerraWeb applications.

1.2 High level Requirements Statement

Some of the high level requirements of this project consist of using a computer running SuSE Linux, having an Apache web server running, using a MySQL data base and program using HTML and Java Technologies. We are required to provide a secure and user-friendly way to interface to web applications and manage users.

1.3 Assumptions and Dependencies

Listed below are assumptions that have been considered.

- TerraWeb applications will be using same technologies so that the interfacing will not be a problem.
- Server exists.
- Users have access to standard browser (i.e. use of links, no use of lynx).
- Direct access to server is available.
- Server has direct access to TerraWeb server (i.e. not going through firewalls).

2.1 Background/History

The U.S. Geological Survey was created by congress in 1879 as a science agency for the Department of the Interior. The USGS serves as a national science provider and fact-finding agency that provides a scientific understanding about natural resource conditions, issues, and problems. USGS scientists collect, monitor and analyzes large amounts of data about the Earth and solar system. The government and citizens in all walks of life for various reasons including to address pressing social issues use the information the USGS produces. The USGS uses the vast scientific expertise for a wide range of scientific solutions and products ranging from maps to restoration of everglades to natural hazards.

The USGS Terrestrial Remote Sensing Team at the Flagstaff Field Center consists of a four-member group: Pat Chavez *(Remote Sensing Scientist and Group Leader),* Stuart Sides *(Computer Scientist),* Deborah Lee Soltesz *(Web Mistress), and* Miguel Velasco *(Image Processing Specialist)*. They work with satellite multispectral, airborne photos, shipborne sidescan sonar, and DEM digital images. This team does such things as digital mosaicking, extraction and mapping of earth science information, geometric and radiometric calibration and corrections, and multitemporal change detection. The team has set up TerraWeb as a way for people to access this information along with a way to organize and manage some of their data.

Currently USGS TerraWeb applications have minimal security. Users are not required to log on to access these web applications. No current user management system is in place. Data management and data analysis/manipulation is the main function of many of these applications, and it is imperative that if work is going to be done using these systems that there be some sort of security standards. These TerraWeb applications are fairly new, therefore application uses and functions are evolving to the groups needs.

2.2 Business Issues

- Security is the biggest business issue that is faced in this project.
- There needs to be a user-friendly interface.
- There will be NO browser specific tags, and NO cookies.
- The project must comply with the Rehabilitation Act of 1973, Amendments of 1998, Section 508.
- Since the project is all web based it is important that the server is accessible and fast enough to handle a given number of users.
- Some of the other important business issues related to this project are how to allow access to multiple web-applications, and how to effectively manage users and the data that they have access to.

2.3 Value of a Technology Solution

This project would facilitate/support a way for users to securely and easily have access TerraWeb applications. The client would have a generic interface to all of the current and future web applications. The TerraUser solution would be a cost effective and easily modifiable solution to current and future needs. It does not have the extreme cost or confusing complexity of commercially available solutions to this problem.

2.4 Competitive Products/Research

We are not aware of competitive products that meet the specialized needs of the project.

2.5 Business Environment

Since the technologies that the project is being developed on are cutting edge, there are always going to be changes to the solution, thus we expect changes in the requirements. Since this project is designed as a interface to applications that deal with scientific data management and analysis we will use a modular design. User expectations are always high when it comes to security and user management. There is a demand for our product in many areas within user market and beyond.

3. PRODUCT SOLUTION STATEMENT

3.1 Proposed use of Technology

Product description

There will be two parts to this TerraUser interactive web application: a stand-alone part for administrators to manage users and permissions, and an invisible application that lets other applications to connect to get specific user information. There will be a variety of information that will be stored about the user including but not limited to:

- Who the user is
- What the user's personal preferences are (look and feel of application)
- What team user belongs to
- Priority level for running processes
- Applications the user has access to
- Level of access

The system will be built on a SuSE Linux server running an Apache web server. TerraUser application will have a MySQL database to store user information in.

Figure 1. Below shows a rough diagram of the TerraUser interface. Users will have to log in through a browser. They will then have access to TerraWeb applications, a user preference page, etc.

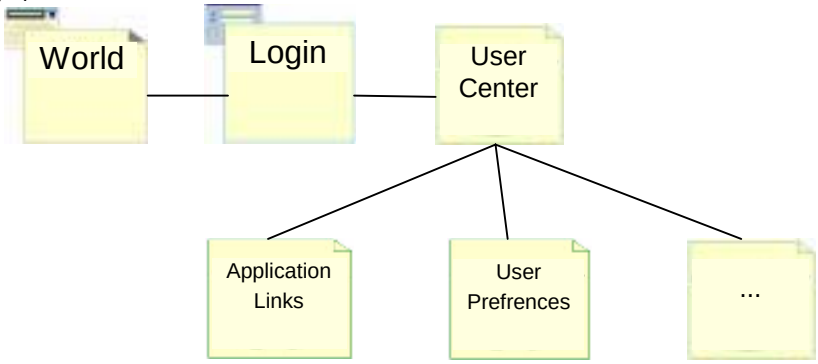

Figure 1: High-level Interface Overview

Figure 2. Below shows a rough overview of the TerraUser application. Users will use a secure socket layer to login to the TerraUser application. The TerraUser application will communicate to a MySQL database using JDBC. And the TerraUser application will be able to send requested information to the TerraWeb applications.

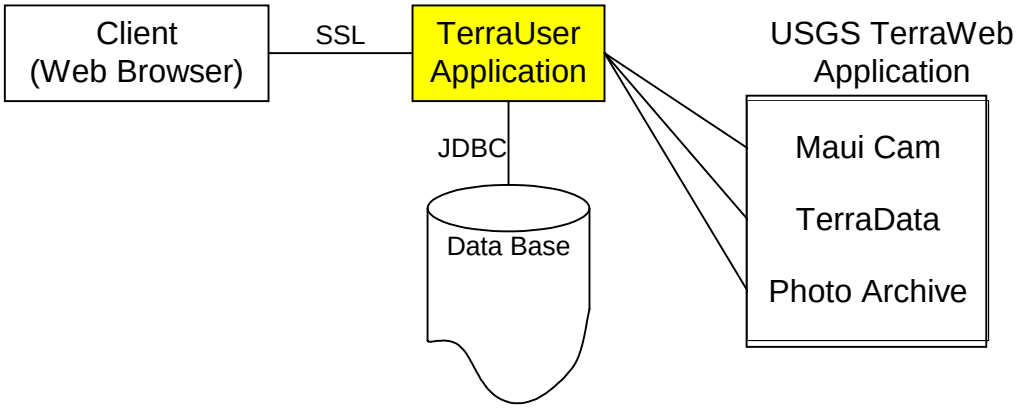

Figure 2: High-level Application overview

4. REQUIREMENTS

4.1 Goals

To provide security for TerraWeb applications
To provide flexibility for user customizations
To provide user management
To design a system that is scalable to future needs

4.2 Functional Requirements

Functional requirements consist of services the system should provide and how the system will behave.

The overall functionality must:

- Provide a secure interface for users to login to TerraWeb applications
- Provide a centralized way to manage users and their access/priority level
- Must be all web-based
- The user must supply a user name and password to be granted access to applications
- The user will specify personal preferences
- The admin will specify users, access levels, priority levels, and applications
- The application must store all this data in its database

Must provide following specific functions:

1. **User Accounts**
   - It shall provide storage of user information including but not limited to: user login name, password, priority level, access rights, group membership, and interface preferences.
   - The accounts will be administered through an administrator interface.

2. **Centralized User Login**
   - Access to the software will require use of user login name and password.
   - It shall provide users the ability to change the password at any time.
   - It shall use encryption to send data.
   - Administrators may set expiration times for passwords and will force users to change their passwords after the expiration time.

3. **Interactive Web Application for Administrators**
   - An independent application will be available exclusively to administrators.
   - It shall provide the ability to add or delete information that can be stored for users.
   - It shall provide the ability to alter information stored for any given user.
   - It shall provide the ability to add or delete users.
   - It shall provide options to set the password expiration time.
   - It shall provide monitoring of user activities and the option to enable output of activities to a log file.

4. **Interactive Web Application for Users**

   - The application available to users shall use secure login.
   - It shall provide access to the Maui Cam, TerraData, and Photo Archive applications.
   - It shall provide user interface customization stored with their user data.
   - It shall allow applications to retrieve the user's customization.

- It shall users to view and manage their system accounts in one place.
- It shall allow users to use TerraWeb applications they have been granted access to.
- It shall allow users the option to change their password.

Description and Priority

Functional Client-side requirements

There are three kinds of users:

- *Administrators:* Users who use the administration application to edit user information in the database. They enter data such as who the user is, what privileges the user has and the priority level of the user.
- *Editors:* Users belonging to a specific work group or multiple work groups who have access to all information belonging to their group. These users also have read access to information that is marked public by other groups.
- *Guests:* Users who are allowed very limited access and information to applications. These users can only search and read information that is marked as public.

Administrators

Administrators purpose is to manage users and teams and what information they can access. The following table is a list of the identified requirements for the administrator.

Requirements prioritized with 1 = Crucial, 2 = Very high, 3 = High, 4 = Average, 5 = Low, 6 = Very low. Difficulty is rated from 1 to 10 where 10 = very difficult and 1 = very easy.

Requirement	Priority	Difficulty	Risk
Add a new user	1	2	
Delete existing user	1	2	
Update user information	1	2	
Add/Update/Delete team information	1	2	
Log off all users	3	8	
Post MOTD (include MOTD in the glossary)	5	1	Requires sending message to users, brief delay, disconnecting users, and locking the system so new users can't log on.
View active users or logs	2	2	
Add information fields	1	4	
Set password reset/expire	1	5	

Table 3: Administrator Requirements

Editors

Editors are the main users of the interface and applications. These users do the actual work. The following table is a list of the identified requirements for the editor.

Requirements prioritized with 1 = Crucial, 2 = Very high, 3 = High, 4 = Average, 5 = Low, 6 = Very low.
Difficulty is rated from 1 to 10 where 10 = very difficult and 1 = very easy.

Requirement	Priority	Difficulty	Risk
Change password	1	2	
Change preference	1	4	
Access to applications	1	8	
Search option	5	8	
E-mail to administrator	2	1	

Table 4: Editor Requirements

Guest

Guests are just able to view certain information and have limited access to applications. The following table is a list of the identified requirements for the Guest.

Requirements prioritized with 1 = Crucial, 2 = Very high, 3 = High, 4 = Average, 5 = Low, 6 = Very low.
Difficulty is rated from 1 to 10 where 10 = very difficult and 1 = very easy.

Requirement	Priority	Difficulty	Risk
View application data that is marked public	1	8	
Search for public information	1	5	
Post messages to administrators or teams	5	1	

Table 5: Guest Requirements

External Interface Requirements

User Interfaces

The TerraUser software will be accessed on the Internet through a web-browser. Users must have an Internet connection and a standard browser to access the software.

Hardware Interfaces

The TerraUser software is a database driven web-application. The software will use the most current standards for implementing JSP pages accessing a MySQL database via JDBC. All HTTP transfers (data over the internet) and SQL access (data from a local database) will be handled by internal functions of the technologies being used.

Software Interfaces

The TerraUser software will act as an interface between the user and applications they have access to. The applications can be only be accessed through the software. When the user selects an application to run, the software will send required information. At any time the application is running, a request from the application can be made back to

the software to gain any new or different information needed. The request can be made from any of the applications. A single feature in the software will handle all requests. When new applications need to interface to the software, the form of the request will be used in the new application and the software will be ready for the new application. The requests for information can be requested only with valid user login.

Communications Interface

The TerraUser software will use a web-based interface. The software will generate dynamic web pages viewed via HTTP/HTTPS. The software will communicate with the users web browser using the most current HTTP/HTTPS and HTML standards. All communication from the software to the user will be prompted by user input. All communication from the user will be done through entering data and selecting items on the web pages dynamically created by the software.

Design and Implementation Constraints

Below are the constraints that have been proposed by the client as well as those which reflect project domain specifications.

General security constraint

The system will have to secure user information sent through the Internet. This will be achieved use of secure HTTPS protocol.

Technical constraints

The TerraUser software must meet the following minimal requirements:

- The System will be designed to be scalable to meet future needs of the client.
- The implementation must utilize specific technologies provided on the server.

The following is a brief summary of the technology required by the project and available on the server:

Category	Technology Used
Operating System	SuSE Linux
Web Server	Apache
Java Server	Apache Tomcat
Server Side Interfacing	Java, JDBC, JSP, JavaScript
Database	MySQL
Security	SSL

Table 2:Technical Requirements

The design must provide a completely web-based interface. All interfaces must meet with HTML 4.0 minimum standards and be in compliance with the Rehabilitation Act of 1973, Amendments of 1998, Section 508.

4.3 Non-Functional Requirements

These are constraints on the services that do not impact the functionality of the product. The system must follow design guidelines presented in the capstone design course. The system must be designed using technical constraints listed in table 2 on page 5.

Delivery

- Documents in electronic forms: PDF, HTML, or MS Word.
- Code should be delivered well commented in electronic form.

Training

- Users should require little to no training to use interface.
- Administrators should require at most an hour worth of training to use application.
- Application developers that want to use interface should require less than an hour of training to interface (Reading documentation should be enough).

Usability

- User logins should take no longer than 30 seconds.
- Access to applications should be seem less
- Errors should give a friendly error message and direct toward help.
- Basic help should be available.
- Users can only be logged in once.
- Message will be displayed when server is unavailable for logins.

5. SPECIFICATIONS

5.1 Performance Requirements

1. System development must follow the technical tools/languages as specified by USGS sponsor. See section 2.5 for technical constraints.
2. System must be user-friendly for non-technical users.
    - The system will use a simple web-based interface.
    - It shall be accessible by all standard methods of Internet access.
    - Should require little to no training to use
3. System must be maintainable.
    - The code will be well commented.
    - Documentation will be created about the software design and implementation.

4. System should be accessible from any machine.
    - The system will reside on a network visible to the Internet.
5. System must be highly secured.
    - The system will enforce user login.
    - It shall use encryption to send information across networks.
6. System must be scalable.
    - The system will use a modular design.
    - The system will use a simple and well-documented interface between modules and external software.

5.2 Business Rules

1. System must comply with the Rehabilitation Act of 1973, Amendments of 1998, Section 508
2. System must adhere to accessibility and government guidelines (System must not use cookies, etc)
3. System must not require specific browser to be run.
4. System development/integration/testing must be completed by April 26, 2002, for the Capstone Project Conference.

5.3 User Documentation

1. All the documentation must be in PDF, HTML, or Microsoft Word.
2. All deliverable documentation should be submitted electronically.
3. Documentation and reports should be formatted so they will make usable two-sided hardcopies.
4. Documentation should be readable on a computer screen.
5. Documentation on setup/implementation of system, database, and security must be provided.
6. Code must be well commented.
7. Manuals for programmers, system/database/web administrators must be provided.
8. Online, context-sensitive help must be provided.

6. RISK ANALYSIS

6.1 Software Risks

After analyzing the problem, Team TerraUser had identified the following as being potential risks:

Risk	Risk Type	Risk Description
Changes in technology	Business	Some new technology might come out that supersedes the technology on which the system is built.
Product competition	Business	Product might already exist.
Time Management	Project	Group members all have different and busy schedules, finding time to meet might be difficult.
Hardware issues	Project	Hardware, which is essential for the project, might fail or be unavailable.
Requirements change	Project and Product	Numerous changes on the requirements that were not anticipated.
Security	Project and Product	Security issues with bugs, viruses, hackers and/or changing security standards in the field.
Compatibility	Project and Product	Compatibility issues on web server, operating system, browser, and database.
Interface specification	Project and Product	Interfacing to a variety of USGS TerraWeb applications. Essential interface specifications not available on schedule.
Data management	Business	Different TerraWeb applications might need TerraUser to keep track of some unique piece of information.
Size underestimate	Project and Product	Underestimate of size and complexity of system.
Lack of experience	Project	There is a lot of technology that group members need to learn and master in order to successfully complete the project.

Table 6: Potential Risks

6.2 Risk Analysis

From looking at detailed information about the project, the process and the team we have come up with an analysis of the probability and seriousness of potential risks.

Risk	Probability	Effects
Time to develop software is underestimated.	High	Serious
Learning curve is high on required technology.	Moderate	Tolerable
The size of the software is underestimated.	High	Tolerable
The network or database or server cannot process as many transactions as expected.	Low	Catastrophic
Cannot perform open searches on database.	Low	Serious
Unexpectedly high number bugs to fix during test phase.	Moderate	Tolerable
Key team members are unavailable or ill at critical time in project.	Moderate	Serious
Changes to the requirements are made causing major design and implementation changes.	Moderate	Serious

Table 7: Risk Analysis

6.3 Risk Management Strategies

Key risks have been identified and possible strategies have been identified to manage these risks. Solutions to some potential risks follow:

Risk	Risk Analysis/Mitigation
Hardware failure on server.	Could lose valuable data and code if there was a major hardware failure on server. Make a back up plan of critical files and document the setup.
Failure to find most efficient and effective way to interface from lack of experience.	Team members will research and try to get up to date on interfacing issues.
Efficient data management can be a challenge.	Sometimes data management can get messy and confusing. One way to avoid this is to follow guidelines and standards set forth by the industry.
Consistency among browsers is related to how the web-application functions.	Decide on some sort of standards and stick to them.
Overwhelming amount of bugs found in software.	If testing of software is done consistently and intensely through out the process, major bugs can be minimized and avoided.
Underestimated development time.	Work harder on prioritizing important tasks to be completed.

Table 8: Risk Management

The clients need for web application security is currently not being met, by completing this project we will have added value to the existing system. The technology is available for our team to use to complete the task. We have designed the system to meet organizational and business objectives. The system will be both cost effective and can be operational within the scheduled time. The system will be an addition to the current system and scalable for future needs.

7.1 Economics

Our client has stated that her group does not have the money to buy costly software to solve this problem. Additionally, she doesn't need overly complex and difficult software to deal with the situation. It is a very desirable and viable situation for the client to sponsor a NAU capstone project to get some of her software needs met. Another big bonus of this situation is that the client will have all the code and be able to easily modify and customize it in the future. The client is also flexible enough to wait for the software to be completed delivered at the end of April 2002.

7.2 Technical Feasibility

The required technology exists to build the product, however, much learning and research is needed to keep up with the incredible pace of all the changes in these modern technologies. A development server is needed with all the technologies as stated in section 2.4 of this document. The development server will need to be connected to the USGS network and be accessible to all group members.

7.3 Resource Availability

The project is easily viable within the scope of the Capstone project duration. The team members have a wide variety of backgrounds and experiences that will lead to the success of the project. All the team members are dedicated to the success and completion of the project.

Team TerraUser sees this project as being a valuable contribution for USGS TerraWeb applications. The business issues make sense. No major roadblocks are predicted. We have already successfully:

- Identified project objectives
- Looked at most viable solution
- Evaluated alternatives
- Focused on possible obstacles

We agree that the project should be done, can be done, and will be done.

APPENDICES

These appendices contain supplementary materials that are used to describe project requirements.

APPENDIX A: GLOSSARY

Document Conventions

Definitions, acronyms and abbreviations for a number of terms are provided because of formal and technical nature of this document.

Acceptance Test

A formal test conducted by the end user of a system, to determine if the system works according to specifications and should be accepted.

Apache

The Apache Project is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful, and freely available source code implementation of an HTTP (Web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the server and its related documentation. These volunteers are known as the Apache Group.

Authentication

Verification of identity as a security measure. Passwords and digital signatures are forms of authentication.

CGI

Common Gateway Interface - A way of interfacing computer programs with HTTP or WWW servers, so that a server can offer interactive sites instead of just static text and images.

CGI script

Common Gateway Interface script - A program that is run on a Web server, in response to input from a browser. The CGI script is the link between the server and a program running on the system; for example, database CGI scripts are used with interactive forms.

CSS

Cascading Style Sheets - style sheet mechanism that has been specifically developed for Web page designers and users. Style sheets describe how documents are presented on screens, in print, and even in spoken voice. Style sheets allow the user to change the appearance of hundreds of Web pages by changing just one file. A style sheet is made up of rules that tell a browser how to present a document. Numerous properties may be defined for an element; each property is given a value.

HTML

Hypertext Markup Language - The language used to create World Wide Web pages, with hyperlinks and markup for text formatting (different heading styles, bold, italic, numbered lists, insertion of images, etc.).

HTTP

Hypertext Transfer Protocol - The protocol most often used to transfer information from World Wide Web servers to browsers, which is why Web addresses begin with http://, also called Hypertext Transport Protocol. It conventionally uses port 80.

Java

A simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, multithreaded, dynamic, buzzword-compliant, general-purpose programming language developed by Sun Microsystems in 1995(?). Java supports programming for the Internet in the form of platform-independent Java "applets".

Interface

A shared boundary where two or more systems meet; or the means by which communication is achieved at this boundary. An interface can be between hardware and hardware (such as sockets and plugs, or electrical signals), hardware and software, software and software, human and computer (such as a mouse or keyboard and display screen).

JavaScript

A cross-platform WWW scripting language from Netscape Communications, very popular because it is simple and easy to learn. It can be included in an HTML file by using the tag <script language="JavaScript">.

JSP

JavaServer Pages - A freely available specification for extending the Java Servlet API to generate dynamic web pages on a web server. Industry leaders wrote the JSP specification as part of the Java development program.

JDBC

Java Database Connectivity Pages – Part of the Java Development Kit that defines an application programming interface for Java for standard SQL access to databases from java programs.

Linux

An Open Source implementation of UNIX created by Linus Torvalds, which runs on many different hardware platforms including Intel, Sparc, PowerPC, and Alpha Processors. Hundreds of application programs have been written for Linux, some of these by the GNU project. Linux and Linux tools can be downloaded via the Internet or BBS for free, or purchased as part of a distribution on a CD-ROM.

MOTD

Message of the Day.

MySQL

MySQL is a true multi-user, multi-threaded SQL (Structured Query Language) database server. SQL is the most popular database language in the world. MySQL is a client/server implementation that consists of a server daemon mysqld and many different client programs and libraries. The main goals of MySQL are speed, robustness and ease of use.

PDF

The file extension for a Portable Document Format file.

Perl

Perl is a general-purpose programming language invented in 1987 by Larry Wall. With over one million users worldwide, it has become the language of choice for World Wide Web development, text processing, Internet services, mail filtering, graphical programming, systems administration, and every other task requiring portable and easily developed solutions.

RCS

Revision Control System - A version control system that automates the storing, retrieval, logging, identification, and merging of revisions. RCS is useful for text that is revised frequently, for example programs, documentation, graphics, papers, and form letters.

SCCS

Source Code Control System - A popular code management system for Unix systems.

SSL

Secure Sockets Layer - A protocol from Netscape Communications Corporation, which is designed to provide secure communications on the Internet.

TerraWeb

Title of website and web applications that have been developed by Deborah L. Soltesz. These applications and web pages are designed to support a group of USGS scientists and computer scientists that are working on terrestrial remote sensing (http://terraweb.wr.usgs.gov/).

USGS

United States Geological Survey.

User

An individual who uses a computer, program, network, or related service for work or entertainment; usually there is a distinction between a user and a programmer or other person who works with the computer on an expert or technical level.

Definitions from: http://www.computeruser.com/resources/dictionary/