Verso Project

Application Report

Tero Hänninen
Juho Nieminen
Marko Peltola
Heikki Salo

Version 0.0.5
Public
29.4.2010

University of Jyväskylä

Department of Mathematical Information Technology

Jyväskylä

Authors:

Tero Hänninen	tejohann@jyu.fi	0400-240468
Juho Nieminen	juho.nieminen@jyu.fi	050-3831825
Marko Peltola	marko.peltola@jyu.fi	041-4498622
Heikki Salo	heikki.ao.salo@iki.fi	050-3397894

Customers:

Tapani Tarvainen	tt@it.jyu.fi	014-2602752
Ville Tirronen	ville.e.t.tirronen@jyu.fi	014-2604987
Tero Tuovinen	Tero.Tuovinen@jyu.fi	050-4413685

Instructors:

Jukka-Pekka Santanen	santanen@mit.jyu.fi	014-2602766
Antti-Juhani Kaijanaho	antti-juhani.kaijanaho@jyu.fi	014-2602756

Contact information:

Email lists verso@korppi.jyu.fi and konkari_opetus@korppi.jyu.fi.

Email archives https://korppi.jyu.fi/kotka/servlet/list-archive/verso/ and https://korppi.jyu.fi/kotka/servlet/list-archive/verso_opetus/

Workroom AgC 222.2, tel. 014-1234567.

Acceptor	**Date**	**Signature**	**Clarification**
Project manager	__.__.2010		
Customer	__.__.2010		
Instructor	__.__.2010		

Version history

Version	Date	Modications	Modifier
0.0.1	20.4.2010	Created report template.	JN
0.0.2	21.4.2010	Created the initial table of contents and wrote the introduction.	JN
0.0.3	22.4.2010	Wrote the backgroud chapter.	JN
0.0.4	28.4.2010	Finished the background chapter and wrote contents to the user inteface chapter.	JN
0.0.5	29.4.2010	Continued the user interface chapter.	JN

Contents

1 Introction

In a work community, sharing information increases productivity [1]. For this reason the Department of Mathematical Information Technology in University of Jyväskylä decided to start working on a good way for reseachers to their share source code among themselves. The idea was first developed by Ville Tirronen who made a prototype software that was tested with poor success. The idea was then proposed for a student project that was starting in the MIT deparment. The idea was approved and the project, soon to be known as Verso, started to further develop the idea of source code publishing software.

Verso project produced a web application that enables users to share their source code and maintain version history of their work. The software is based on Gitorious, which was chosen to be the starting point of the development because it covered the largest amount of requirements out of the other compared available software. The job of Verso project was to implement the missing requirements [2] to Gitorious. The web application was developed with Ruby On Rails framework for Linux servers and PCs. It was named as YouSource by one of the customers, Tero Tuovinen.

This document describes the implementation of the requirements specified in the requirement specification [2]. In chapter 2 the essential terminology used in the document is explained. Chapter 3 tells more about the background of the project. Chapter 4 describes the user interface of the application on the parts that were modified in the project. Chapter 5 explains the inner structure of the application. Chapter 6 specifies the programming practices used during the project and chapter 7 tells how testing was carried out. In chapter 8 it is considered how the objectives of the application were fullfilled and finally in chapter 9 the installation guide for the software is presented.

2 Terminology

YouSource is the name of the developed application. The name was introduced by Tero Tuovinen.

Commit contains file modification data and a log message from the user describing the changes.

Project is a user created project in the Verso system which has one or many repositories.

Repository refers to a Git repository which belongs to a project.

3 Background

Thic chapter goes briefly through the background of the developed software.

3.1 Need for a common publishing channel

Currently, at the MIT department in University of Jyväskylä, there is no common system for version control. This causes a number pf problems. For instance, when a worker leaves the deparment all the source code he has done might be lost too. The lack of a single version control system also prevents workers from knowing who is doing what, which may lead to producing overlapping work. Furthermore, the current disorganized situation presents licencing problems in which one is unaware who owns a piece of source code and how can it be used.

The aim was to get as many people to use proper version control as possible. Therefore, a system that can be used in many different ways to support different users was needed. A WWW-interface was planned for the people unfamiliar with version control. A command line interface was ment for the more experienced and even features that will adapt to people's current unique working methods should be implemented, such as reading and mirroring a zip-archive at supplied URL.

3.2 Gitorious

When the key requirements for the needed software were put together, it soon became apparent that software closely fitting for the requirements already existed. One noted application was GitHub which basically covered all the key requirements. However, GitHub isn't open source and buying it would be too expensive. More alternatives were reviewed and finally the project group end up with Gitorious.

Gitorious had most of the key requirements already implemented. It supports a widely used version control application called Git, it has a WWW-interface that covers most of the main features, all the information about the projects and repositories is easily obtained. However, some of the requested features were missing from Gitorious. It didn't support private projects, it didn't save project information to the repository itself and it didn’t support any update methods other than Git.

4 User interface

The user interface of YouSource was developed using html elements and a cascading style sheet (css). Most of the look of Gitorious was left untouched but almost all the new features needed new user interface implementation. This chapter describes the user interface changes that were done to Gitorious. Not only the web pages related to the new features were changed but also minor changes were made here and there to improve usability.

4.1 Logging in

Gitorious uses email for logging user in tp the system. This was concidered inconsistent because all the current web applications in University of Jyväskylä use a username for logging in. Therefore, YouSource should use the username too instead of email. Gitorious also sends an activation email after a new user has registered. This is not needed in YouSource since users can't register to the site by themselves. After the first login the site creates a new user to its database.

One of the main application in the university is Korppi, which provides various kinds of services for students, employees and guests. Korppi also provides LDAP authentication, and this was used as a method to log users in to YouSource. The LDAP authentication made it possible to not store user passwords at all to YouSource, since Korppi already has that information. This lead to removing all the UI elements that handled with passwords.

4.2 Creating a repository

The way a user can create a repository has changed a lot from Gitorious to YouSource. Gitorious offered only one way to create a repository while in youSource a user has several ways to do it. A user can initialize the repository with a zip file uploaded from his computer, or he can set up a mirror repository with providing a url to a zip file or a SVN repository. If the user prefers not to use any of these options, a normal empty repository will be created.

4.3 Repository browser

Gitorious doesn't offer any way to browse and search for repositories. In YouSource this was corrected and a page displaying all the repositories on the site was created. In YouSource this page is one of the four main pages which are activities, projects, repositories and teams. All the main pages have a link in the common header in each page.

The repositories page consists of a list of repositories and a side bar. The repository list is in the order the repositories are updated (latest first). In the side bar there is a list of the most active repositories from the past two weeks and a list of the user's own active repositories. The side bar also provides a link to create a new repository.

4.4 Updating repository with a zip package

On the repository page the side bar contains links to operational pages concerning the repository. One of these links is update repository with zip which leads to a form on a new page. The form accepts an url of a zip file or a local zip file for the update process. The form is also asking the user in which branch does he want the zip file to be pushed (updated). The dedault branch is the master branch if that exists. Otherwise, the default is a new branch but the name for the new branch is suggested as master.

4.5 Adding single files to a repository

Gitorious offers a way to browse the contents of a repository with a source tree view. This view was updated so that at the bottom of the page there is a small form which accepts a local file. After sending the form the file will be added the the folder that the source view is currently in. This form also allows updating files if a file with existing file name in the repository is offered.

4.6 Creating a mirror repository

While creating a repository it is possible to provide a url that leads to a zip file or a SVN repository. That url is called a mirror url because it will be stored to the repository for a daily update. If a url for a zip file is provided the repository will be created normally with Git and the contents of the zip will be added to it. On the other hand, if the user selects SVN mirroring, the repository will be created by cloning the SVN repository.

A script takes care of the daily update. It will go through all the mirror repositories in YouSource and checks if there's any changes in the source file or repository at the mirror url. The mirror url can be changed at any time on the repository edit page which can be accessed through the side bar of the repository page.

4.7 Private repository

A repository can be marked as private (as opposed to public) when the repository is being created or later on the repository edit page. A private repository means that it will only be shown to people that have view rights to it. A viewer is a new type of user priviledge that was developed into YouSource. The owner of a repository can modity who can view his repository in the manage collaborators page. The view rights can be given to a user or a team (latter will set all the members of the team as viewers).

4.8 Private project

Much like repositories, projects can be marked as private aswell. This can be done when the project is created or later on the project edit page. However, projects have a three step privacy while repositories have two step privacy. A project can be set to be visible to everyone, visible to users that are logged in or visible only to the members of the project. Members can be added and removed on the project members page which can be accessed through the side bar on the project page. It should be noted, that a project which is only visible to members is also visible to users who have gained view rights to one of the repositories in the project. That kind of user still can't view the other repositories in the project however.

4.9 Usability modifications

While the main focus of the project group was on developing the requested features, on the way there came up a few usability issues concerning the user interface. The project group also consulted a usability expert Meeri Mäntylä during the developing process and got valuable feedback from the initial users and from the usability testing sessions.

The most notable changes are the header and footer from which the header was totally redesigned. The menu in header was centered and the user control links were separated from the main manu. Auri Kaihlavirta supplied a logo, color theme and a bookmark icon for the website.

YouSource relies on SSH key authentication. Users are asked to upload a public SSH key before they can make a project. The SSH key generation process has proven to be difficult for unexperienced users during usability testing [4]. For this reason, the SSH key help was remarkably improved.

The usalibity testing sessions brought up another issue concerning a help box [**?**]. In Gitorious, when a user creates a new repository, a getting started message is shown until the initial commit is made to the repository. After that there is no way to bring the useful getting started message back. In YouSource this was corrected so that a getting started button is always shown and the getting started message can be viewed any time by clicking the button.

Meeri Mäntylä reviewed YouSource and gave instructions to the project group how to improve the usability of the application. Based on her advice, Gitorious' term dashboard (user's home page) was changed to the term home. Form labels were modified so that they have a colon at the end (e.g. Project name:). The difference between a project and a repository was increased by adding a small label on top of the project and repository names to incidate which one is in question.

The project group found a few usability flaws in Gitorious during their own use of the softawre. Buttons to create a new project and a new repository were added to project and repository pages respectively. Buttons to delete a project or repository were added to their own pages. The confirmation message for deleting a project was unified with the repository deletion confirmation.

5 Application structure

5.1 Operation, structure and components

5.2 Class structure

5.3 Configuration

5.4 Starting the application

5.5 Header, footer and breadcrumb

5.6 Sidebar and main content

6 Programming practices

6.1 Formatting of source code

6.2 Naming practices

6.3 Commenting of source code

6.4 Grouping of source code

6.5 Developing platform

7 Testing

7.1 Testing practices

8 Realization of objectives

8.1 Realization of requirements

8.2 Insufficient implementations

8.3 Further development ideas

8.4 Complications of implementation

8.5 Modification of implemtentation during the project

9 Installation guide

10 Sources

[1] Sanna Talja, *Information sharing in academic communities: Types and levels of collaboration in information seeking and use*, 2002, New Review of Information Behaviour Research, vol. 3, pages 143–160.

[2] Tero Hänninen & Juho Nieminen & Marko Peltola & Heikki Salo, *Requirement Specification, Verso Project*, 2010.

[3] Tero Hänninen & Juho Nieminen & Marko Peltola & Heikki Salo, *Memo, 2. usability testing, Verso Project*, 2010.

[4] Tero Hänninen & Juho Nieminen & Marko Peltola & Heikki Salo, *Memo, 1. usability testing, Verso Project*, 2010.

9