Verso Project

Application Report

Tero Hänninen
Juho Nieminen
Marko Peltola
Heikki Salo

Version 0.1.4
Public
18.5.2010

University of Jyväskylä

Department of Mathematical Information Technology

Jyväskylä

Project contact information

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:

Ville Tirronen	ville.e.t.tirronen@jyu.fi	014-2604987
Tero Tuovinen	tero.tuovinen@jyu.fi	050-4413685
Paavo Nieminen	nieminen@jyu.fi	040-5768507
Tapani Tarvainen	tt@it.jyu.fi	014-2602752

Instructors:

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

Contact information:

Email lists	verso@korppi.jyu.fi and yousource-users.group@korppi.jyu.fi.
Email archives	https://korppi.jyu.fi/kotka/servlet/list-archive/verso/ and https://korppi.jyu.fi/kotka/servlet/list-archive/yousource-users.group/
Workroom	AgC 222.2, tel. 014-2604963.

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 background chapter.	JN
0.0.4	28.4.2010	Finished the background chapter and wrote contents to the user interface chapter.	JN
0.1.0	29.4.2010	Continued the user interface chapter.	JN
0.1.1	3.5.2010	Corrected typos and started the structure chapter and programmin practises chapter.	JN
0.1.2	4.5.2010	Started the testing chapter.	JN
0.1.3	5.5.2010	Fixed appearance and added notes concerning the contents.	JN
0.1.4	6.5.2010	Started the realization of objectives chapter.	JN

Contents

1 Introduction

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 researchers 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 department. 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]. The decisions behind the practices and implementation can be found from the project report [3] 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 fulfilled and finally in chapter 9 the installation guide for the software is presented.

2 Terminology

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

Git push uploads the new commits in the local repository to a remote repository. It can be thought as releasing a version of a code.

KorppiLDAP is an authentication and directory access method. It is used in YouSource for authenticating the users.

Owner is a user or a team (which is a group of users). An owner has one or many projects and repositories.

Private project means that the project is only seen by its members.

Private repository means that the repository is only seen by users that have one of the following rights to it: view, review, commit or administrate.

Project is a user created project in the Verso system which has one or many repositories. It can have many attributes, for example a description, a home page URL and keywords. A project is owned by a user or a team.

Project group refers to the developers in Verso project: Tero Hänninen, Juho Nieminen, Marko Peltola and Heikki Salo.

Public means that the item (a repository or a project) is seen by all users.

Repository refers to a Git repository which belongs to a project. It stores data and the version history of that data. It can have several attributes, for example a description and a license. A repository is owned by a user or a team.

Ruby On Rails is an open source web application framework for Ruby programming language.

Viewer is a status that a user can have for a repository. It grants the ability to see the repository even if it is marked as private.

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

3 Background

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

3.1 Publishing Channel for Source Codes

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 department 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 licensing 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 meant 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 Starting Point: 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 is not open source and buying it would be too expensive. More alternatives were reviewed [4] and finally the project group ended up with Gitorious [5].

Gitorious is an open source application for hosting projects that use Git. It stores users' repositories and provides useful tools to manage them. Gitorious encourages users to collaborate with each other which gives it a feel of a social networking website. Other reviewed software that came close to the initial requirements were FusionForge, InDefero, GNU Savannah, Project Kenai, Fedora Hosted and KnowledgeForge. The problems with these were inactivity, difficult repository creation process, no activity view for repositories and limited search functions.

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 or repositories , it didn't save project information to the repository itself and it didn't support any update methods other than Git. All these features were added to YouSource. In addition, the logging in was changed to be similar to other services of the university and usability was increased with user interface modifications.

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 Sitemap

Since YouSource is based on a fully functional web application it has a lot of pages. Figure 4.1 shows nearly all the pages and describes their relations. Not all links are visible to simplify the picture. In the picture, the horizontal axis displays the relative path of the pages and vertical axis displays the logical dependencies between the pages. For instance, a project has many repositories and a branch has many commits.

4.2 Header, Footer and Breadcrumb

The header, the footer and the breadcrumb sections offer navigation and user control. The header is located at the top of each page and contains the main menu and the user menu. The main menu contains links to all the main pages of the application and the user menu contains links to user information and user control. For users not logged in the user menu contains only a link to login page.

The footer is located at the bottom of each page. It contains a menu with links to informative pages (about, FAQ, contact). The breadcrumb navigation is located right under the header but it is not visible on all pages. It contains the logical path to the currently visible page. For example: project / repository / branch / source tree. The breadcrumb is not visible on root pages like the four main pages (activities, projects, repositories, teams) or project creation and team creation pages. A full view of a page in YouSource is shown in Figure 4.2.

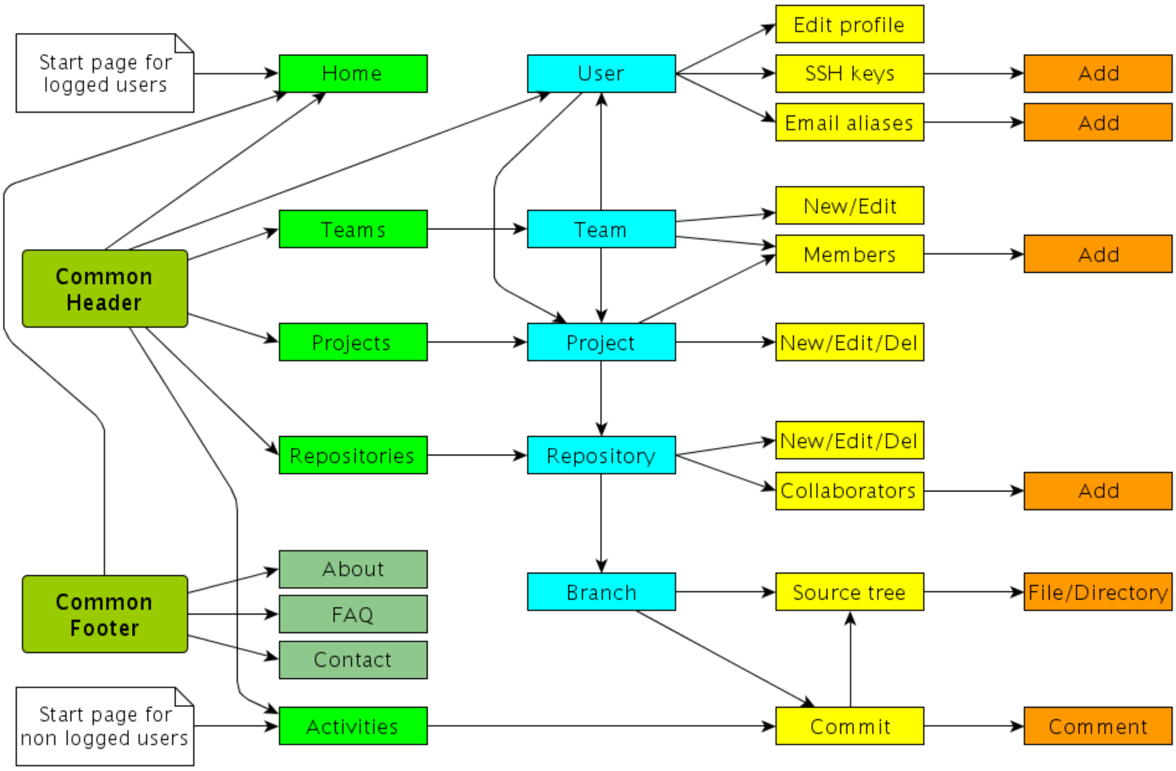


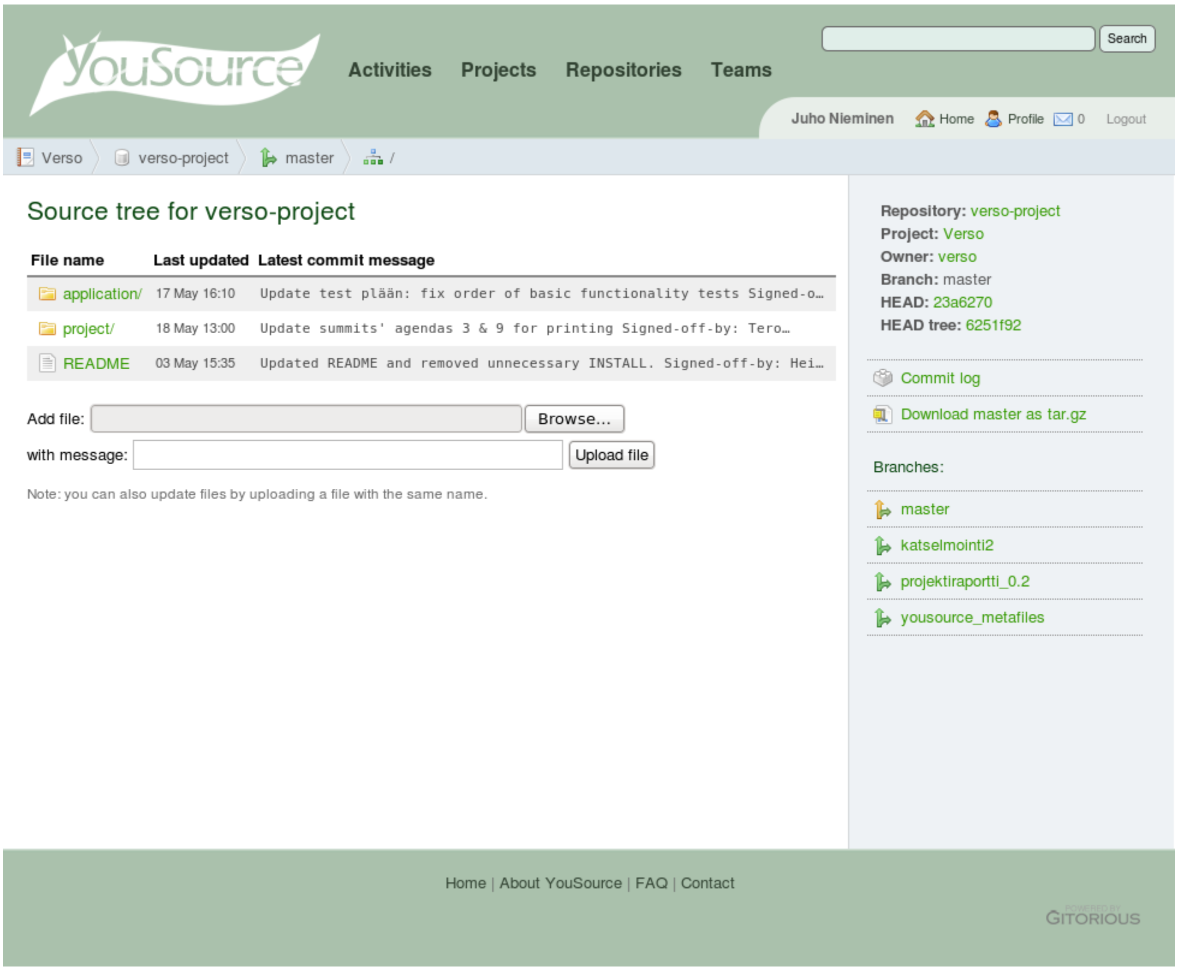
Figure 4.1: Simplified sitemap of YouSource.

4.3 Sidebar and Main Content

The sidebar is visible on most of the pages in the application. It contains additional information for the main content section and links management pages. The main content presents the most useful information on the page like the site and project activities. All the forms that are used to create projects, repositories and teams are also shown in the main content area as well as all the warning and notification messages. The main content and sidebar are visible in the full page view in the Figure 4.2.

4.4 Logging in

Gitorious uses email for logging user into the system. This was considered 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



Source tree for verso-project

File name	Last updated	Latest commit message
application/	17 May 16:10	`Update test plään: fix order of basic functionality tests Signed-o…`
project/	18 May 13:00	`Update summits' agendas 3 & 9 for printing Signed-off-by: Tero…`
README	03 May 15:35	`Updated README and removed unnecessary INSTALL. Signed-off-by: Hei…`

Add file:

with message:

Note: you can also update files by uploading a file with the same name.

Repository: verso-project
Project: Verso
Owner: verso
Branch: master
HEAD: 23a6270
HEAD tree: 6251f92

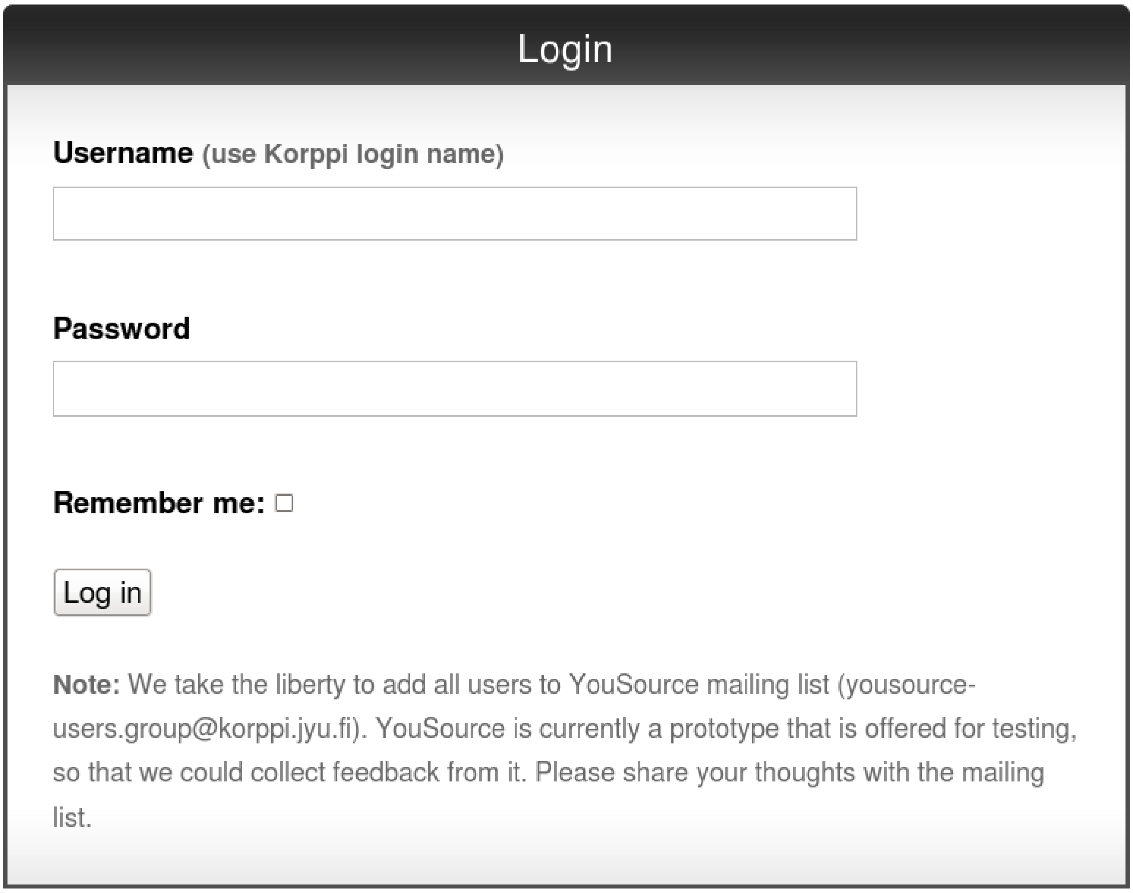
Branches:

- master
- katselmointi2
- projektiraportti_0.2
- yousource_metafiles

Figure 4.2: A page view showing the general sections: header, footer breadcrumb, sidebar and main content.

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 led to removing all the UI elements that handled passwords except for the login page which is shown in the Figure 4.3.



Login

Username (use Korppi login name)

Password

Remember me: □

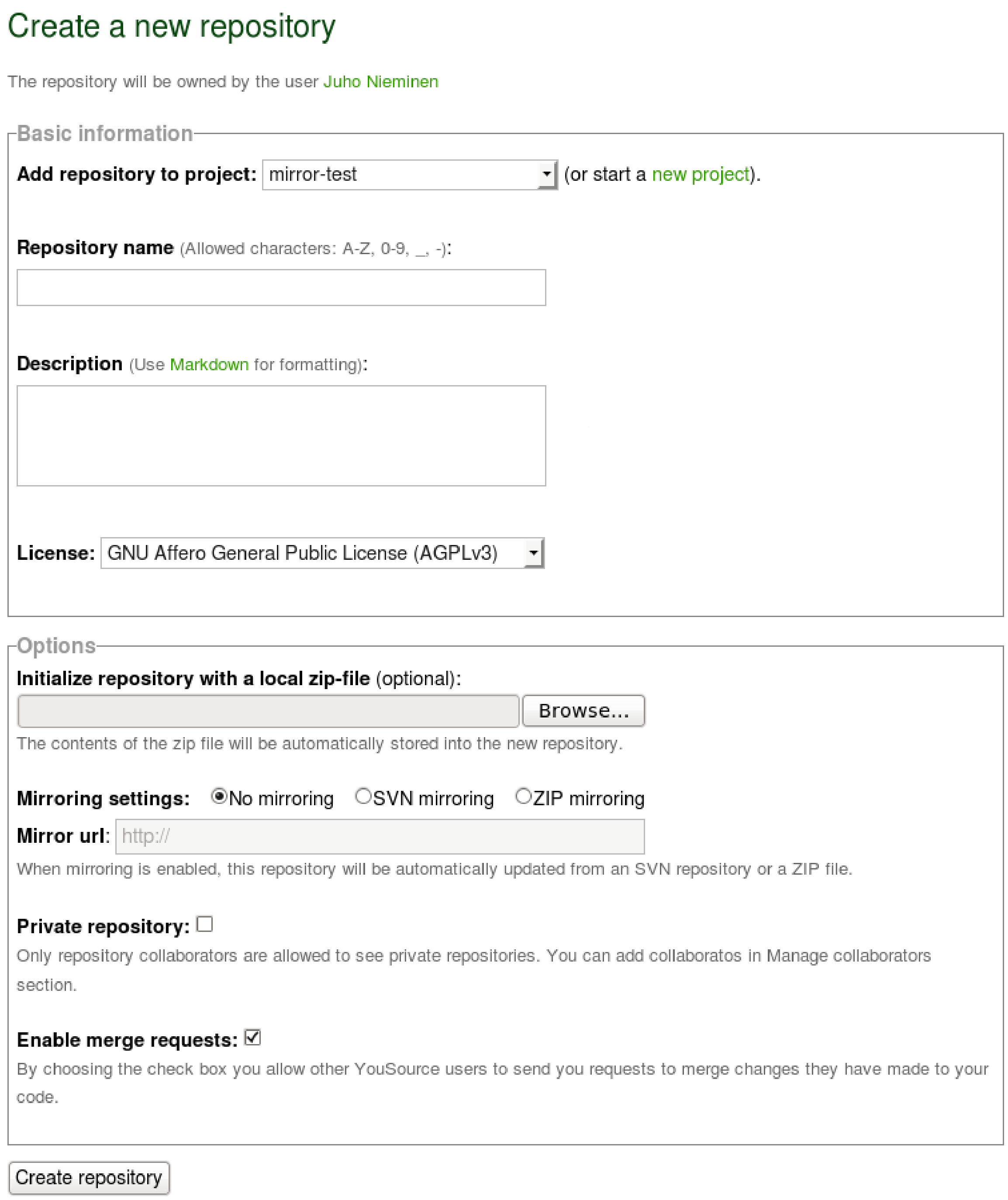
Log in

Note: We take the liberty to add all users to YouSource mailing list (yousource-users.group@korppi.jyu.fi). YouSource is currently a prototype that is offered for testing, so that we could collect feedback from it. Please share your thoughts with the mailing list.

Figure 4.3: A partial view of the login page.

4.5 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. Figure 4.4 displays the form that is used to create a new repository.



Create a new repository

The repository will be owned by the user Juho Nieminen

Basic information

Add repository to project: mirror-test (or start a new project).

Repository name (Allowed characters: A-Z, 0-9, _, -):

Description (Use Markdown for formatting):

License: GNU Affero General Public License (AGPLv3)

Options

Initialize repository with a local zip-file (optional):

The contents of the zip file will be automatically stored into the new repository.

Mirroring settings: ● No mirroring ○ SVN mirroring ○ ZIP mirroring

Mirror url: http://

When mirroring is enabled, this repository will be automatically updated from an SVN repository or a ZIP file.

Private repository: □

Only repository collaborators are allowed to see private repositories. You can add collaboratos in Manage collaborators section.

Enable merge requests: ☑

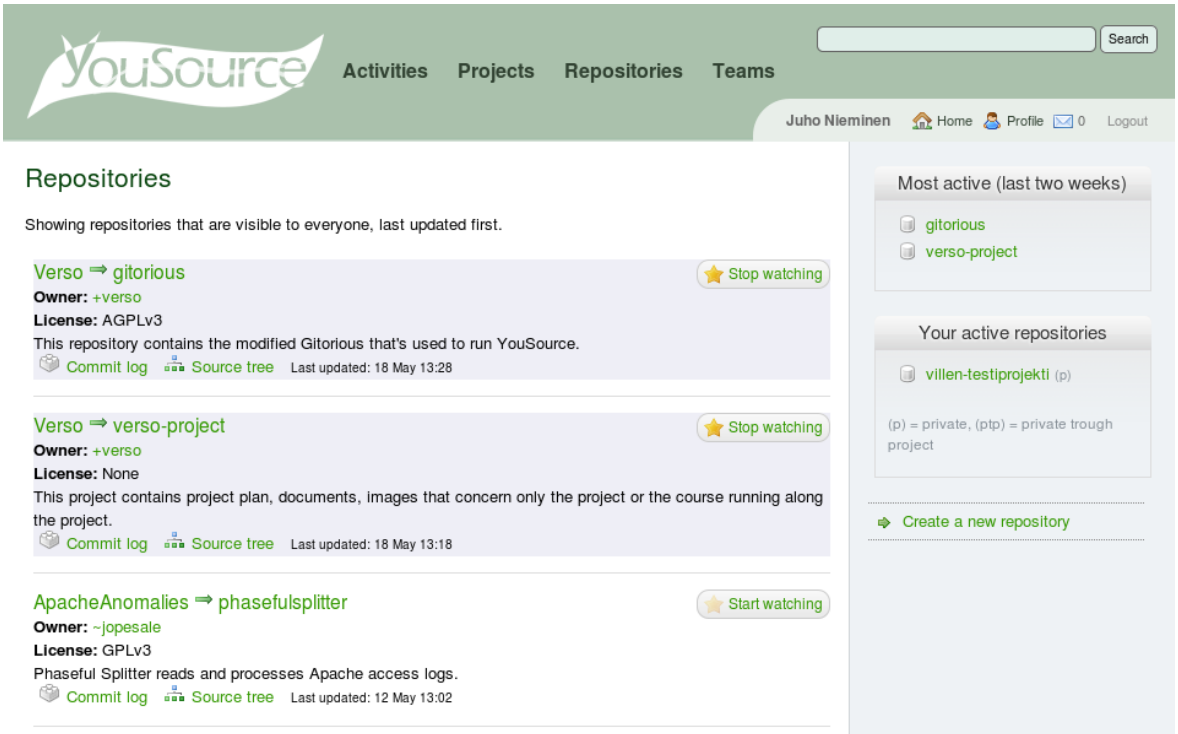
By choosing the check box you allow other YouSource users to send you requests to merge changes they have made to your code.

Figure 4.4: A partial view of the create a new repository page.

4.6 Repository Browser

Gitorious doesn't offer a listing page for repositories as it does for projects. In YouSource this was corrected and a page displaying all the public 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). The list is paginated for twenty repositories which means that only twenty repositories are shown at once and the rest can be viewed through page number links at the bottom of the list. If a user is logged in, the repositories in which the user has commit rights are highlighted as is seen in the Figure 4.5. In the sidebar there is a list of the most active repositories from the past two weeks and a list of the user's own active repositories (if logged in). The sidebar also provides a link to create a new repository.



Repositories

Showing repositories that are visible to everyone, last updated first.

Verso ⇒ gitorious
Owner: +verso
License: AGPLv3
This repository contains the modified Gitorious that's used to run YouSource.
Commit log Source tree Last updated: 18 May 13:28

Verso ⇒ verso-project
Owner: +verso
License: None
This project contains project plan, documents, images that concern only the project or the course running along the project.
Commit log Source tree Last updated: 18 May 13:18

ApacheAnomalies ⇒ phasefulsplitter
Owner: ~jopesale
License: GPLv3
Phaseful Splitter reads and processes Apache access logs.
Commit log Source tree Last updated: 12 May 13:02

Most active (last two weeks)

- gitorious
- verso-project

Your active repositories

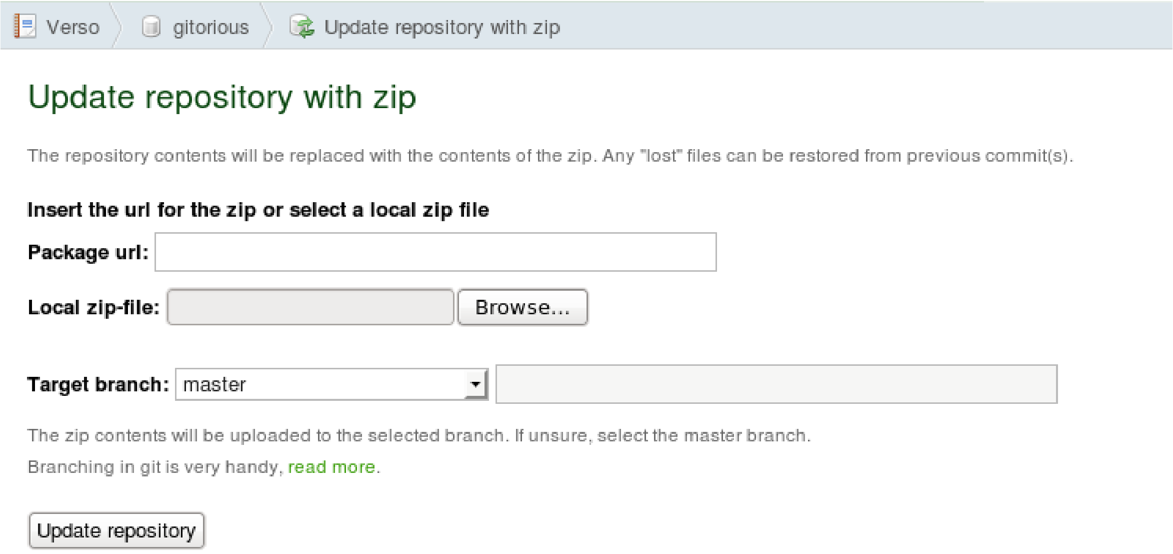
- villen-testiprojekti (p)

(p) = private, (ptp) = private trough project

Figure 4.5: A partial view of the repository browser page.

4.7 Updating Repository with a Zip Package

On the repository page the sidebar 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 (Figure 4.6). The form accepts a URL of a zip file or a path to 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 default 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.



Update repository with zip

The repository contents will be replaced with the contents of the zip. Any "lost" files can be restored from previous commit(s).

Insert the url for the zip or select a local zip file

Package url:

Local zip-file:

Target branch: master

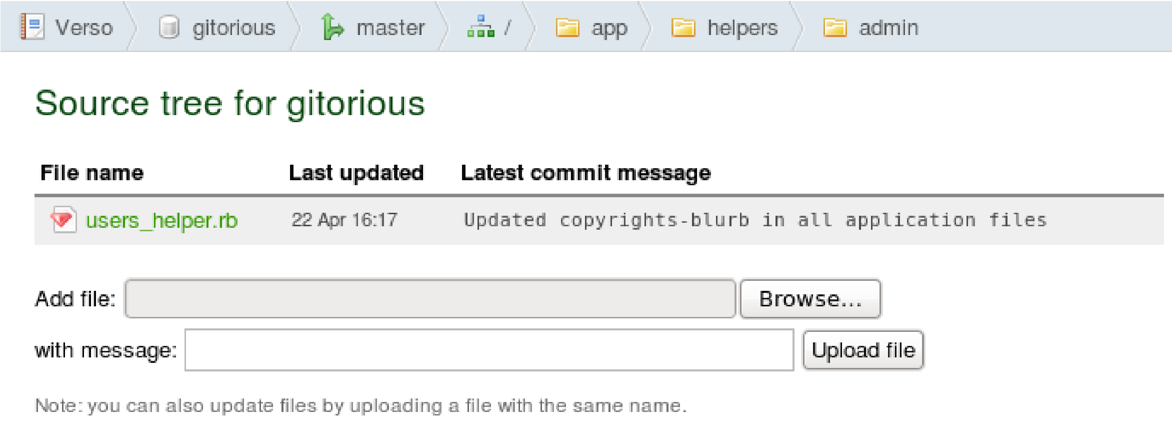
The zip contents will be uploaded to the selected branch. If unsure, select the master branch.

Branching in git is very handy, read more.

Figure 4.6: A partial view of the update repository with zip page.

4.8 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 (Figure 4.7). 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.



Verso › gitorious › master › / › app › helpers › admin

Source tree for gitorious

File name	Last updated	Latest commit message
users_helper.rb	22 Apr 16:17	Updated copyrights-blurb in all application files

Add file: [ ] Browse...

with message: [ ] Upload file

Note: you can also update files by uploading a file with the same name.

Figure 4.7: A partial view of the source tree page that shows the small form for adding sigle files to the current directory.

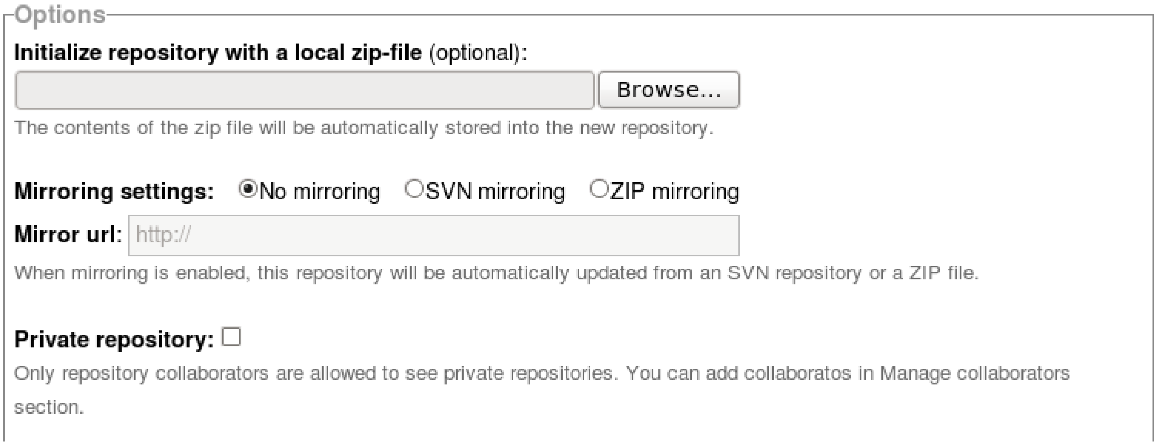
4.9 Creating a Mirror Repository

While creating a repository it is possible to provide a URL that leads to a zip file or a SVN repository (Figure 4.8). 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 sidebar of the repository page.

4.10 Private Repository

A repository can be marked as private (as opposed to public) when the repository is being created (Figure 4.8) 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 privilege that was developed into YouSource. The owner of a repository can modify 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). However, a viewer can only access the repository page, not



Options

Initialize repository with a local zip-file (optional):

Browse...

The contents of the zip file will be automatically stored into the new repository.

Mirroring settings: ● No mirroring ○ SVN mirroring ○ ZIP mirroring

Mirror url: http://

When mirroring is enabled, this repository will be automatically updated from an SVN repository or a ZIP file.

Private repository: □

Only repository collaborators are allowed to see private repositories. You can add collaboratos in Manage collaborators section.

Figure 4.8: A partial view of the create a new repository page showing the options for creating a mirror repository and a private repository.

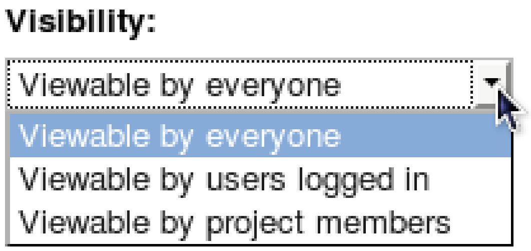
the repository data. Those privileges require committer or administrator rights to the repository.

4.11 Private Project

Much like repositories, projects can be marked as private as well. 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 (Figure 4.9. Members can be added and removed on the project members page which can be accessed through the sidebar 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.

4.12 Metafiles

Not all user interface changes concern the WWW-interface. One improvement to the use of command line interface with YouSource was implemented. The repository name, description and the option to allow merge requests are editable through the command line interface. This is achieved by adding a branch called yousource_metafiles



Visibility: Viewable by everyone

Figure 4.9: A partial view of the create a new project page showing the options for creating a private project.

to every repository created in YouSource. The branch contains a plain text file which specifies the description and a YML file witch sets the other options. These files are synchronized with the repository's options so that changes in the mentioned files show up in the repository's info and vice versa.

4.13 Usability Modifications to the WWW-interface

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 menu. Auri Kaihlavirta supplied a logo, color theme and a bookmark icon for the website. The new header is shown in the Figure 4.2 and for comparison the Gitorious header is shown in the Firuge 4.10.

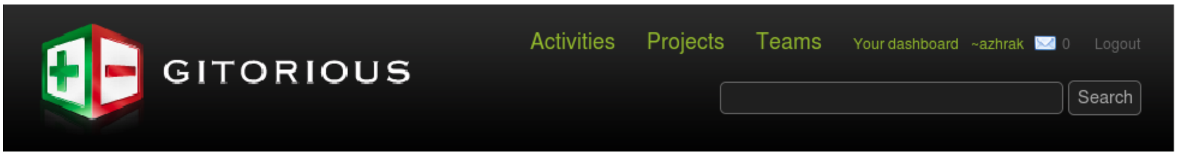


Figure 4.10: Site header of Gitorious when logged in.

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 inexperienced users during usability testing [6]. For this reason, the SSH key help was remarkably improved.

The usability testing sessions brought up another issue concerning a help box [7]. In Gitorious, when a user creates a new repository, a getting started message is shown until an 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 indicate which one is in question.

The project group found a few usability flaws in Gitorious during their own use of the software. 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

This chapter describes the different components in YouSource and their relations to each other. YouSource is mainly a server application because of its web service nature and the number of components it utilizes. The application uses Ruby on Rails libraries, MySQL database, UltraSphinx search engine, Stomp queuing server, Poller daemon to execute the queues, Git daemon for file download service and a number of external Ruby libraries (most notably Grit to handle Git functions).

5.1 Components

5.2 Class Structure

6 Programming Practices

6.1 Formatting Practices

6.2 Naming Practices

6.3 Commenting Practices

6.4 Grouping Practices

6.5 Development Platform

No single developing platform was used during the project. Each member of the project used a text editor of his choice to edit the source code. Gedit, Vim and Notepad++ were the most used editors. The requirements for the source code were that it should use the selected character encoding (UTF-8). Version control was handled with Git and YouSource itself making the project self hosting which was one of the initial goals.

Testing and debugging was carried out with Firefox 3.5.9 and Konqueror 4.4.2 browsers, Firebug 1.5.3 plug-in and Ruby on Rails console. The browsers were used to test the user interface and the general functionality of the application. The Firebug plug-in was used to debug Javascript and CSS. The Ruby on Rails console was used to debug new features.

7 Testing

In this chapter the testing in Verso project is described. The developed software, YouSource, was tested during the project by the project group, customers, tutors, initial users and two usability test users. The customers started to use the software as soon as it was set up in the test server. At that point no modifications were made to Gitorious. The initial users were invited to use the software after most of the key requirements were implemented. The usability tests took place at the final phases of development.

7.1 Testing Practices

The project group tested YouSource constantly during the development. The application was used for maintaining and keeping the version history of its own source code. In other words, YouSource was in active and practical use by the project group during the whole project. This practice revealed several bugs and usability issues from the application and also produced new ideas for future development.

When new features were implemented they were instantly tested by its developer and at least one other project member. The tests were based on the functional requirements supplied by the customer. No automatic testing was exploited, although the Ruby on Rails environment supports it. After the initial testing on the work machines the new features were added to the test server which was accessible for the group of initial users and the customers.

Occasionally the test server turned testing into a challenge. The problem was that it behaved differently compared to the work machines. A fine working feature on a work machine would present an error at the test server. Therefore, some bugs were not spotted but until they were out for the users to find. Luckily Gitorious served a feature that let all the errors happening on the test server to be emailed to the project group. This feature enabled a quick response to errors and helped to discover errors that were hard to find.

7.2 Usability Testing

YouSource's ease of use was tested in two usability testing sessions [6][7]. Before the sessions not much had been done to improve the usability of the application. The focus of development had been on implementing the required new features. Still, as one of the main ideas behind the project was to encourage more people to use version control, it was necessary to put effort into usability too.

The two usability testing sessions were conducted very much the same way but the test users very different. The first session had a test user who was an active programmer and had used to operate with command line and version control. The second testing session had an inexperienced test user instead. The testing sessions brought up a number of usability issues. Some of the problems were detected on both sessions and some came up only on either one of the two.

8 Realization of Objectives

This chapter describes the requirements set to YouSource and how well they were achieved. The main features that were developed were: login using KorppiLDAP, private projects and repositories, a repository browser page and different updating and mirroring methods for repositories. Some of the features that were left for future development were: possibility to comment and certificate source code, button for automatic merging, editing of text based files via the WWW-interface and requiring the user's SSH key only when truly needed.

8.1 Realization of Requirements

All of the mandatory and nearly all of the important feature requirements [2] of the application were fulfilled in the project. Possibility to authenticate users with Kerberos and requiring the user's SSH key only when truly needed were considered important but were left out because of lack of time. Out of the features marked as useful only the feature that shows an abbreviation of the repository's license was implemented.

8.2 Insufficient Implementations

8.3 Further Development Ideas

8.4 Complications of Implementation

8.5 Modifications of Implementation during the Project

9 Installation Guide

9.1 Configuration of the Application

9.2 Starting the Application

10 Summary

11 References

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

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

[3] Heikki Salo, *Project Report, Verso Project*, 2010.

[4] Tero Hänninen, *Comparison of Platforms, Verso Project*, 2010

[5] Johan Sørensen, *About Gitorious*, 2009.

[6] Juho Nieminen, *Memo, 2. usability testing, Verso Project*, 2010.

[7] Juho Nieminen, *Memo, 1. usability testing, Verso Project*, 2010.

9