FLOOD-serv

D3.5 Semantic Wiki

V 2.0

Project acronym:	**FLOOD-serv**
Project full title:	**Public FLOOD Emergency and Awareness SERvice**
Grant agreement no.:	**693599**
Responsible:	**SIVECO**
Contributors:	**SIVECO**
Document Reference:	**D3.5**
Dissemination Level:	**PU**
Version:	**Final**
Date:	**31/01/2018**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 693599

Table of contents

List of tables

List of abbreviations

Abbreviation	Explanation
FLOOD-serv	Public FLOOD Emergency and Awareness SERVice
JSON	JavaScript Object Notation
OWL	Web Ontology Language
PHP	PHP: Hypertext Preprocesor
RDF	Resource Description Framework Schema
SMW	Semantic MediaWiki
SSO	Single Sign On
XML	Extensible Markup Language

Executive summary

This document reports on the technical implementation of the FLOOD-serv Semantic Wiki (SW)solution. We explain the hardware and software infrastructure used and the overall architectural structure of the SW. FLOOD-serv SW is based mainly on MediaWiki and Semantic MediaWiki technologies. From a content point of view, the SW has four types of content/articles:

1. Articles about Floods from a physical science perspective;
2. Articles about Flood Risk Management;
3. Articles about Geographically, Legally, Institutionally contextualized information;
4. Articles related to the FLOOD-serv Project;

Our Consortium worked collaboratively to identify relevant content and write articles about various topics, resulting over 100 articles in the SW. Much of the content was translated in the 5 languages of the Pilot Cities, resulting over 300 articles including translations.

The SW solution was thoroughly tested internally by SIVECO (before the complex piloting process that took place under WP5). All elicited requirements were implemented. At the time of finalizing version 1.0 of this report, one functionality (related to user administration) was not implemented (since it required Single Sign On integration which was to be implemented under the integration Task 4.5. This functionality was later implemented and is reported under D4.5 and D5.4.

1 Introduction

1.1 Document Purpose and Scope

This document, D3.5, is an outcome of Task 3.7, within Work Package 3 of the FLOOD-serv Project. The purpose of this document is to report on the implementation and release of the FLOOD-serv Semantic Wiki (SW) component of the FLOOD-serv System, based on the user requirements developed in D3.1 and D3.2. The SW contains general information and knowledge about floods and flood management but also specific and contextualized knowledge related to the FLOOD-serv Project, Flood Risk Management systems in the countries of the Project Pilot cities, etc. The SW is dedicated for use by both specialists on the one hand, and regular citizens on the other. It is aimed informing the public and raising awareness about floods and flood risk management.

1.2 Target Audience

This document is publicly available. It is available for members of the review committee on behalf of the European Commission to serve in the review process, but also to the general public who are interested in floods, the FLOOD-serv Project or semantic wikis.

1.3 How Final Review Observations Were Addresses

NOTE: This document was modified following the Final Review feedback. This is the final version after modifications.

Reviewers' observations	Explanations as to how observations are addressed
more substantial description of the content of listed components should be provided in the Deliverables D3.3, D3.5 - D3.7 accordingly to the remarks of the present report and its Annex 1 - Deliverables due for the Period 2/Final review.	Technical explanations about the component have been added in Chapter 2, *Technical Description*. A Chapter about SW Content has been added: Chapter 3, Semantic Wiki Content.
The deliverable describes Semantic Wiki component of FLOOD-serv platform. Out of 10 user requirements, 9 were implemented, and 1 (user administration) was postponed until the implementation of the Single Sign On at the level of the entire FLOOD-serv Portal, it is however no indications (on the Deliverable submission date Jan.3st, 2018) when this implementation will take place.	Indeed, at the moment of submitting this report, integration (and SSO and User Management were not finalized). We now explained in this document in Section 4.1, on page 22, and in Section 4.7 at page 72 that this was postponed for T4.5 and details are present in D4.4, D4.5 and D5.4.
More detailed information on the testing process is required.	Section 4.1 gives an Overview of Testing Methodology and Results.
It is regrettable no access was provided in order to allow reviewers accessing and evaluating the content of Semantic Wiki Tool. The report foot page requires correction.	Previous version gave access information for the test environment (still available but with no updated content). Current version gives access information for the production environment, see Section 2.1, *Access Information*.

The document has the following structure:

Chapter 1: Introduction

Chapter 2: Technical Description

Chapter 3: Semantic Wiki Content;

Chapter 4: Testing and Test Results;

Chapter 5: Conclusions

2 Technical Description

2.1 Access Information

Initial test development url (still functional):
http://195.82.131.198/semantic/index.php/Main_Page

Credentials: the user can create an account.

Final production url: https://wiki.floodserv.eu/index.php/Main_Page

Access credentials and method:

The User needs to access the FLOOD-serv Portal url: https://floodserv.eu/

Log in using user's user name and password. If the user has no account, he/she needs to create one.

In the Main Menu click "SEMANTIC WIKI".

2.2 Technologies Used

2.2.1 Physical hosting infrastructure

The Semantic Wiki Server is hosted in a virtual server (the physical infrastructure on SIVECO's premises) with the following characteristics.

Processor: 4 CPUs x 2 cores,2.4 GHz per core

Memory: 16 GB

Hard disk capacity: 120 GB (not including backup)

BANDWIDTH: at least 10 MBps

Operating system: Unix based Ubuntu

The virtual environment is scalable; if more resources become necessary they can be allocated.

2.2.2 Software Technologies

2.2.2.1 Operating system and software packages

Install Ubuntu Server 16.04 as operating system and on top of that install Apache, MySql, PHP and Memcache

During the installation of Ubuntu we have installed also Apache, MySql and PHP.

In order to run properly, MediaWiki needs PHP version 5.5.9 or later is required

On PHP the following extensions were enabled:

- **Perl Compatible Regular Expressions (PCRE)**
- **Session**

- **Standard PHP Library**
- **JSON**
- **mbstring**
- **fileinfo**

Note that all of these are enabled in PHP by default.

Also you need to enable the optional **PHP intl extension** "to handle Unicode normalization".

In most Debian/Ubuntu-based distros, this is in the php5-intl package.

Also **PHP OpenSSL** extension must be enabled .

Either the **php5-mysql** or **php5-mysqlnd** package is required

Take in consideration that some features require PHP functions that execute external processes, like image thumbnailing, that some in some configuration is usually disabled. This has surfaced specially on file uploads.

For VisualEditor is required additional **PHP features**, like **libcurl** support (php5-curl on Debian/Ubuntu-based distros).

Redis is an in-memory key-value store known for its flexibility, performance, and wide language support. For improved performance Semantic MediaWiki needs Redis.

To configure Redis as a cache we have made following update to configuration file located on /etc/redis/redis.conf.

We have configured the max memory for Redis as well as how Redis will select what to remove when the max memory is reached, by adding the following lines at the end of the file:

maxmemory 128mb

maxmemory-policy allkeys-lru

Save and close the file, then restart the Redis service:

sudo systemctl restart redis-server.service

Next, enable Redis on system boot:

sudo systemctl enable redis-server.service

2.2.2.2 MediaWiki and Semantic Mediawiki

Technical implementation of the Component involved the scouting, choice, instalation, adaptation, configuration and integration of a series of existing open source technologies. The major technologies chosen and implemented are MediaWiki: which is collaborative documentation platform supported by a large community of users and developers (which also powers Wikipedia). This is a “regular wiki” technology, allowing users to develop content in a dictionary/encyclopedia style. In addition, a major extension to this technology is Semantic MediaWiki, which allows more advanced semantic annotations and querying of content.

MediaWiki is a free open-source wiki software and as all wikis it allows everyday users to create and edit webpage content in any browser without knowledge in CSS and HTML. Mediawiki supports this open editing function where anyone can add and edit content, which the software would write in its database but without deleting the old version. Thus in case of certain abuse of this great functionality, the administrators could revert to the original

unedited version. MediaWiki is also the most famous wiki engine as it is used to power Wikipedia.

Semantic MediaWiki (SMW) is an extension of **MediaWiki** – that helps to search, organise, tag, browse, evaluate, and share the wiki's content. While traditional wikis contain only text which computers can neither understand nor evaluate, SMW adds semantic annotations that allow a wiki to function as a collaborative database. Semantic MediaWiki was first released in 2005, and currently has over ten developers, and is in use on hundreds of sites. In addition, a large number of related extensions have been created that extend the ability to edit, display and browse through the data stored by SMW: the term "Semantic MediaWiki" is sometimes used to refer to this entire family of extensions.

Semantic MediaWiki enables wikis to make their knowledge computer-processable, so that you can find and display the answer to this question - and to many more.

Semantic MediaWiki introduces some additional markup into the wiki-text which allows users to add "semantic annotations" to the wiki. While this at first appears to make things more complex, it can also greatly simplify the structure of the wiki, help users to find more information in less time, and improve the overall quality and consistency of the wiki. Here are some of the benefits of using SMW:

- **Automatically-generated lists.**
- **Visual display of information.**
- **Improved data structure.**
- **Searching information.**
- **External reuse.**
- **Integrate and mash-up data.**

Other extensions used are:

Extension name	Description	URL
Babel	adds a parser function to replace the *old* Babel system that completely relied on templates. If an unrecognized language parameter is specified, it will see if there is an existing template with the name and include that	https://www.mediawiki.org/wiki/Extension:Babel
Cite	allows a user to create references as footnotes on a page. It adds two parser hooks to MediaWiki, <ref> and <references />; these operate together to add citations to pages.	https://www.mediawiki.org/wiki/Extension:Cite
Bootstrap	provides the Bootstrap 4 web front-end framework to MediaWiki. It can be activated directly or used from skins or extensions.	https://www.mediawiki.org/wiki/Extension:Bootstrap
CleanChanges	is based on enhanced changes list, but it tries to be more concise by hiding less important information by default. It needs JavaScript to be fully functional. It works best in wikis where the number of *changes per user* is high.	https://www.mediawiki.org/wiki/Extension:CleanChanges

CiteThisPage	previously known as SpecialCite.php, is an extension that creates a special page (Special:CiteThisPage) and toolbox link to it for pages in the main namespace. Given an article name, this extension generates citations to it in a variety of styles.	https://www.mediawiki.org/wiki/Extension:CiteThisPage
CLDR	contains local language names for different languages, countries, currencies, and time units extracted from CLDR data	https://www.mediawiki.org/wiki/Extension:CLDR
Confirmedit	extension lets you use various different CAPTCHA techniques, to try to prevent spambots and other automated tools from editing your wiki, as well as to foil automated login attempts that try to guess passwords. ConfirmEdit ships with several techniques/modules to generate captcha.	https://www.mediawiki.org/wiki/Extension:ConfirmEdit
Gadgets	provides a way for users to pick JavaScript or CSS based "gadgets" that other wiki users provide	https://www.mediawiki.org/wiki/Extension:Gadgets
ImageMap	allows clickable image maps. An image map is a list of coordinates in a specific image, which hyperlinks areas of the image to multiple destinations (in contrast to a normal image link, in which the entire area of the image links to a single destination).	https://www.mediawiki.org/wiki/Extension:ImageMap
InputBox	adds already created HTML forms to wiki pages. Users can "complete" a form (entering text, selecting menu items, etc.) by entering text into the box.	https://www.mediawiki.org/wiki/Extension:InputBox
InterWiki	allows the InterWiki list in the database to be maintained from a list of entries in a file instead of running an update script or using a special page. This has the advantage of allowing a number of wikis installed on the same server to share the same interwiki list simplifying maintenance.	https://www.mediawiki.org/wiki/Extension:InterWiki
LanguageSelector	provides detection of the interface language to use for anonymous visitors, and a menu for selecting the user language on each page, for logged in users as well as anonymous visitors. It also provides an additional tag, <languageselector>, for embedding the selector into wiki pages and system messages.	https://www.mediawiki.org/wiki/Extension:LanguageSelector
LocalisationUpdate	allows to update the localizations for MediaWiki messages at any time, without needing to upgrade the MediaWiki software	https://www.mediawiki.org/wiki/Extension:LocalisationUpdate
Mpdf	lets you export the printable version of a page as a PDF file. To do so it uses uses the "mPDF PHP class" in version 6.1 for HTML to PDF conversion.	https://www.mediawiki.org/wiki/Extension:Mpdf
Nuke	makes it possible for sysops to mass delete pages.	https://www.mediawiki.org/wiki/Extension:Nuke

OpenID Connect	extends the PluggableAuth extension to provide authentication using OpenID Connect.	https://www.mediawiki.org/wiki/Extension:OpenID_Connect
Page Forms	(known until November 2016 as **Semantic Forms**) is an extension to MediaWiki that allows users to add, edit and query data using forms. It was originally created as an offshoot of the Semantic MediaWiki extension, to be able to edit templates that store their parameters via SMW, which is why it was originally called "Semantic Forms". However, it can now also work with the alternative Cargo extension, or with neither extension installed.	https://www.mediawiki.org/wiki/Extension:Page_Forms
ParserFunctions	enhances the wikitext parser with helpful functions, mostly related to logic and string-handling. Since MediaWiki 1.15, ParserFunctions has incorporated most (but not all) of the functions from the StringFunctions extension, which may be enabled or disabled (See the install instructions below).	https://www.mediawiki.org/wiki/Extension:ParserFunctions
PluggableAuth	provides a framework for creating authentication and authorization extensions. **PluggableAuth** provides the shared code necessary to implement these extensions, which are referred to below as authentication plugins and authorization plugins. **PluggableAuth** must be used with exactly one authentication plugin and zero or more authorization plugins.	https://www.mediawiki.org/wiki/Extension:PluggableAuth
Poem	allows easy formatting of poems and similar material within Wikitext. Once the extension is enabled, you can put any block of text within <poem></poem> tags	https://www.mediawiki.org/wiki/Extension:Poem
Renameuser	provides a special page which allows authorized users to rename user accounts. This will cause page histories, etc. to be updated. If you want to rename an account "into" another, already existing one, you need the User Merge and Delete extension.	https://www.mediawiki.org/wiki/Extension:Renameuser
Semantic Cite	allows to manage citation resources using semantic annotations.	https://www.mediawiki.org/wiki/Extension:Semantic_Cite
SyntaxHighlight	formerly known as SyntaxHighlight_GeSHi, provides rich formatting of source code using the <syntaxhighlight> tag. It is powered by the Pygments library and supports hundreds of different programming languages and file formats.	https://www.mediawiki.org/wiki/Extension:SyntaxHighlight
TitleBlacklist	allows wiki administrators to block the creation, movement and upload of pages, the title of which matches one or more regular expressions, as well as blocking creation of accounts with matching	https://www.mediawiki.org/wiki/Extension:TitleBlacklist

	usernames.	
Translate	makes MediaWiki a powerful tool to translate every kind of text. It's used especially to translate software and to manage multilingual wikis in a sensible way.	https://www.mediawiki.org/wiki/Extension:Translate
Universal Language Selector	allows users to select a language and configure its support in an easy way. Where used, it also ships the functionality of both WebFonts and Narayam extensions (both of which have been deprecated in favour of Universal Language Selector). See Universal Language Selector for background and additional information.	https://www.mediawiki.org/wiki/Extension:UniversalLanguageSelector
WikiEditor	provides an improved interface for editing wikitext. It is the wikitext editing interface that Wikipedia started using in 2010 for desktop users, and so it is sometimes called the "*2010 wikitext editor*". The extension used to provide "Labs" features for "publication" and "preview" steps, but these are removed from REL1_31 onwards	https://www.mediawiki.org/wiki/Extension:WikiEditor

Table 1: Installed Extensions to MediaWiki

2.3 Component structure

The Semantic Media Wiki user functions can be divided into the following groups:

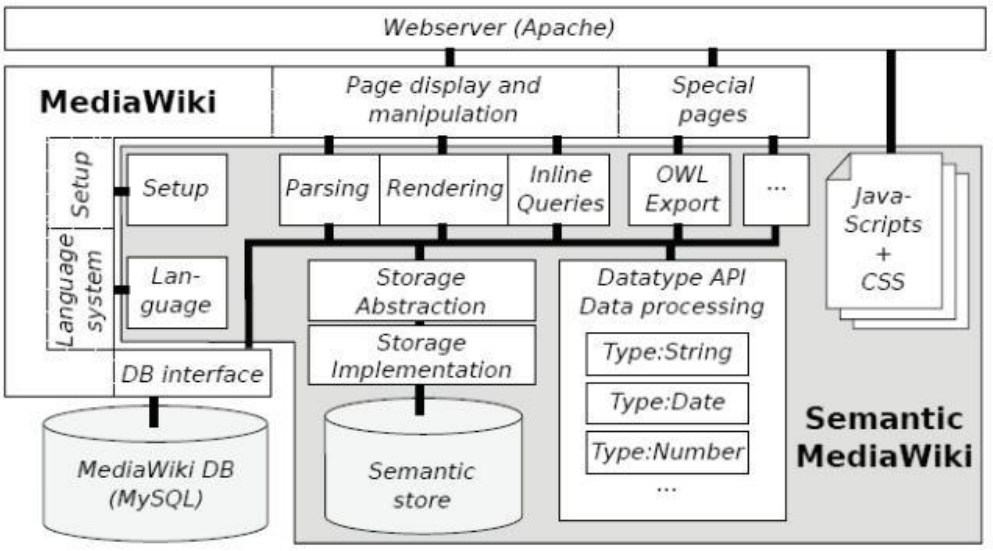


Figure 1: Semantic Wiki Component Structure

2.3.1 Data input

The data input function is used for gathering of information, being the main layer between the user client and the business and storage layer. It is composed of the following data types:

- Text – corresponds to the HTML text input
- Textarea – corresponds to the HTML textarea input and it can be also be accompanied by a WYSIWYG (what you see is what you get) editor to enable the input of rich text content.
- Combobox – a combination of an autocomplete input and a drop down to let users both search for values but also see them all at once
- Tokens – multiple value text input that allows a structured way of adding and rearranging data
- Radiobutton – corresponding to the radiobutton HTML element it lets users see a list of values from which they can choose one
- Dropdown – it corresponds to the dropdown HTML element and it let's users see a list of values from which they can select one
- Checkboxes – corresponding to the checkbox HTML element let's users see a list of values from which they can choose multiple elements.
- Tree – the three element contains a hierarchical list of elements and with the help of checkbox elements it allows users to see and select multiple values on different branches.

2.3.2 **Data inspection and output**

The component is also part of the user interaction with the system. It is responsible for allowing the browsing of input data and special pages. The component communicates directly with the business layer and outputs the information as it is and also allowing user actions.

2.3.3 **Query answering and query formatting**

The component in mainly based on the business layer (Wiki Layer) and the communication of it with the repository. It allows data acquisition from the repository on user request and also the manipulation of these requests.

2.3.4 **Data export and import**

Part of the business layer (Wiki Layer) the component allows bulk addition and output of information from different sources with a standard format (XML and CSV). The data is here processed and mapped on the standard architecture to allow the reuse of it in other similar systems.

- Maintenance tools

The component contains specific administration functions to allow the maintainability of the system. These functions consist of:

- Backup of the repository and rebuilding the repository thru RDF export/import of existing triples
- Search table indexing and rebuilding of search indexes
- Property usage statistics and the rebuild of the property usage statistics
- Configuration management
- Migration management

2.3.5 **Storage**

The repository layer is responsible for querying and modifying semantic data and properties. The Semantic Media Wiki module implements two types of storage engines:

-Structured data using SQL based database schema

-Semantic storage using SPARQL as a data store

3 Semantic Wiki Content

The SW contains general information and knowledge about floods and flood management but also specific and contextualized knowledge related to the FLOOD-serv Project, Flood Risk Management systems in the countries of the Project Pilot cities, etc. The content is developed collaboratively and the end result is open ended; the SW is never final but can be improved continuously by users.

The areas of knowledge covered by the SW are rather wide topics of evolving human knowledge, possibly to be extended without any obvious limit. Therefore, it was important to establish criteria for scope delimitation and topic prioritization based on project objectives and identified user needs. Initially (in D3.2) content was planned according to some general criteria of:

1. Relevance to project and project concerns: semantic data will be selected based on general thematic relevance to the project concerns related to emergency management in case of floods. Relevance will be judged by expert SW content developers and validated with designers and developers on other components, and representatives of pilot cities.
2. Applicability to identified, generated or used data: a very concrete criterion of inclusion is based on available data sources and variables developed by project components or imported from external sources.
3. Prior existence of semantic or ontological data and level of effort needed for further or new development: the SW component will not start from scratch but build (as per project requirements) on existing semantic and ontological data and prior research in relevant fields.

To advance point 3, SIVECO and DDNI have conducted a literature study or relevant research on the topic of SW in FRM, or related topics like ontologies, vulnerability, emergency management, etc. It should be noted that the literature directly relevant for our concerns (i.e simultaneously approaching semantic wikis or ontologies and FLR) is rather "thin", which speaks to the innovativeness of having this king of a semantic wiki. Here is a selected list of consulted literature:

Andreas H. Schumann (auth.), A. H. S. (eds. . (2011) *Flood Risk Assessment and Management: How to Specify Hydrological Loads, Their Consequences and Uncertainties*. 1st edn. Springer Netherlands. Available at: http://gen.lib.rus.ec/book/index.php?md5=DF5B3C589F3D31D77CED8D383B436E08.

Begum, S., Stive, M. J. F. and Hall, J. W. (eds) (2007) *Flood Risk Management in Europe: Innovation in Policy and Practice*. Dordrecht: Springer (Advances in Natural and Technological Hazards Research).

De Wrachien, D. *et al.* (2012) 'Ontology For Flood Management: A Proposal', *WIT Transactions on Ecology and the Environment*, 159, pp. 3–13.

Environment Agency England and The RRC (UK) (2007) *RiverWiki*. Available at: https://restorerivers.eu/wiki/.

Galton, A. and Worboys, M. (2011) 'An Ontology of Information for Emergency Management', in *Proceedings of the 8th International ISCRAM Conference*. *ISCRAM Conference*, Lisbon. Available at: http://oldway.org/publications/ISCRAM-122-final.pdf.

Garrido, J. and Requena, I. (2011) 'Proposal of ontology for environmental impact assessment: An application with knowledge mobilization', *Expert Systems with Applications*, 38(3), pp. 2462–2472.

Garrido, J., Requena, I. and Mambretti, S. (2012) 'Semantic model for flood management', *Journal of Hydroinformatics*, 14(4), pp. 918–936.

Grenon, P. and Smith, B. (2004) 'SNAP and SPAN: Towards Dynamic Spatial Ontology', *Spatial Cognition and Computation*, 4(1), pp. 69–103.

Gruber, T. R. (1995) 'Toward principles for the design of ontologies used for knowledge sharing?', *International Journal of Human-Computer Studies*, 43(5), pp. 907–928. Available at: http://www.sciencedirect.com/science/article/pii/S1071581985710816.

Huner, K. M. (2009) 'The Effect of Using a Semantic Wiki for Metadata Management: A Controlled Experiment', in *Proceedings of the 42nd Hawaii International Conference on System Sciences*. IEEE Computer Society, pp. 1–9.

Huner, K. M., Otto, B. and Osterle, H. (2011) 'Collaborative management of business metadata', *International Journal of Business Metadata*, 31, pp. 366–373.

Khazai, B. *et al.* (2014) 'VuWiki: An Ontology-Based Semantic Wiki for Vulnerability Assessments', *International Journal of Disaster Risk Science*, 5(1), pp. 55–73. Available at: https://link.springer.com/article/10.1007/s13753-014-0010-9.

Province of Zeeland and HZ University of Applied Sciences (2013) *Flood Aware Wiki*. Available at: http://floodawarewiki.eu/ (Accessed: 16 August 2017).

Simonović, S. P. (2013) *Floods in a Changing Climate: Risk Management*. 1st edn. Cambridge University Press (International Hydrology Series). Available at: http://gen.lib.rus.ec/book/index.php?md5=49C9EC2F9C2C4DD35FDDD2626B852016.

Smith, K. and Ward, R. (1998) *Floods: Physical Processes and Human Impacts*. 1st edn. Wiley.

The assessment of relevance applicability of topics was an ongoing issue. A working group of partners working on SW content was established and worked throughout the project. The following partners participated in the workgroup:

1. SIVECO, coordinator;
2. DDNI;
3. EXDWARF
4. BILBAO;
5. BRATISLAVA
6. GENOVA;
7. TULCEA;
8. CMVNF

The group worked on a list of topics divided in 4 broad areas:

1. Floods& Physical sciences;
2. FRM;
3. Geographical&Legal&Institutional information;
4. FLOOD-serv Project specific information;

Main Categories Used:

- City

- Country;
- County
- Legal Norm;
- Region;

Main properties used:

- Has country;
- Has county;
- Has region;
- Has definition;
- Has short description;
- Has year;
- Legislation type;
- Is vulnerable to;
- Issued by;
- Has number;
- Has population;
- Has area;

As of reporting there are a number of 107 terms/articles in the SW.

Content was developed initially in English. After the content reached a certain point, the working group also worked on translations of content (usually each partner translating from English in their native language). With translations the total of articles reached over 300 (not all articles are translated in all languages, information of local relevance is present only in English and local language).

4 Testing and Test Results

4.1 Overview of Testing Methodology and Results

Initial tests of the Semantic Wiki tool were carried out as unit tests executed by internal staff of the implementer of the component (Siveco). Tests were aimed at verifying all functional user requirements as defined in D3.1 and D3.2. Therefore the benchmark against which all functionalities were tested were the user requirements. Table 2, below, presents an overview of the user requirements/user stories as presented in D3.1 and D3.2.

User Story Code	*User requirement code*	*Name*	*User story*
USSW01	SWRQ02	Search	As a citizen / content editor / administrator I want to search through the semantic wiki for terms or definitions so that I can access specific information.
USSW02		Data export	As an Administrator I want to export data from semantic wiki into OWL, RDF and XML formats.
USSW03	SWRQ08	Content editing	As a content editor / administrator I want to create content / edit / publish content so that I can improve the terms, definitions and vocabularies.
USSW04	SWRQ03 SWRQ04 SWRQ05 SWRQ06 SWRQ07	Content translation	As a content editor / administrator I want to translate content into one of the accepted languages so that it can be accessed into that language.
USSW05		Usage and editing statistics	As a content editor / administrator I want to consult statistical reports regarding the wiki component so that I can keep track of what had happened with the content.
USSW06		User administration	As an administrator I want to create / modify / delete profiles and users accounts so that I can manage the users' situation.
USSW07		User suspension	As an administrator I want to suspend a user account so that I am allowed to decide over users accounts.
USSW08		Ontology management	As an Administrator I want to create / modify / delete ontology so that I can manage the ontologies.
USSW09		Content organization	As an Administrator I want to organize the wiki's content so that I am allowed to change the structure and content position on wiki's pages.
USSW10	SWRQ01	Content approval	As an administrator I want to approve the created / edited content so that I am responsible for the changes on wiki's content.

Table 2: Overview of User Stories

To test functionalities against these user requirements, a set of Test Scenarios and Test Cases were created in standardized formats. Test Scenarios are broader descriptions of what needs to be tested, and what should be the end result of tests. Test Cases describe step by step how tests are to be carried out: what are the steps followed by the user and that how the system should respond. The main instruments used in directly in testing are Test Cases, while Test Scenarios serve as a container structure and an instrument against which the Test Cases can be checked.

[a] Explanations as to why these Test Cases are missing are provided in Section 4.1.1, below.

Table 3, below, presents an overview of test cases.

User Story	Test Cases
Search	Search for existing entry
	Search for keyword
	Search for non-existing keyword
Data export	Export wiki data into .rdf file
Content editing	Create content
	Edit content
	Publish content
Content translation	Content translation
Usage and editing statistics	View statistics
User administration	NONE[a]
User suspension	Suspension of a user account
Ontology Management	NONE[a]
Content organization	NONE[a]
Content approval	Approve content

[a] Explanations as to why these Test Cases are missing are provided in Section 4.1.1, below.

Table 3: Test Cases Overview

After internal testing, later tests were run in the piloting phase by external users (pilot city employees, stakeholders, citizens; see D5.1-D5.4). These tests used test cases based on the same test cases used for internal testing.

Tests were carried out by professional testers employed by the implementer of the component (SIVECO). This set of tests is the final set of tests executed at the end of the implementation period. Testers carried out the tests by following the test cases (developed at the beginning of the implementation period) and executing them step by step, and then reporting the results.

Table 4, below, displays the Test Cases and an overview of the results of tests.

User Story	Test Cases	Test Pass	Comment
Search	Search for existing entry	✓	
	Search for keyword	✓	
	Search for non-existing keyword	✓	
Data export	Export wiki data into .rdf file	✓	
Content editing	Create content	✓	
	Edit content	✓	
	Publish content	✓	
Content translation	Content translation	✓	
Usage and editing statistics	View statistics	✓	
User administration	NONE		Implementation moved under T4.5 (M16-M24).
User suspension	Suspension of a user account	✓	
Ontology Management	NONE		Redundant requirement
Content organization	NONE		Redundant requirement
Content approval	Approve content	✓	

Table 4: Overview of Test Results

For a total of 10 user requirements a number of 11 test cases were developed and tested. Of all tests executed all passed the tests.

For a few requirements/functionalities no test cases were developed nor tests executed. In the cases of Ontology Management and Content Approval these requirements – due to how the Semantic Mediawiki solution functionalities are organized –were condidered redundant with other functionalities. I.e. in implementing those other functionalities (particularly, Content editing, and Data Export) these functionalities were implemented as well.

With regard to User Administration, it was considered that this functionality overlaps with similar functionalities of the FLOOD-serv Portal. Since the overall System is planned to have a Single Sign On solution, then user administration should occur at the level of the integrated solution (in the Portal) and not at the level of the components. Consequently the implementation of this functionality was postponed until the integration process takes place

within Task 4.5 (M16-M24). Further on, this functionality was later tested and piloted as shown in D5.4 as part of the Portal/SSO Module.

4.1.1 Search: Test Scenario

TEST SCENARIO – SEMANTIC WIKI			
Scenario Code:	**TS-SW-SF-001**	**Version:**	**1.0**
Title:	**Search Functionality**		
Description:	**Search through the semantic wiki for terms or definitions in order to access specific information.**		
Preconditions:	User does not need to be authenticated into system		

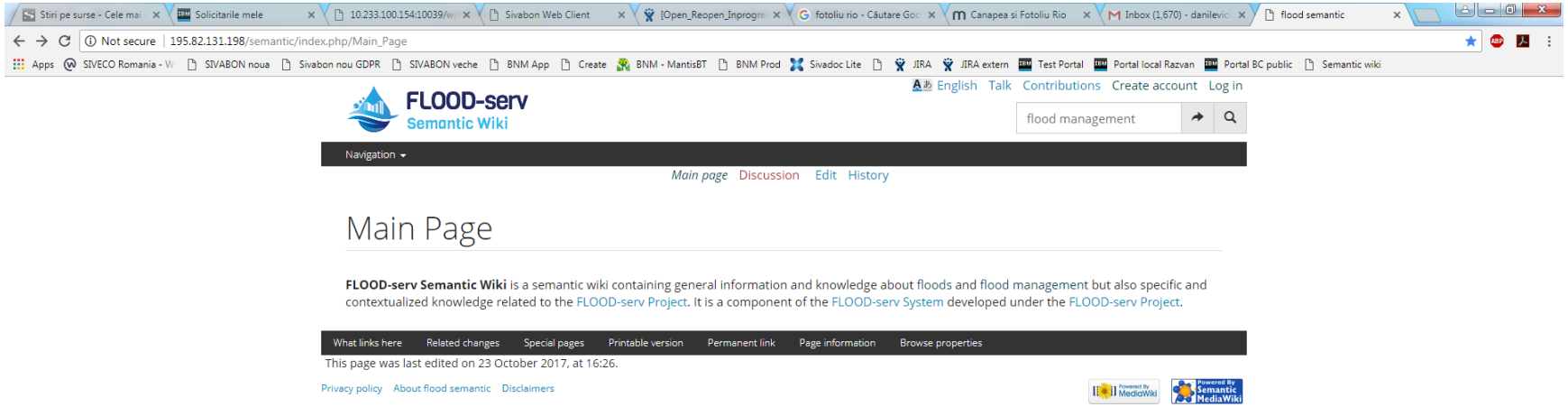
Test Case	Description	Results
TC-SW-SF-001	Search through the semantic wiki for a specific existing term/entry	The explanation for this term is shown
TC-SW-SF-002	Search through the semantic wiki for a specific keyword which exists in various articles/entries but is not an entry itself.	All content (list of entries) containing the specific keyword is shown.
TC-SW-SF-003	Search through the semantic wiki for a specific term which does not exist in the semantic wiki content.	No content for the specific term is shown

4.1.1.1 Test Case: Search for existing entry

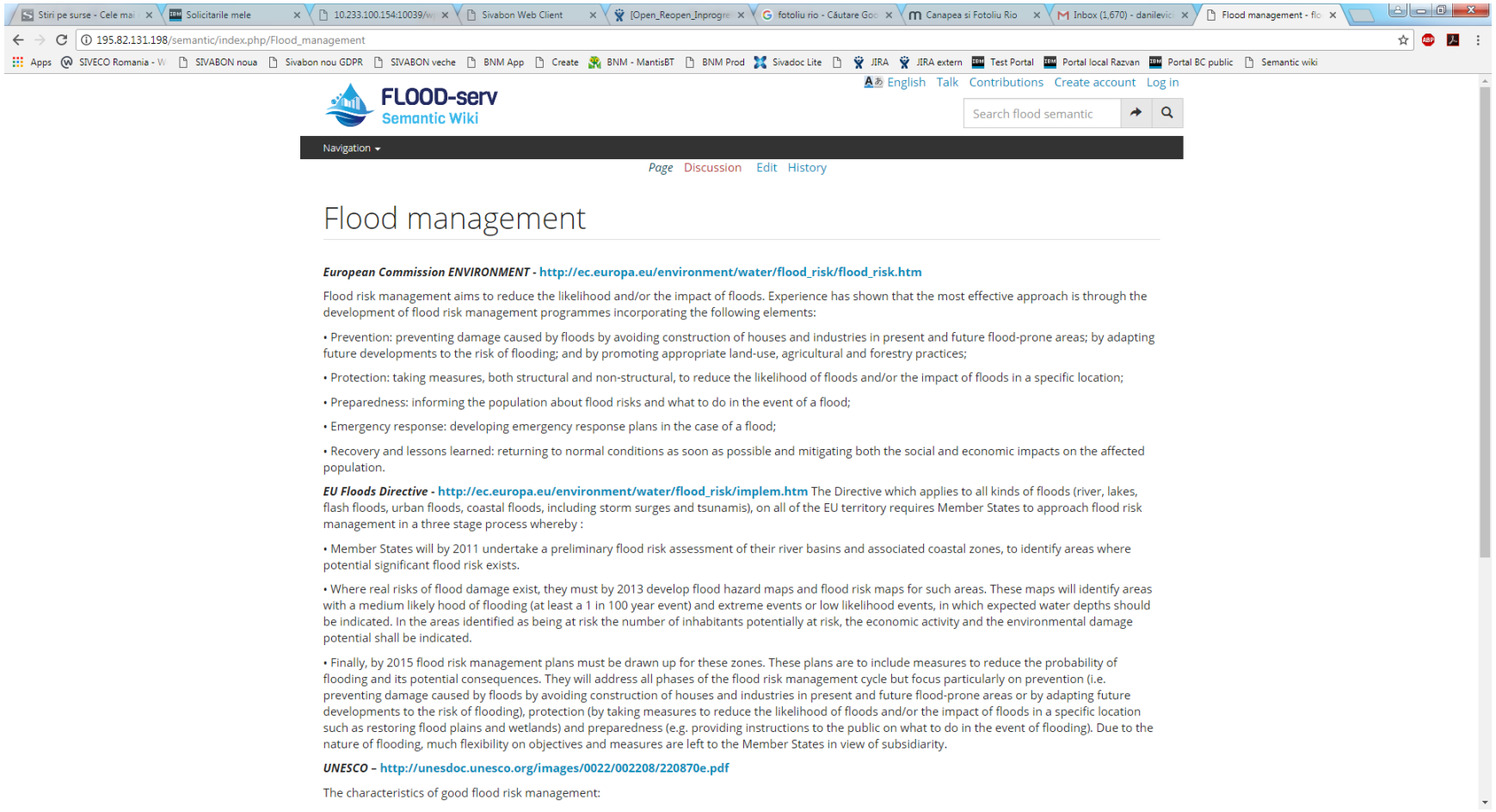
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-SF-001**	**Version:**	**1.0**
Title:	**Search for existing entry**	**Date:**	
Descriptions:	**Search through the semantic wiki for a specific existing term/entry**		
Preconditions:	User does not need to be autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Enter the term into the “Search flood semantic” box	-	
2	Use ENTER or Click the button	The content for the simple term is shown	

4.1.1.2 Test Results: Search for existing entry

Step 1



Step 2

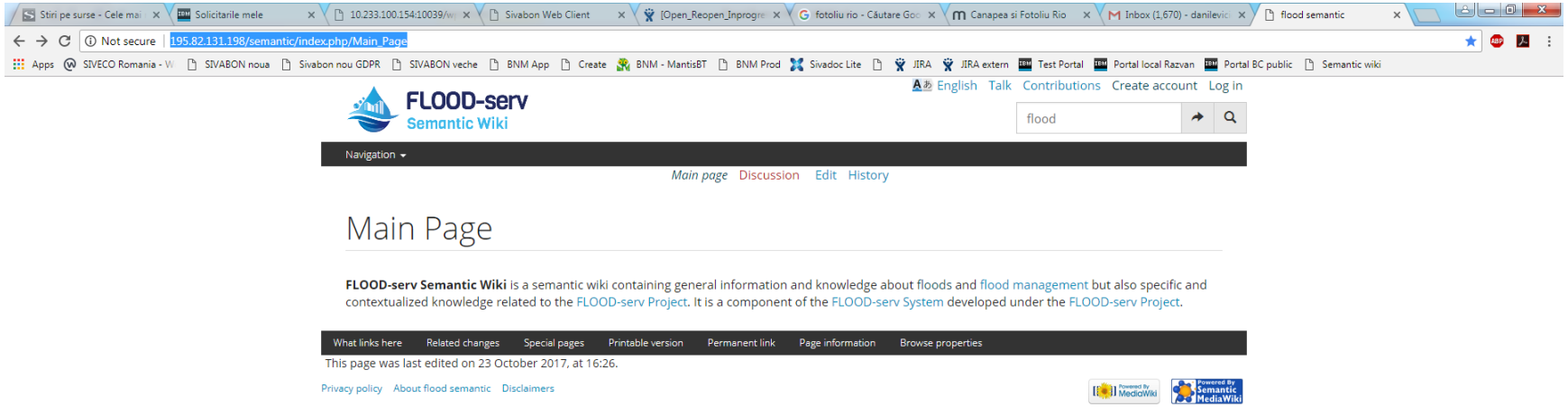


4.1.1.3 Test Case: Search for keyword

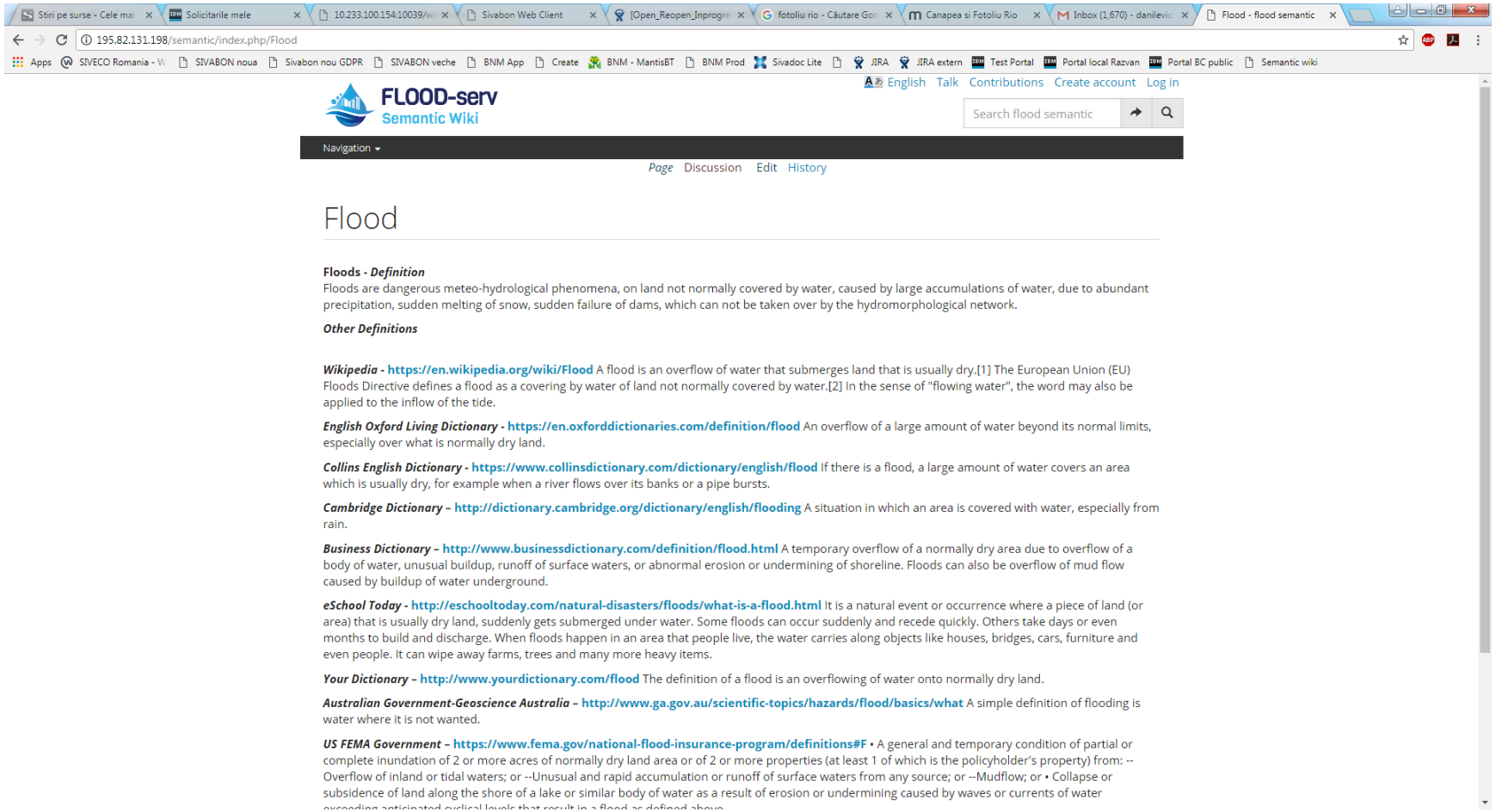
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-SF-002**	**Version:**	**1.0**
Title:	**Search for keyword**		
Descriptions:	**The case test describes the search through the semantic wiki for a specific keyword which exists in various articles/entries but is not an entry itself.**		
Preconditions:	User authenticated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Enter the term into the "Search flood semantic" box	-	
2	Use ENTER or Click the button	All content (list of entries) containing the specific keyword is shown.	

4.1.1.4 Test Results: Search for keyword

Step 1



Step 2

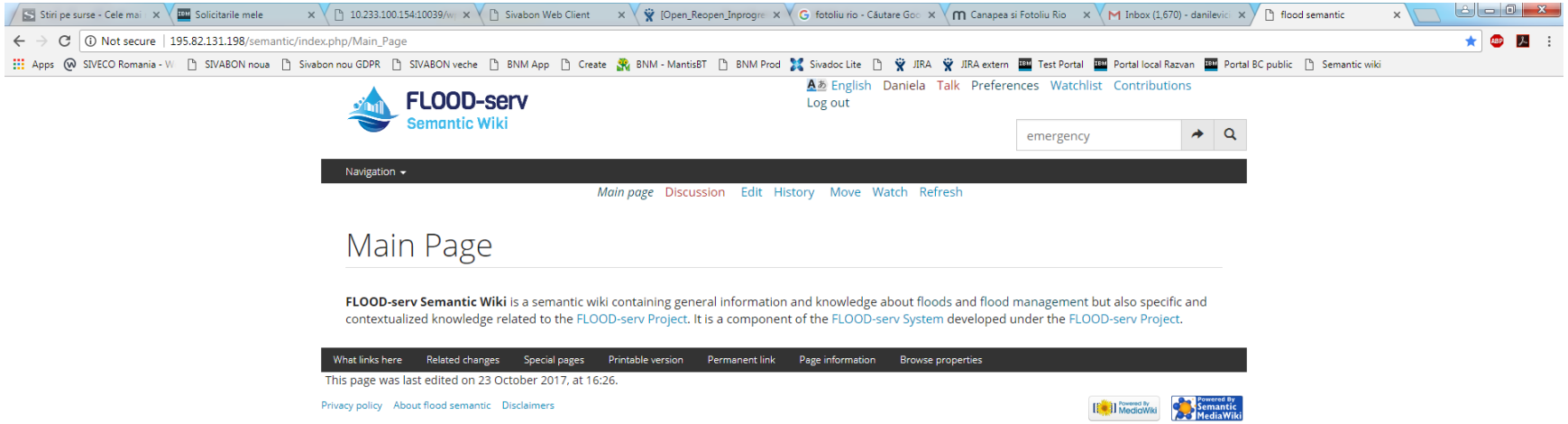


4.1.1.5 Test Case: Search for non-existing keyword

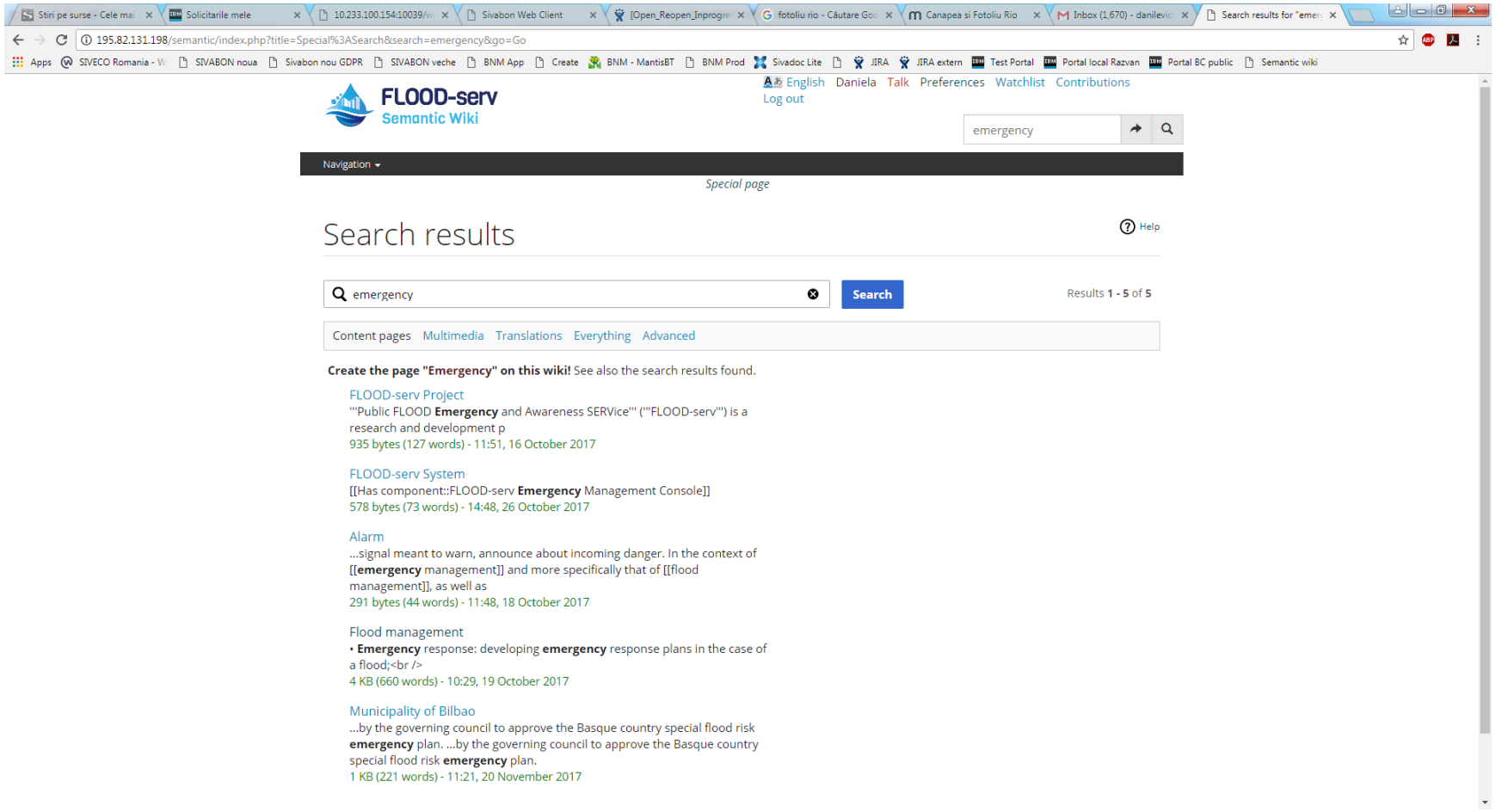
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-SF-003**	**Version:**	**1.0**
Title:	**Search for non-existing keyword**	**Date:**	
Descriptions:	**The case test describes the search through the semantic wiki for a specific term which does not exist in the semantic wiki content.**		
Preconditions:	User does not need to be authenticated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Enter the term into the “Search flood semantic” box	-	
2	Use ENTER or Click the button	No content for the simple term is shown	

4.1.1.6 Test Results: Search for non-existing keyword

Step 1



Step 2



4.2 Export content: Test Scenario

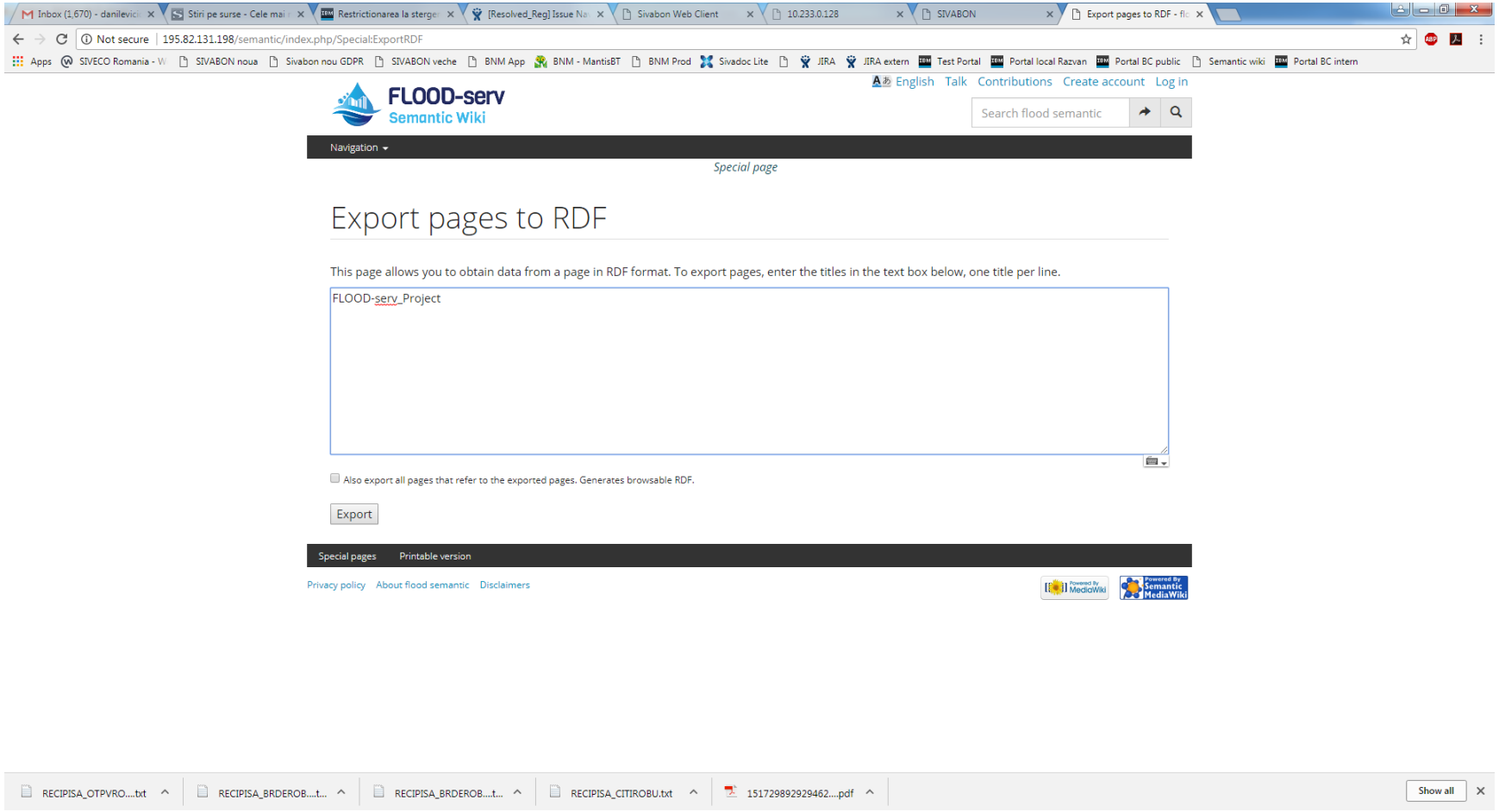
TEST SCENARIO – SEMANTIC WIKI			
Scenario Code:	**TS-SW-EX-001**	**Version:**	**1.0**
Title:	**Export wiki data**	**Date:**	
Description:	Export wiki data into file		
Preconditions:	User autenthicated into system		
Test Case	**Description**	**Results**	
TC-SW-EX-001	Export wiki data into .rdf file	The exported .rdf file contain data and can be used	

4.2.1.1 Test Case: Export wiki data into .rdf file

TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-EX-001**	**Version:**	**1.0**
Title:	**Export wiki data into .rdf file**	**Date:**	
Descriptions:	**The case test describes how to export the content from a wiki page to a .rdf file format**		
Preconditions:	User autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Go to the functionality of Export	As screen to introduce the name of page to be exported will be open	
2	Enter the page to be exported	No content for the simple term is shown	
3	Click Export	The content is exported into the required format	
4	Click Save as	Save the file on the disk	
5	Open the exported file	The appropriate content is in the file	

4.2.1.2 Test Results: Export data into rdf file

Steps 1 and 2



Step 3

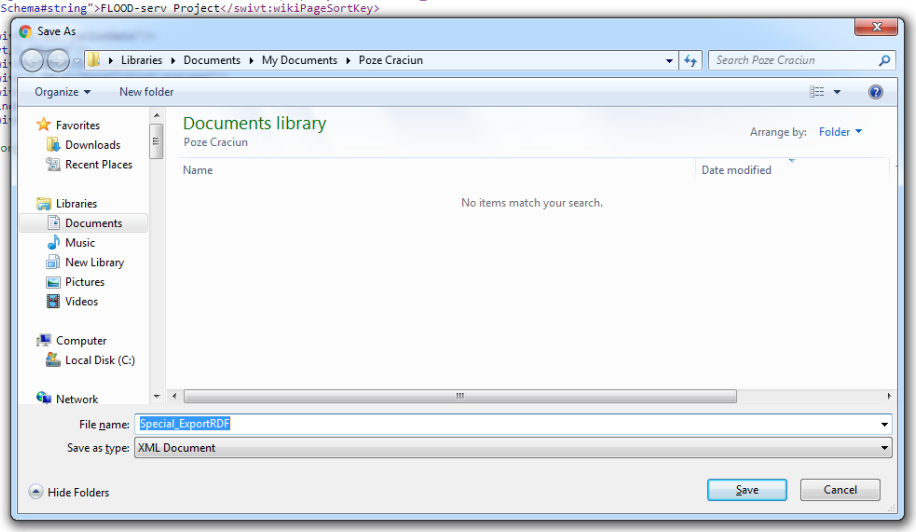
This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:swivt="http://semantic-mediawiki.org/swivt/1.0#"
 xmlns:wiki="http://195.82.131.198/semantic/index.php/Special:URIResolver/" xmlns:category="http://195.82.131.198/semantic/index.php/Special:URIResolver/Category-3A" xmlns:property="http://195.82.131.198/semantic/index.php/Special:URIResolver/Property-3A">
 ▼<owl:Ontology rdf:about="http://195.82.131.198/semantic/index.php/Special:ExportRDF/FLOOD-2Dserv_Project">
    <swivt:creationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2018-01-30T13:05:31+02:00</swivt:creationDate>
    <owl:imports rdf:resource="http://semantic-mediawiki.org/swivt/1.0"/>
  </owl:Ontology>
 ▼<swivt:Subject rdf:about="http://195.82.131.198/semantic/index.php/Special:URIResolver/FLOOD-2Dserv_Project">
    <rdfs:label>FLOOD-serv Project</rdfs:label>
    <rdfs:isDefinedBy rdf:resource="http://195.82.131.198/semantic/index.php/Special:ExportRDF/FLOOD-2Dserv_Project"/>
    <swivt:page rdf:resource="http://195.82.131.198/semantic/index.php/FLOOD-2Dserv_Project"/>
    <swivt:wikiNamespace rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">0</swivt:wikiNamespace>
    <swivt:wikiPageContentLanguage rdf:datatype="http://www.w3.org/2001/XMLSchema#string">en</swivt:wikiPageContentLanguage>
    <swivt:wikiPageModificationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2017-10-16T09:51:11Z</swivt:wikiPageModificationDate>
    <property:Modification_date-23aux rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2458042.910544</property:Modification_date-23aux>
    <swivt:wikiPageSortKey rdf:datatype="http://www.w3.org/2001/XMLSchema#string">FLOOD-serv Project</swivt:wikiPageSortKey>
  </swivt:Subject>
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#creationDate"/>
  <owl:ObjectProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#page"/>
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiNamespace"/>
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiPageContentLanguage"/>
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiPageModificationDate"/>
  <owl:DatatypeProperty rdf:about="http://195.82.131.198/semantic/index.php/Special:URIResolver/Property-3AModification_date-23aux"/>
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiPageSortKey"/>
 ▼<!--
     Created by Semantic MediaWiki, https://www.semantic-mediawiki.org/
  -->
</rdf:RDF>
```

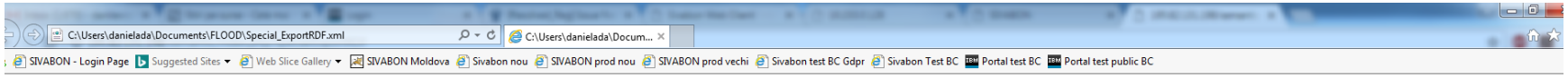
Step 4

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:swivt="http://semantic-mediawiki.org/swivt/1.0#"
xmlns:wiki="http://195.82.131.198/semantic/index.php/Special:URIResolver/" xmlns:category="http://195.82.131.198/semantic/index.php/Special:URIResolver/Category-3A" xmlns:property="http://195.82.131.198/semantic/index.php/Special:URIResolver/Property-3A">
▼<owl:Ontology rdf:about="http://195.82.131.198/semantic/index.php/Special:ExportRDF/FLOOD-2Dserv_Project">
    <swivt:creationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2018-01-30T13:05:31+02:00</swivt:creationDate>
    <owl:imports rdf:resource="http://semantic-mediawiki.org/swivt/1.0"/>
  </owl:Ontology>
▼<swivt:Subject rdf:about="http://195.82.131.198/semantic/index.php/Special:URIResolver/FLOOD-2Dserv_Project">
    <rdfs:label>FLOOD-serv Project</rdfs:label>
    <rdfs:isDefinedBy rdf:resource="http://195.82.131.198/semantic/index.php/Special:ExportRDF/FLOOD-2Dserv_Project"/>
    <swivt:page rdf:resource="http://195.82.131.198/semantic/index.php/FLOOD-2Dserv_Project"/>
    <swivt:wikiNamespace rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">0</swivt:wikiNamespace>
    <swivt:wikiPageContentLanguage rdf:datatype="http://www.w3.org/2001/XMLSchema#string">en</swivt:wikiPageContentLanguage>
    <swivt:wikiPageModificationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2017-10-16T09:51:11Z</swivt:wikiPageModificationDate>
    <property:Modification_date-23aux rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2458042.910544</property:Modification_date-23aux>
    <swivt:wikiPageSortKey rdf:datatype="http://www.w3.org/2001/XMLSchema#string">FLOOD-serv Project</swivt:wikiPageSortKey>
  </swivt:Subject>
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swi
  <owl:ObjectProperty rdf:about="http://semantic-mediawiki.org/swivt
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swi
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swi
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swi
  <owl:DatatypeProperty rdf:about="http://195.82.131.198/semantic/in
  <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swi
▼<!--
     Created by Semantic MediaWiki, https://www.semantic-mediawiki.o
  -->
</rdf:RDF>
```

Step 5




```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE RDF>
<rdf:RDF xmlns:property="http://195.82.131.198/semantic/index.php/Special:URIResolver/Property-3A" xmlns:category="http://195.82.131.198/semantic/index.php/Special:URIResolver/Category-3A"
xmlns:wiki="http://195.82.131.198/semantic/index.php/Special:URIResolver/" xmlns:swivt="http://semantic-mediawiki.org/swivt/1.0#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  - <owl:Ontology rdf:about="http://195.82.131.198/semantic/index.php/Special:ExportRDF/FLOOD-2Dserv_Project">
        <swivt:creationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2018-01-30T13:05:31+02:00</swivt:creationDate>
        <owl:imports rdf:resource="http://semantic-mediawiki.org/swivt/1.0"/>
    </owl:Ontology>
  - <swivt:Subject rdf:about="http://195.82.131.198/semantic/index.php/Special:URIResolver/FLOOD-2Dserv_Project">
        <rdfs:label>FLOOD-serv Project</rdfs:label>
        <rdfs:isDefinedBy rdf:resource="http://195.82.131.198/semantic/index.php/Special:ExportRDF/FLOOD-2Dserv_Project"/>
        <swivt:page rdf:resource="http://195.82.131.198/semantic/index.php/FLOOD-2Dserv_Project"/>
        <swivt:wikiNamespace rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">0</swivt:wikiNamespace>
        <swivt:wikiPageContentLanguage rdf:datatype="http://www.w3.org/2001/XMLSchema#string">en</swivt:wikiPageContentLanguage>
        <swivt:wikiPageModificationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2017-10-16T09:51:11Z</swivt:wikiPageModificationDate>
        <property:Modification_date-23aux rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2458042.910544</property:Modification_date-23aux>
        <swivt:wikiPageSortKey rdf:datatype="http://www.w3.org/2001/XMLSchema#string">FLOOD-serv Project</swivt:wikiPageSortKey>
    </swivt:Subject>
    <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#creationDate"/>
    <owl:ObjectProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#page"/>
    <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiNamespace"/>
    <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiPageContentLanguage"/>
    <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiPageModificationDate"/>
    <owl:DatatypeProperty rdf:about="http://195.82.131.198/semantic/index.php/Special:URIResolver/Property-3AModification_date-23aux"/>
    <owl:DatatypeProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#wikiPageSortKey"/>
        <!-- Created by Semantic MediaWiki, https://www.semantic-mediawiki.org/ -->
</rdf:RDF>
```

4.3 Content editing: Test Scenario

TEST SCENARIO – SEMANTIC WIKI			
Scenario Code:	**TS-SW-MC-001**	**Version:**	**1.0**
Title:	**Content editing**	**Date:**	
Description:	Create content / edit / publish content in order to improve the terms, definitions and vocabularies.		
Preconditions:	User autenthicated into system		

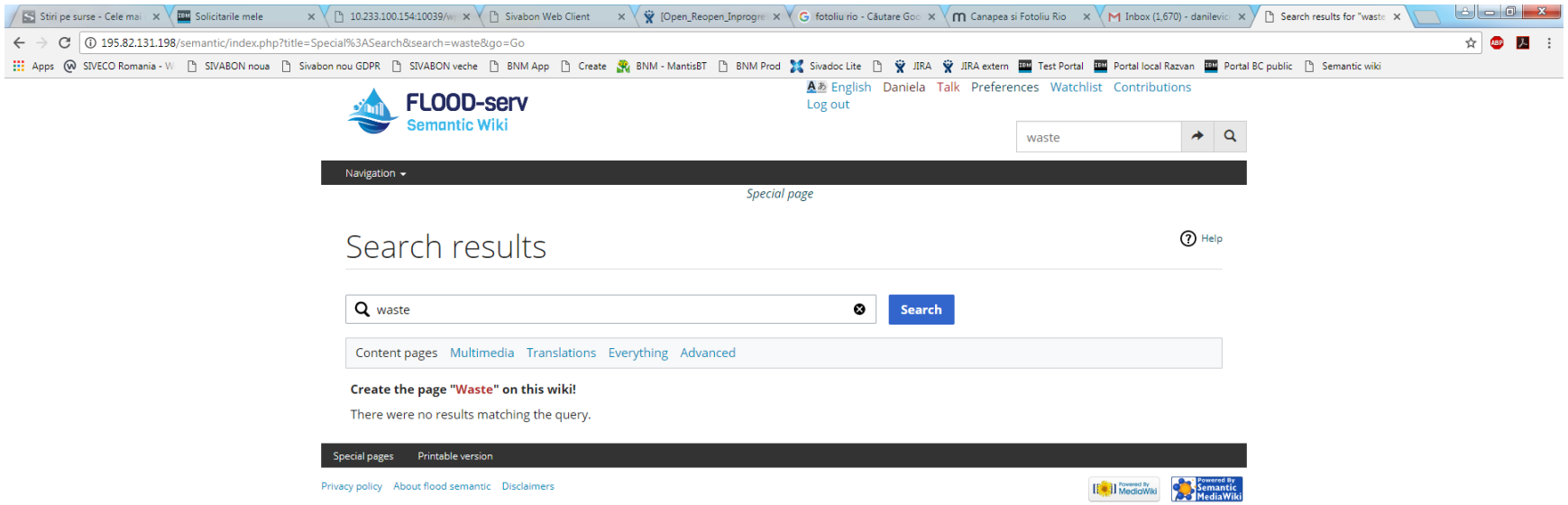
Test Case	**Description**	**Results**
TC-SW-MC-001	Create content	Specific area for content creation is enabled and the user can add data
TC-SW-MC-002	Edit content	Specific area is enabled for editing and the user can perform any change
TC-SW-MC-003	Publish content	The content is published in the semantic wiki
TC-SW-MC-004	Content visualisation by the administrator	The content created / edited / published by the content editor can be visualized by the administrator

4.3.1.1 Test Case: Create content

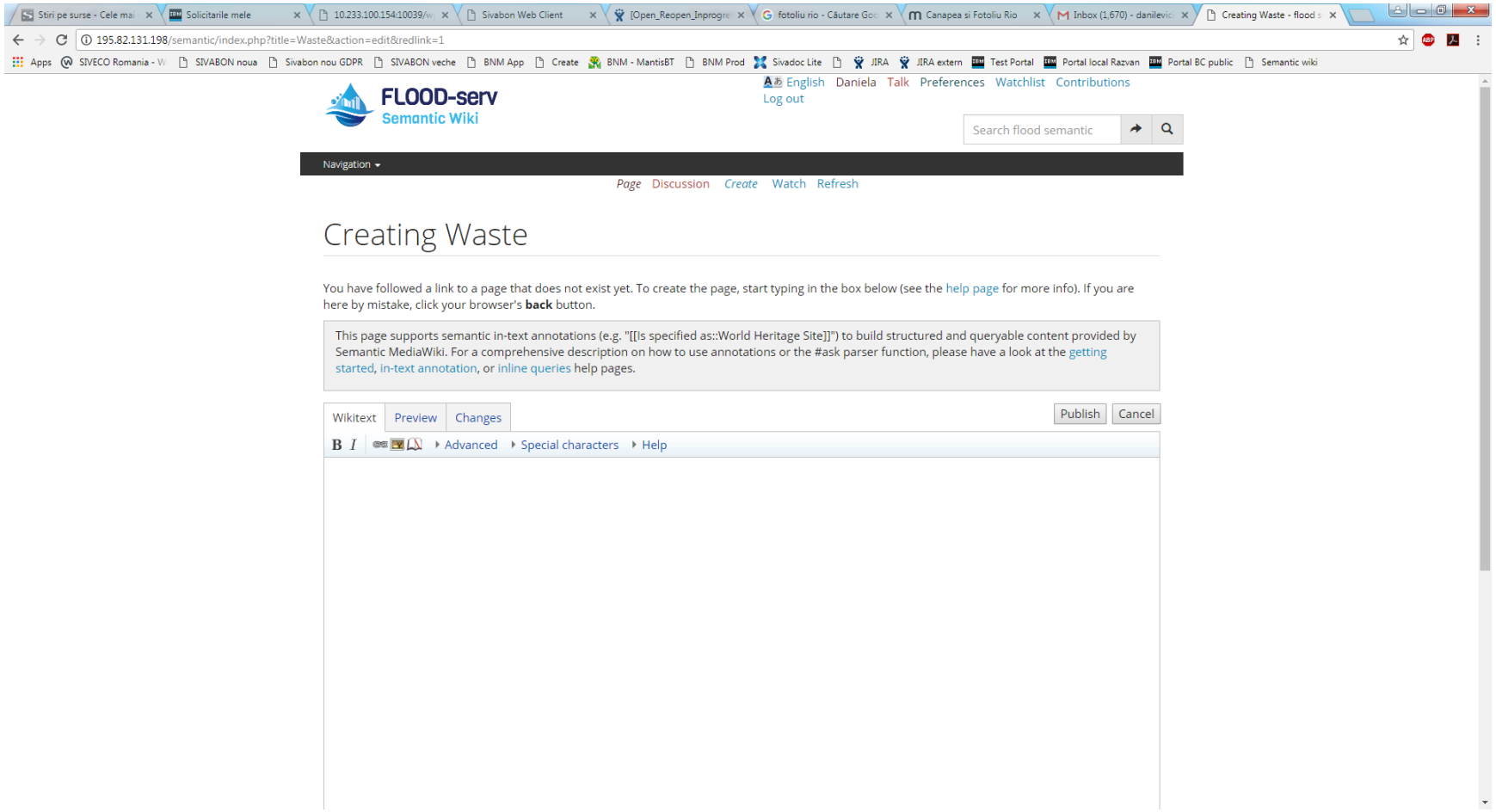
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-MC-001**	**Version:**	**1.0**
Title:	**Create content**	**Date:**	
Descriptions:	**The case test describes creation of content for a term which does not exists into the ontologies**		
Preconditions:	User autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Enter the term into the “Search flood semantic” box	-	
2	Use ENTER or Click the button	No content for the simple term is shown	
3	Click link from *Create the page „….” on this wiki*	Specific area for content creation is enabled and the user can add data	
4	Enter the specific data	-	
3	Click *Save page*	The created content is saved	

4.3.1.2 Test Results: Create content

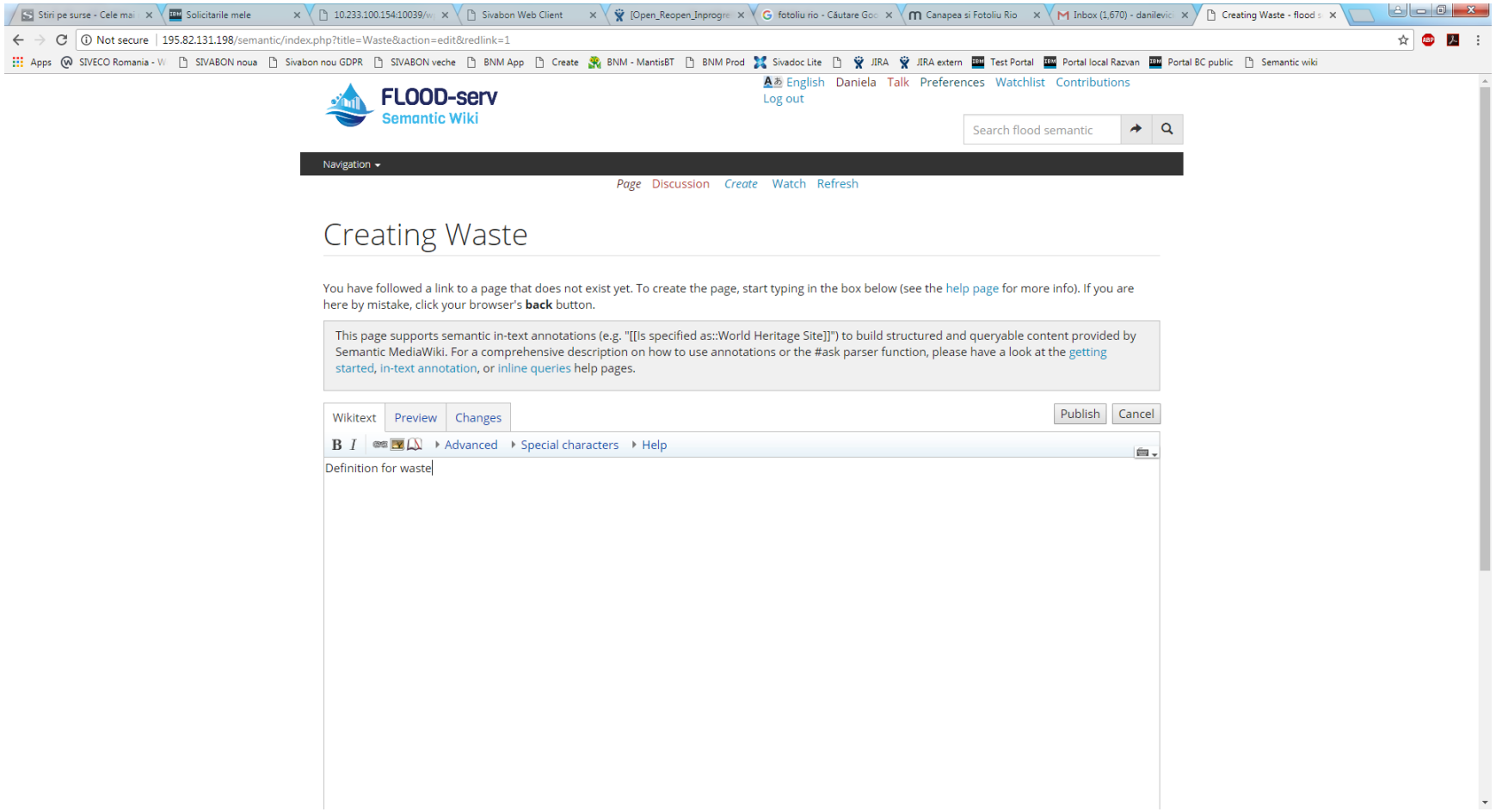
Steps 1 & 2



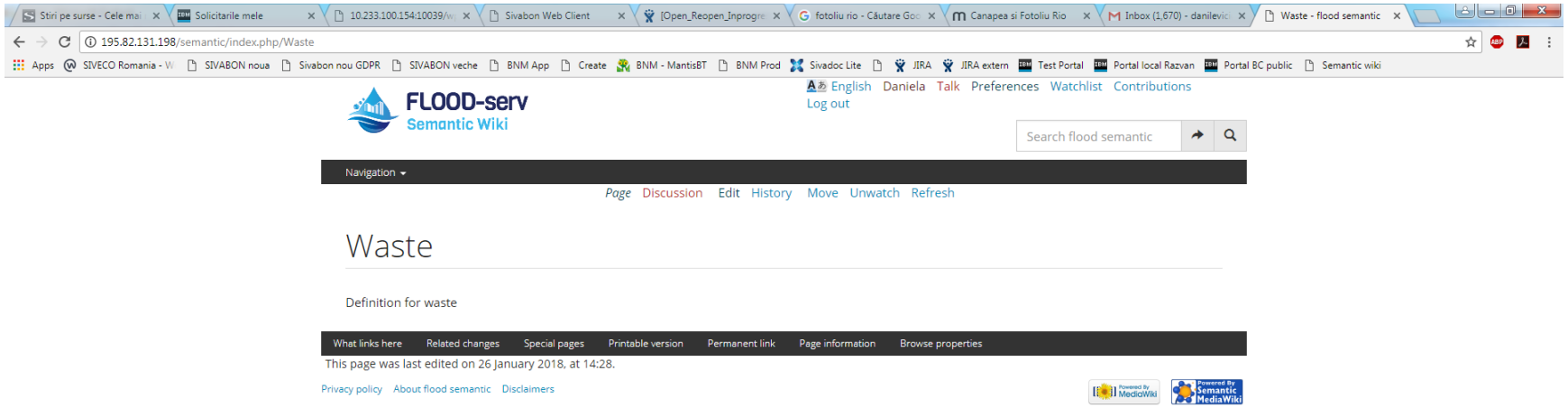
Step 3



Step 4



Step 5

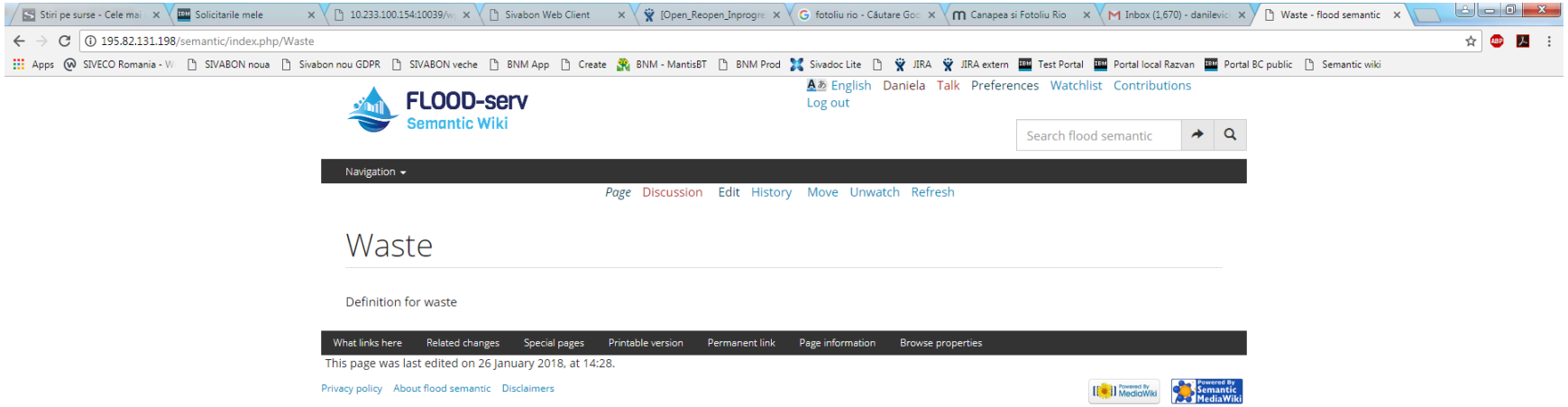


4.3.1.3 Test Case: Edit content

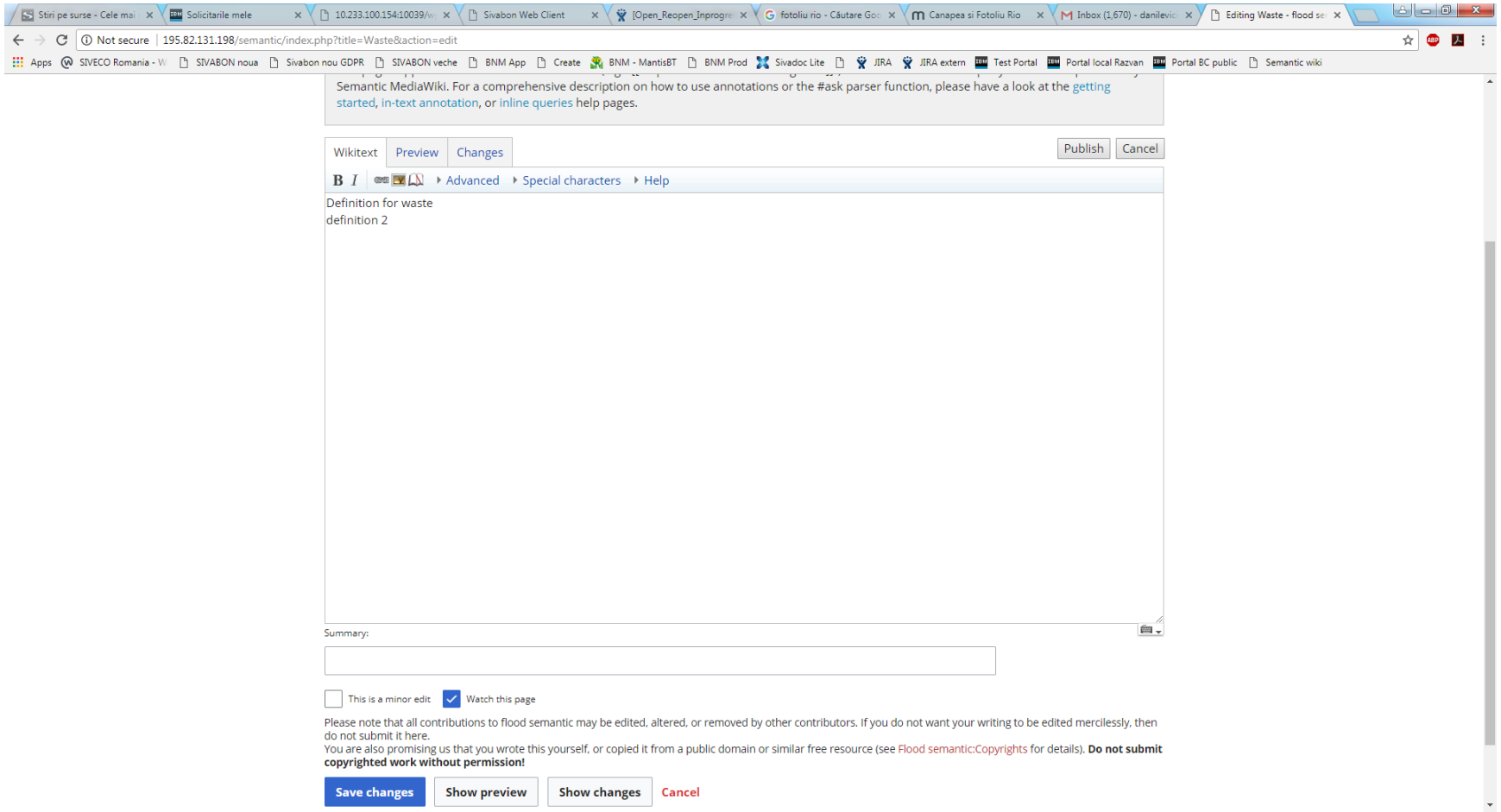
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-MC-002**	**Version:**	**1.0**
Title:	**Edit content**	**Date:**	
Descriptions:	**The case test describes how to edit the content for a term from the ontologies**		
Preconditions:	User autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Enter the term into the “Search flood semantic” box	-	
2	Use ENTER or Click the button	The content for the term is shown	
3	Click *Edit*	Specific area is enabled for editing and the user can perform any change	
4	Enter the changes	-	
3	Click *Save change*	The modified content is saved	

4.3.1.4 Test Results: Edit content

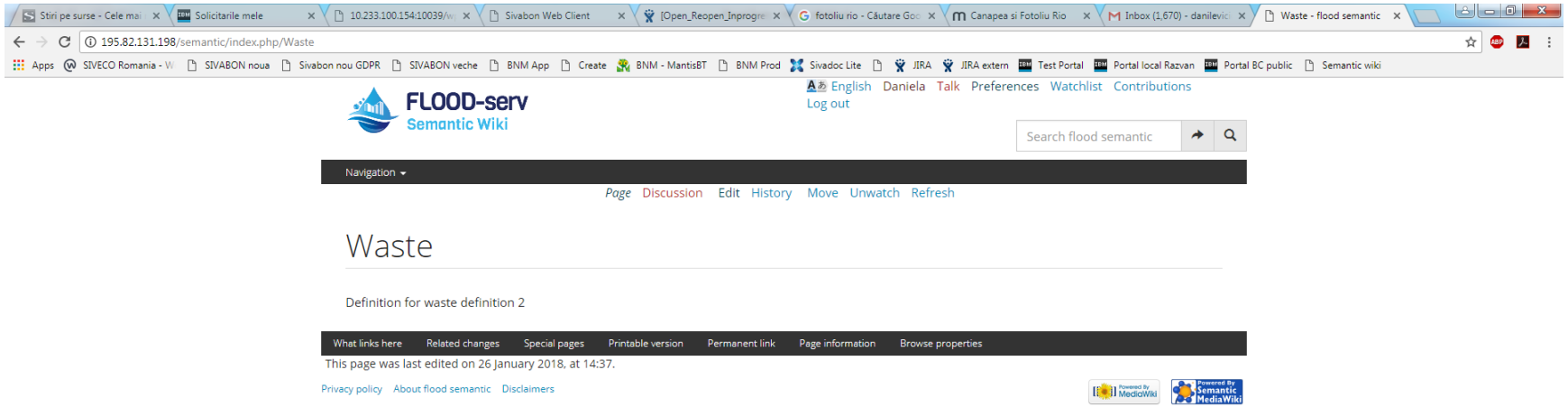
Step 2



Step 3



Step 5

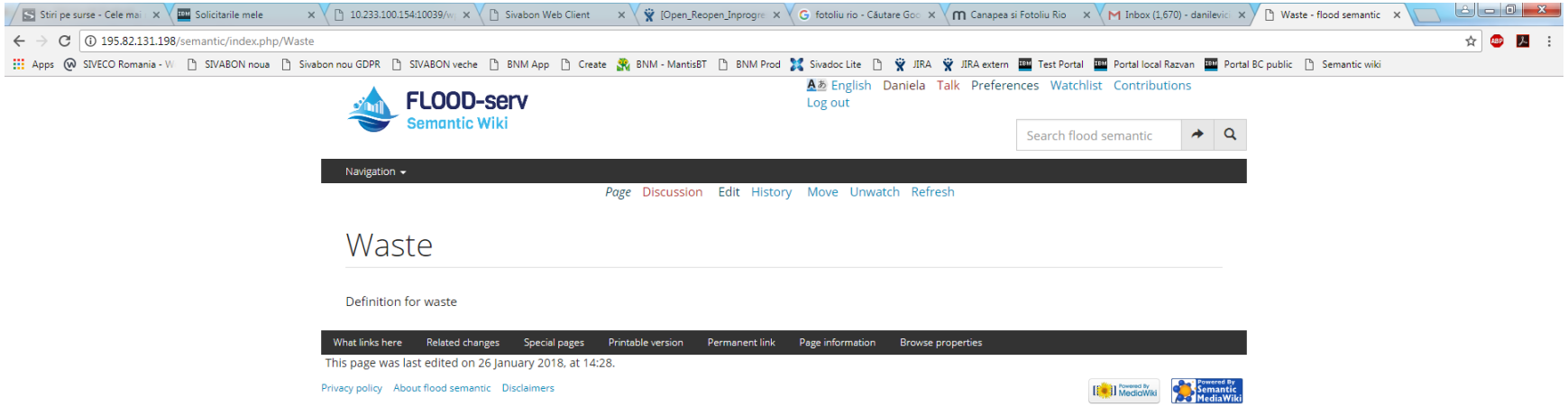


4.3.1.5 Test Case: Publish content

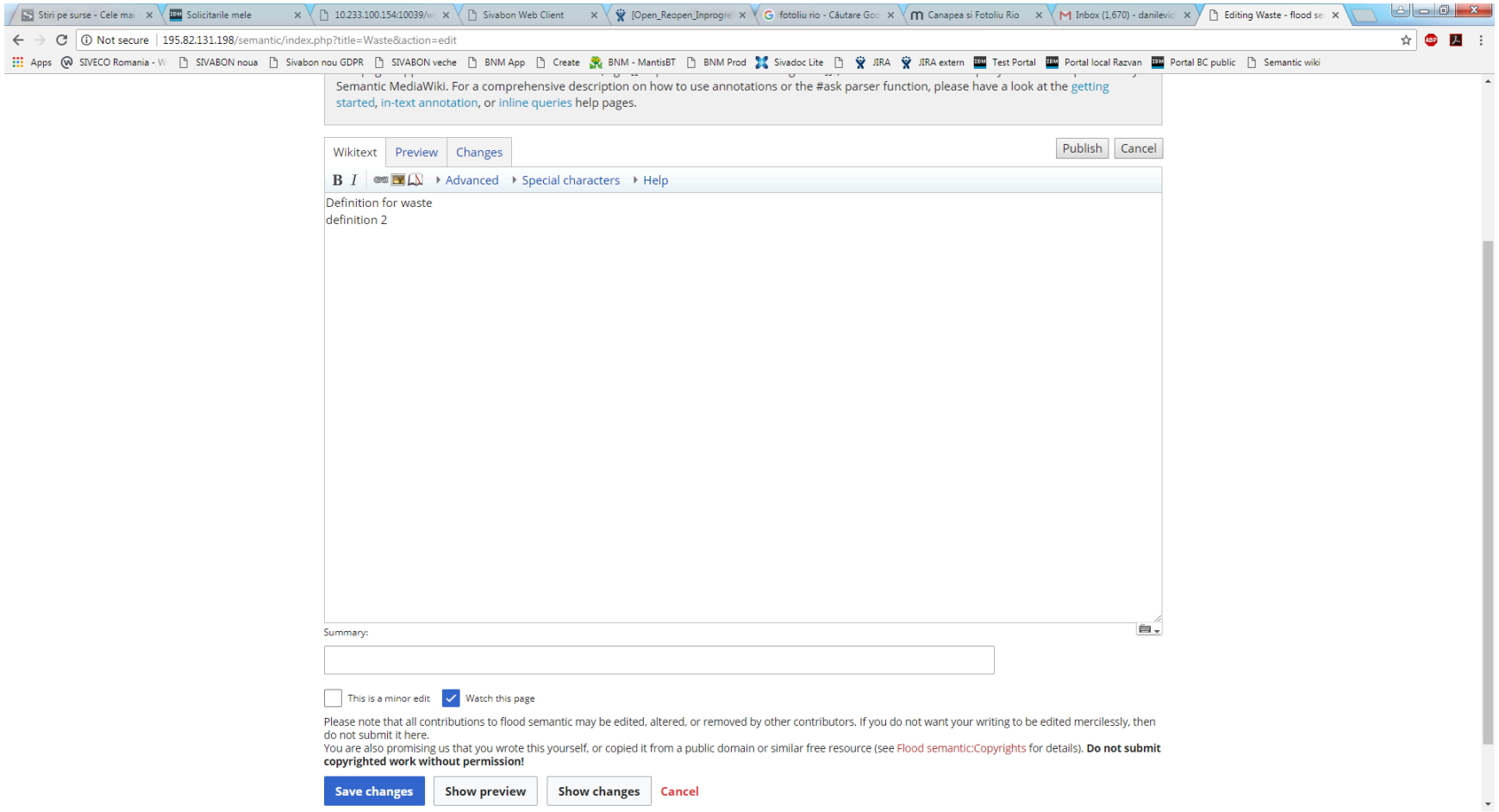
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-MC-003**	**Version:**	**1.0**
Title:	**Publish content**	**Date:**	
Descriptions:	**The case test describes how to publish the content for a term from the ontologies**		
Preconditions:	User autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Enter the term into the “Search flood semantic” box	-	
2	Use ENTER or Click the button	The content for the term is shown	
3	Click *Edit*	Specific area is enabled for editing and the user can publish the content	
4	Click Publish	-	
5	Complete summary	-	
6	Click Publish		

4.3.1.6 Test Results: Publish content

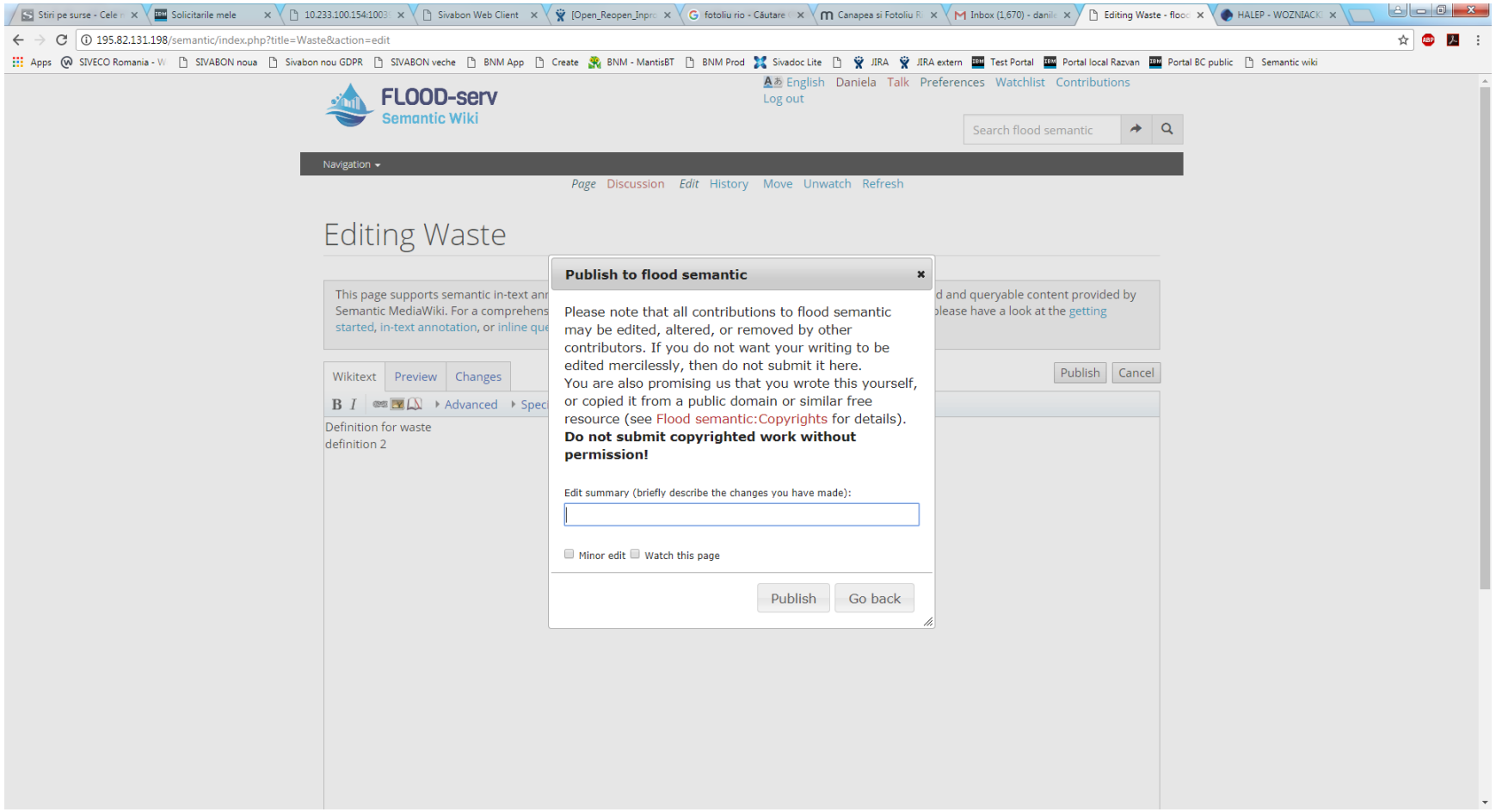
Step 2



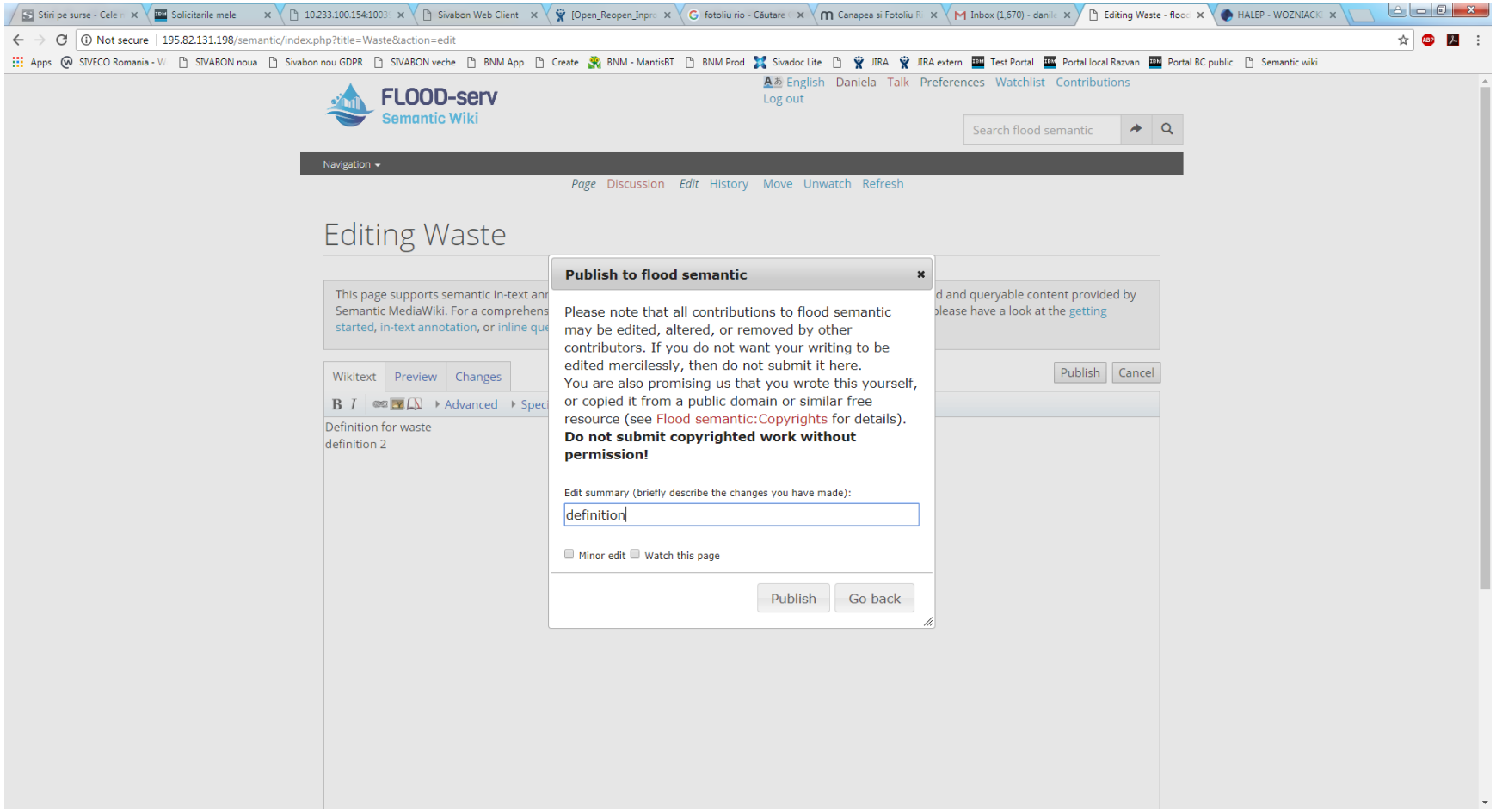
Step 3



Step 4



Step 5



4.4 Content translation: Test Scenario

TEST SCENARIO – SEMANTIC WIKI			
Scenario Code:	**TS-SW-CT-001**	**Version:**	**1.0**
Title:	**Content translation**	**Date:**	
Description:	The test scenario allows the admin to translate the content in one of the accepted languages		
Preconditions:	User admin autenthicated into system		

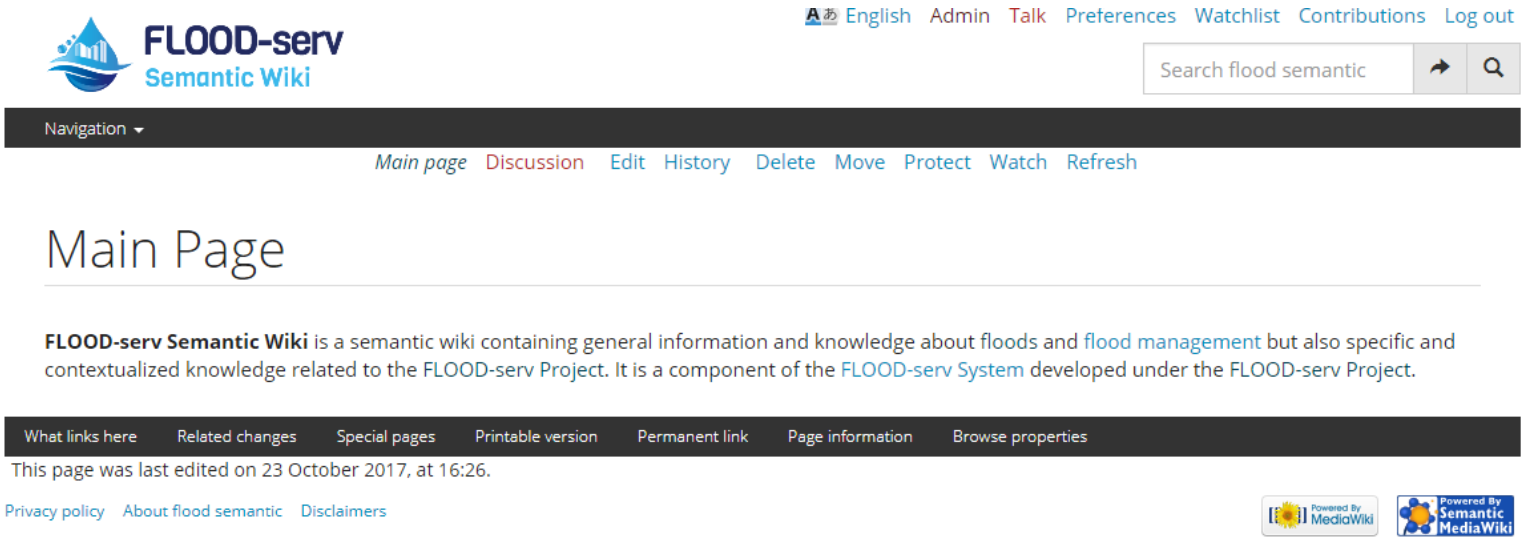
Test Case	**Description**	**Results**
TC-SW-CT-001	Content translation	The content is translated in one of the accepted languages

4.4.1.1 Test Case: Content translation

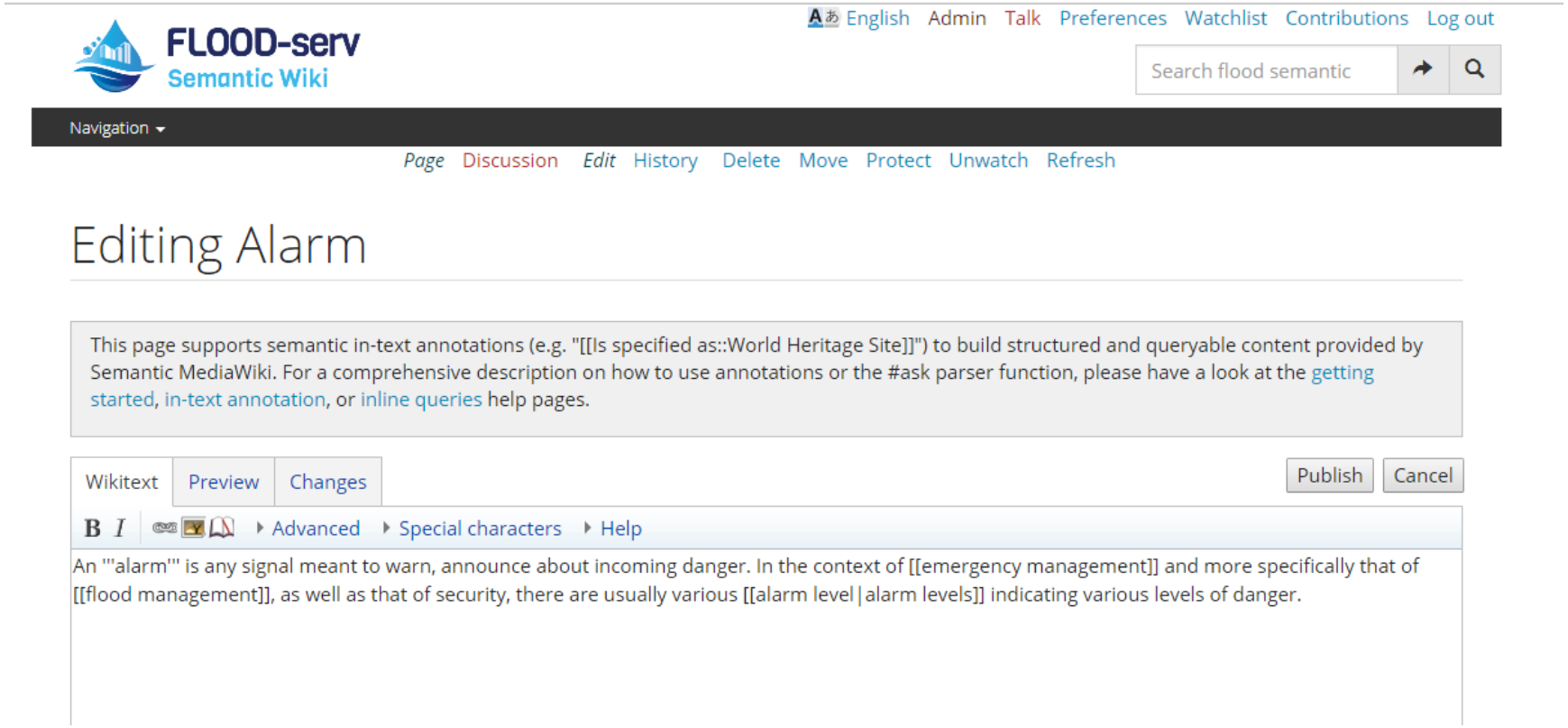
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-CT-001**	**Version:**	**1.0**
Title:	**Content translation**	**Date:**	
Descriptions:	**The case test describes how to translate the content**		
Preconditions:	User admin autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Enter a specific term and click Search	Find the definition for this term	
2	Click Edit	A screen with the possibility of editing. The content is open.	
3	Mark text for translation (add tag <transalte>….content… </translate>)	-	
4	Click Publish	The page is redirected to the main page	
5	Click Translate the page	The page with the translation tools is open	
6	Choose the Language for translation	-	
7	Click the „edit” icon for editing the title/definition translation	A screen for entering/modifying the text in choosen language is open. The text can be added.	
8	Click Save translation	The translation is saved.	

4.4.1.2 Test Results: Content translation

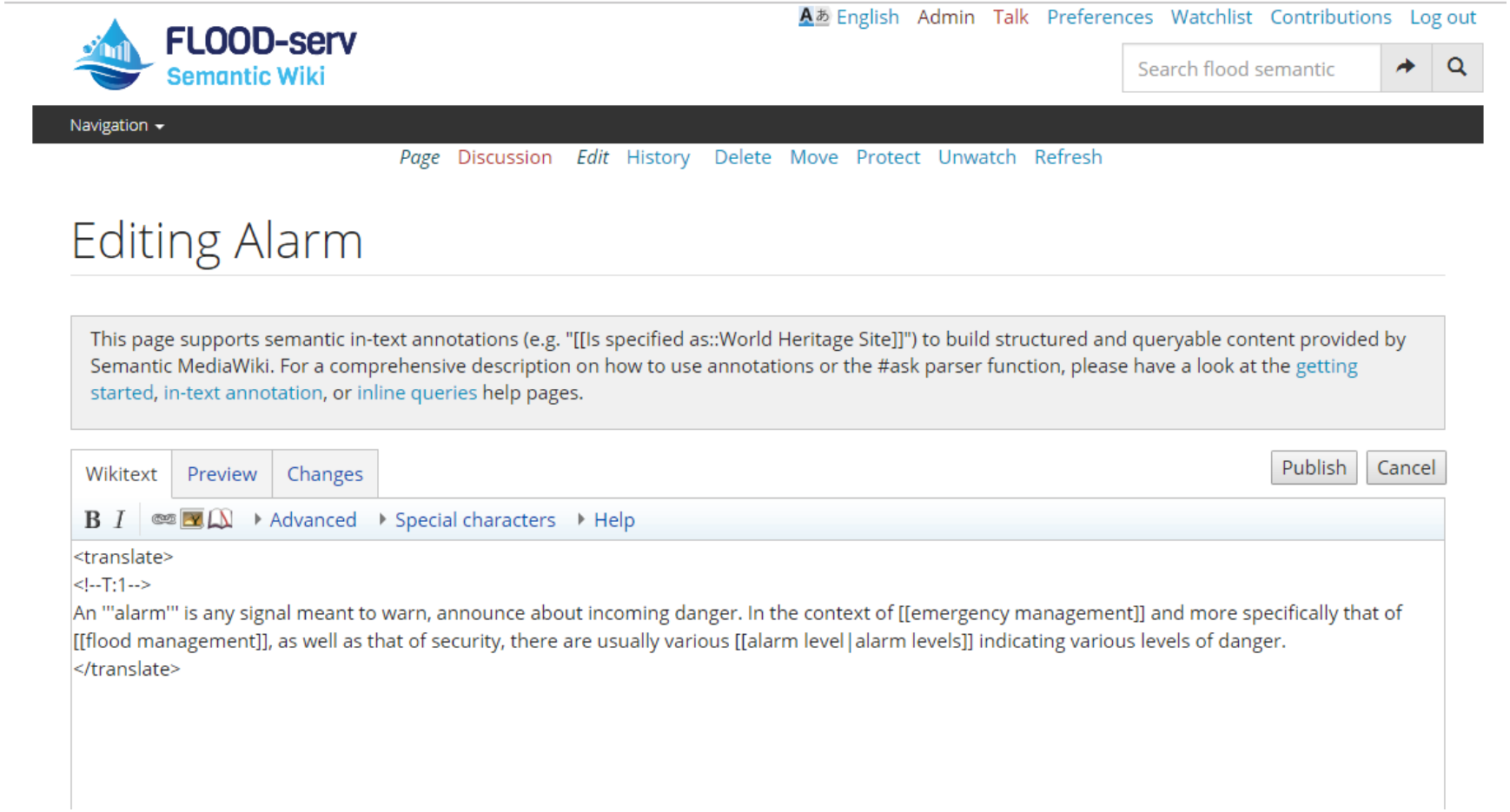
Step 1



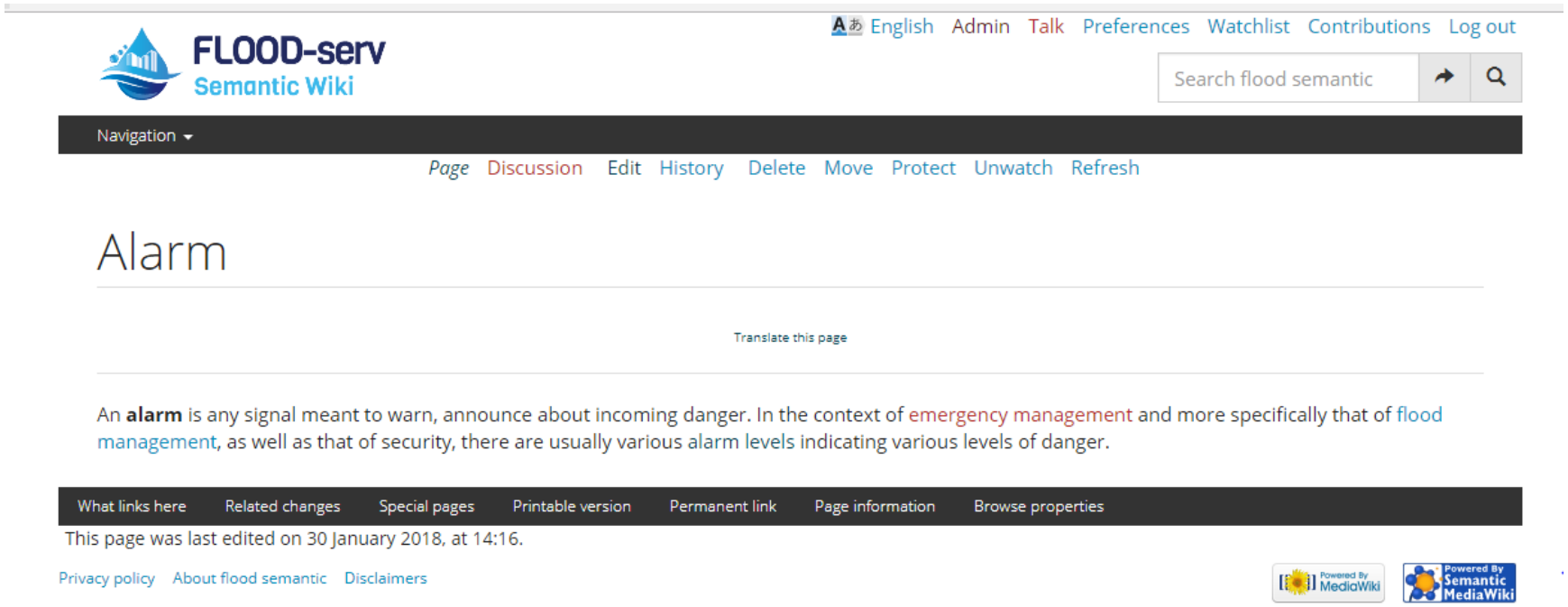
Step 2



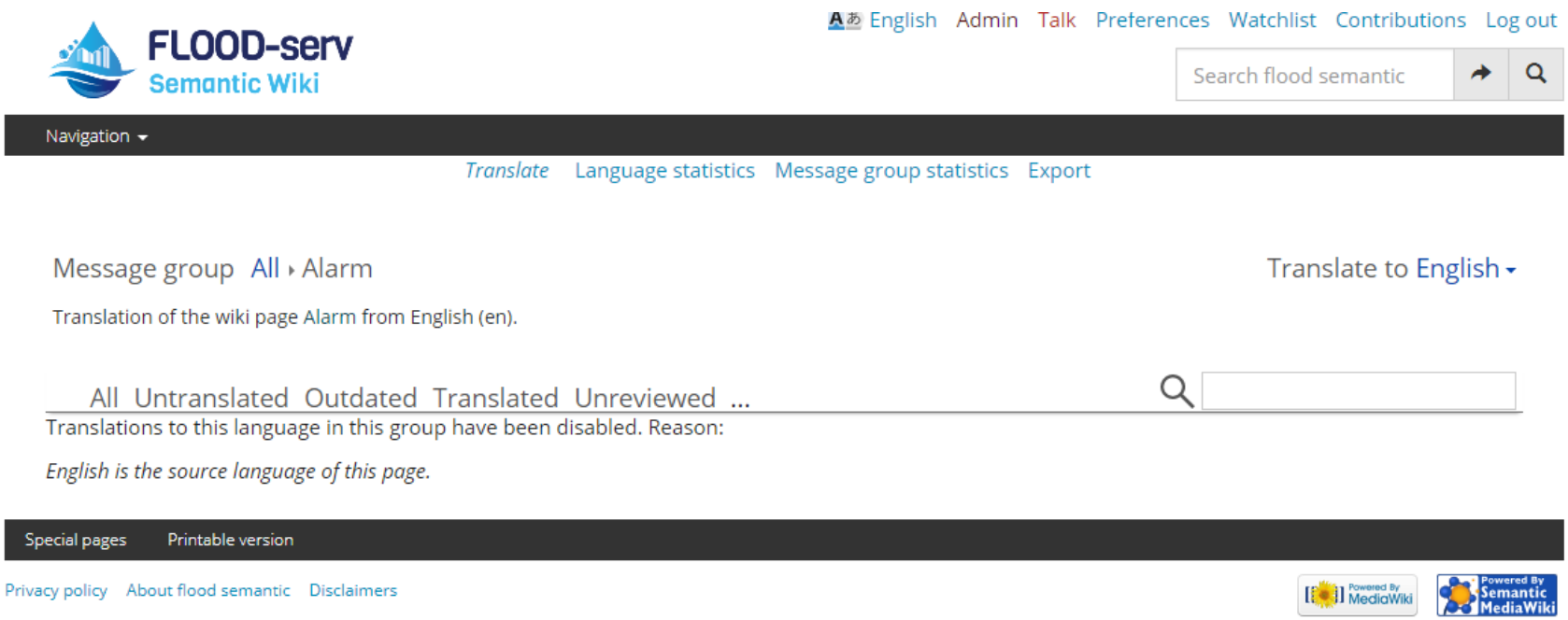
Step 3



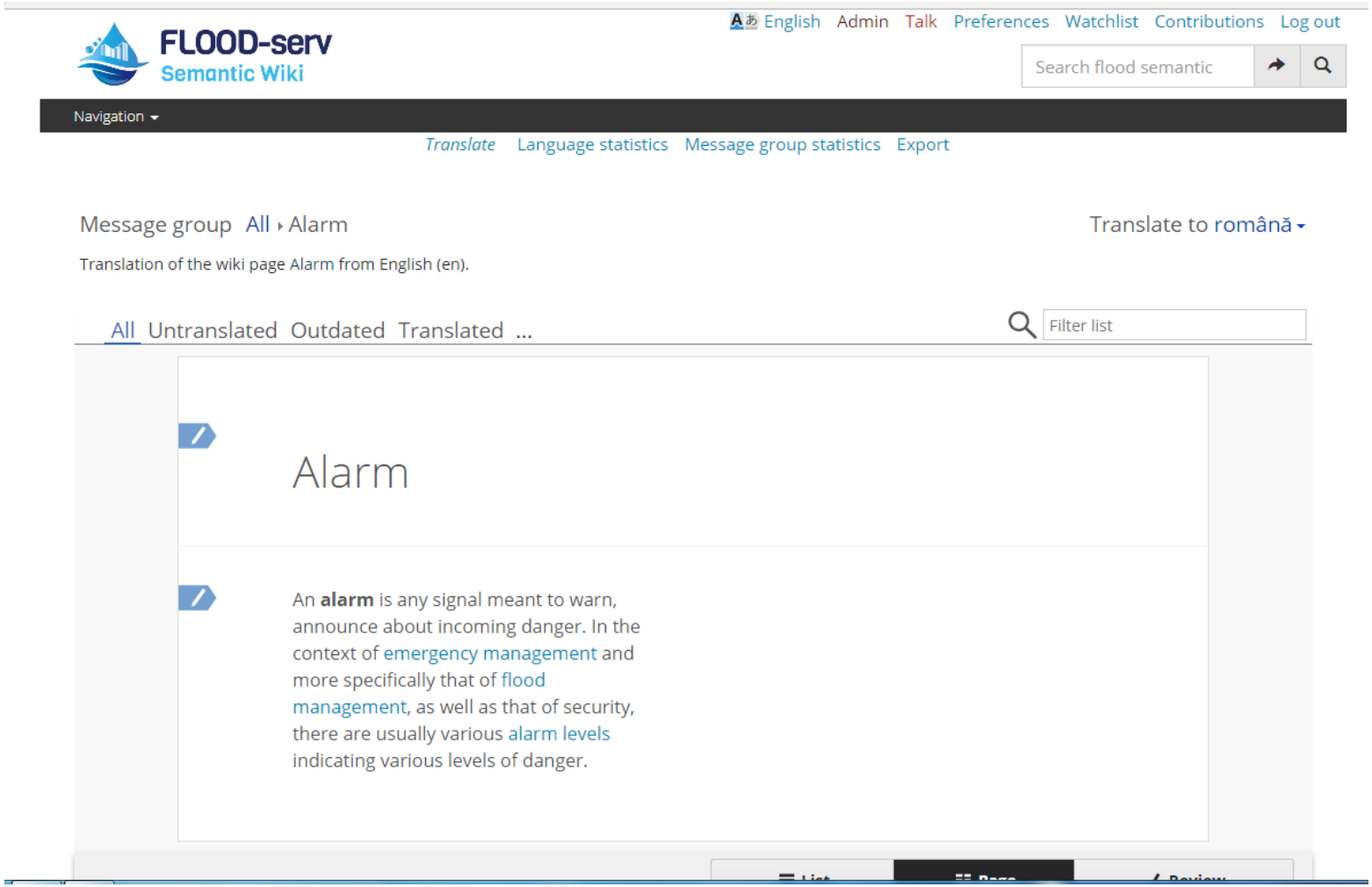
Step 4



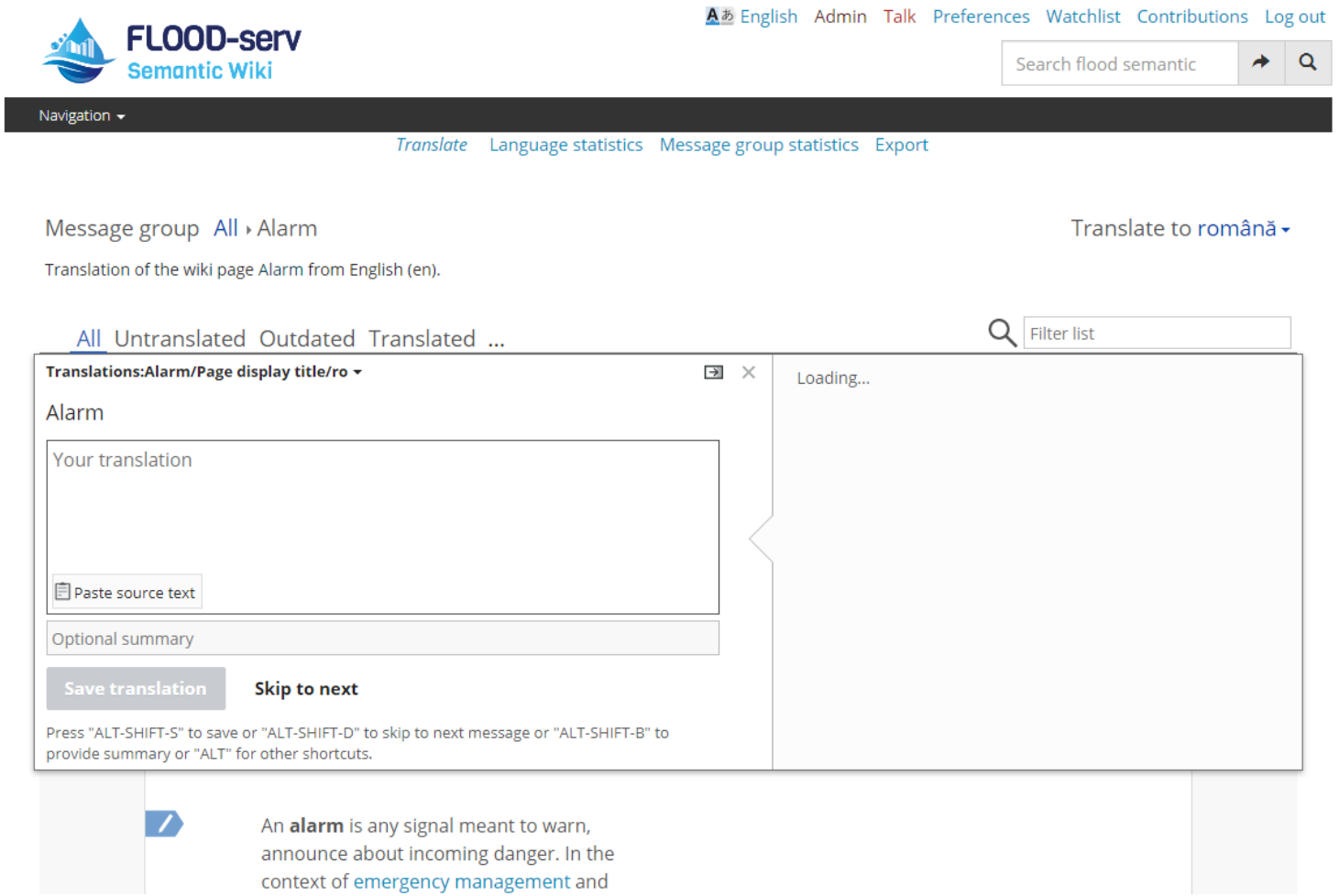
Step 5



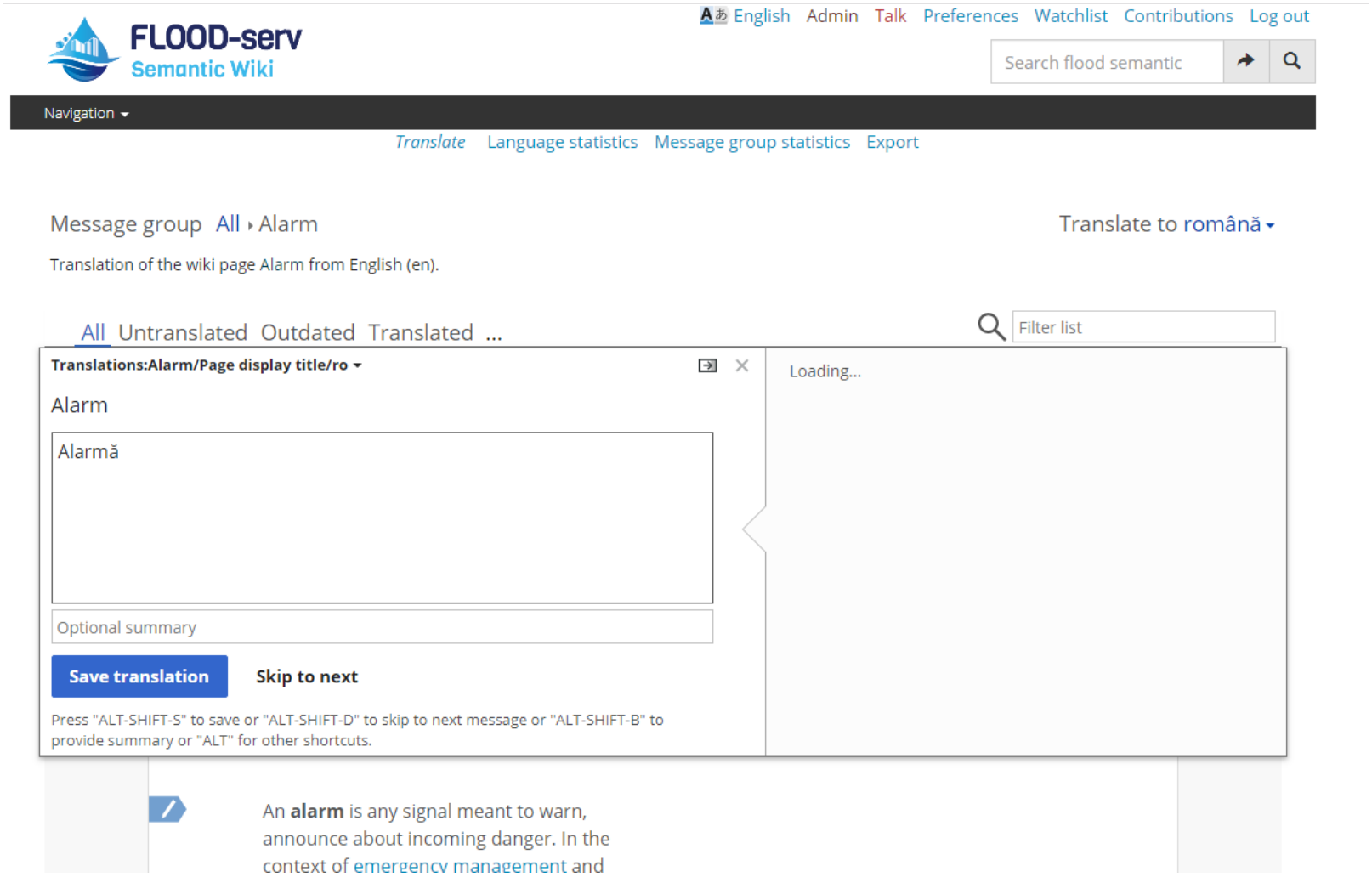
Step 6



Step 7



Step 8



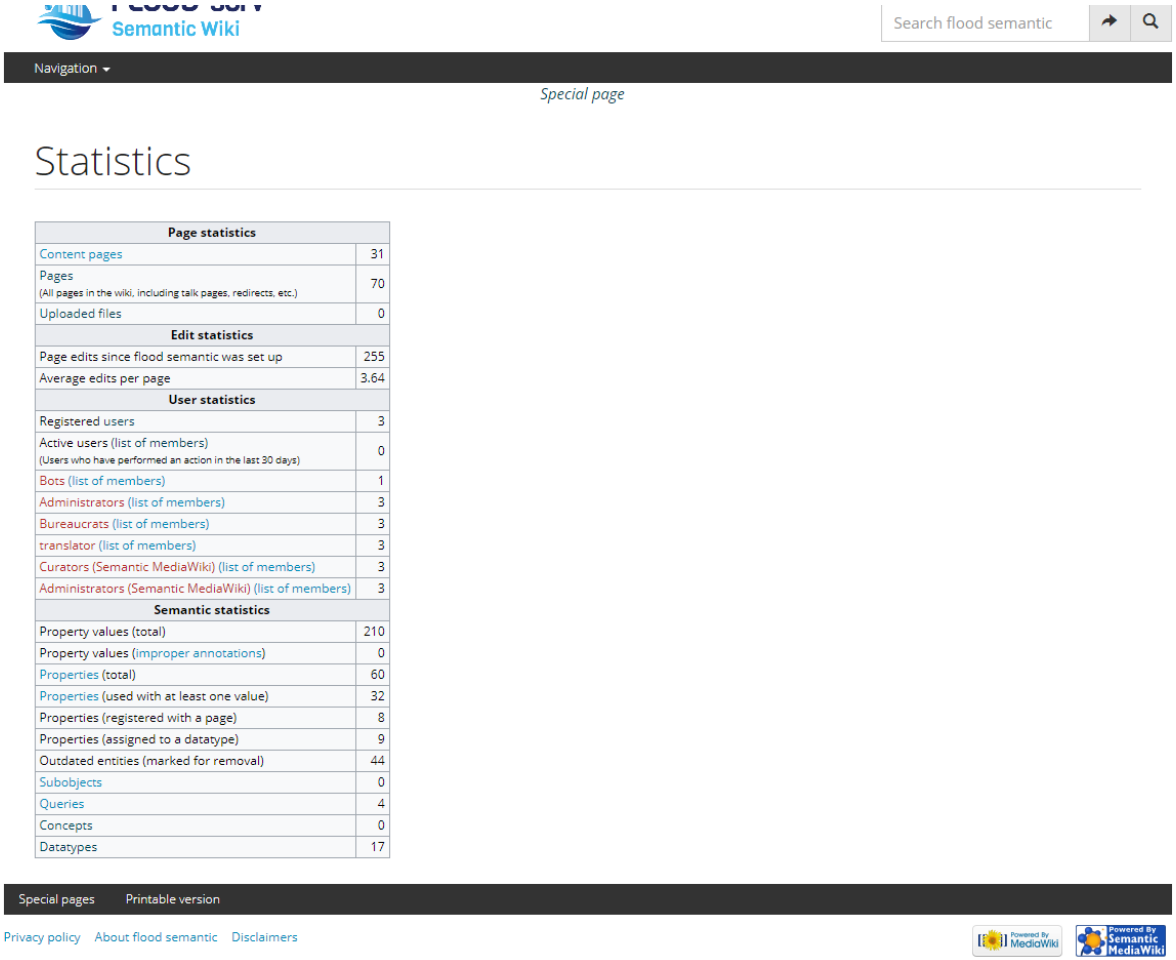
4.5 Usage Statistics:

TEST SCENARIO – SEMANTIC WIKI			
Scenario Code:	**TS-SW-US-001**	**Version:**	**1.0**
Title:	**Usage statistics**	**Date:**	
Description:	Reports and statistics about usage of the Semantic Wiki and content editing.		
Preconditions:	User admin authenticated into system		
Test Case	**Description**	**Results**	
TC-SW-US-001	View statistics		
	View semantic statistics		

4.5.1.1 Test Case: View statistics

TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-US-001**	**Version:**	**1.0**
Title:	**View statistics**	**Date:**	
Descriptions:	**View page, edit, user and semantic statistics**		
Preconditions:	User admin authenticated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	In the Special Pages list/Data and tools group press "Statistics"	A page containing a table with page, edit, user and semantic statistics opens.	

4.5.1.2 Test Results: View statistics



4.6 Suspension of an account: Test Scenario

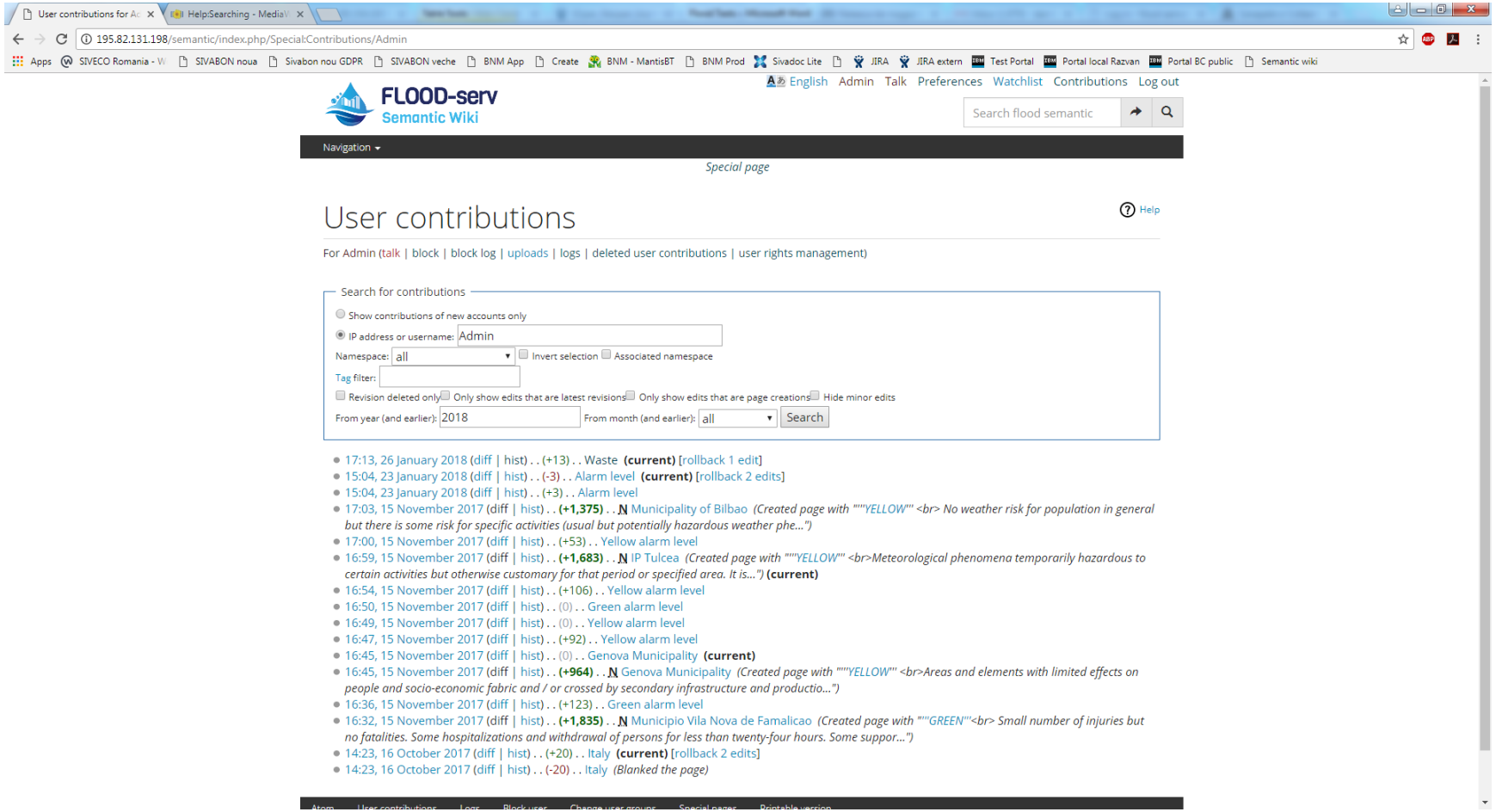
TEST SCENARIO – SEMANTIC WIKI			
Scenario Code:	**TS-SW-SA-001**	**Version:**	**1.0**
Title:	**Suspension of an user account**	**Date:**	
Description:	User account suspension		
Preconditions:	User admin autenthicated into system		
Test Case	**Description**	**Results**	
TC-SW-SA-001	Suspension of an user account	The user account is suspended and the respective user is not able to access their account.	

4.6.1.1 Test Case: Suspension of a user account

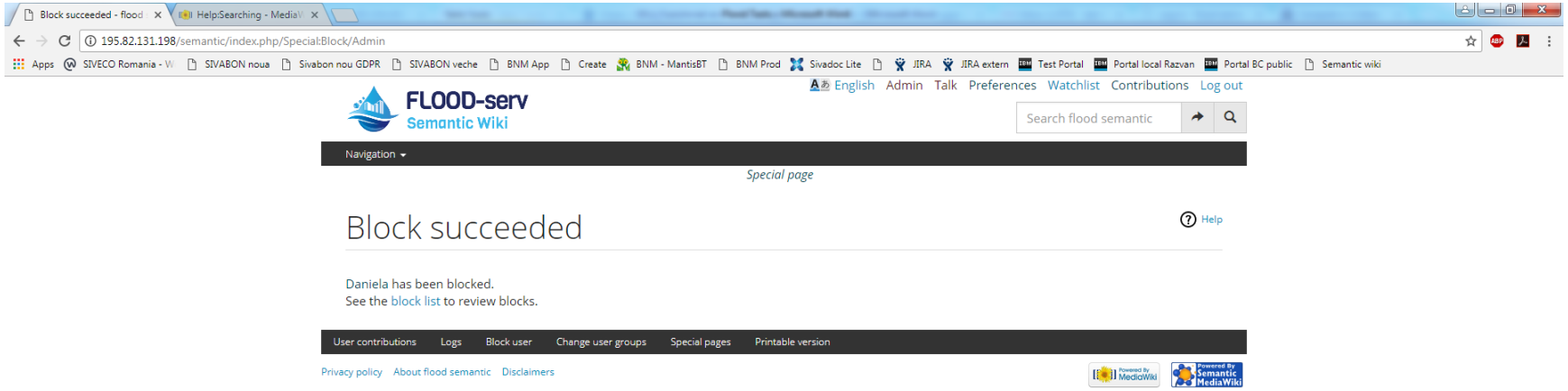
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-MC-001**	**Version:**	**1.0**
Title:	**Suspension of an user account**	**Date:**	
Descriptions:	**The case test describes how to suspend an user account**		
Preconditions:	User admin autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Click Contributions	Open the list of all contributions in the system	
2	Click Block	Open the specific screen for this functionality	
3	Enter the name of user to be blocked and the additional information	-	
4	Click Block the user	-	

4.6.1.2 Test Results: Suspension of a user account

Step 1



Step 2



4.7 User Administration: Postponed implementation

POSTPONED

The base technology implemented, MediaWiki and Semantic MediaWiki, by default assumes that users are created on a self-service basis and also modifications to user data (name, email) are made by the users themselves. While in principle the SIVECO team could have developed the ability for Administrators to create users, such development was deemed unnecessary since the intent is to have Single Sign On (SSO) and centralized user creation in the FLOOD-serv Portal. Consequently we decided to postpone the implementation of user administration functions until the implementation of SSO is made.

User administration functionalities have been implemented in the FLOOD-serv Portal (SSO Module, functionalities: USFP17, USFP18, USFP19, USFP20, USFP14) under WP4, Task 4.4. See D4.4, Chapter 7 and D5.4 (sections 3.2.1, 3.3.1, 3.4.1, 3.5.1, 3.6.1, and 6.1.2)

4.8 Ontology Management: Redundant Requirement

User Requirement: As an Administrator I want to create / modify / delete ontology so that I can manage the ontologies.

In light of current understanding of the issue, we consider that this requirement is superfluous, overlapping with the data export requirement. That is because the Semantic Wiki content can be exported as RDF-OWL ontology, and by exporting and re-exporting the user can create and modify ontologies.Modifications of ontologies resulting can be performed by either modifying the export parameters (e.g. the pages being exported) or by modifying the content of the SW and then re-exporting it.

4.9 Content Organization: Redundant Requirement

In light of current understanding of the issue, we consider this requirement as superfluous and included in the Edit/Modify content requirement. By editing a page any user with editing rights and the administrator can organize that page’s content, change the structure, move the content’s position within the page.

4.10 Approve content: Test Scenario

TEST SCENARIO – SEMANTIC WIKI			
Scenario Code:	**TS-SW-AC-001**	**Version:**	**1.0**
Title:	**Approval of the content**	**Date:**	
Description:	The test scenario allows the admin to approve the created/edited contents		
Preconditions:	User admin autenthicated into system		

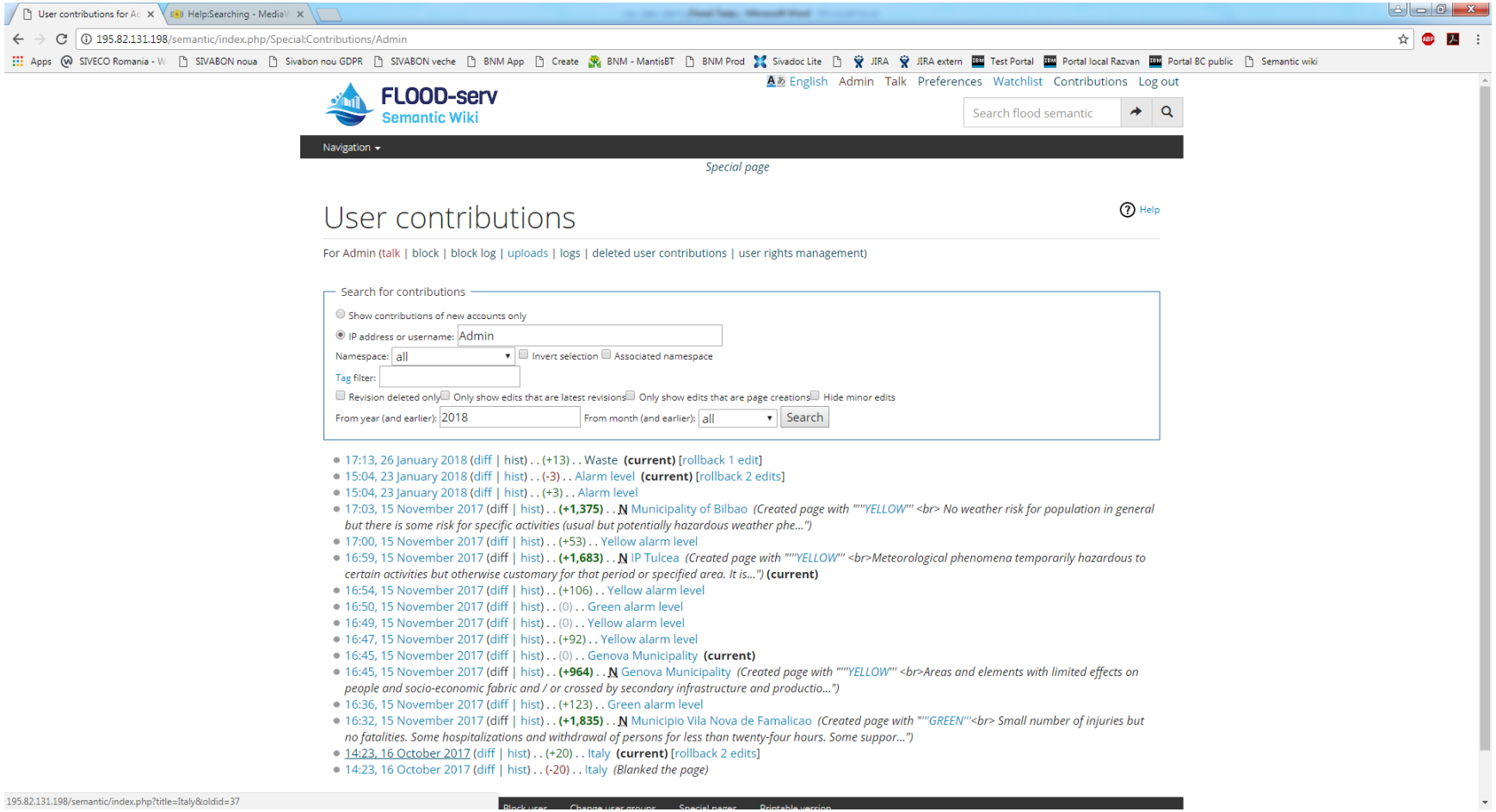
Test Case	**Description**	**Results**
TC-SW-AC-001	Approval of content	The content is modified accordingly with the approval

4.10.1 **Test Case: Approve content**

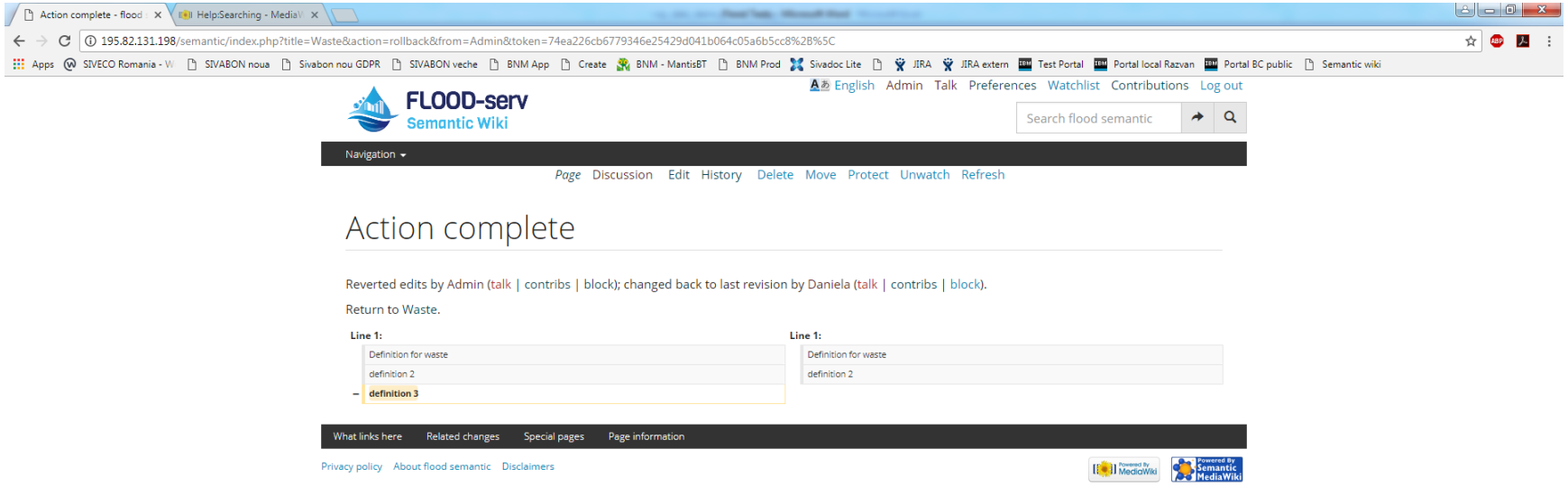
TEST CASE – SEMANTIC WIKI			
Test Case Code:	**TC-SW-AC-001**	**Version:**	**1.0**
Title:	**Approval of the content**	**Date:**	
Descriptions:	**The case test describes how to approve the content**		
Preconditions:	User admin autenthicated into system		
Steps	**Actions and Data**	**Expected Results**	**Outcome**
1	Click Contributions	Open the list of all contributions in the system	
2	Click Rollback where the modifications on the content are not correct	The modifications made are undo	

4.10.2 **Test Results: Approve content**

Step 1



Step 2



5 Conclusions

This document reports on the technical implementation of the Semantic Wiki Component. Out of 10 user requirements, 9 were implemented, and 1 (user administration) was postponed until the implementation of the Single Sign On at the level of the entire FLOOD-serv Portal. Of the 9 implemented requirements, 2 were considered redundant with others.