

Project Proposal: Add New License Submittal Feature to Online Tools

Abstract

The aim of this project is to design and develop new features to an existing SPDX web application (<https://github.com/spdx/spdx-online-tools>), that will allow software licenses producers to generate new license requests submissions in a more robust and user-friendly way. These new features are going to enhance the process that the SPDX legal team follows to check new license request submissions as well as to generate the new licenses XML files. These improvements will reduce the time the process of enriching the SPDX licenses list takes (<https://github.com/spdx/license-list-XML>).

Motivation

Currently, the process of adding new licenses to the *SPDX license list* is tedious and involves several steps, which include: sending the possible new license information through email to the SPDX legal team; generating a XML file manually after validation and submitting the license in the proper XML format to the SPDX license-list-XML GitHub repository.

An open-source tool provided to enhance and *pseudo automate* the current process of enriching the SPDX licenses list, would be helpful to direct the human resources used to execute this task to another assignment.

Description and Technical Overview

At the time this proposal is written, adding new licenses to the SPDX software licenses list requires the licenses producers to send emails containing the licenses information to the SPDX legal team. Then, the legal team checks both valid and invalid licenses information. Finally, if the information is valid, the legal team manually generates an XML file containing the license information, to be submitted to the licenses list. This entire process takes too much time and has certain sensitive to errors steps. The objective of this project is to develop a more robust method of submitting new licenses requests to the SPDX legal team and reduce the time that takes reviewing for valid submissions to the legal team.

To achieve this, we are going to improve the current web application (<https://github.com/spdx/spdx-online-tools>). We are going to design and develop a form that will be used to submit new license requests. The submitted licenses are going to be compared with the existing licenses to avoid duplication. In addition to this, we are going to allow the

SPDX legal team to manage non-reviewed requests and either approve or deny them. Furthermore, the licenses producers will track their requests approval status by email notifications. Finally, the legal team will be able to automatically generate the XML files of the past approved licenses.

Web Application Development Tools

Front End: The development of the visual design component of the application will be using the web standards: HTML5, CSS3 and Javascript. To give the application a responsive and good-looking design on different displays (e.g. Mobile, Desktop, Tablet), Bootstrap will be used as front-end framework.

Back End: The development of the back-end component will be done using Django Web Framework. Also, the Relational Database Management System (RDBMS) engine will be PostgreSQL.

Use Cases

The uses cases primary actors are shown below. Both of them are users that will need to be logged in. However, the producer login might be in discussion.

Use Cases Primary Actors		
	Actor	Description
001	Producer	User who submits the licenses requests.
002	Consumer	User who views licenses requests and manage them accordingly. A consumer is a SPDX Legal team member.

The use cases related to the new features to be added to the existing SPDX web tool, are shown below.

Submit a license request	
ID	UC-001
Title	Submit a license request
Primary Actor	Producer, Consumer
Description	A producer or consumer desires to add a new software license to the SPDX licenses list.

Precondition	1.1- The producer is logged in. 1.2- The producer has filled all the necessary fields about the license.
Postcondition	1.1- The license request is added to the submitted license requests list. 1.2- The license has a “non-reviewed” state.

View submitted license requests	
ID	UC-002
Title	View submitted license requests.
Primary Actor	Consumer
Description	A consumer views a list of all the submitted license requests by their state. These states can be “non reviewed”, “approved” or “denied”.
Precondition	2.1- The consumer is logged in.
Postcondition	2.1- The consumer can choose to view the non-reviewed requests. 2.2- The consumer can choose to view the approved requests. 2.3- The consumer can choose to view the denied requests. 2.4- The consumer can search a license by its full name.

Approve a license request	
ID	UC-003
Title	Approve a license request.
Primary Actor	Consumer
Description	A consumer reviews a submitted license request and approve its addition to the SPDX licenses list.
Precondition	3.1- The consumer is logged in. 3.2- The consumer has checked the license validity.
Postcondition	3.1- A notification about the approval is sent to the producer who submitted the license. 3.2- The license request state is changed from “non-reviewed” to

	“approved”.
--	-------------

Deny a license request	
ID	UC-004
Title	Deny a license request.
Primary Actor	Consumer
Description	A consumer reviews a submitted license request and deny its addition to the SPDX licenses list.
Precondition	4.1- The consumer is logged in. 4.2- The consumer has checked the license validity.
Postcondition	4.1- A notification about the denial is sent to the producer who submitted the license. 4.2- The license request state is changed from “non-reviewed” to “denied”.

View a license request information	
ID	UC-005
Title	View a license request information.
Primary Actor	Consumer
Description	A consumer selects a license request in the license requests list to view its information.
Precondition	5.1- The consumer is logged in. 5.2- The license is part of the license requests list.
Postcondition	5.1- The consumer can view the producer username who generated the request. 5.2- If the state of the request is different than “non-reviewed”, then it is visible the consumer’s username that checked the request.

Generate a license XML file	
ID	UC-006
Title	Generate license XML file.
Primary Actor	Consumer
Description	A consumer generates a valid license XML file.
Precondition	6.1- The consumer is logged in. 6.2- The license request has an “approved” state.
Postcondition	6.1- The generated XML file is automatically downloaded.

Front-End Description and Mockup

The visual component of the application new features will be responsive and it consists on adding 3 views to the existing SPDX tool:

- *Submit New License View*
- *View License Requests List View*
- *View License Request Information View*

Submit New License View: This view is available for any logged in user, by clicking the respective tab button “Submit New License” located in the menu visible in any page of the site at the top. This view contains a form that includes the required fields to submit a new license (Fig. 1). The source/URL field, includes an *add* button to increase the amount of the license web pages URL.

OBS: If there is any problem on the constraint to be logged in to submit a license request, an additional email field would be added to the form. It is because it could be useful to notify the producer about the approval or denial of its request.

SPDX Validation Tool About Validate Compare Convert Check License Submit New License

Submit a New License to the SPDX Licenses List

Full Name	<input type="text" value="License Full Name"/>
Short Identifier	<input type="text" value="License Short Identifier"/>
Source / URL	<input type="text" value="Other web pages for this license"/>
OSI Approved	<input type="text" value="Yes"/> ▼
Notes	<input type="text" value="License notes"/>
Standard License Header	<input type="text" value="Standard License Header"/>
Text	<input type="text" value="Full license text of the license"/>

Figure 1. Submit New License View.

View License Requests List View: Only consumers have access to this view, by clicking the respective tab button “License Request” located in the menu visible in any page of the site at the top. At the beginning the user need to select a filter in a combobox to show the list-view below (Fig. 2). The options contained in the combobox are “Non-Reviewed”, “Approved” and “Denied”. The user will be able to see a list of requests ordered from newest to oldest. However, the there will be also a search bar to find specific requests by their license full name. Moreover, in order to enhance the user experience, the requests will be distributed in several pages to avoid having too many rows in the list-view of one single page.

SPDX Validation Tool About validate Compare Convert Check License Submit New License License Requests

View the License Requests List

Select the state of the requests you want to view

State of request ▼

Full name	Short Identifier	Submission date
BSD Zero Clause License	0BSD	18-03-2018
Adobe Glyph List License	Adobe-Glyph	16-03-2018
Academic Free License v2.1	AFL-2.1	16-03-2018
GNU Affero General Public License v3.0	AGPL-3.0	12-03-2018

Page
1 - 2 - 3

Figure 2. License Requests List View.

View License Request Information View: Only consumers have access to this view, by clicking any row of the list-view shown in the License Requests List (Fig. 2). This view shows the information related to the request selected in the view shown in Fig. 2. If the selected request has the state “non-reviewed”, then it is shown two buttons to approve or deny the request. If the selected request has a different state than “non-reviewed”, then it is shown the consumer’s username who reviewed the request. In any case, it is always shown the username of the request’s producer. Also, it is shown a back button that shows the requests list-view.

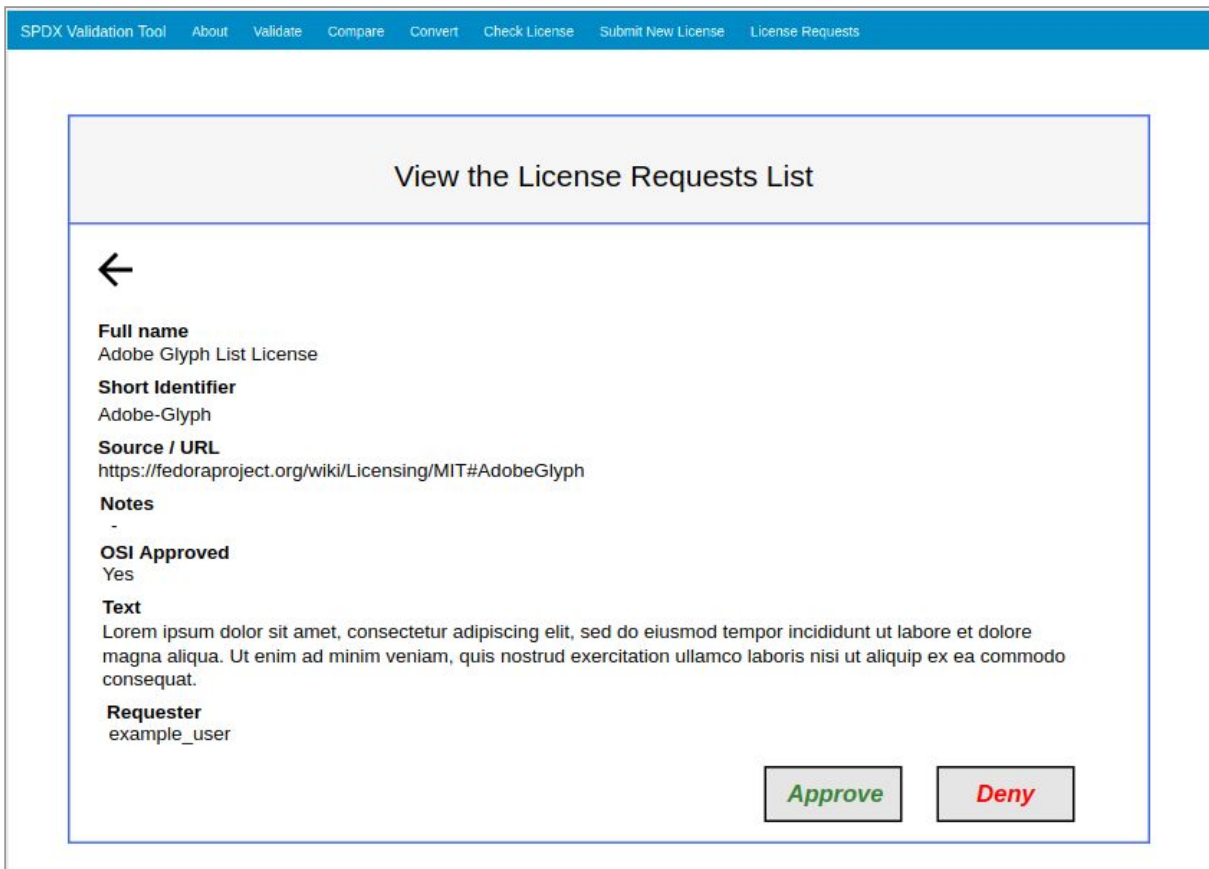


Figure 3. Non-Approved License Request Information Example.

Submission Process

The new software license request submission and review process is shown in Fig. 4. The new process includes automatic email notifications to both consumers and producers.

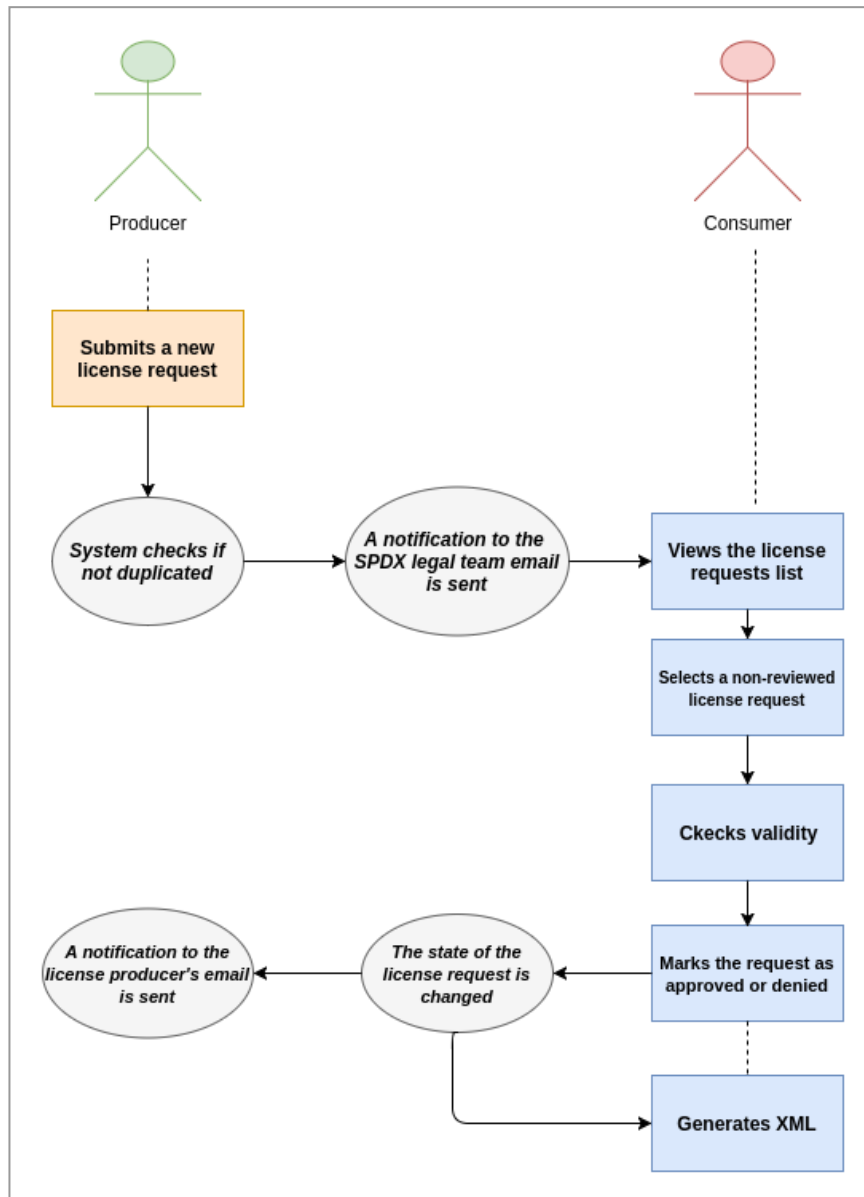


Figure 4. New License Request Submission Process.

Tentative Project Timeline and Deliverables

The project will use an Agile Scrum methodology. For that reason, the project timeline is based on a two-week deliverables schedule as shown in the table below. This plan has been developed considering semester midterm and final evaluations at my university.

Dates	Activity
May 14 - May 27	Community Bonding. Use cases validation. Concreting proposal details and development.

May 28 - June 10	Prepare coding environment. Review existing tool's code. Front-end-development
June 11 - June 24	Front-end Development.
June 25 - July 8	Back-end Development.
July 9 - July 22	Session Management. Communicating frond-end with back-end.
July 23 - August 5	Documentation. <i>Automatic GitHub pull requests (if possible)</i>
August 6 - August 10	Rigorous Testing. Performance analysis. Bugs Fixing.

Table 1. Tentative Project Timeline and Deliverables.

Future Work

Future work plan includes the generation of automatic pull requests to the licenses XML GitHub repository (<https://github.com/spdx/license-list-XML>) after a license request approval. If possible, this feature would be reviewed during this project's Documentation phase, as shown in Table 1.

Student Information

Full name: Galo Daniel Castillo Lopez

Residence place: Guayaquil, Ecuador

Institution: Escuela Superior Politécnica del Litoral

Email: gadacast@fiec.espol.edu.ec

Hangouts email: gadacast@fiec.espol.edu.ec

Cell-Phone: (+593) 95-951-7136

GitHub username: gaalocastillo

Brief Resume: Undergraduate Computer Science 8th semester student at an Ecuadorian University, ESPOL.

SPDX Questions

1.- When did you first hear about SPDX?

The first time I heard about SPDX was in 2017, when looking for GSoC projects to apply at GSoC 2017. However, I did not apply to any project for that GSoC edition.

2.- Describe your participation in our community (bug fixes, communication via mailing lists, IRC and Gitter).

I participated in the SPDX's Gitter community during the GSoC 2018 *Students Application* phase. During this period, I interacted with mentors and other students interested in applying to SPDX's GSoC projects.

3.- In less than 3 sentences, why should we pick YOU?

I am sure I would be a valuable asset for the organization given my experience working with agile methodologies and Django web framework. Being accepted will greatly contribute in my professional career, boosting my technical and non-technical abilities and give me valuable experience on open source projects.

About myself

1.- Describe any plans you have for the summer in addition to GSoC (classes, thesis, job, vacation, etc.)

During the 2018 summer I will be attending three courses at ESPOL, which are part of my degree studies. However, those would be the only activities in which I would be working on besides this project.

2.- What programming projects have you completed? What are some of the similarities of your previous projects to your proposal?

About a year ago, I took the Web Applications Development course at ESPOL, which is part of my degree studies. In this course, I presented *HelpMapp*, a web application to manage donations after natural disasters. Similar to the current SPDX tool, Helpmapp uses Django Web Framework for the backend programming. Also, I am working on EspolGuide, a campus outdoor mapping mobile application. EspolGuide uses an Agile Scrum methodology, which include a two-week sprints schedule, similar to this project proposal.

3.- What are your favorite programming tools (editor, etc.)?

My favorite programming tool is the Sublime Text editor. Moreover, I prefer using GitHub for version control, instead of other similar tools.

4.- Describe any work on other open-source projects.

This proposal project, if accepted, would lead me to work on my first open-source project. This project is a great opportunity for me to have my first open-source development experience.

5.- List other GSoC projects you are applying to.

This project is the only one I am applying to for the current GSoC edition. Also, this is the first time I am applying to a GSoC project.