Proposal for Elixir Idea #2

Vulnerability report system in hex.pm website

Table of contents

Personal information.

Name:
Lois Soto López.

Student timeline:
- Degree in Computer Engineering at Universidade de A Coruña, Spain, September 2016 - Present

Email addresses:
loissotolopez@gmail.com
loissoto@outlook.com

GitHub user:
https://github.com/LoisSotoLopez (non-academic projects only).

Abstract.

The idea of this project is to design and create all needed components for a vulnerabilities report system in Hex packages integrated in the https://hex.pm/ website.
The essential objective of the system would be to provide a method for the user to inform and stay informed about the status of vulnerability reports and packages, allowing him to personalize his experience with a *followed packages* list.

Project planning.

Website functionalities:

With the project idea and the essential objective described in the previous section, the following functionalities for the website are proposed:

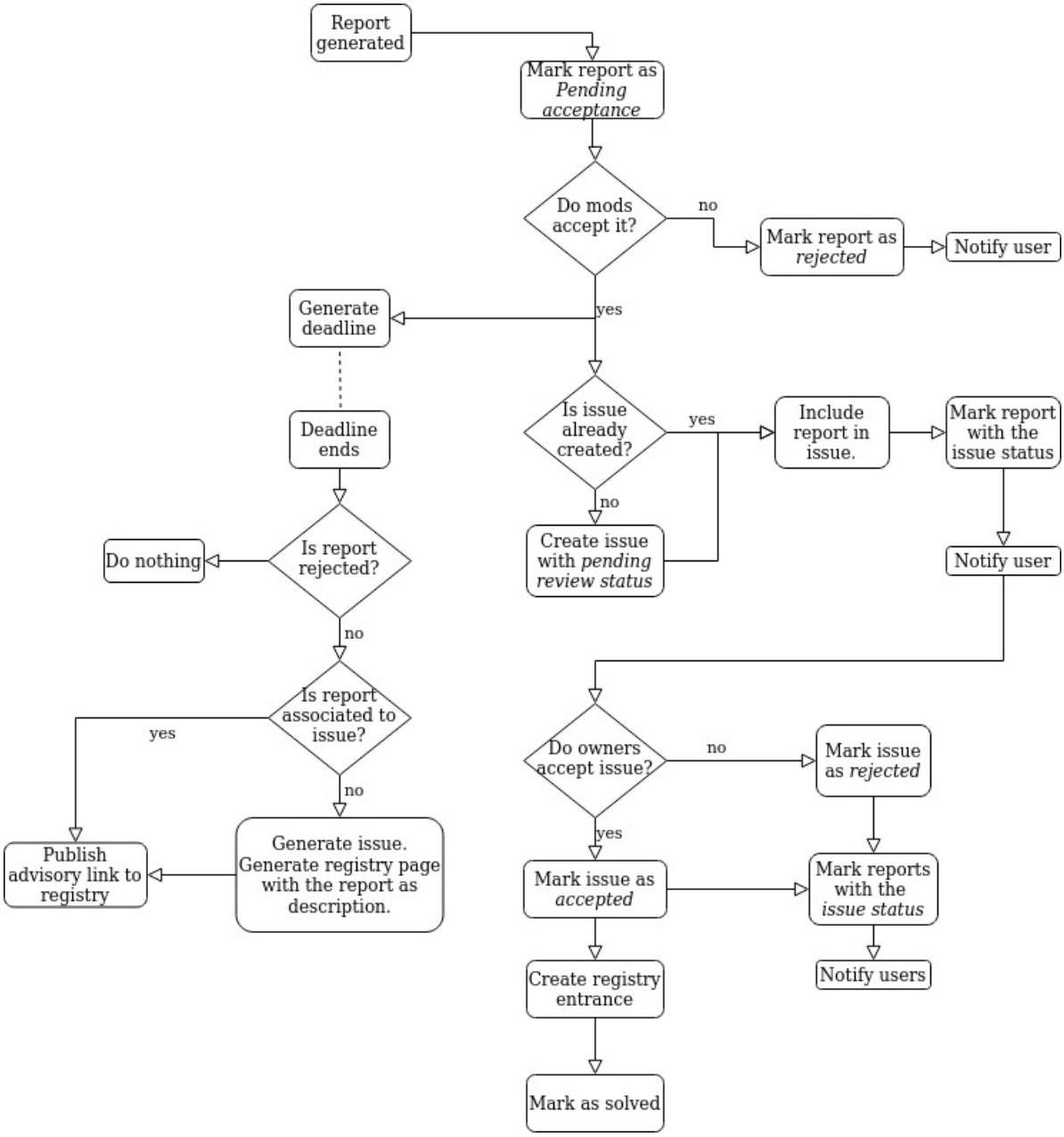
- A **vulnerabilities registry** where each package vulnerability has its entrance; something like the chromium bugs list (https://bugs.chromium.org/p/chromium/issues/list). The possibility of allowing users to write comments has been considered, but I am not sure this fits with the purpose of the project.

- Concerning the group of trusted administrators (aka. **the admins**):

    - A view of all reports, and the hability to group reports concerning the same security issue. This view would also include those groups of reports.
    - A report specific view, and the ability to *accept* or *reject* a report. This view should also allow admins to provide a “report rejection reason” in case the report is rejected.

In any case, the user would receive a notification about the report status, that could be *rejected* if the report has been rejected or *Pendent of review* if the report has been accepted by the admins.

- Concerning the site user (aka. **the user**):
  - Ability to report vulnerabilities on a package, providing its identifier and a description detailed enough to replicate the problem.
  - The ability to follow packages through a button on the package page, and a list of followed packages under a specific section on the user's profile. Some kind of visual alert should appear on the users dropdown menu (on the page navbar) to make them now there is a followed package alert. For each package, the user should be able to:
    - Get to the vulnerabilities registry package page.
    - Retrieve the reports submitted and the current vulnerability report status (Pendent of acceptance, Pendent of review, accepted, rejected, solved).
- Concerning the package owners (aka. **the owners**):
  - A view of all accepted reports for their packages. This view would include an agile way to create a **vulnerability issue**, and group all related reports in it. The owners would also be able to reject a report/vulnerability issue if the vulnerability associated is not a real one (the users think it is but is actually intentional). In this case all reports associated to that vulnerability issue would be marked as *rejected*. If the owners consider the issue is real it must be accepted, which would result on all reports associated to be marked as *accepted*. The package owners must create the vulnerability registry entrance correspondent with the issue.
  - The ability to modify vulnerabilities registry entrances for their packages and to mark them as fixed if needed. This would result on the reports associated with the vulnerability issue to be marked as *solved*.



Integrations:

It has been considered the possibility to integrate the previous funtionalities with both Hex.audit and Hex.retire tasks. Users should be notified about the reitement of a package they follow the same way they are when one of their report status changes.
The retirement *reason* message could reference a vulnerability registry entry.

Biographical information.

Even though the main scope of the academic speciallity I am studying at the moment focuses on network protocols and design, virtualization and comunication technologies, throughout the career I have also taken courses about Application Integration, Human-Machine Interfaces and Integative Programming.
On my own I have done some toy projects with a few web frameworks and I have found Phoenix to

be the most enjoyable to work with.
Based on previous experiences with similar (but less complex) projects, I think that with some help on the first stages of the process I would be able to achieve the materialization of the funtionalities previously described.