Sprint 4 Review Report
USSF GRC Controls - Kubernetes

Sprint Dates:

November 8th, 2024 - November 15th, 2024: Sprint 4
November 15th, 2024 - November 22th, 2024: Sprint 4 Retrospective

Links to Important Resources:
Heroku App: app link
GitHub Repository: USSF-CSCE606
GitHub Project: USSF

Team Roles:

- **Product Owner**: Medha Kaushika Podipireddi
- **Scrum Master** : Vasudha Devarakonda
- **Developers:** Sahil Fayaz, Shravan Bhat, Aditya Gourishetty, Maitreya Niranjan, Duy Vu, Medha Kaushika Podipireddi

Sprint Goal:

In Sprint 4, we focused on adding more granular permissions for sharing across users and also user levels. At user levels, we have admins who can delete the existing users from the database i.e., remove access to the product, and at a general user level we have owners and collaborators for sharing images. Further, we fixed bugs; show the vulnerability report immediately without reloading and do not reload the page when the user is downloading the report. Additionally, we focused on increasing the accessibility of the product by adding “help” features for easy navigation and color blindness feasibility options. We refined the user interface as well and made it more consistent.

Sprint Summary and Achievements:

In Sprint 4, our team successfully advanced the GRC Control application by implementing features that enhance the user management of the applications. Specifically three types of users were created; admin, image owners and general users. Further general users will be given permission to either view or edit the image based on the permissions given by the image owners. Further the application supports scanning of private registry images. With this feature

users can scan the images in a private registry by adding their credentials(temporary and not saved). By adding this feature, the application supports all types of container images. Further, the view of the vulnerability report was improved where users can now filter and sort for an easy and structured view of the report. Additionally, considering the request from the client, the accessibility of the application was improved. Navigations were added to each element of the application and was further tested with NVDA to ensure accessibility to impaired users. Additionally, color schema and styling was selected to improve color blind accessibility.

The application is currently hosted on Heroku. We have also achieved good code quality having used rubocop, rubycritic, and code coverage for Rspec and Cucumber test cases. All the committed stories for the sprint have been marked as done.

Story	**Story Points**	**Sub Tasks**	**Sub Task Points**	**Assigned To**	**Status**
Add user types	3	Add admin, general users(owners and collaborators)	3	Sahil	Done ✅
Refine UIs	2	Make buttons and texts more consistent across different pages on the same and different devices.	1	Aditya	Done ✅
		Add a description of image tags	1	Aditya	Done ✅
Refine functionalities and actions	3	See the vulnerability report on the screen immediately without reloading	2	Shravan	Done ✅
		Fix downloading report feature	1	Aditya	Done ✅
Refine Access Control	3	Include granular sharing options like edit, and view for the run-time	2	Duy Vu	Done ✅

		objects			
		Remove the delete option in the collaborator view, only owners can delete the containers	1	Duy Vu	Done ✅
Make the application Accessible	3	Show help: what a certain action or a field does while hovering over them	2	Maitreya	Done ✅
		Make the UI accessible to color blindness, especially the report page.	1	Maitreya	Done ✅
Add Sort feature based on some properties	1	Ascending and Descending Sort dropdown in vulnerability report per properties	1	Shravan	Done ✅
Add feature to scan from private registry	3	Scan images in private registry	3	Medha	Done ✅

See the Appendix for story descriptions.

Overall Project Contributions

Name	Total Points By Individual	Total Points of the Team
Aditya Gourishetty	9	63
Duy Vu	9	63
Medha Kaushika Podipireddi	9	63

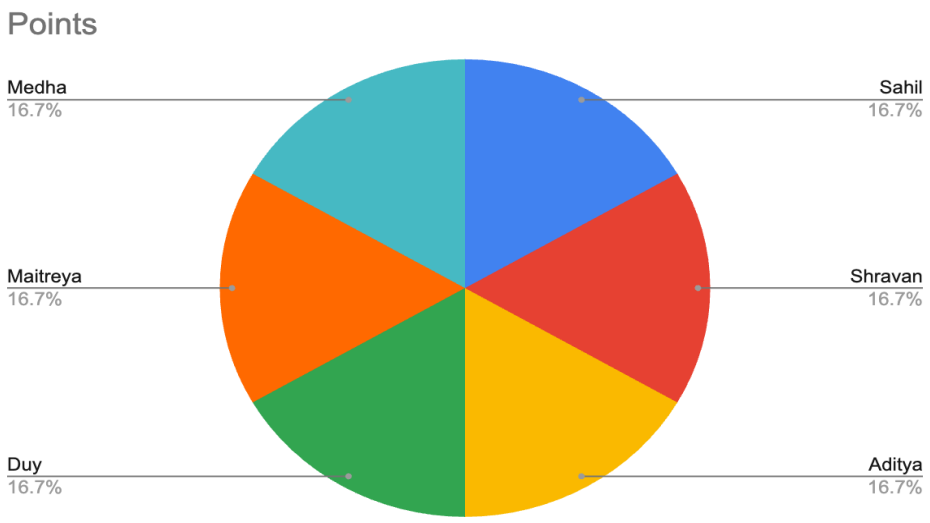
Maitreya Niranjan	9	63
Sahil Fayaz	9	63
Shravan Bhat	9	63
Vasudha Devarakonda	9	63

Sprint Backlog Items and Status:

In this sprint, small bugs like loading were added after the sprint was closed based on the client feedback because this was the last sprint and the team wanted to deliver a refined product.

Apart from this all the other planned features and tasks for Sprint 4, mentioned in the table above, were completed in the scope of the sprint

Team member contributions:



Medha: Implemented the feature to scan and rescan private registry images that takes in credentials of the private registry and runs accordingly.

Sahil: Developed user management features of the applications by making admin and general users options.

Aditya: Fixed the UI buttons and text inconsistencies and further improved the download report feature

Shravan: Fixed the reloading bug of the report and added sort and filter features for the report

Duy: Added permissions for specific types of users and improved sharing options.

Maitreya: Improved the accessibility of the application by adding navigation and by improving the color scheme of the application.

Burn Down Chart:

The y-axis represents the story points committed for the sprint.

Design Diagrams and UI Mockups:

Document of Changes:

In Sprint 4, the primary change was to add new users - admin and image owners. This improved the user management feature of the application. Additionally the new version of the application also supports private registry images. Further, the report view was further enhanced to add filters and sorting mechanisms. Additionally, the accessibility of the applications was improved.

Aside from this addition, there were no other changes to the design, stories, mock-ups, storyboards, or code organization in Sprint 4 compared to Sprint 3.

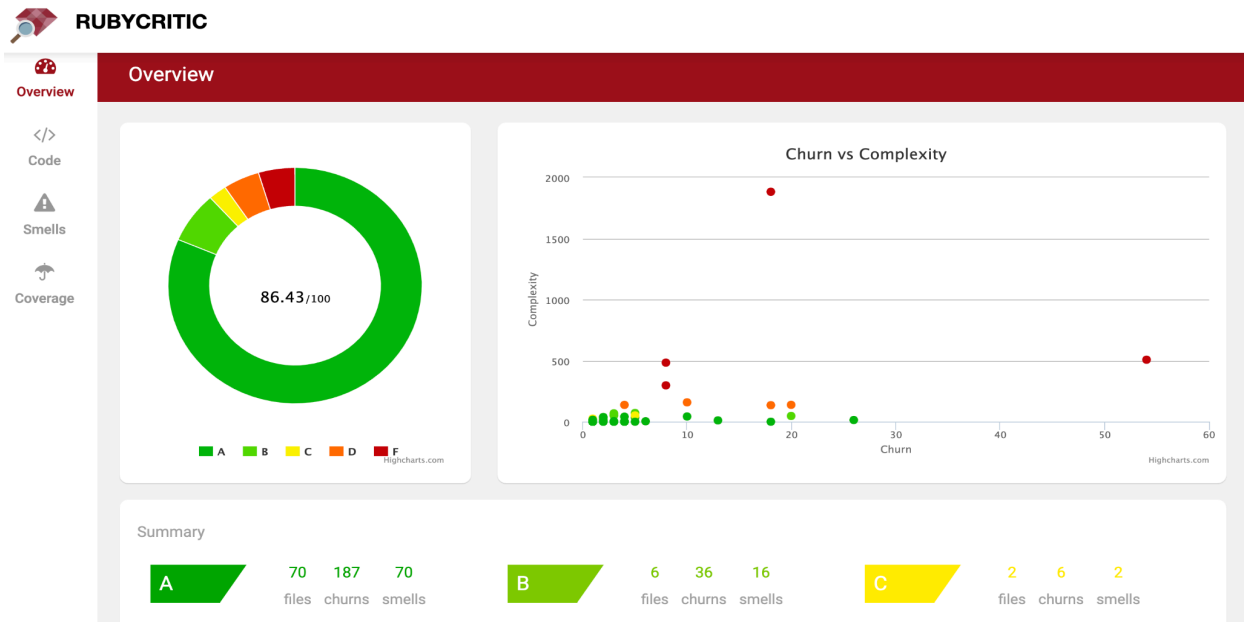
Code Evaluations:

Rspec and Cucumber Coverage:

```
2024-11-21T16:51:55-06:00       INFO    Number of language-specific files       num=0
.deleting
.....................................................................................

Deprecation Warnings:

Rails 7.1 has deprecated the singular fixture_path in favour of an array.You should migrate to plural:


If you need more of the backtrace for any of these deprecations to
identify where to make the necessary changes, you can configure
`config.raise_errors_for_deprecations!`, and it will turn the
deprecation warnings into errors, giving you the full backtrace.

1 deprecation warning total

Finished in 5.36 seconds (files took 1.8 seconds to load)
103 examples, 0 failures

Coverage report generated for Cucumber Features, RSpec to /Users/vasudhadevarakonda/Documents/USSF-CSCE606/coverage.
Line Coverage: 93.24% (345 / 370)
(base)  vasudhadevarakonda@Vasudhas-MacBook-Air  ~/Documents/USSF-CSCE606  main
```

```
25 scenarios (25 passed)
174 steps (174 passed)
0m7.310s

  Share your Cucumber Report with your team at https://reports.cucumber.io

  Command line option:    --publish
  Environment variable:   CUCUMBER_PUBLISH_ENABLED=true
  cucumber.yml:           default: --publish

  More information at https://cucumber.io/docs/cucumber/environment-variables/

  To disable this message, specify CUCUMBER_PUBLISH_QUIET=true or use the
  --publish-quiet option. You can also add this to your cucumber.yml:
  default: --publish-quiet

Coverage report generated for Cucumber Features, RSpec to /Users/vasudhadevarakonda/Documents/USSF-CSCE606/coverage.
Line Coverage: 93.24% (345 / 370)
```

RubyCritic:



Client Sprint MVP meeting information:

Date: 11/21/2024
Time: 18:15 - 19:00
Location: Google Meet
Attendees: Prof. Shreyas Kumar, Major Eric Griffin, Oslyn Marie, Aditya Gourishetty, Maitreya Niranjan, Duy Vu, Medha Kaushika Podipireddi, Sahil Fayaz, Shravan Bhat, Vasudha Devarakonda

The team presented the latest version of the application to Prof. Sheyras Kumar and the USSF team, showcasing the application navigation, scanning, and downloading functionality, different types of users(admin and regular), accessibility of the application for color blindness, user feasibility of usage of the application, and access control features across users. We also presented security due diligence of the application along with answers to the additional requirements raised by Lt. Theresa Kopecky in the previous sprint(Sprint 2).

The additional requirements were:

- One page overview about building a service like Trivy that includes details like estimated time of building such service, complexities involved in it, etc.,
- How is our application different from any other automated vulnerability scanning applications available?
- What is the scalability of the application and Trivy(third-party scanner)?

The USSF clients were impressed by the application and there were no further improvements suggested. They wanted us to present the application demo to their seniors on the day of the final presentation in class so that their team could draw up tasks for next semester.

Appendix:

Story Descriptions:

Feature: Add User types

- **As a user** of the application
- **So that** I can have a centralized governance of the product
- **I want** an “admin” user and a “general” user who can share their applications with each other

Feature: Refine UIs

- **As a user** of the application
- **So that** I can have a better user interface

- **I want** consistency in my views irrespective of the device we are viewing it on.

Feature: Refine functionalities and actions

- **As a user** of the application
- **So that** I can see the report
- **I want** consistency in my actions .

Feature: Refine Access Control

- **As a user** of the application
- **So that** I can take action based on the permissions granted
- **I want** to edit or view the image page as per sharing options.

Feature: Make the application Accessible

- **As a user** of the application
- **So that** I can make the application more accessible to end users
- **I want** a help to navigate and accommodate multiple people

Feature: Add Sort feature based on some properties

- **As a user** of the application
- **So that** I can sort my vulnerability report based on various properties together instead of the existing severity-only filter
- **I want** ascending and descending sort ability per properties available in the report.

Feature: Add feature to scan private images

- **As a user** of the application
- **So that** I can scan image from private registry
- **I want** to able to fetch the image from private registry and check for vulnerabilities