

Sprint 4 Plan - [USSF] GRC Control Kubernetes

Team Roles

- **Product Owner (Iteration 4):** Medha Kaushika Podipireddi
 - **Scrum Master (Iteration 4):** Vasudha Devarakonda
-

User Stories

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

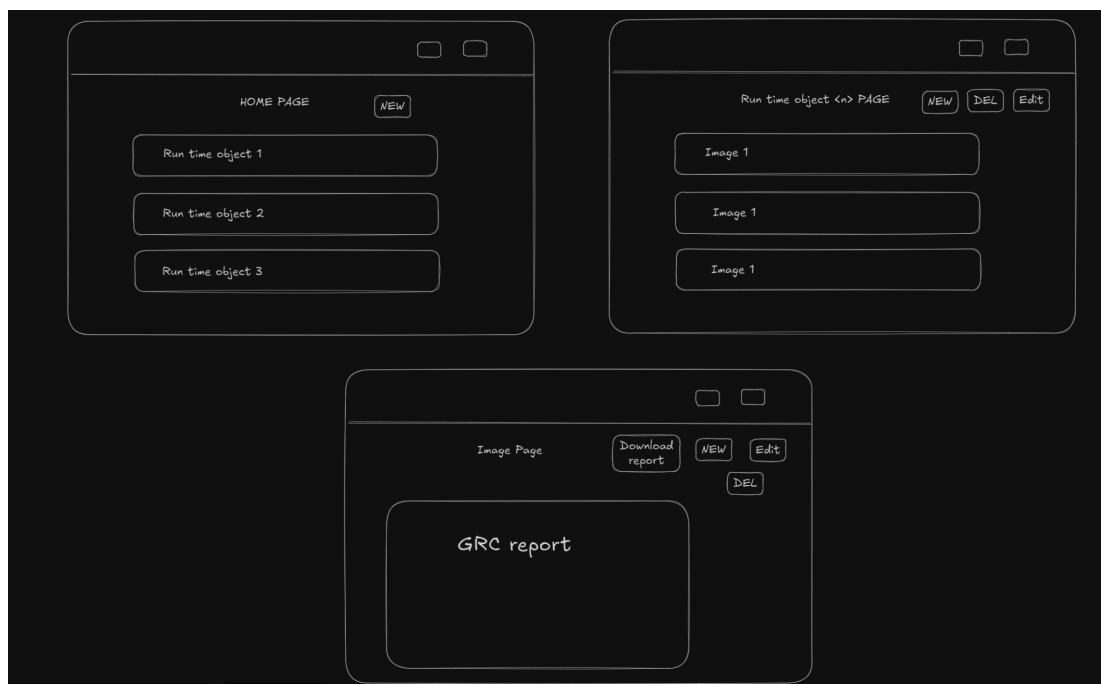
- **As a user** of the application
- **So that** I can sort my vulnerability report based on various properties together instead of the existing severity-only filter
- **I want** ascending and descending sort ability per properties available in the report.

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

UI Mock-up

We will be having the same UIs but we will be refining them, for instance, dynamic font sizes



Sprint Goal

In Sprint 4, we will focus on adding more granular permissions for sharing across users and also user levels. At user levels, we will have admins who can delete the existing users from the database i.e., removing access to the product, and at a general user level where we will have

owners and collaborators for sharing images. Further, we will be fixing bugs; show the vulnerability report immediately without reloading and do not reload the page when the user the downloading the report. Additionally, we will be focusing on increasing the accessibility of the product by adding “help” features for easy navigation and color blindness feasibility options. We will be refining the user interface as well.

Stories Pulled into Sprint

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

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

Note (Time estimate) : 3 points is equivalent to 5 days, 2 points is equivalent to 3 days and 1 point is equivalent to 1 day

Links to Key Resources

- **GitHub Repo:** [USSF-CSCE606](#)
- **Project Management Page:** [USSF-CSCE606-Project-Management](#)
- **Slack Workspace:** [Fall 2024 CSCE606: USSF GRC Control Kubernetes](#)
- **Deployed App URL:** [USSF-GRC](#)

Grading Approach (if applicable)

Apart from the SaaS application UI and backend, we would be exploring third-party libraries for conducting the GRC scan against a Docker Image.

