

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: AmLight Kytos-NG

Team Members: Pedro Garcia, Ruben Cerna, Enrique Alfonso, Christopher Alvarez

Product Owner(s): Vasilka Chergarova

Mentor(s): Vinicius Arcanjo

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

The Kytos-NG project represents an innovative approach to network management through the development of a Software-Defined Networking (SDN) controller built on the principles of modularity, scalability, and extensibility. Leveraging the Kytos framework, the controller offers a comprehensive platform for orchestrating network operations, managing network devices, and implementing custom network services. Through a microservices architecture and RESTful API endpoints, Kytos-NG provides seamless integration with existing network infrastructures and enables developers to extend its functionality through custom plugins. This abstract highlights the project's focus on open-source collaboration, leveraging Python programming language, OpenFlow protocol, and modern software development tools to deliver a robust and flexible SDN solution. With its emphasis on modularity, scalability, and extensibility, Kytos-NG aims to revolutionize network management practices and empower organizations to build and operate dynamic and adaptive networks.

Table of Contents

INTRODUCTION

.....	4	
CURRENT SYSTEM		
.....	5	
PURPOSE OF NEW SYSTEM		
.....	5	

USER STORIES

USER STORIES		
.....	6	
IMPLEMENTED USER STORIES		
.....		6

PROJECT PLAN

HARDWARE AND SOFTWARE RESOURCES		
.....		10
SPRINTS PLAN		
.....	11	
<i>Sprint 1</i>		
.....	11	
<i>Sprint 2</i>		
.....	11	
<i>Sprint 3</i>		
.....	11	
<i>Sprint 4</i>		
.....	11	
<i>Sprint 5</i>		
.....	12	

<i>Sprint 6</i>	
.....	
..... 12	
<i>Sprint 7</i>	
.....	
..... 12	
SYSTEM DESIGN	
SYSTEM DESIGN	
.....	
..... 13	
DESIGN PATTERNS	
.....	
...	13
SYSTEM VALIDATION	
.....	
..... 14	
GLOSSARY	
.....	
..... 15	
APPENDIX	
.....	
.....	16
APPENDIX A - USER INTERFACE DESIGN	
..... 16	
APPENDIX B - SPRINT REVIEW REPORTS	
.....	17
APPENDIX C - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	
.....	20
REFERENCES	
.....	
.....	21

INTRODUCTION

Our goal with Kytos-NG was to finalize one of the Network Applications (NApps) already in use, namely SDNtrace. The aim of these NApps is to segment the various functions of

the webpage into separate sections, allowing for greater flexibility and independent scaling. Our objective was to enhance the modularity and scalability of the Sdntrace application within Kytos-NG, enabling more efficient management of its different functionalities.

Current System

The majority of the system is constructed using Python, encompassing everything from the backend architecture to the API connections and the startup files. For the frontend, we employ HTML, CSS, and JavaScript, leveraging the Vue.js framework. This framework enables us to utilize modern libraries like jQuery, facilitating seamless API responses. Through this approach, we craft responsive user interfaces that enhance the user experience.

Purpose of New System

A toolbar has been designed and implemented with several key features to enhance the user experience in network tracing operations. It includes a comprehensive set of input rows necessary for initiating traces, ensuring users have all required fields readily available. Additionally, a search button has been added, allowing users to efficiently generate traces based on the inputs provided. Upon executing a trace, a k-info panel opens to display the results of the latest trace. This panel includes a refetch button to update the results with the most current data. The toolbar also features a reset button that enables users to clear all input rows and start a new quickly. Moreover, an 'all traces' button is present both on the toolbar and within the single trace k-info panel. This button activates another k-info panel that provides an overview of all past traces, offering users easy access to and management of their tracing history.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

User Story 1: Review the specifications and functional requirements for the toolbar to ensure complete understanding of its capabilities and intended uses.

User Story 2: Design the layout of the toolbar to include input rows, ensuring a user-friendly interface for data entry.

User Story 3: Develop the front-end code for input rows in the toolbar to allow users to enter and edit tracing data.

User Story 4: Implement validation for each input row to ensure the data entered is within the expected format and range.

User Story 5: Create a search button on the toolbar that initiates the tracing process based on user input.

User Story 6: Develop backend logic to process tracing requests when the search button is clicked.

User Story 7: Integrate the search functionality with existing databases or APIs to retrieve trace results.

User Story 8: Design and implement a reset button that clears all inputs in the toolbar with a single action.

User Story 9: Ensure the reset button resets the state of the application without reloading the page.

User Story 10: Add an 'all traces' button to the toolbar for displaying a panel with historical trace results.

User Story 11: Implement the functionality to retrieve and display all traces from the system when the 'all traces' button is clicked.

User Story 12: Design the information panel to list all traces in a user-friendly and accessible manner.

User Story 13: Implement pagination or scrolling for the information panel if the number of traces is large.

User Story 14: Test the input rows for usability to ensure they meet the needs of various users.

User Story 15: Conduct usability testing on the toolbar to refine the interface based on user feedback.

User Story 16: Write documentation on how to use the toolbar effectively for new users.

User Story 17: Review the security implications of inputting and storing tracing data.

User Story 18: Implement security measures such as data validation and sanitization to protect against SQL injection and other attacks.

User Story 19: Optimize the performance of the search functionality to handle large volumes of data efficiently.

User Story 20: Develop error handling routines for the toolbar to manage failed traces or input errors gracefully.

User Story 21: Implement session management to preserve user input and trace results over the session lifecycle.

User Story 22: Integrate the toolbar with existing tools or platforms, ensuring compatibility and functionality.

User Story 23: Configure settings to customize the behavior of the toolbar based on user preferences or roles.

User Story 24: Deploy the toolbar in a testing environment to validate its functionality under real-world conditions.

User Story 25: Monitor the toolbar's performance post-deployment to identify any potential issues or areas for improvement.

User Story 26: Update the toolbar based on user feedback and emerging needs to ensure it remains relevant and useful.

User Story 27: Train support staff on how to troubleshoot issues with the toolbar to ensure user issues are resolved quickly.

User Story 28: Develop a mobile-responsive design to ensure the toolbar is accessible on smartphones and tablets.

User Story 29: Perform accessibility testing to ensure the toolbar is usable by people.

User Story 30: Plan and implement a feedback mechanism within the toolbar so users can report issues or suggest improvements directly.

PROJECT PLAN

The planning phase of the project was crucial in ensuring the successful realization of our goals within the Kytos-NG project. We embraced agile development techniques, which necessitated meticulous sprint planning to effectively manage our tasks and deliverables. Each sprint was carefully organized to prioritize key features and address critical issues, allowing us to iteratively enhance the platform's functionality and usability.

Throughout the sprint planning sessions, we collaborated closely with stakeholders to define clear objectives and establish realistic timelines for each iteration. By breaking down our project into manageable tasks and setting achievable goals for each sprint, we maintained a steady pace of progress and remained adaptable to changing requirements and priorities.

In addition to agile methodologies, careful consideration was given to the selection of components, both software and hardware, that would form the foundation of our project. We conducted thorough research and evaluation to identify technologies that aligned with our project's objectives and requirements.

For software components, we opted for frameworks and libraries that offered robust functionality, scalability, and flexibility. By leveraging open-source solutions and building upon existing tools and platforms, we were able to accelerate development and minimize overhead costs.

In terms of hardware, we carefully assessed our infrastructure needs and selected reliable and cost-effective resources to support our application's deployment and operation. This included provisioning servers, networking equipment, and other hardware components necessary to ensure optimal performance and reliability.

Overall, the planning phase of the project played a pivotal role in laying the groundwork for our development efforts and setting the stage for success. Through agile methodologies and thoughtful component selection, we were able to navigate the complexities of the Kytos-NG project and deliver a robust and feature-rich solution to our users.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- Hardware Resources:
 - Servers: High-performance servers may be used to host the Kytos-NG controller and associated applications.
 - Switches: SDN-compatible switches are necessary components for building the software-defined network infrastructure.
 - Routers: Depending on the network architecture, routers may be employed to manage traffic between different network segments or domains.
 - Network Monitoring Tools: Hardware devices for network monitoring and traffic analysis may be utilized to collect data for the SDN controller.
- Software Resources:
 - Linux Distribution: Linux-based operating systems such as Ubuntu Server or CentOS were used as the primary operating system for hosting the Kytos-NG controller and associated applications.
 - Python was the primary programming language used for developing the Kytos-NG controller and its plugins. Python's simplicity, readability, and extensive library support make it well-suited for rapid development and prototyping in SDN environments.
 - Kytos Framework: The Kytos framework serves as the foundation for building the Kytos-NG controller. This open-source framework provides a modular and extensible platform for developing SDN controllers, allowing developers to implement custom network management applications and services.

Sprint Plan

Sprint 1: Planning and Setup

User Story 1: Review the specifications and functional requirements.

User Story 17: Review the security implications.

User Story 22: Integrate the toolbar with existing tools or platforms.

User Story 23: Configure settings for toolbar behavior based on user preferences or roles.

Sprint 2: Frontend Development

User Story 2: Design the layout of the toolbar.

User Story 3: Develop the front-end code for input rows.

User Story 8: Design and implement the reset button.

User Story 10: Add an 'all traces' button to the toolbar.

Sprint 3: Backend Development

User Story 6: Develop backend logic for tracing requests.

User Story 7: Integrate search functionality with databases or APIs.

User Story 11: Implement functionality to retrieve and display all traces.

Sprint 4: Testing and Optimization

User Story 14: Test input rows for usability.

User Story 15: Conduct usability testing on the toolbar.

User Story 19: Optimize search functionality for performance.

User Story 20: Develop error handling routines.

Sprint 5: Documentation and Deployment

User Story 16: Write documentation for toolbar usage.

User Story 24: Deploy the toolbar in a testing environment.

User Story 27: Train support staff on troubleshooting.

User Story 30: Plan and implement a feedback mechanism.

Sprint 6: Enhancements

User Story 4: Implement validation for input rows.

User Story 13: Implement pagination or scrolling for trace results.

User Story 28: Develop a mobile-responsive design.

User Story 29: Perform accessibility testing.

Sprint 7: Finalization and Review

User Story 9: Ensure reset button functionality without page reload.

User Story 18: Implement security measures.

User Story 21: Implement session management.

User Story 25: Monitor toolbar performance post-deployment.

User Story 26: Update toolbar based on user feedback and needs.

SYSTEM DESIGN

Kytos Sdn platform is written in python, including all of their Napss. The user interfaces were written on javascript using the vue.js framework. Kytos uses many functionalities to provide a framework for developing SDN applications and services and make it as easy as possible. They use a controller such as Openflow and or other controllers. They also use other tools such as southbound Apis, northbound Apis to enable communication between the sdn controller and application or devices

Design Patterns

In order to create the user interface for the Sdntrace Napp we had to create a k-toolbar and k-info-panel. The toolbar will be used to input all the information required, to create a trace that will be shown in the k-info-panel.

SYSTEM VALIDATION

Functional Testing:

We conducted functional testing to verify that each feature of the toolbar functions as intended. This involved testing all user interactions, such as entering tracing data, initiating a trace, resetting inputs, and viewing traces.

Functional testing also included verifying the behavior of buttons like the search button, reset button, and 'all traces' button, ensuring they perform their respective actions accurately.

Input Validation:

We implemented validation mechanisms to ensure that the data entered into the input rows meets the expected format and range. This validation helps prevent errors and ensures the integrity of the tracing data.

Input validation tests were performed to validate various scenarios, such as entering invalid data, leaving fields blank, and entering data outside the acceptable range.

Performance Testing:

Performance testing was carried out to evaluate the responsiveness and efficiency of the toolbar under different load conditions. This involved testing the search functionality with varying volumes of tracing data to ensure it performs optimally.

Optimization techniques were applied to enhance the performance of critical functions, such as search queries and data retrieval.

Usability Testing:

Usability testing was conducted to assess the overall user experience of the toolbar. This involved gathering feedback from users regarding the toolbar's layout, ease of use, and intuitiveness.

Based on user feedback, refinements were made to the toolbar interface to improve usability and user satisfaction.

Compatibility Testing:

Compatibility testing was performed to ensure that the toolbar functions correctly across different browsers, devices, and operating systems. This involved testing the toolbar on various platforms to identify and resolve any compatibility issues.

GLOSSARY

SDN (Software-Defined Networking): An approach to network management that allows dynamic, programmatically efficient network configuration in order to improve network performance and monitoring using software-based controllers.

Kytos-NG: An extension of the Kytos SDN platform, specifically designed to be more scalable, robust, and user-friendly, facilitating easier network management and development of network applications.

Kytos Framework: An open-source Python-based framework for building SDN controllers. It is known for its modular architecture that allows developers to create custom solutions tailored to specific network requirements.

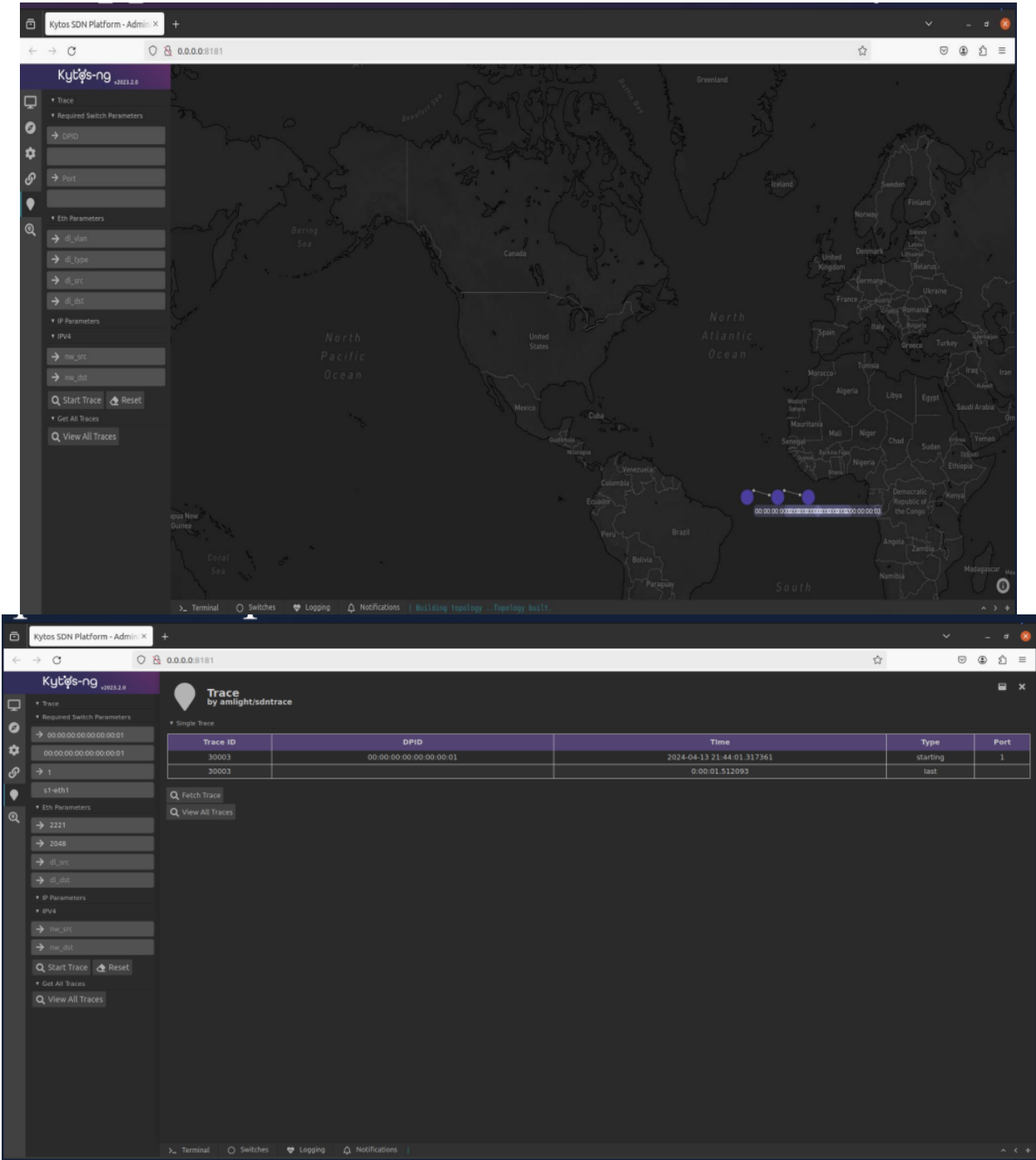
OpenFlow: A communications protocol that gives access to the forwarding plane of a network switch or router over the network, allowing the path of network packets through the network of switches to be determined by software.

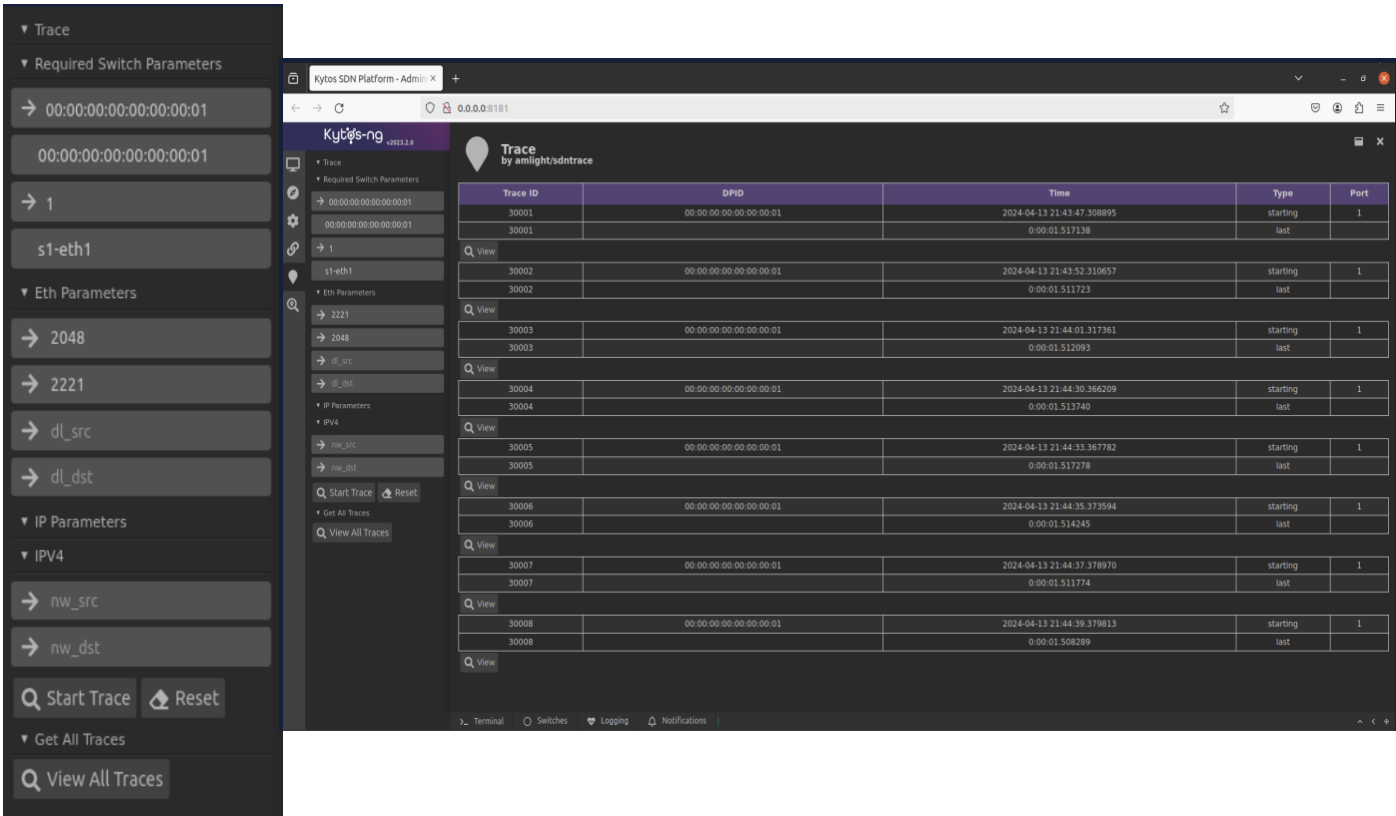
Docker: A software platform that allows developers to build, test, and deploy applications quickly through containers. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.

Git: A distributed version control system used to track changes in source code during software development.

APPENDIX

Appendix A - User Interface Design





Appendix B - Sprint Review Reports

Sprint 1 Review Report: Planning and Setup

Summary: Sprint 1 focused on project planning and setup. We reviewed project specifications, addressed security implications, and initiated integration with existing platforms. Although initial setup took longer than expected, we're now ready to proceed with detailed design and development planning.

Next Steps:

Address compatibility issues with existing platforms.

Proceed with frontend and backend development planning.

Begin initial implementation of frontend and backend components.

Sprint 2 Review Report: Frontend Development

Summary: Sprint 2 concentrated on frontend development tasks. We successfully designed the toolbar layout, implemented frontend code for input rows, and added reset and 'all traces' buttons. Despite layout challenges, we achieved significant progress in creating a user-friendly interface.

Next Steps:

Proceed with backend development for tracing functionality.

Conduct usability testing to gather feedback.

Begin documentation for toolbar usage.

Sprint 3 Review Report: Backend Development

Summary: Sprint 3 focused on backend development tasks. We developed backend logic for tracing requests, integrated search functionality with databases or APIs, and implemented functionality to retrieve and display traces. Despite some integration challenges, significant backend functionality was achieved.

Next Steps:

Conduct integration testing to ensure seamless functionality.

Begin optimization of search functionality for performance.

Initiate error handling implementation.

Sprint 4 Review Report: Testing and Optimization

Summary: Sprint 4 concentrated on testing and optimization. We conducted functional testing, input validation, and security testing to ensure robustness. Additionally, performance testing revealed areas for optimization, and usability testing provided valuable feedback for interface refinement.

Next Steps:

Implement optimization measures for search functionality.

Finalize error handling implementation.

Prepare documentation for release.

Sprint 5 Review Report: Documentation and Deployment

Summary: Sprint 5 focused on documentation and deployment tasks. We finalized documentation for toolbar usage and troubleshooting, and successfully deployed the toolbar in a testing environment. Feedback mechanisms were also implemented to gather user input for future improvements.

Next Steps:

Monitor toolbar performance post-deployment.

Update documentation based on user feedback.

Prepare for full-scale deployment.

Sprint 6 Review Report: Enhancements

Summary: Sprint 6 concentrated on enhancing toolbar functionality. We implemented input validation, pagination for trace results, and a mobile-responsive design. Accessibility testing was also conducted to ensure inclusivity.

Next Steps:

Perform final testing and bug fixes.

Prepare for full-scale deployment.

Plan for future enhancements based on user feedback.

Sprint 7 Review Report: Finalization and Review

Summary: Sprint 7 focused on finalizing the toolbar and reviewing project outcomes. We ensured the reset button functionality without page reload, implemented security measures, and finalized session management. Post-deployment monitoring revealed areas for improvement, and user feedback guided future updates.

Next Steps:

Address post-deployment issues and optimizations.

Prepare for full-scale deployment.

Evaluate project outcomes and lessons learned for future projects

Appendix C - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Kytos.ng Development Environment Installation Guide

<https://www.dropbox.com/scl/fi/mion5ua2y4dt7si89bqs8/CIARA-Labs-Tutorial.docx?rlkey=a72hsf0jl1agf8l7hh2lfc5285&dl=0>

REFERENCES

SLACK Kytos	https://kytos-ng.slack.com
Kytos Github	https://github.com/kytos-ng/
Kytos onboarding	https://hackmd.io/XpXMuGMESz-jCUdwN5TE7g
Kytos WEB UI docs	https://github.com/kytos-ng/kytos/blob/master/docs/developer/web-ui.rst
	https://github.com/kytos-ng/kytos/blob/master/docs/developer/components.rst
API Kytos	https://kytos-ng.github.io/api/api.html
Pull request and CLA agreement	https://kytos-ng.github.io/procedures/prs.html
Javascript V2	https://vuejs.org/v2/guide/
Tutorial: Vue js	https://www.vuemastery.com/courses/intro-to-vue-js/vue-instance/
	https://www.linkedin.com/learning/learning-vue-js-8602681/is-vue-for-you?autoplay=true&u=42567804
	https://www.linkedin.com/learning/vue-js-2-essential-training/why-you-should-learn-vue-js?autoplay=true&u=42567804
Setting up your dev environment	https://github.com/kytos-ng/documentation/blob/master/tutorials/naps/development_environment_setup.rst
Introduction to SDN (Software-Defined Networking)	https://www.youtube.com/watch?v=DiChnu_PAzA
Introduction to OpenFlow	https://www.youtube.com/watch?v=l25Ukkmk6Sk
AmLight ExP Network	https://www.youtube.com/watch?v=nu3kByNloPI