Working as a Software Engineer Trainee at Nokia Finland

Khang Duong

Haaga-Helia University of Applied Sciences
Haaga-Helia Bachelor's Degree
Bachelor of Business Information Technology
2024

Abstract

Author(s)
Khang Duong

Degree
Bachelor of Business Information Technology

Report/Thesis Title
Working as a Software Engineer Trainee at Nokia Finland

Number of pages and appendix pages
64 + 0

This thesis, "Working as a Software Engineer Trainee at Nokia Finland " presents an 8-week diary-style exploration of a traineeship, focusing on daily experiences, challenges, and professional growth at Nokia. The central objectives revolve around two themes: adaptability and learning agility, and effective cross-team collaboration, chosen to elucidate the diverse nature of a Software Engineer Trainee's role.

Situated within Nokia's DCAP department, the trainee contributes to both the Simca and Inspector teams, engaging in real-time data processing and front-end development with the latest enterprise library, respectively. The theoretical framework integrates concepts of adaptability, learning agility, collaboration, and technical mastery, guiding the analysis of daily duties.

The methodology includes daily reflections, weekly assessments, and mentor feedback. Results and discussions, presented in the past tense, focus on identified development areas, challenges, and skill evolution in programming, communication, and problem-solving.

The abstract, adhering to guidelines, uses neutral English, passive voice, and the third person. It stands as an independent entity, summarizing the entire report concisely. The timeline covers 8 weeks, capturing day-to-day transformations. This abstract provides a nuanced glimpse into the trainee's professional journey, contributing insights into the dynamic realm of code and culture at Nokia Finland.

Contents

Introduction

The focus of this thesis is a working diary that delves into an in-depth analysis and reporting of my daily experiences during an 8-week period as a Software Engineer Trainee at Nokia Finland. The decision to choose a working diary format stems from the desire to offer a personal, authentic, and reflective account of my journey, capturing the nuances of daily work, challenges faced, and personal growth achieved in a dynamic and innovative environment. The choice of a working diary format was motivated by the recognition of the unique value that personal narratives bring to understanding the professional journey of a Software Engineer Trainee. This format allows for a detailed exploration of the daily experiences, providing a genuine insight into the intricacies of the role and the dynamic nature of the work environment.

The employer for this traineeship is Nokia, a global leader in technology and telecommunications. Nokia is renowned for its contributions to the evolution of mobile communication and continues to be at the forefront of technological innovation. The work environment at Nokia Finland is characterized by its positive atmosphere, with friendly colleagues and supportive management. Coffee and snacks are provided on the floor, contributing to a collaborative and enjoyable workspace.

As a Software Engineer Trainee, I am currently involved in two teams within the DCAP (Data Collections and Analytics Platform) department: Simca and Health Check (Inspector) UI. The Simca team focuses on presenting different statistics such as counter graphs, tables, parameters, and logs from Imsiproviders, SPD (SIMCA Probe Device). Centralized O&M Function (COF) works as a data source to Web UI, while the Health Check UI team concentrates on give overview of LTE Network health near real time. These roles allow for a well-rounded experience in both backend and frontend development, utilizing C# and React, Javascript technologies.

I have been actively performing my current job tasks since joining the Simca team in September 2023, and subsequently, the Health Check UI team in December 2023.

The framework used daily:
1. Locally installed software: Visual Studio Code, Visual Studio, Git, Node, Teams, Microsoft Outlook.
1. Tools and services: Gitlab, Jira
2. Technologies and frameworks: C#, Wireshark, SharpPcap, .NET, Javascript, Typescript, Reactjs, Jest.

Simca:

- *Expertise:* Network programming, real-time data processing, UI design.

- *Tasks:* Designing a program to read network data directly from the server's built-in network interface card and implementing real-time traffic filtering. Contribute in Simca UI to create new features, writing test cases.

Inspector:

- *Expertise:* Front-end development, UI design.

- *Tasks:* Refactoring front-end code for the UI project and updating to the latest enterprise library.

Due to restrictions imposed by Nokia's data privacy policy, I am unable to discuss specific details about the business or the solutions we are working on. Consequently, there will be no sources related to the Nokia Design System or Nokia Library. Instead, I will provide descriptions based on my own observations and understanding of these aspects. The focus of my thesis will primarily be on reviewing my daily tasks and analyzing the approaches I take when completing these tasks. Additionally, I will reflect on the new technologies and knowledge I have gained during my time as a Software Engineer Trainee at Nokia. While I may not be able to delve into proprietary details, I will strive to offer valuable insights and reflections based on my experiences within the confines of the data privacy policy.

In this thesis, I draw upon multiple sources from various domains relevant to software engineering and related technologies. The sources include authoritative documentation sites such as Jira, which provide essential information on project management and software development processes. Additionally, I refer to professional blogs, such as Alex Anie's article on the "UseLocation Hook in React Router" on Medium, to gain insights into practical implementation techniques and best practices. For a deeper understanding of version control systems, I consult resources like the "Git Clone Command" tutorial on Studytonight and articles discussing the differences between Git merge and rebase on Data Science Simplified. These sources provide valuable insights into Git workflows and strategies for efficient collaboration within development teams. In addition to version control, I explore networking protocols and technologies through resources such as the "GTPv2 - GPRS Tunnelling Protocol Version 2" glossary entry on Mpirical and documentation on the "PFCP Protocol" from Nokia. These sources deepen my understanding of network infrastructure and protocols essential for modern software development.

Furthermore, I refer to tutorials and guides on software deployment and containerization technologies provided by platforms like Docker and Packet.net, enabling me to explore efficient deployment strategies and infrastructure management techniques.To enhance my knowledge of frontend development, I delve into articles discussing JavaScript frameworks like React and tools like Redux Toolkit, as well as emerging trends such as micro frontends. Sources such as the article on "A Simple Explanation of React.useEffect()" by Dmitri Pavlutin on Dmitri Pavlutin Blog and Cam Jackson's piece on "Micro Frontends" on Martinfowler.com provide valuable insights into these topics. In instances where I encounter bugs or challenges, whether stemming from npm packages or my own oversight, I turn to Stack Overflow for references and solutions. As a professional online community for developers, Stack Overflow serves as a valuable platform for knowledge exchange and collaboration. Here, developers can ask questions, share insights, and engage in discussions, often finding solutions to their queries readily available. Leveraging this resource not only saves time and effort in debugging but also fosters a sense of community and collective learning within the developer community.

During the monitoring period of my thesis work, I hope to see significant development in the following areas:

- **Programming Skills:** Improvement in both C# and JavaScript, particularly in the context of real-world projects.

- **Communication Skills:** Enhanced collaboration with two distinct teams, fostering effective communication.

- **Problem-Solving Skills:** The ability to tackle novel challenges, particularly in the Simca team where tasks differ significantly from my background.

- **Self-Learning Skills:** Adapting to new and unfamiliar tasks, reflecting a proactive approach to continuous learning.

From a sustainability perspective, the thesis work aims to contribute to the sustainable development of skills and knowledge. By focusing on improving programming, communication, and problem-solving skills, the goal is to create a sustainable foundation for ongoing professional growth and adaptability.

The boundaries of the thesis work include:

- **Time Constraints:** The thesis work is limited to the 8-week monitoring period.

- **Focus on Work Experience:** The primary focus is on daily work experiences, challenges, and growth as a Software Engineer Trainee.

- **Confidentiality:** Any sensitive or confidential information related to Nokia's projects will be excluded to maintain confidentiality and adhere to ethical considerations.

These boundaries help maintain the scope and integrity of the thesis while providing a meaningful and insightful reflection on the traineeship experience.

Professional concepts	Description
Adaptability and Learning Agility	Continuous learning and cognitive flexibility. Being resilient and open to new experiences.
Effective Cross-Team Collaboration	Emphasizes communication, interpersonal skills, conflict resolution, and shared goals.
Technical Mastery in C# and JavaScript	Involves understanding syntax, algorithms, data structures, and best practices in coding.
DCAP	Data Collections and Analytics Platform
Permission	Permission in Software refers to the level of access and actions a user or group of users is allowed to perform within the system.
UI	User Interface
SPD	SIMCA uses Napatech Network Interface Card for data capturing. SIMCA Probe Device (SPD) contains the Napatech® card and data provider solution. Otherwise SIMCA can use server's built-in Network Interface Card (NIC) for data capturing. The Network Interface Card will perform the same functionality as Napatech card.

ImsiProvider	IMSI Provider receives S11 / N4 messages from SPD and keeps track of subscriber sessions.
COF	Centralized O&M Function (COF) works as a central piece on the back end. COF is responsible for dataflow between SIMCA and outer connections such as O&M Mediator and Web UI.
Health Check UI	Give overview of LTE Network health near real time
CSS	Cascading Style Sheets (CSS) is a language for describing the appearance of a document written in a markup language like HTML. (W3school 2022, CSS Tutorial)
Git	A free and open-source distributed version control system that can handle modest to extremely big projects quickly and efficiently. (Git-scm.com, Git)
Javascript	A lightweight, interpreted programming language. While JavaScript is best known as a scripting language for Web pages, it is also used in a variety of non-browser settings, including Node.js, Apache CouchDB, and Adobe Acrobat. (Developer.mozilla.org 2021. JavasScript)
C#	Object-oriented programming language developed by Microsoft as part of the .NET framework. It is primarily used for building a wide range of applications, including desktop, web, mobile, and gaming applications. While JavaScript is often associated with scripting for web pages, C# is used for a broader spectrum of development tasks.
React	The Progressive JavaScript Framework is a set of tools for developing progressive web applications. React is a web-based UI framework that is progressive and incrementally adaptable. (github.com/reactjs, React)

Table 1: Key professional concepts

Description of the initial situation

Analysis of your current work

In Nokia, our development team operates within a structured framework, with tasks managed through Jira, a comprehensive project management tool. Over the course of my engagement with the team, I've been entrusted with a variety of specific tasks aimed at contributing to the development and enhancement of our software products.

Key tasks in my current role include:

Setting up Local Development Environment: When I join older projects, getting everything set up can be quite a task. Take the Simca project, for example: It was put together using .NET 5 a while back, but now we're looking at moving it over to .NET 7. This switch means we've got to make a bunch of adjustments and updates to keep everything running smoothly. It's a good reminder of how important it is to keep up with the latest tech trends and make sure our projects stay current.

Designing New Features: Being involved in enhancing our software products means tackling several tasks. It starts with understanding what users need, diving into their requirements deeply. Then, it's about coming up with new ideas that not only meet those needs but also make the user experience even better. This brainstorming involves teamwork, research, and bouncing ideas around. Once we've got a solid plan, we break it down into specific tasks for development. This includes detailing the technical side and figuring out how to put everything into action. Throughout this process, clear communication and teamwork are key to making sure everything runs smoothly and the new features are a success.

Refactoring the Codebase: One of my daily tasks is refactoring the codebase using the newest Nokia's library for User Experience. The goal is to ensure that the new codebase, utilizing the updated library, maintains the same functionality, features, and user interface as the old one. This involves thorough research into the documentation of the new library to understand its capabilities and how it aligns with our existing codebase. Once I've familiarized myself with the library, I apply it to the new codebase, making necessary adjustments and optimizations to ensure a seamless transition while preserving the integrity of our user experience.

Code Review: While at Nokia, my role extends beyond developing my assigned tasks; I'm also responsible for reviewing and testing the work of others. In contrast to my previous company, where we had dedicated testers for our features, here I wear both hats: developer and tester. To excel as a reviewer, I rely on my strong programming skills and keen eye for clean code to pinpoint coding mistakes that others might overlook. It's crucial for me to ensure that the code written by fellow developers is clear and readable.

Testing for Other Developers: My typical testing process involves evaluating Docker images that other developers have pushed into the pipeline. I utilize these Docker images to conduct comprehensive end-to-end testing on the real server environment. As an end-user, I thoroughly assess all functionalities, ensuring they perform as expected and meet the desired specifications. Once testing is complete, I revert the server back to the previous version, maintaining stability and consistency in our environment. This workflow allows me to validate the integrity and functionality of our software updates before deploying them into production.

Evaluation and development plan

My colleagues have given me good feedback on the task that I've completed. I believe there are areas that I would need to improve. As a novice, I acknowledge that my performance still relies heavily on instructions from co-workers or written documentation. While I am eager to learn and develop my skills, I often find myself needing guidance and support to complete tasks effectively. My level of proficiency in tasks such as setting up local development environments and designing new features is still in the early stages. Overall, while I am committed to improving and gaining more experience, I recognize that I am currently at a novice level in terms of meeting the competency requirements of my job.

To effectively fulfil these tasks, a diverse set of knowledge and skills is required. This includes proficiency in programming languages such as C# and JavaScript, familiarity with development tools and frameworks, understanding of software design principles, and strong problem-solving and analytical abilities.

In terms of professional development, I am currently at a stage where I am actively seeking to expand my skill set and deepen my expertise in areas such as software architecture, design patterns, and emerging technologies. By investing in continuous learning and skill development, I aim to stay abreast of industry trends and advancements and contribute effectively to the success of our projects.

As I reflect on my current level of competence, I recognize both strengths and areas for growth. By leveraging my existing skills and proactively addressing areas for improvement, I am committed to delivering high-quality solutions that meet the demands of my role and contribute to the overall success of our team and organization.

Stakeholders

My personal work can affect the following groups:

Simca and Inspector Team:
- Product owner: who works as a bridge between the development team and the stakeholders. He also has the role as a project manager in my team.
- Line Manager: who responsible for keeping communication flowing smoothly throughout the entire team.
- Technical Lead: who setting up and organizing weekly meetings for everybody to update their work.
- Back-end developers: who ensure the behind-the-scenes aspects of our software, such as managing databases, processing requests, and ensuring that everything on the server side runs smoothly.
- Front-end developers: who handling the user-interface of the product
- Testers: who oversee the automation of the pipeline, ensuring that code changes are continuously integrated, tested, and deployed in an efficient and reliable manner.

Other stakeholders:

They can come from areas such as marketing, sales, customer support, or any other department. As a member of the development team, I may need to collaborate with them on various aspects of our projects. This collaboration could involve gathering requirements, seeking support for tasks, or coordinating efforts to achieve project goals. Effective communication and collaboration with these

stakeholders are essential for the success of our projects and the overall functioning of the organization.

Interaction situations

My primary interaction skills at work are primarily devoted to online meetings. Most of my coworkers choose online work because it fits in better with their work requirements, and my project team was initially composed of people from several departments located in different parts of the world. Everyone is therefore accustomed to and at ease with conducting meetings and phone calls online. In addition, I try to visit the office since I wanted to experience the life of a Software Engineer at Nokia's office (One of the biggest offices in Helsinki Finland).

I usually have weekly meetings with Simca team on Monday and Thursday, and I also have daily meetings (except Monday) with the Inspector team to talk about any problems or challenges we're facing. Communication channels that we use daily includes Microsoft Teams or Email. Moreover, our department have a weekly catchup meeting called DCAP Finland (Coffee break and rumor sessions), and our whole business unit have a monthly catchup meeting as well, these two meeting are usually occurring on Friday. Every week, my Line Manager and I have a catch-up meeting where we talk about my overall progress and growth within the team as well as any issues I'm having at work. I find that the main challenge I face is dealing with online communication. Since people have more flexibility with their workday when they're not in the office, it can be hard to get in touch with them when I need to discuss something. They're not always online or quick to respond to messages. This is especially tricky when I need to show or explain something to them, which is much easier to do in person.

Firstly, I aim to enhance my digital communication skills, ensuring that my messages are clear and to the point. This should help reduce the need for immediate clarifications. Secondly, I'm focusing on being patient and adaptable since responses might not come as quickly as they would in an office. Also, dealing with stakeholders across different time zones has been challenging. To handle this, I'm improving my organizational skills and scheduling communications to fit everyone's local working hours. This involves careful planning ahead, considering time zone variations and potential scheduling clashes.

Diary entries

Observation week 1 (December 25 – December 29, 2023)

Monday 25th December 2023

It's been approximately four months since I began my traineeship at Nokia Finland, and during this time, I've gradually adapted to the working environment. Although I've made some progress in getting used to the dynamics of the company, being part of a large organization like Nokia means that my daily tasks are constantly evolving, especially since I'm part of a highly dynamic development team.

Upon starting my traineeship, the company provided me with essential office equipment to facilitate my work, including a Lenovo ThinkPad (ThinkPad T14s Gen 2), a smart mouse (Logitech M185), a keyboard (Microsoft Designer), and a set of headphones (Jabra Evolve2 65). Adjusting to the new laptop, particularly because I'm accustomed to using a MacBook, took some time. It's also my first experience with a laptop running on the Windows operating system, which required additional adaptation. Furthermore, the keyboard layout using the Nordic configuration presented a new challenge for me.

Since today is Monday, my schedule primarily revolves around my involvement with the Simca team. We had a scheduled meeting at 9:30 AM, where our Team Lead, orchestrated the discussion and delegated tasks among team members. Our conversation entered on our ongoing work and the challenges we've been facing, particularly regarding the installation and piloting of the Simca product. This week, my primary task is to design an independent program capable of listening to a network interface card while operating in promiscuous mode. To accomplish this successfully, I recognize the importance of comprehensively understanding the workflow and intricacies of the Simca product. Thus, I've initiated research on the architecture of Simca to gain a deeper understanding of its operations.

During my discussion with Tatu, he emphasized the significance of delving deeper into networking concepts and modifying the SPD component within Simca. Currently, Simca relies on the Napatech card to receive Core network data, leveraging its powerful capabilities for real-time traffic filtering. My assignment involves designing a version of Simca that can function seamlessly without the Napatech card, utilizing the server's built-in network card instead.

Following the discussion with my Team Lead, I embarked on an extensive research journey, focusing on understanding network data, network interface cards, techniques for reading data from network interface cards, and the specific functionalities of the Napatech card. Given the complexity of the task at hand, I typically dedicate my Mondays solely to tasks related to the Simca project, ensuring I allocate sufficient time and attention to its successful execution.

Tuesday 26th December 2023

Today, I made the decision to head into the office, opting for a change of scenery from my usual remote work setup. Typically, I find myself at the office about three or four days per week, finding the balance between collaborative in-person interactions and focused remote work.

My day kicked off with a meeting with the Inspector team, a group I recently joined just last week. During our discussion, we identified our primary task as the refactoring of the codebase for the Inspector User Interface project. Our goal is to transition from using the existing cfswidget library to

the newest library provided by Nokia, known as CCFK. This transition involves migrating various components of the codebase to leverage the functionalities offered by CCFK. **In this project, my role centers around refactoring and testing two key components: the Report and Alarm pages. These components are built using Reactjs and JavaScript. For the coming week, I've decided to focus my efforts on refactoring the Report pages.**

With guidance from my Team Lead on the project, I received some helpful hints to kickstart my tasks. They suggested starting with "git clone, npm install, npm start" as key steps. Following this advice, I began my research to understand these commands and their significance within the context of our project. Firstly, I delved into "git clone," which involves creating a copy of the project repository from the version control system, Git, onto my local machine. This step ensures that I have access to the latest version of the project codebase to work on.
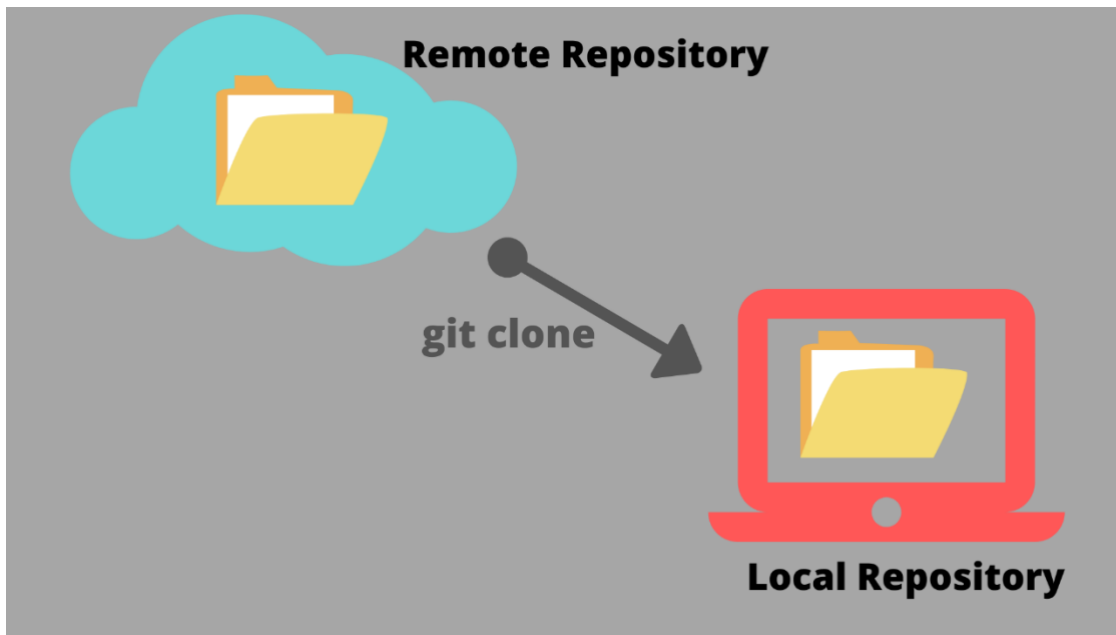

Figure 1. How to clone a reposistory from git ("Git Clone Command" 2023)

After clone the Inspector project which is named basic-hcguiv2 to my local laptop, I started to investigate the package.json file of the project. Before start to use npm (Node Package Manager), I need to download the right version of it.
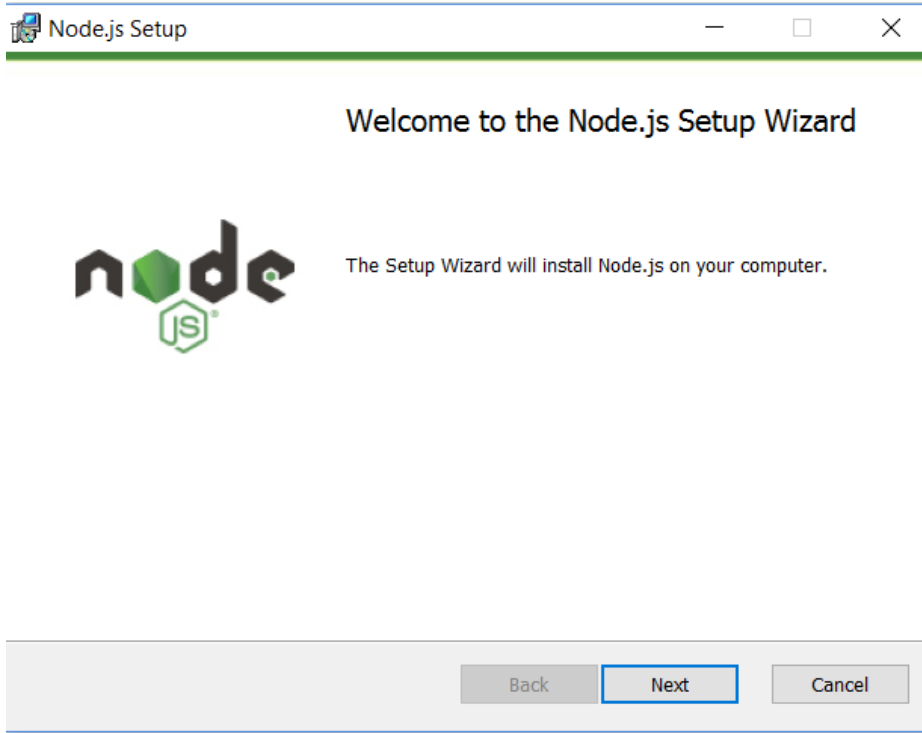
Figure 2. Install Node.js and NPM ("How to Install Node.js and NPM on Windows and Mac?" Jaydeep Patadiya, 2023)

"npm install" is used to install all the necessary dependencies and packages required for the project to run smoothly. It ensures that I have all the required libraries and tools in place to execute the project successfully. Finally, I run the "npm start" command to launches the development server, enabling me to view and interact with the project in a local environment. It's a crucial step in the development process, allowing me to test and iterate on the codebase effectively.

Wednesday 27th December 2023

Today I continue with daily meeting of Inspector team, we discussed about that my Team Lead created Jira ticket for me where I can log hours that I spend for this project. Jira is a software application developed by the Australian software company Atlassian that allows teams to track issues, manage projects, and automate workflows (Jira, 2022).

In our development process, we primarily communicate through Microsoft Teams and manage our projects using Jira. When we need to provide updates on the tickets we're working on, we usually add them to the comment section of the respective ticket for documentation purposes. If there's any information that needs to be shared with other team members, we mention them in the comments, triggering an automatic email notification to the relevant team member. This helps ensure that everyone stays informed and updated on the progress of our work. When started to look into the ticket, I encountered an issue with a Jira ticket where I couldn't change the status from "Open" to "In Progress" or log hours. It seemed like I didn't have the necessary permissions to make these changes. I decided to reach out to my Team Lead for assistance, and he advised me to contact my Line Manager to resolve the problem.
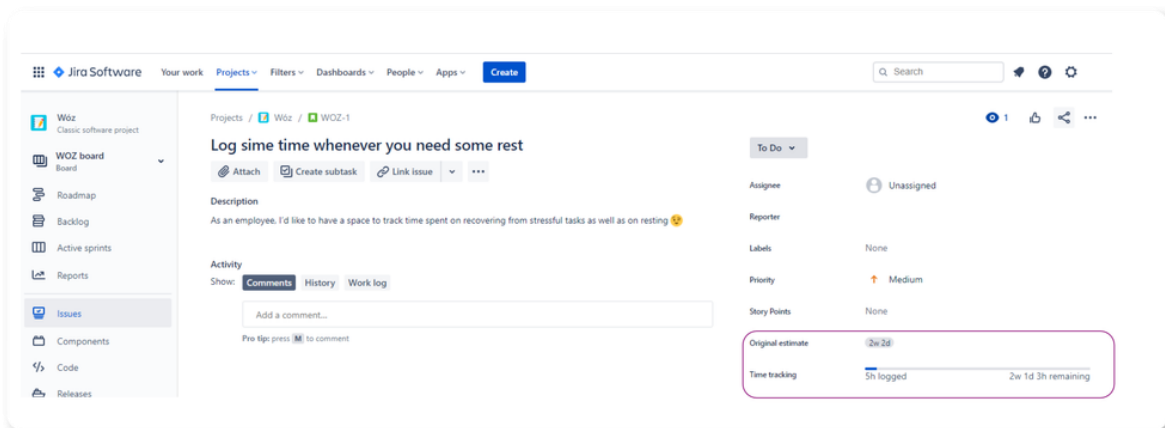
Figure 3: Logged hours in Jira ("How to view all logged time by users in Jira?" 2021)

In a large organization like Nokia, ensuring security and permissions for every application or tool is crucial. Following my Team Lead's suggestion, I contacted my Line Manager, who directed me to the Program Manager responsible for handling access requests for Jira tickets. After reaching out to the Program Manager and explaining my issue, he promptly added me to the DCAP group in Fusion access, granting me the necessary permissions. As a result, I was finally able to successfully update the Jira ticket as needed. This experience highlighted the importance of proper access management in ensuring smooth workflow processes within the company.

Thursday 28th December 2023

Since today is Thursday, I have two meetings scheduled with both the Simca and Inspector teams. The Simca meeting starts at 9 AM, where we discuss our progress since Monday. During this meeting, I reported to my Team Lead about my research on the Simca product and how the data flow operates internally. My Team Lead provided me with a lot of encouragement, stating that I have made good progress. He suggested that it's now time to test the real network data from the Remote Desktop. My task is to connect to the Remote Desktop to run the SPD (Simca Probe Device) components and observe how data is captured by the Napatech card. The Napatech Card is located within one of our host machines in the Espoo labs. This is why I need to connect to the

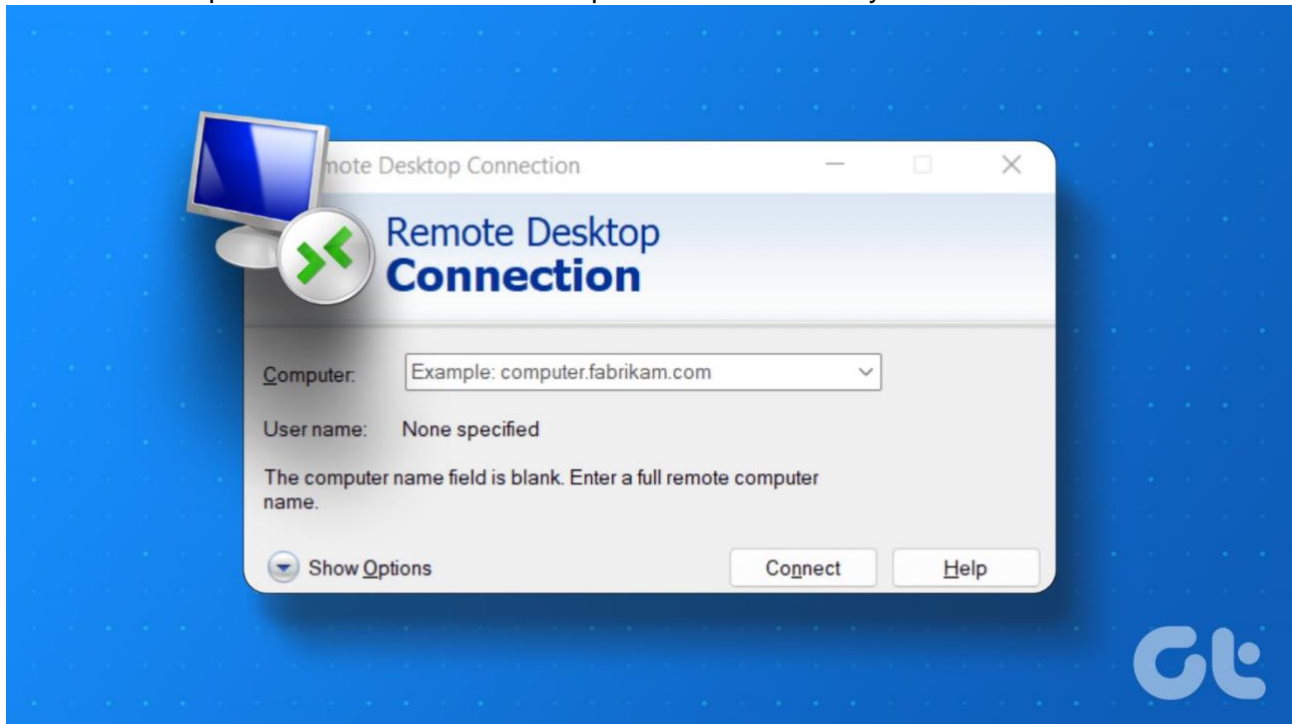Remote Desktop to access and utilize the Napatech card effectively.

Figure 4: Connect to Remote Desktop ("8 Quick Ways to Open Remote Desktop Connection Tool in Windows 11", Abhishek Mishra, 2023)

After access to Remote Desktop, I started to run SPD component and receive the GTPV2 and PFCP network data. GTPV2 stands for GPRS Tunneling Protocol V2 which is 4G data. In LTE, GTP tunnels manage communication between nodes. Each tunnel has a TEID, IP address, and UDP port number. GTPv2's updated control plane facilitates control messages between MME, S-GW, PDN GW, etc. (MPIRICAL, 2024). PFCP stands for Packet Fowarding Control Protocol which is 5G data. The PFCP protocol runs on top of UDP/IP. The UDP destination port for a request message uses port 8805, which is the registered port number for the PFCP protocol (Nokia, 2023).

At Nokia, my workflow offers a considerable degree of flexibility. Given that I'm fully engaged in two projects simultaneously, I've structured my workweek to ensure I allocate sufficient time and attention to each. Typically, I dedicate Mondays and Thursdays to tasks related to the Simca project, reserving the remaining days for Inspector-related work. This approach allows me to maintain a balanced focus and effectively manage the demands of both projects.

Following the Simca meetings on Mondays, I transition seamlessly into meetings with the Inspector teams. During these sessions, I update my Team Lead on the progress made, including the successful access to Jira tickets, which we both find gratifying. I reassure him of my commitment to diligently tackle the tasks at hand, emphasizing my intention to prioritize Simca responsibilities for the remainder of the day before delving into Inspector tasks the following day.This structured approach not only ensures that I can address urgent matters promptly but also enables me to allocate sufficient time and attention to each project, maximizing productivity and ensuring successful outcomes. By maintaining this balance and clear communication with my team, I strive to contribute effectively to the progress and success of both the Simca and Inspector projects at Nokia.

Friday 29th December 2023

Today is packed with meetings for me, starting with one with the Inspector team and then a department-wide gathering later in the day. Fridays are particularly enjoyable for me because it means a trip to the office, where I eagerly anticipate the coffee break. It's a fantastic opportunity to mingle with colleagues from other floors and catch up with what's happening around the office.

My day kicked off with the Inspector meeting, where we welcomed Trainee A from Tampere, who has joined our project. She'll be focusing on Topology views, adding another valuable member to our team. However, a slight hiccup arose when Trainee A and I realized that we both had conflicting modifications to the ImpactDialog file, affecting both the Report and Topology views. Recognizing the potential for conflict and the time it could take to resolve, I took the initiative to propose a solution during the meeting.
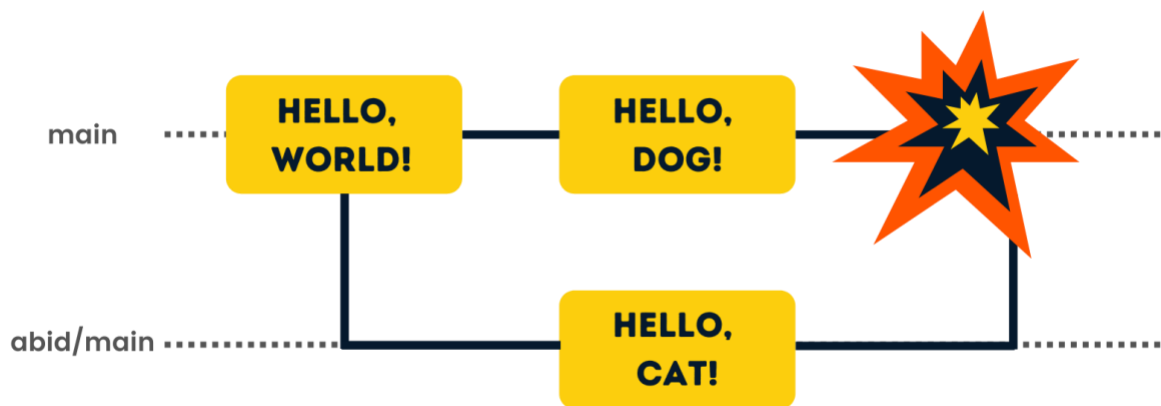

Figure 5: Conflict in Git ("How to Resolve Merge Conflicts in Git Tutorial", 2022)

I suggested that I take on the responsibility of modifying the ImpactDialog file, relieving Trainee A of the burden and streamlining our workflow. This way, Trainee A could simply pull my branch and integrate it into her task without any complications. Fortunately, my team agreed with this approach, and we swiftly moved forward with our respective tasks, ensuring smooth progress and effective collaboration within the team.

In the afternoon, following our invigorating coffee break session, I returned to my workstation to delve into the Reports view task. First and foremost, I encountered an issue as the old library (cfswidget) ceased to function. To address this, I attempted to transition the entire setup to the new CCFK library.

However, upon implementation, an error surfaced indicating that the state was undefined, consequently preventing the Report page from loading. Employing the F12 key on the keyboard to access the website's inspect feature, I identified that the problem stemmed from the incorrect rendering of the location state within the components. Promptly reaching out to my teammates for assistance, we convened for a brief meeting to brainstorm potential solutions. One suggestion that emerged was to utilize the useLocation hook from "react-router-dom". This hook serves to retrieve the current location of a React component, leveraging the location object to access key details such as pathname, state, hash, and search (Anie, 2023).

Despite it being my first time utilizing the useLocation hook, I welcomed the opportunity to expand my technical skill set. After implementing the useLocation hook to declare the location variable, I successfully resolved the issue, ensuring the Reports pages rendered correctly in the web

browser. Today marked a milestone as I made my inaugural commit to my branch in GitLab, succinctly documenting the progress with the commit message "Make the Reports view show".

Week 1 evaluation

During the first week at Nokia, I embarked on a journey of familiarizing myself with the company's work culture, projects, and responsibilities. This weekly evaluation aims to reflect on my experiences, identify areas for growth, and set goals for improvement in the upcoming weeks.

In the Simca project, I delved into researching the product architecture, particularly focusing on understanding the data flow within the system. This involved attending meetings, communicating with team members, and familiarizing myself with tools such as Jira for project management. Additionally, I initiated the process of setting up my development environment and began exploring the codebase.
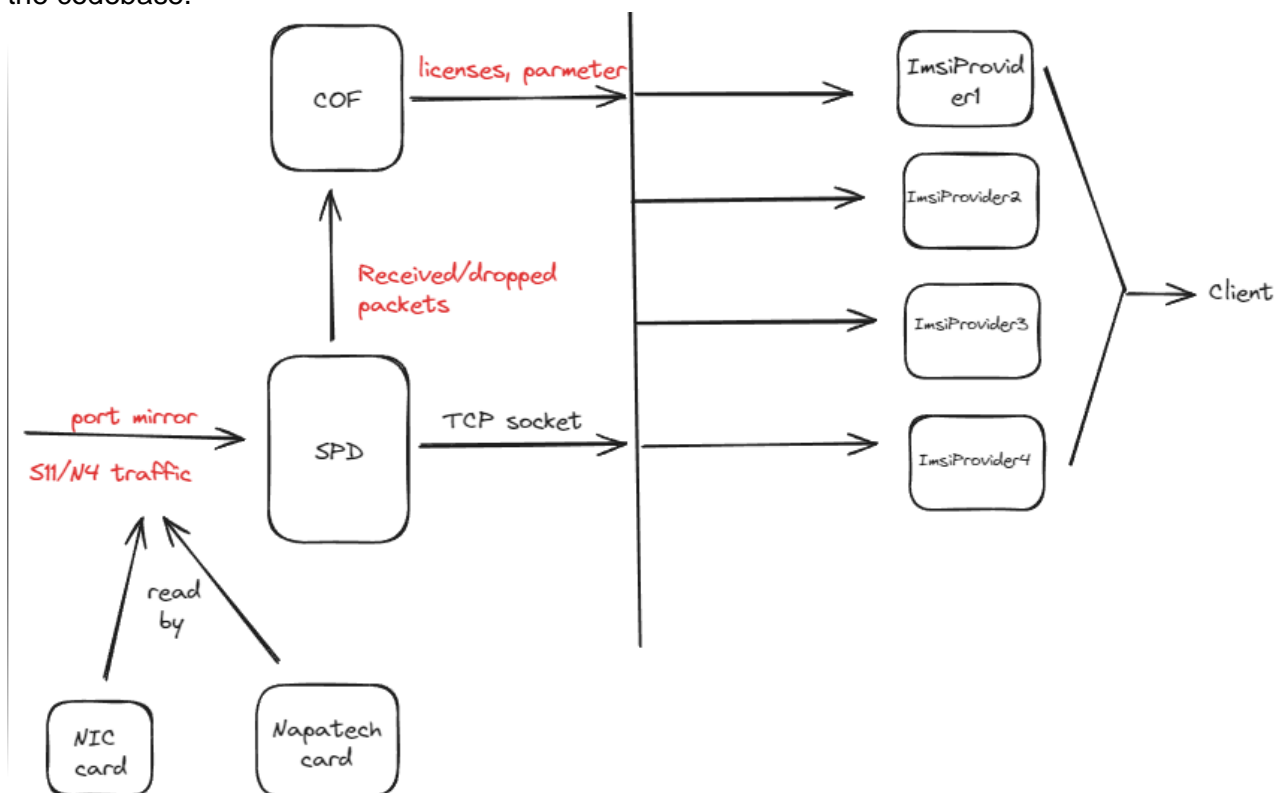
Figure 6: Simca product architecture created by me after researching and summarizing.

In the Inspector project, I encountered a collaborative challenge early on when Trainee A and I discovered conflicting modifications to a shared file. Demonstrating proactive problem-solving skills, I proposed a solution during our team meeting, which was well-received by the team. This experience highlighted the importance of effective communication and collaboration within the team.

In reflecting on my experiences, I found guidance from external sources invaluable. Books such as "The Agile Samurai" by Jonathan Rasmusson and "Clean Code" by Robert C. Martin provided insights into Agile methodologies and software development best practices, which I applied to my work at Nokia. Additionally, articles from industry blogs such as Medium's "Better Programming"

enhanced my understanding of project management tools and collaborative strategies. Comparing my approach to challenges encountered during the week with strategies proposed in external sources revealed alignment with established practices in software engineering. For example, techniques suggested in "The Agile Samurai" for managing complex project architectures resonated with my efforts to understand the data flow within the Simca project. Similarly, insights from "Clean Code" influenced my approach to code review and collaboration in the Inspector project.

Overall, my first week was characterized by a steep learning curve as I navigated through new projects, tools, and workflows. Reflecting on my experiences, I recognize the importance of continuous learning and adaptation in a dynamic work environment. Lessons learned during the first week, including the value of effective communication, proactive problem-solving, and seeking guidance from external sources, will inform my approach to future projects and professional development. Despite encountering some challenges, I embraced them as opportunities for growth and learning.

Looking ahead to the next week, I aim to focus on refining my technical skills and deepening my understanding of the projects I'm involved in. Specifically, I plan to:
- Further familiarize myself with the Simca product architecture and data flow to enhance my contributions to the project.
- Improve my proficiency in using development tools such as GitLab and Jira to streamline my workflow and collaboration with team members.
- Proactively seek mentorship and guidance from senior team members to accelerate my learning and development.
- Continuously communicate with my team members to ensure alignment and clarity on project goals and tasks.

By proactively addressing these areas for improvement, I am confident that I can enhance my performance and contribute more effectively to the success of the projects in the upcoming week.

Observation week 2 (January 1 – January 5, 2024)

Monday 1st January 2024

Today marks the first day of the year, and like many, the majority have opted to stay away from the office. However, I've decided to take a different approach. With a positive attitude, I made my way from the train station to the office, feeling energized and ready to start the year off on a productive note.

As I walked, I thought about my goals for the coming year. One thing I've resolved to do is spend more time at the office. I've always found that I work more efficiently and comfortably in the office environment, so I want to take advantage of that.

Despite it being a holiday and most people being away, I felt a sense of purpose being in the office. My first task of the day was a meeting with the Simca team. During the meeting, I had the opportunity to share a program I've been working on. Using SharpPcap and PacketDotnet, I demonstrated how I could read and filter UDP and TCP packets from the network interface card of my local laptop.
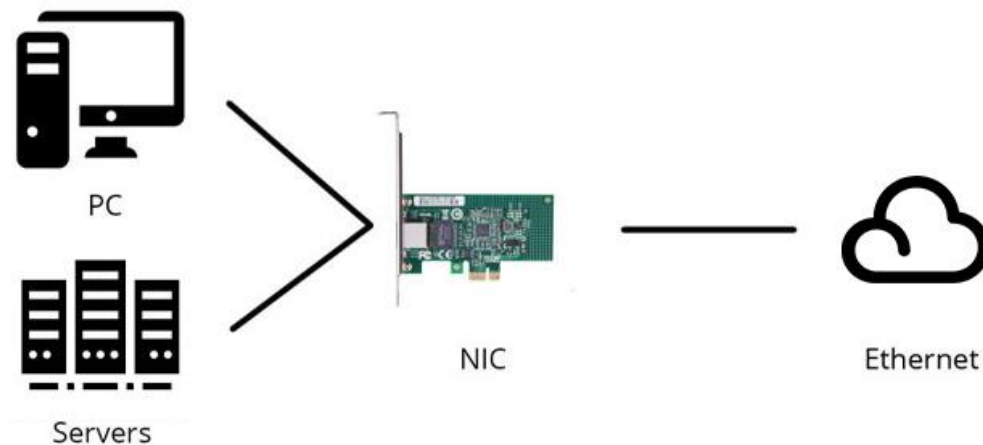
Figure 7: Network Interface Card – NIC ("What Is Network Interface Card: Purposes, Functions & Types, Irving, 2021)

The reason behind my choice of the SharpPcap library stems from our project requirements, particularly the need for the Simca product to operate seamlessly across various operating systems, with a special emphasis on Linux compatibility. SharpPcap presents an ideal solution as it is a cross-platform packet capture framework designed for the .NET environment, compatible with Windows, Mac, and Linux systems.

Developed by Chris Morgan in 2022, SharpPcap offers a robust API for capturing, injecting, analyzing, and constructing packets using any .NET language, such as C#. Its compatibility with .NET 7, the version utilized in our product, makes it a fitting choice for our project needs.

In my efforts to integrate SharpPcap into our project, I have taken several steps. Firstly, I created NICDriver components to facilitate testing of the SharpPcap framework. These components enable the identification and enumeration of all network interface cards present in the machine. Subsequently, I initiated data capture from these cards, focusing on both dropped and received packets.

To enhance the functionality of the captured data, I also leveraged the PacketDotnet library. This allowed for manipulation and filtering of the data obtained from the network interface cards, further optimizing its utility within the Simca product. The data I received is UDP and TCP network packages.

Tuesday 2nd January 2024

My Tuesday started with our regular Inspector meeting at 10:30 AM, where we delved into the tasks planned for the week ahead. It was an opportunity for me to provide updates on my progress to the team.

During the previous week, I had embarked on a significant task: updating the code to ensure all tabs could be displayed. This involved creating a branch dedicated to the task and delving into the intricacies of tab implementation. One challenge I encountered was the disparity in how tabs were utilized between the cfswidget and CCFK library. Consequently, I needed to conduct thorough research to understand the documentation and refactor the code accordingly to ensure seamless

tab functionality in the browser. I also make changes to the ReportMenu component so it can show.
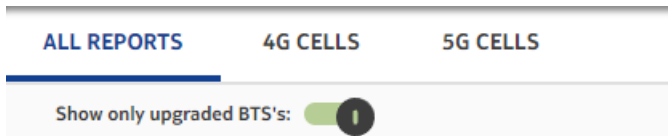

Figure 8: Tabs in Inspector project

This quite good progress from my site, I received good feedback from the team, and continue the good work. Despite the challenges, I made significant strides towards achieving the objective, laying a solid foundation for further progress in the coming days.

After the Inspector meeting, I had lunch with my Line Manager and a friend from our Trainee hub. We took the opportunity to reflect on our respective journeys at Nokia. It was a diverse group - my Line Manager, with over 20 years of experience, my friend Trainee B, who had been with Nokia for almost a year, and myself, with just four months under my belt.

During our conversation, I candidly shared my initial apprehensions about joining Nokia. Coming from a background that differed significantly from the tasks assigned to me here, I initially felt a bit lost and uncertain. However, despite the initial challenges, I made a conscious decision to embrace the opportunity and confront the tasks head-on. I expressed to my Line Manager and friend that while I may have felt nervous at the beginning, I was eager to learn and grow.

I also took the initiative to request additional tasks related to my background. I believed that by expanding my responsibilities to areas more aligned with my expertise, I could contribute more effectively while also broadening my skill set. This drive to learn and challenge myself was one of the reasons why I chose to be a part of both the Simca and Inspector teams. I saw it as an opportunity to immerse myself in diverse projects and gain valuable experience across various domains.

Wednesday 3rd January 2024

Today began with the daily Inspector meeting at 10:30 AM, where our team lead assigned me a new task: to enhance the appearance of the Expansion Panels on the Reports page. Currently, they are not displayed correctly and require fixing to ensure they are presented in the right way. Here's a snapshot of how they currently appear, highlighting the need for improvement: [insert snapshot/image].

This task presents an exciting challenge, as it requires both attention to detail and creativity to ensure the Expansion Panels are visually appealing and functional. I look forward to diving into this task and collaborating with my team to deliver a solution that meets our quality standards and enhances the user experience.
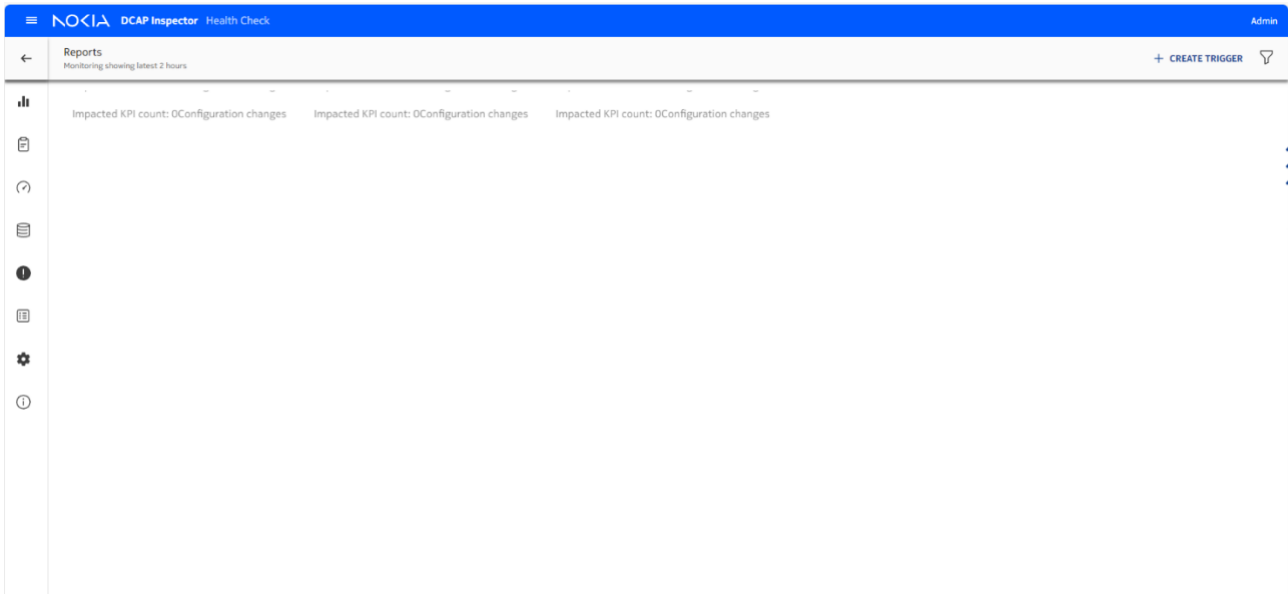
Figure 9: Error in the User Interface

With the introduction of the new Expansion Panels component from the CCFK library, there's a notable change in the code structure. The new component comprises three main elements: Container, Header, and Body. This revamped structure offers enhanced flexibility and functionality compared to the old library. In the past, when working with the old library, modifications to the layout, color, size, and styling of the Expansion Panels were primarily achieved through CSS. This approach allowed for customization tailored to the specific requirements of the user experience, ensuring a visually appealing and cohesive design.

However, with the transition to the CCFK library and its new component structure, the process of modifying the Expansion Panels may involve a different approach. While CSS will still play a role in styling, the new code structure may offer additional customization options and require adjustments to the styling techniques previously employed.

Thursday 4th January 2024

Starting my morning with the Simca meeting at 9:00 AM, I received valuable feedback from my team lead regarding my program's successful reception of UDP and TCP packages from the Network Interface Card. Building on last Monday's presentation, my team lead provided guidance on how to adapt my program to real-world scenarios, including understanding data transfer processes, data sources, and testing protocols in the lab. This became my primary focus for the day. Before delving into the new tasks, I needed to address the procedural aspect of managing my codebase. This involved merging or rebasing the latest changes from the master branch, a common task in large-scale projects with multiple developers. Reflecting on my experience, I recalled using "git rebase," but I recognized the need to understand the difference between "git rebase" and "git merge."
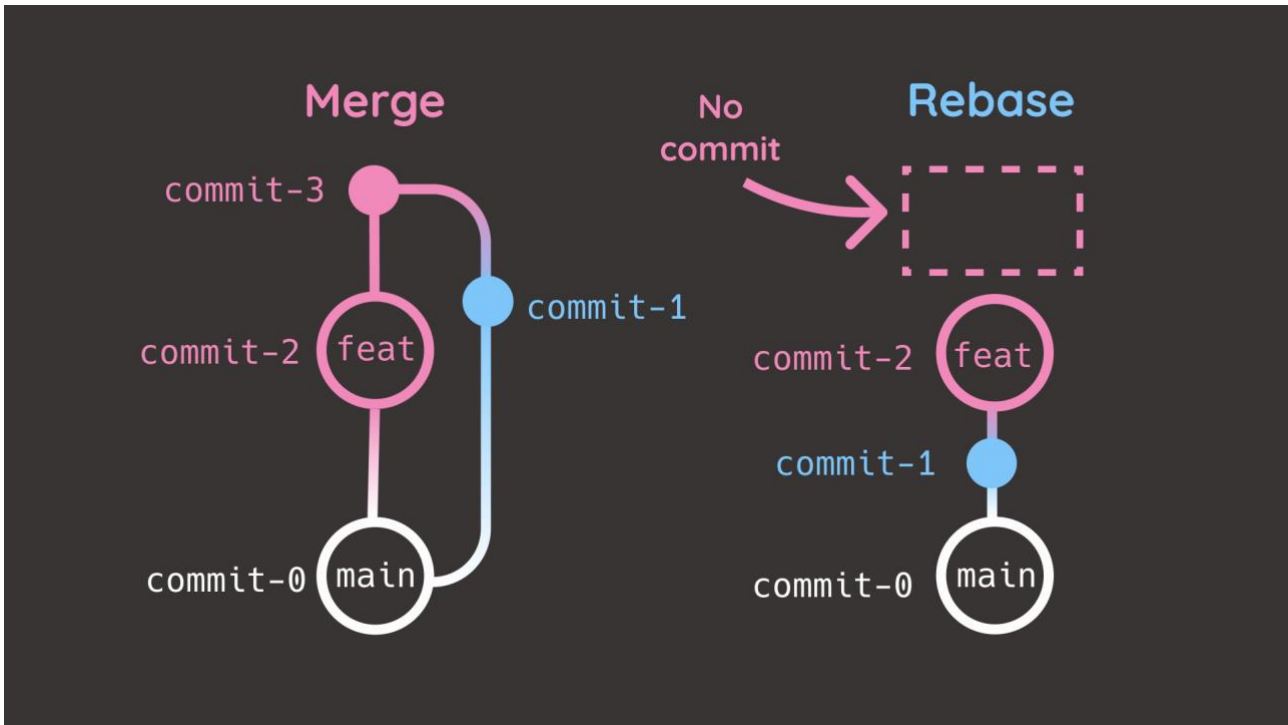
Figure 10: Git Merge vs Git Rebase ("Git Merge vs Git Rebase: Which One to Use", 2023)

While both commands aim to integrate changes from one branch into another, they do so in distinct ways. "Git merge" combines the changes from a source branch into the current branch, creating a new commit that preserves the history of both branches. On the other hand, "git rebase" rewrites the commit history by incorporating changes from the source branch onto the current branch, resulting in a linear commit history.

Understanding these differences is crucial for maintaining a clean and organized commit history, facilitating collaboration among team members, and ensuring the smooth integration of changes into the project. Armed with this knowledge, I proceeded to execute the necessary steps to synchronize my branch with the latest changes from the master branch, laying the groundwork for the day's tasks ahead.

Friday 5th January 2024

Friday began with the Inspector meeting, during which I had the opportunity to showcase the improvements I made to the Expansion Panels component. By addressing layout issues and ensuring the page's resemblance to the old version, I aimed to maintain consistency and familiarity for users. Feeling confident in the changes I implemented, I took the initiative to create my first Merge Request, enabling my team to review and provide feedback on the modifications. Additionally, I took the time to acknowledge the efforts of my team members by giving a thumbs up to some of their Merge Requests. Recognizing and appreciating the contributions of others fosters a positive and collaborative work environment, encouraging teamwork and mutual support within the team.

Today, I tackled an important CSS task: making the Reports page scrollable. To achieve this, I conducted research on CSS overflow properties and implemented a solution to create a seamless scrolling experience for users. My first step was to wrap all the Report Data Grid elements into a div container. This provided a structured approach to managing the content and enabled precise

control over the overflow behavior. Next, I applied the overflow Y property with a value of "auto" to the div container. This setting instructs the browser to automatically add vertical scrolling when the content exceeds the available space. As a result, users can effortlessly navigate through the Reports page, even when dealing with extensive datasets or expanded content.
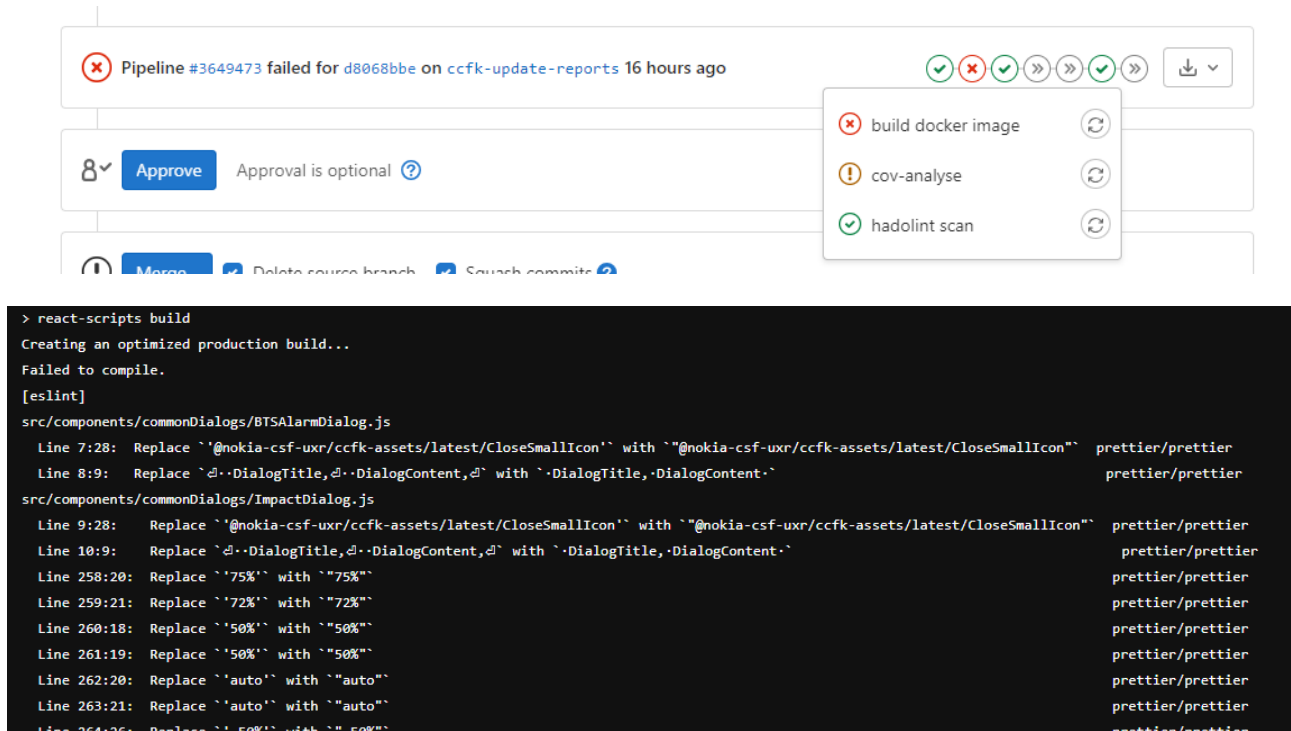

Figure 11: Failed pipeline for my first merge request.

Unfortunately, my first pipeline encountered a setback as it failed during the process of building the Docker image. Upon investigation, I discovered that the issue stemmed from a misconfiguration in my Visual Studio Code settings, which prevented Prettier from being applied correctly. In addition to resolving the Prettier configuration, I also identified the need to update my codebase with the latest changes from the master branch. However, before proceeding with the update, it was essential to address another issue: the End of Line (EOL) sequence.

The failure in the pipeline highlighted the presence of CRLF (Carriage Return Line Feed) line endings in the codebase, which were incompatible with the Unix-based systems commonly used in Docker environments. To rectify this, I needed to convert the line endings from CRLF to LF (Line Feed), ensuring compatibility and smooth operation of the Docker build process.

Week 2 evaluation

During Week 2 at Nokia, significant progress was made in advancing the Simca and Inspector projects. This weekly evaluation reflects on the experiences of the week, identifies areas of growth, and sets goals for improvement in alignment with the objectives established in Week 1.

In the Inspector project, I addressed a critical task of enhancing the Expansion Panels on the Reports page. By leveraging CSS techniques and diligently working on the code structure, I succeeded in making significant improvements to the page layout, achieving a level of similarity to the old version. This culminated in the creation of my first Merge Request, signaling a milestone in my contributions to the project.

This week provided ample opportunities for honing technical skills. I delved into CSS overflow properties to make the Reports page scrollable, demonstrating an understanding of advanced CSS concepts. Additionally, I navigated challenges with Visual Studio Code settings and resolved issues related to line endings, showcasing problem-solving abilities and adaptability in managing codebase configurations. The failure of my first pipeline due to issues with Docker image building served as a significant challenge. Identifying and resolving misconfigurations in Visual Studio Code settings and addressing line ending inconsistencies were crucial steps in overcoming this setback.

In reflecting on my experiences, insights from external sources played a crucial role in guiding my approach to technical challenges and project management. Books such as "CSS Mastery" by Andy Budd (2006) and "Docker Deep Dive" by Nigel Poulton (2017) provided invaluable insights into advanced CSS techniques and Docker image building processes, respectively. Additionally, articles from industry blogs such as CSS-Tricks enhanced my understanding of CSS concepts.

The exploration of Docker, as detailed in "Docker Deep Dive," offered a comprehensive understanding of containerization principles and the benefits it brings to software development. By containerizing applications, Docker enables consistent deployment across different environments, facilitates dependency management, and enhances scalability and portability. Moreover, Docker's lightweight nature and resource isolation capabilities contribute to efficient resource utilization and improved application performance. In contrast to traditional virtualization methods, Docker containers share the host system's kernel, resulting in reduced overhead and faster startup times. This approach streamlines the development workflow, as developers can build, ship, and run applications seamlessly across various environments, from local development machines to production servers. However, it's essential to acknowledge that Docker is not the only solution for containerization. Alternatives such as Podman, which offers a daemonless architecture and enhanced security features, and Kubernetes, a container orchestration platform for managing containerized applications, provide viable alternatives depending on specific project requirements and use cases.

Comparing my approach to challenges encountered during the week with strategies proposed in external sources revealed alignment with established practices in software engineering. Techniques suggested in "CSS Mastery" for advanced CSS concepts resonated with my efforts to make the Reports page scrollable, while insights from "Docker Deep Dive" influenced my approach to resolving Docker image building issues. By leveraging the insights from these external sources, I was able to adopt best practices and industry-standard approaches, contributing to successful project outcomes and my professional development.

Balancing tasks between the Simca and Inspector projects posed a challenge in effectively managing time and priorities. Striking the right balance between project commitments while ensuring quality contributions to both teams required careful planning and organization. To further enhance my contributions to the projects, I aim to continue expanding my technical knowledge and skills. This includes exploring advanced CSS techniques, deepening understanding of Docker image building processes, and refining proficiency in version control systems such as Git.

Strengthening communication and collaboration within the team remains a priority. Actively participating in code reviews, providing constructive feedback, and fostering a supportive team environment are areas where I seek to improve. Focus on implementing the guidance provided by my team lead and successfully adapting the program to real-world scenarios. This involves testing the program with real data from the lab and ensuring its seamless integration into the project. Continue refining the Expansion Panels on the Reports page based on feedback received from the team. Actively participate in code reviews and address any identified issues to ensure the

successful completion of the Merge Request. Dedicate time to further exploring CSS concepts, Docker image building processes, and Git workflows. Strive to deepen understanding and proficiency in these areas to facilitate smoother project execution.

In conclusion, Week 2 provided valuable learning experiences and opportunities for growth. By leveraging insights from external sources, reflecting on challenges encountered, and setting clear goals for improvement, I am confident in my ability to contribute effectively to the success of projects at Nokia in the future weeks.

Observation week 3 (January 8 – January 12, 2024)

Monday 8th January 2024
Today marks the beginning of the second week after the new year, and with the return of colleagues from their holiday break, there's a renewed sense of energy in the office. Personally, I find myself becoming increasingly familiar with the dynamics of both projects, which enables me to approach tasks more confidently and efficiently. My morning commenced with the Simca team meeting at 9:30 AM, where we collectively discussed the tasks planned for the week ahead. This provided valuable insights into the overall project trajectory and allowed each team member to align their individual goals with the team's objectives. Following the team meeting, I had a private discussion with my team leader to delve deeper into the specifics of my assigned task.

This week, I am tasked with receiving mock network data to test the functionality of my program in reading data from the Network Interface Card (NIC). However, before proceeding with the testing, it is imperative to disable the usage of the Napatech card by configuring relevant properties in the project. This preparatory step ensures that the testing environment accurately reflects real-world conditions, setting the stage for effective evaluation of the program's performance. Each component in Simca product has its own Parameter Class where we can add properties to decide what parameter used inside the program. I have added a Boolean called UseNapatechCard for the SPD component so the other developer can decide when to use Napatech card or not.

stream_gtpv2	Wireshark capture file	1,667 KB	No	6,914 KB	76%
stream_pfcp	Wireshark capture file	1,179 KB	No	3,249 KB	64%
stream2_cmu_gre4	Wireshark capture file	5,626 KB	No	29,510 KB	81%

Figure 12: Mock data from Nokia network lab.

After adjusting the properties to alter the Napatech card's usage, the subsequent step involves configuring the system to receive the simulated data on the Network Interface Card (NIC) of your local laptop. Since the simulated data is sent from the Nokia network lab, the process entails establishing the data transmission mechanism, ensuring network connectivity, determining the appropriate protocol, setting up data routing, conducting testing and validation, and troubleshooting any encountered issues. This comprehensive approach ensures the seamless reception of simulated data on your local laptop's NIC, facilitating the evaluation of your program's functionality in processing the received data.

Tuesday 9th January 2024

The day kicked off with the Inspector meeting where the team reviewed my latest merge request. Several key points were highlighted for improvement this week: removing unnecessary white gaps, addressing duplicated code, enabling the ability to open multiple panels, and ensuring functionality

to open the Impact Dialog. These tasks form the focal points of my agenda, aimed at enhancing the project's efficiency and user experience.

24409:80018	Impacted KPI count: 32
41302:5276	Impacted KPI count: 4
12345:0	Impacted KPI count: 0
12345:0	Impacted KPI count: 0
12345:1	Impacted KPI count: 0
12345:2	

BTS sw version is

Report check point is **14-12 04:30 +00:00**

Impacted KPI count before vs after check point is **0**

KPI trigger count after check point is **0**

Symptom data collection **Not done** after check point

BTS active alarms count after check point is **0**

Configuration change count done after check point is **1564**

12345:3	Impacted KPI count: 0
12345:4	Impacted KPI count: 0
12345:5	Impacted KPI count: 0
12345:6	Impacted KPI count: 0
12345:7	Impacted KPI count: 0
12345:8	Impacted KPI count: 0

Figure 13: Changes need to be fixed in Report pages.

I began by prioritizing the display of the Impact Dialog. Aligning with the structure of other components in the new CCFK library, the Dialog component comprises three main sections: Dialog Title, Dialog Content, and Dialog Footer. To ensure seamless integration, I embarked on redesigning the code to adapt it to the new Dialog structure.

Moreover, I leveraged the Base DataGrid, a standardized component developed by my team lead, to enhance the quality and readability of the code. By incorporating this base code into all common Dialogs across the project, I not only streamlined the development process but also optimized consistency and maintainability. This practice underscores the significance of creating custom, reusable code components, a skill that I have cultivated and applied effectively within the project, facilitating efficient code management, and fostering scalability.

Wednesday 10th January 2024

Wednesday commenced with the Inspector meeting at 10:30 AM, where my teammate, Software Engineer A, presented a merge request for the Dashboard page, awaiting team review. Observing that his merge request was dependent on the common Dialogs I was currently refining, I felt a surge of motivation to prioritize my task. During the meeting, I updated the team on my progress, mentioning that I was actively addressing the comments provided by our team lead on the merge request. Despite the reassurance from our team lead that there was no need to rush, as everything was proceeding according to the timeline, I remained committed to maximizing my productivity.

Today, I continued refining the Scf Dialog and Trigger Alarm Dialog, following the modifications made to the Impact Dialog yesterday. The tasks remained largely similar to yesterday's work, involving replicating the design of the Impact Dialog for these components. While these tasks didn't demand much in terms of logical thinking or brainstorming, they did require significant time and attention to detail.

As the day end at 7 PM, with the office emptied of colleagues who had already departed for home, I finally completed the redesign of all the common Dialogs. This achievement meant that my teammates could now easily access and utilize the updated code for their own tasks. Despite feeling tired and exhausted, the satisfaction of knowing that my work would benefit my colleagues and contribute to our collective progress brought a sense of fulfillment and happiness. Today's efforts underscored the importance of teamwork and collaboration in achieving our shared goals.

Thursday 11th January 2024

Thursday began with the Simca meeting at 9 AM, followed by the Inspector meeting at 10:30 AM. During the Simca meeting, my team lead provided me with a keyword for sending network data to the NIC: tcpreplay. My task for the day was to research and understand tcpreplay, as well as how to utilize it effectively for our purposes.

Upon reporting to the Inspector team that I had successfully fixed all common Dialogs the previous day, the team expressed surprise and appreciation for the prompt completion of the task. With the team planning to dedicate the day to reviewing my code, I found myself with some extra time to focus on tasks within the Simca project. This provided an opportunity to make significant progress on the assigned task while awaiting feedback from the Inspector team.

Tcpreplay is a tool used for replaying network traffic captured in pcap format. It allows you to reproduce recorded network traffic on a network interface to test various network devices, intrusion detection systems, and other networking equipment. This tool is particularly useful for network administrators and security professionals for testing the resilience and effectiveness of their network infrastructure and security measures (Aaron, 2024).

Hence, I encountered an issue when attempting to utilize tcpreplay through my WSL account. Within the WSL environment, I discovered that only three Network Interface Cards (NICs) were available: lo (loopback), eth0 (Ethernet), and docker0 interfaces. This stark contrasted with the NICs listed in my local laptop's Windows Device Manager. Recognizing this disparity, I decided to seek clarification from my team lead. Fortunately, my suspicions were validated when my team lead confirmed that WSL was not suitable for this task. Instead, I was advised to install Oracle VM VirtualBox, where I could set up a Linux OS and configure the network adapter in bridge mode. This solution would enable me to effectively utilize tcpreplay and overcome the limitations posed by the WSL environment.
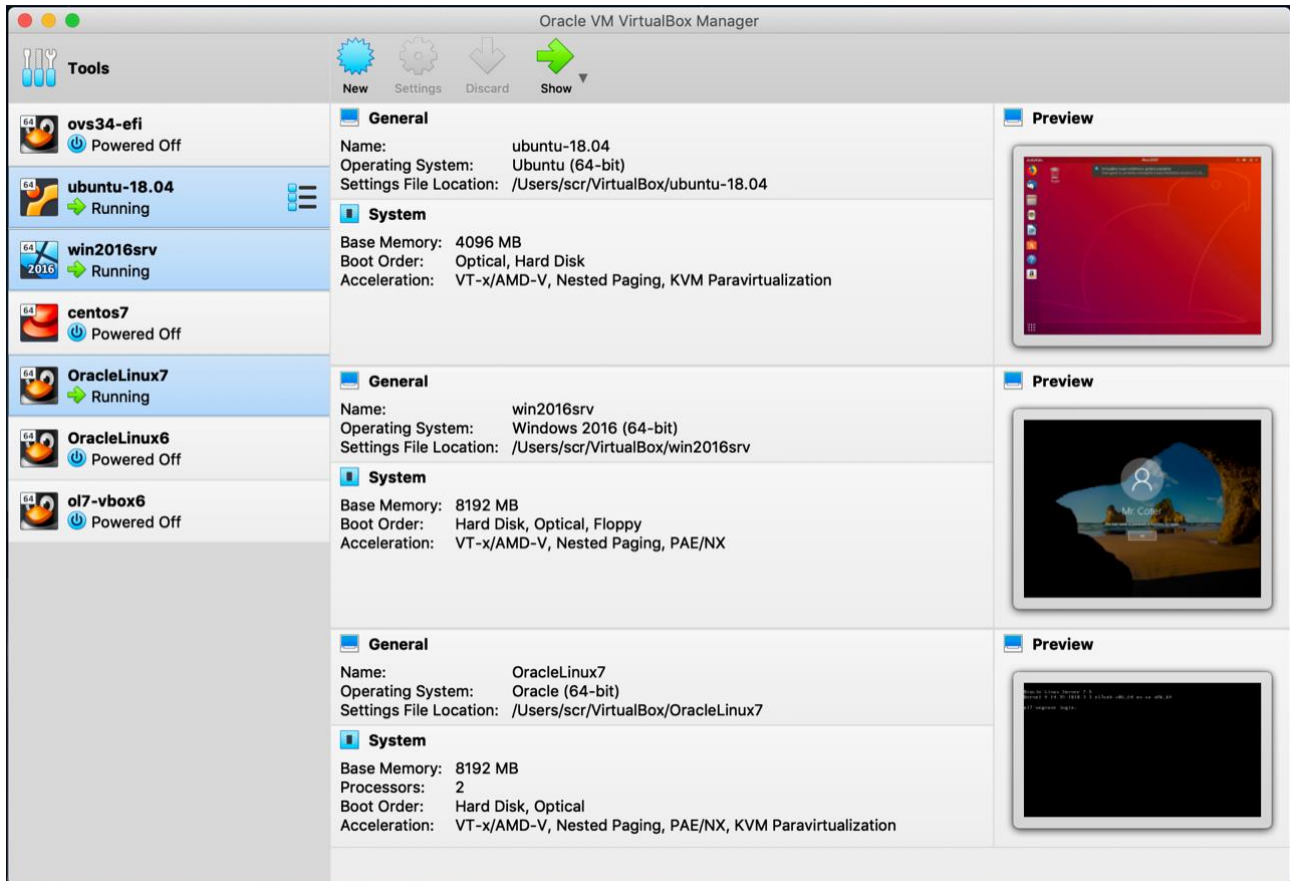
Figure 14: Oracle VM VirtualBox.

Indeed, deploying a Linux OS within Oracle VM VirtualBox proved to be an excellent solution. With this setup, leveraging tcpreplay became more straightforward, allowing me to effectively monitor how the data was being sent to the NIC. By configuring the network adapter of the virtual OS to bridge with the NIC in my local laptop, I established seamless communication between the virtual environment and the hardware, facilitating smooth data transmission and analysis. This approach not only resolved the initial challenges but also provided a robust platform for future network-related tasks.

Friday 12th January 2024

The final day of the week began with the Inspector meeting in the morning. During the meeting, my refactored code for all common dialogs was reviewed and approved. It was a significant milestone as all my teammates, including Software Engineer A and Trainee B, whose tasks were dependent on the common dialogs, could now easily access and utilize my code. This was fantastic news and a testament to the collaborative effort within the team.

With the common dialogs now successfully integrated into the project, my remaining task was to address the issue of multiple expanded rows for the Report page. Once this task is completed, the Reports page would be fully functional and ready for use. To accomplish the task, I create a new function called handleExpansion with the id parameter. The function is handling the expanded state used in Expansion Panels. In addition, it also set the status of Expansion Panels to open whenever the function is called.

In the afternoon, I participated in a department-wide meeting where various topics were discussed, ranging from rumors to the upcoming coffee break schedule. These moments provided a welcome break from work-related discussions, allowing for more relaxed conversations.

However, amidst the casual atmosphere, our Line Manager shared news that the number of trainees in our department this year would be significantly lower than last year. Consequently, there was a possibility that some individuals might not have the opportunity to extend their contracts. Concerned about this development, I reached out to my Line Manager to inquire whether contract extensions were contingent on performance. Reassuringly, my Line Manager affirmed that my performance had been commendable, with positive feedback from teammates. He emphasized that any decision regarding contract extensions was due to financial constraints within the company, rather than any shortcomings on our part.

Week 3 evaluation

As the third week draws to a close, it's an opportune moment to delve deeper into the accomplishments and experiences that have shaped this period. This week has been particularly rich in terms of both professional growth and personal reflection.

One of the most significant achievements of this week lies in the progress made within the Simca project. I successfully tackled the challenge of sending data to the NIC, a crucial step in enhancing the functionality and efficiency of the project. This endeavor required meticulous research, problem-solving, and technical expertise, ultimately culminating in a solution that aligns with project objectives and requirements. Similarly, in the Inspector project, I dedicated substantial effort to completing the overhaul of all common dialogs. This task demanded meticulous attention to detail, as well as a keen understanding of the project's architecture and design principles. By refining and standardizing these components, I not only improved the overall code quality but also empowered my teammates to work more efficiently and collaboratively.

Throughout Week 3, insights gleaned from external sources have significantly enriched my understanding of project management and technical challenges. Particularly noteworthy is the contribution of "Ethernet: The Definitive Guide" by Charles E. Spurgeon (2000), which provided invaluable insights into network technologies and the crucial role of Network Interface Cards (NICs). Understanding the technical intricacies outlined in this book has enhanced my comprehension of the requirements and considerations within the Simca project, particularly regarding data transmission to the NIC. By leveraging the insights from this source, I was better equipped to address challenges effectively and align my solutions with industry best practices. When comparing my approach to challenges encountered during Week 3 with strategies proposed in external sources, I observed a strong alignment with established practices in software engineering. This validation underscores the effectiveness of leveraging external insights to inform decision-making and problem-solving processes. By drawing upon insights from reputable sources, I was able to navigate complex technical challenges with confidence and ensure that my solutions adhered to industry standards and best practices. This alignment not only validates the efficacy of external sources but also highlights the importance of continuous learning and knowledge integration in achieving project success.

Beyond these technical achievements, this week also provided ample opportunities for personal and professional growth. Engaging in departmental meetings and discussions allowed me to gain valuable insights into organizational dynamics, challenges, and priorities. These interactions fostered a deeper sense of connection and alignment with the broader goals and vision of the company.
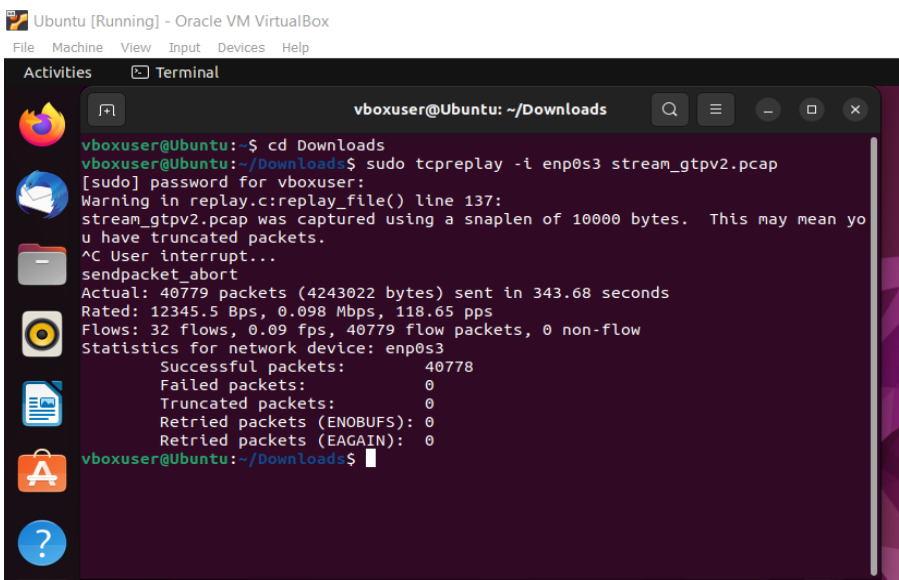
Moreover, amidst the daily demands of project work, I also found moments of reflection and introspection. These moments allowed me to recognize and appreciate my own growth and progress, as well as identify areas for further development and refinement. By embracing a growth mindset and a commitment to continuous improvement, I am confident in my ability to overcome challenges and seize opportunities for learning and advancement.

Looking ahead, I am energized and motivated to build upon the momentum of this week's achievements. I remain steadfast in my dedication to delivering high-quality work, fostering collaboration and teamwork, and contributing positively to the success of both projects and the organization. With each passing week, I am reminded of the transformative power of perseverance, resilience, and a relentless pursuit of excellence.

Observation week 4 (January 15 – January 19, 2024)

Monday 15th January 2024

I kicked off my Monday with a Simca meeting, presenting my project to the entire team. I demonstrated how I could effectively transmit data to my Network Interface Card using Oracle VM VirtualBox, configuring the bridge adapter, and employing tcpreplay for the task. The entire team responded positively to the presentation, with one of my colleagues, Senior Test Engineer A, expressing interest in the tcp replay command for data transmission. I was able to assist him by debugging some issues he encountered while utilizing Oracle VM VirtualBox.

Figure 15: Sending GPTv2 network data successfully from Oracle VM VirtualBox to bridge adapter.

Figure 16: GTPv2 network data captured from Wireshark through WSL bridge adapter from Oracle Vm Virtual Box

After successfully transmitting data to the Network Interface Card, I moved on to designing NIC-StreamClass within the SPD component to attempt reading the data directly from my local laptop. The NIC-StreamClass features a method called public void Device_OnPacketArrival(object sender, PacketCapture e), which mirrors the functionality of NapatechPacketHandler_stream. However, I encountered a challenge: while the SPD in my local environment could reach up to the point of ConsoleTrace.Write("Info", $"Socket account service started at: {myOwnEP}"), it failed to proceed beyond this line. Strangely, I observed that in a Remote Desktop environment, a similar line, ConsoleTrace.Write("Info", $"Client Connected From: {accountSock-et.RemoteEndPoint}"), executed without issue. I began to suspect that Napatech might be causing my local environment to malfunction after the "Socket account service" line. Despite attempting to comment out the line napatechStreamHandler.StartNapatechAndAttachToStreams(this), the issue persisted. Consequently, I scheduled a meeting with my Team Lead to address and resolve this issue.

It transpired that I was attempting to connect to the wrong PC IP address. The address I was trying to connect to belonged to the Remote Desktop environment. To rectify the issue, all I needed to do was replace the PC IP address with the IP address of my local machine, and this simple adjustment resolved the problem, allowing the connection to work smoothly.

Tuesday 16th January 2024

Tuesday began with a meeting with the Inspector team. During the meeting, we delved into the final features required before merging my pull request into the master branch. Specifically, we focused on ensuring that every line in the DataGrid had its own row action pinned to the right to display buttons in the User Interface, and I committed to completing this task. After refining the code in my local environment, I pushed the changes to my branch. Subsequently, I created a merge request to allow other developers on the team to review and verify the modifications.

My team lead scheduled a meeting for tomorrow with the entire team to demonstrate how to utilize our Docker image for testing with real data from the server. Excited about this opportunity, I plan to dedicate the afternoon to researching Docker in preparation for the session. Additionally, I received excellent news: my merge request for the report page passed the pipeline and seamlessly merged into the master branch. This achievement brought me immense satisfaction as it marked the completion of my first task within the Inspector project. Following the successful merge, I promptly updated the status of my Jira ticket to "Done" and logged my hours accordingly.
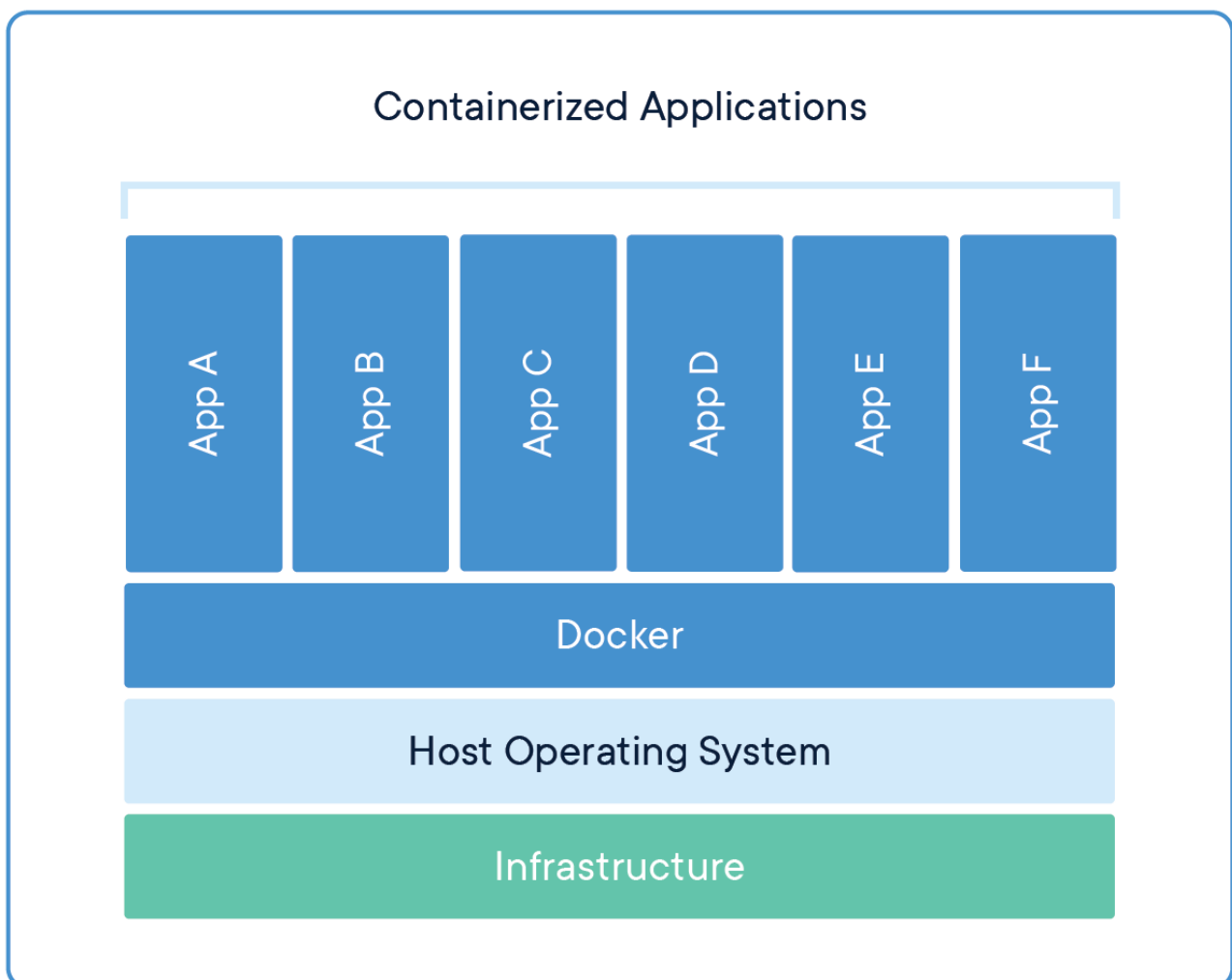

Figure 17: What is docker ("Use containers to Build, Share and Run your applications", 2024)

In my department, Docker serves as a container for all applications, including the Inspector project. To familiarize myself with Docker's functionality, I've been exploring several commonly used commands:

- docker-compose: This command is pivotal for managing multi-container Docker applications. It allows for defining and running multi-container Docker applications using a YAML file to configure the application's services, networks, and volumes (Docker, 2023). With docker-compose, I can efficiently orchestrate the deployment of complex applications.

docker pull: This command is used to fetch Docker images from a registry, such as Docker Hub, onto the local machine. It's a fundamental step in preparing to run containers, ensuring that the required images are readily available for deployment (Docker, 2023).

docker logs: This command provides access to the logs generated by a Docker container (Docker, 2023). By utilizing docker logs, I can monitor and troubleshoot containerized applications, gaining insights into their behavior and performance.

Wednesday 17th January 2024

Today, we opted to forgo the daily meetings with the Inspector team to attend an introduction to testing servers led by our team lead. During the session, I made sure to take note of key steps to enable independent testing in the future without relying on direct instructions from our team lead.

The first step involves gaining access to the server by connecting via VPN. For remote work scenarios, the Cisco Secure Client proves invaluable for accessing the Nokia VPN. Subsequently, we're provided with a URL to access the server's website along with login credentials by our team lead.

The second step entails logging into the server from the command line using the ssh command.

Moving on to the third step, we retrieve the Docker image from the pipeline within our merge request. Then, utilizing the sudo i command, we install the latest version and navigate to /utf_lab_config/composers/. Here, we locate the yml file of our project, named basic_inspector. Once identified, we proceed to edit the Docker image by copying and pasting the Docker image from our merge request.

The final step involves downgrading the current Docker version using docker-compose down, updating the image, and uploading it to the server using docker-compose up. This ensures that the latest version of our merge request code is thoroughly tested on the server.

Overall, this meeting proved to be incredibly insightful as I gained hands-on experience with Docker commands and their practical application in real-world scenarios.

Thursday 18th January 2024

Thursday began with a Simca meeting, during which our team lead shared some exciting news. He mentioned that the previous day, individuals from Singapore had expressed interest in the Simca product. Consequently, he had a meeting scheduled with the sales team and the Singapore representatives. This development was greeted with enthusiasm by the entire team, as it indicated potential new clients and increased interest in our products.

Furthermore, during the meeting, I had the opportunity to present to the team about the progress of my NICStreamClass in the SPD component. I highlighted that it can now effectively read data from the Network Interface Card and replicate the functionality of the Napatech card. This accomplishment marked the completion of 80% of my tasks within the Simca product, much to the team's satisfaction.

However, there are still some remaining tasks to address. Specifically, I need to integrate filtering features into the NICStreamClass to effectively filter GTPv2 and PFCP network data. Additionally, the program must define the data and send it to the client based on the information derived from the Parameter class. These enhancements will further refine the functionality of the product and contribute to its overall effectiveness.

```
"NapatechFilters": {
  //"IP": {
  //  "StreamTag": "ip",
  //  "Configurations": [ "Assign[]=Layer3Protocol==IP" ]
  //}
  //,
  "Gtpv2": {
    "StreamTag": "gtpv2",
    "Configurations": [ "Assign[]=TunnelType==GTPv1v2-C" ]
  },
  "Pfcp": {
    "StreamTag": "pfcp",
    "Configurations": [
      "HashMode=Hash5TupleSorted",
      "DefineMacro(\"UdpSrcPort\",\"Data[DynOffset=DynOffUDPFrame;Offset=0;DataType=ByteStr2]\")",
      "DefineMacro(\"UdpDstPort\",\"Data[DynOffset=DynOffUDPFrame;Offset=2;DataType=ByteStr2]\")",
      "Assign[]=Layer4Protocol==UDP AND (UdpSrcPort==0x2265 OR UdpDstPort==0x2265)"
    ]
  }
}
```

Figure 18: GTPv2 and PFCP network data filtered from Napatech Card.

My task involves replicating the functionality depicted in the figure, where the Napatech Card filters network data and transmits it to the client, but this time utilizing the Network Interface Card (NIC).

Following the Simca meeting, we transitioned into the Inspector meeting where I shared a brief update with the team. I announced my plan to conduct testing on the latest version of the Report page with the server and committed to presenting the results during the next meeting. Additionally, I took the opportunity to ensure that the master branch was up to date. This step was crucial to facilitate smooth collaboration within the team. Specifically, I wanted to ensure that my teammate, Trainee A, could easily pull my code, address any conflicts, and seamlessly continue her work without encountering any obstacles.

Friday 19th January 2024

Today's schedule included two meetings. The morning began with the daily Inspector meeting, where we discussed project updates and ongoing tasks. Later in the afternoon, the entire department gathered for a coffee break meeting, providing an opportunity for informal discussions and camaraderie among colleagues.

Figure 19: Meeting with whole DCAP department.

In the afternoon, I delved into a new task for the Inspector Project: refactoring the Alarms view. The initial issue was raised when the Alarms view failed to display as anticipated. To address this issue, I embarked on replacing all instances of the cfswidgets library with the CCFK library throughout the codebase. Following this adjustment, I proceeded to create a new branch and made my first commitment to initiate the refactoring process. Having successfully completed the Report page, I've noticed a growing familiarity with refactoring the codebase and implementing new features. As a result, encountering bugs in the Alarms page didn't consume much of my time to resolve. This experience has left me feeling positive and confident in my ability to tackle tasks effectively.

Week 4 evaluation

Throughout this week, I encountered diverse challenges, honed various skills, and achieved notable progress in both individual tasks and team collaborations. This evaluation provides an overview of my experiences, accomplishments, and areas for further improvement during Week 4.

One of the key highlights of this week was the consolidation of my technical proficiency and skill development. Having successfully completed the Report page task, I noticed a significant enhancement in my ability to refactor the codebase and implement new features. This newfound familiarity allowed me to address bugs encountered in the Alarms page efficiently, underscoring the effectiveness of my skill development efforts. The seamless transition between tasks and the positive outcomes achieved underscored the progress made in enhancing my technical capabilities.

Throughout Week 4, I actively contributed to the Inspector Project, demonstrating a proactive approach to task completion and collaboration within the team. Notably, I presented my NICStreamClass in the SPD component, showcasing its capability to read data from the Network Interface Card and replicate the functionality of the Napatech card. This accomplishment, alongside the positive reception from the team, marked a significant milestone in my project contributions. Furthermore, I ensured the master branch remained up to date, facilitating smooth collaboration with my teammates and enabling them to seamlessly integrate their work.

The week also presented an opportunity for client engagement and the receipt of positive news. During the Simca meeting, our team lead shared that individuals from Singapore had expressed interest in the Simca product. This development bodes well for the team, indicating potential new clients and increased interest in our products. Additionally, the opportunity to test the latest version of the Report page with the server further demonstrated our commitment to delivering high-quality solutions to our clients. Week 4 provided ample opportunities for personal growth and reflection. The successful completion of tasks, effective collaboration with teammates, and positive outcomes achieved bolstered my confidence and affirmed my capabilities as a Software Engineer Trainee. Furthermore, encountering and resolving challenges, such as the bugs faced in the Alarms page, served as valuable learning experiences, reinforcing the importance of perseverance and problem-solving skills in the software development process.

This week, external sources played a vital role in enriching my understanding of project management and technical challenges. Specifically, "Scrum: The Art of Doing Twice the Work in Half the Time" by Jeff Sutherland (2014) provided insightful strategies for enhancing team collaboration and efficiency within the Scrum framework. Leveraging concepts from this book, I deepened my understanding of Scrum principles and their application in fostering collaborative relationships within the team. Additionally, industry articles from platforms such as Agile Alliance and Scrum.org offered practical insights and case studies on implementing Scrum practices, further augmenting my knowledge and informing my approach to project tasks. Comparing my approach to challenges encountered during Week 4 with strategies proposed in external sources revealed alignment with established practices in agile methodologies, particularly within the Scrum framework. Insights gleaned from Jeff Sutherland (2014) resonated with my goals of fostering collaborative relationships within the team. The principles of transparency, inspection, and adaptation outlined in the Scrum framework were reflected in my efforts to enhance communication, streamline processes, and promote accountability within the team. Additionally, industry articles provided real-world examples of successful Scrum implementations, highlighting the tangible benefits of adopting agile practices in achieving project objectives and driving team performance.

Reflecting on my weekly development goals, I observed significant progress in fostering collaborative relationships within the team. By implementing Scrum ceremonies and embracing agile practices, we achieved greater transparency, improved communication, and enhanced alignment on project goals and priorities. The structured framework provided by Scrum facilitated effective decision-making, accelerated problem-solving, and fostered a culture of continuous improvement within the team. Moving forward, I remain committed to refining our agile processes, leveraging external insights, and embracing new opportunities for growth and learning.

Observation week 5 (January 22 – January 26, 2024)

Monday 24th January 2024

At the start of the week, I participated in the Simca meeting where our team lead outlined our plans for the upcoming release of the Simca product. This includes updates to the Simca server, Simca Dashboard Backend, and Simca Dashboard Frontend. With the Simca architecture divided into three distinct projects, it became evident that there's a significant workload to tackle, particularly within the Simca Dashboard Frontend. Given my background in Frontend Development, I approached my team lead and expressed my interest in contributing to the Simca Dashboard Frontend project. After discussing this with the team, I was granted approval to join the Simca Dashboard Frontend team once I completed my current task.
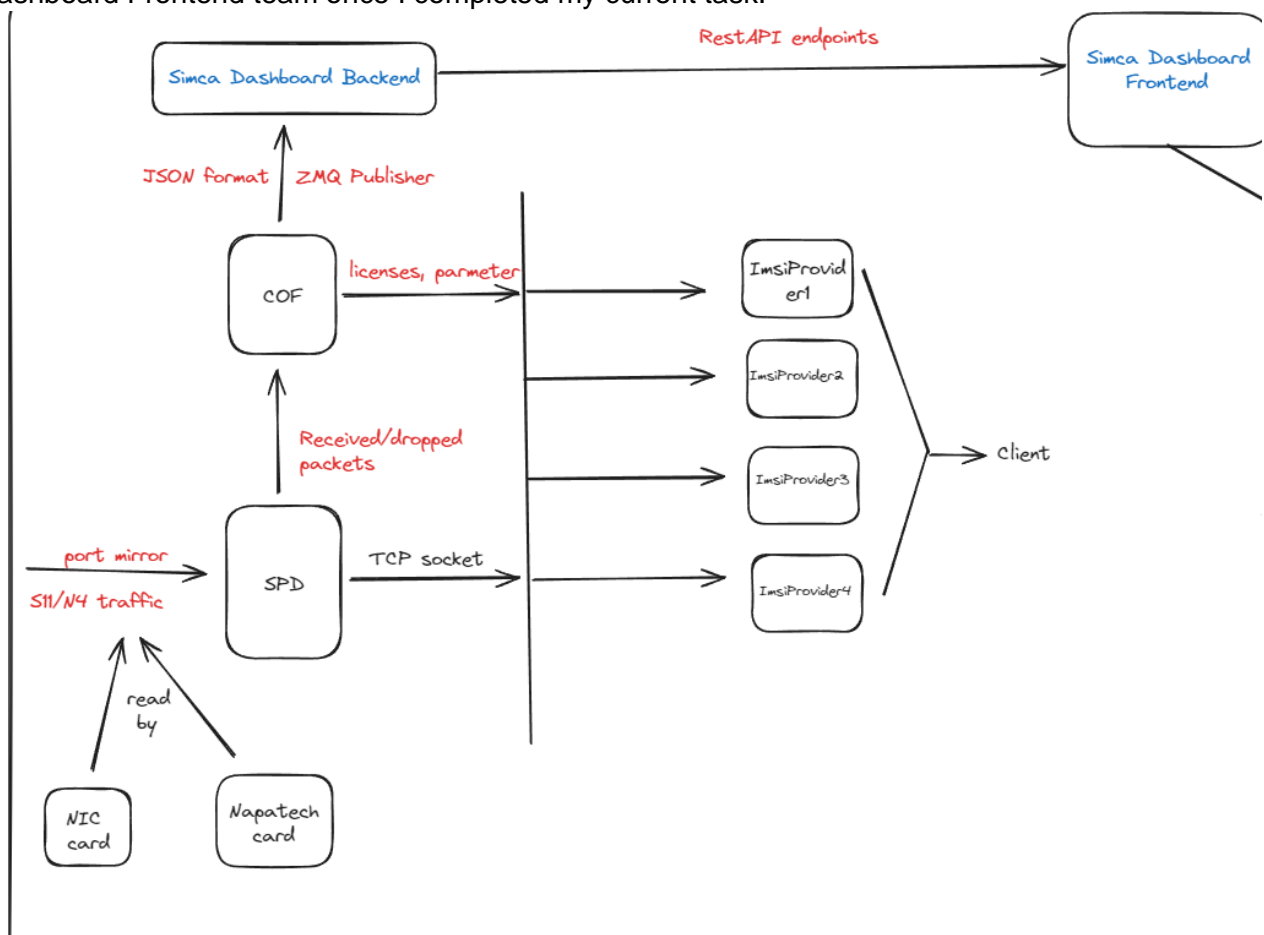

Figure 20: Simca Product Architecture draw by me.

I successfully completed the task of filtering GTPv2 and PFCP by utilizing Packet.Net. The reason why I use Packet.Net because it can support packet formats: TCP and UDP (Morgan, 2023), which is suitable for Simca project. My approach involved associating the stream tag of the data sent to the SPD component with the corresponding port number. Subsequently, I used Packet.Net to extract the received data and compared the packet's source port with the designated port number. Upon validation, I forwarded the captured data to the clients. In the meeting, I presented my solution to the team, outlining the methodology and its efficacy. I'm pleased to report that my presentation was well-received, and my solution was accepted by the team.

Later in the afternoon, my team lead provided me with a zip file containing a collection of tunneled packets, each with a dual IP-layer configuration. Due to complications with the Napatech card's performance in handling these packets, my team lead tasked me with testing the functionality of the Network Interface Card program with this specific packet structure.

To conduct the test, I forwarded the data to the Network Interface Card via Oracle VM Virtual Box. Fortunately, the functionality remained intact. This is because Packet.Net adeptly extracts the source and destination ports from UDP packets, thereby mitigating any potential issues stemming from the dual IP-layer configuration.

Tuesday 25th January 2024

On Tuesday, I kicked off the day with our regular meeting with the Inspector team, during which each member provided updates on their progress from the previous week and outlined their objectives for the current week. In my report, I highlighted my accomplishment of getting the Alarms page to display last week and shared my goal of enhancing its functionality and introducing new features this week. Additionally, I took ownership of a new ticket in Jira related to the Alarms view, initiating its status as "In Progress" and logging the hours I dedicated to it last Friday.

Today's focus revolves around refining the rowCallback function. I'm streamlining it by eliminating any redundant parameters and functionalities. Following constructive feedback from my team lead regarding a past merge request in the Report page, I've undergone significant learning. Specifically, I've grasped the importance of writing clean code, which involves removing unnecessary elements such as blank spaces, console.log statements, and comments. Moreover, I've gained insights into refactoring code to optimize the utilization of duplicated sections. For instance, I've transitioned from using if-else conditional statements to employing switch and case statements within the rowCallback function, enhancing its clarity and readability. This function is pivotal as it triggers whenever a user interacts with the row action buttons within the Data grid, executing the corresponding functionality for each case.

Wednesday 25th January 2024

I started today with daily meeting with Inspector team. After re-designing the rowCallback function yesterday to make all the row action buttons work, I realized one problem with this project is that it still use Javascript with the large legacy codebase. I suggested to the team that should we use Typescript for the whole project. The reason is that Typescript will help us a lot when refactoring, handling error or implementing new features. It will contain type for all the functions, variables. It also make the code clearer and more optimized.

TYPESCRIPT: PROS AND CONS

Pros

+ Static type-checking

+ Improved scalability

+ ES6/ES7 support

+ Improved IntelliSense support

+ Better error handling

Cons

− Stiff learning curve

− Increased complexity

− Slower compilation speeds

− Limited browser support

Figure 21: Pros and Cons of Typescript over Javascript ("Javascript VS. Typescript: Which Is Better for Your Project In 2023?", Evgeniy, 2023)

JavaScript offers dynamic typing, which allows variables to change type during runtime.

TypeScript extends JavaScript's capabilities, making it a superset of the language. This implies that any valid JavaScript code is also valid TypeScript code. However, TypeScript distinguishes itself by offering robust static typing, which aids in identifying errors in code prior to runtime, thereby enhancing code reliability (Evgeniy, 2023). In terms of features, TypeScript emerges as the winner, providing advanced functionalities like interfaces and generics. These features empower developers to create more resilient and maintainable programs, further cementing TypeScript's position as a preferred choice for building complex applications.

My team lead told me that our department goal is to change all the User Interface projects from using JavaScript to Typescript. Additionally, we have a designated repository called ts-monorepo housing various TypeScript projects within our department. While this transition is underway, our specific focus within the Inspector project involves refactoring the entire codebase to align with the latest Nokia library standards. Given the substantial size of the codebase, this undertaking will undoubtedly require significant time and effort, but it stands as a crucial plan for the year ahead. I'm thrilled to have had the opportunity to contribute my thoughts on this matter during our team

discussion today. It's gratifying to see that we're all in agreement and aligned in our approach towards addressing these challenges.
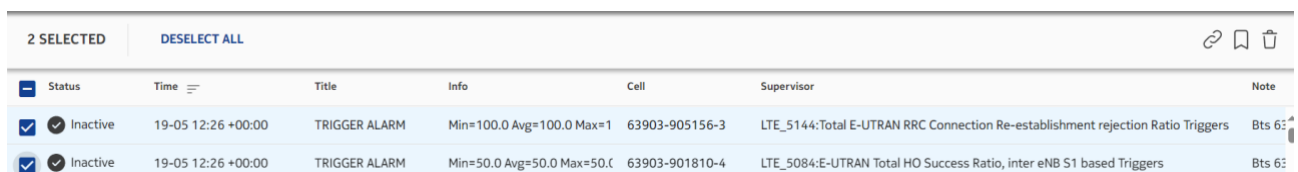
Thursday 26th January 2024

Today was quite eventful with two meetings on the agenda: the weekly meeting with the Simca team and the daily meetings with the Inspector team. In the Simca project, my upcoming task involves implementing statistics functionality for the Network Interface Card. This entails retrieving statistics such as dropped bytes, dropped packets, received bytes, and received packets after filtering the network packages.

In my exploration of SharpPcap's capabilities, I discovered that it facilitates developers in reading dropped and received packets from captured network data. However, it lacks built-in support for printing dropped bytes and dropped packets. Consequently, I'm now challenged to devise a solution to address this gap in functionality without relying on library support.

During the Inspector team meeting, I engaged in a discussion with my teammate, Trainee A, regarding her merged request for the Topology views. As the designated reviewer for her merge request, I found her work to be satisfactory, yet I identified areas where enhancements could be made. Drawing from the lessons learned from my own previous merge request, where my team lead provided valuable guidance on code improvement by eliminating duplication and unnecessary elements, I shared similar insights with Trainee A. However, I must commend Trainee A for her commendable efforts in adhering to the guidelines outlined in the Nokia Design System. By applying best practices from this system to her merge request, she succeeded in making the code more readable and easier to comprehend. This attention to detail and commitment to following established design principles significantly contributes to the overall quality of our project.

Friday 26th January 2024

Today was a busy day with two meetings on the agenda. In the morning, I met with the Inspector team, where I shared my plan to implement the Multi Selection feature for the Alarms view. This feature allows users to select multiple rows, prompting the application to display a toolbar indicating the number of selected rows and offering the same row action buttons as the DataGrid.

Figure 22: Multi Selection Toolbar

To accomplish this task, I need to introduce new functionality to the BaseDataGrid component. This enhancement will enable developers to pass row action buttons props through BaseDataGrid, facilitating the display of the Multi Selection Toolbar when the number of selected rows exceeds one. By creating a base component for commonly used features like this, we can effectively eliminate duplicated code and maintain a cleaner codebase.

Later in the afternoon, during the coffee break with colleagues from the entire DCAP department, I received unexpected news from my Line Manager. He informed us that after 24 years of dedicated service and contribution, he will be leaving Nokia. This news came as a shock, and I couldn't help but feel a sense of sadness. Throughout my time here at Nokia, he has been a source of

encouragement and support, always extending kindness towards me. Our team will soon welcome a new Line Manager next month. It's a significant transition that marks the end of an era. We will have the opportunity to bid farewell to him on his last working day, which is scheduled for next Monday. It will be a chance for us to express our gratitude and appreciation for his guidance and leadership over the years.

Week 5 evaluation

This week commenced with two crucial meetings: the weekly meeting with the Simca team and the daily meetings with the Inspector team. These gatherings served as platforms for progress updates, goal setting, and collaborative discussions. Notably, the alignment achieved during these meetings laid the groundwork for cohesive teamwork and project advancement. Throughout the week, I encountered technical challenges, particularly in the implementation of new features and functionalities. One such challenge was the task of implementing statistics for the Network Interface Card in the Simca project. Despite facing obstacles such as limited library support, I approached the problem with resilience and determination, seeking alternative solutions to achieve the desired outcome.

By embracing Scrum practices, particularly within the Inspector Project led by our team lead acting as the Scrum Master, we have witnessed a significant improvement in team communication, collaboration, and overall project performance. Moving forward, we remain committed to leveraging Scrum principles to drive continuous improvement, deliver value to our stakeholders, and achieve project success.

- **Structured Daily Stand-ups Led by Scrum Master**: With the adoption of Scrum, our team has implemented structured daily stand-up meetings led by our Scrum Master, who also serves as our team lead. These meetings provide a dedicated time for team members to synchronize their activities, discuss progress, and address any impediments. The Scrum Master ensures that the stand-ups remain focused, time-bound, and productive, encouraging open communication and collaboration among team members
- **Focused Sprint Planning Sessions**: Sprint planning sessions have become a cornerstone of our project workflow. Led by our Scrum Master, these sessions involve collaborative discussions to determine the goals for the upcoming sprint, select user stories to be completed, and estimate the effort required for each task. In the absence of a dedicated Product Owner, the team collectively prioritizes tasks based on project objectives and client requirements, ensuring alignment with stakeholder expectations.
- **Retrospectives for Continuous Improvement**: At the end of each sprint, our team conducts retrospectives to reflect on our processes, identify areas for improvement, and implement actionable changes. Led by our Scrum Master, these retrospectives provide a safe space for team members to voice their concerns, share feedback, and suggest enhancements to our workflow. By actively participating in retrospectives, we foster a culture of continuous improvement and adaptability, driving positive changes in our project execution.
- **Increased Collaboration Across Cross-Functional Teams**: The adoption of Scrum has facilitated increased collaboration and communication across cross-functional teams within the Inspector Project. Our team lead, acting as the Scrum Master, facilitates interactions between developers, testers, and other stakeholders, ensuring that everyone remains aligned with project goals and objectives. This collaborative approach promotes knowledge sharing, problem-solving, and collective ownership of project deliverables, resulting in enhanced productivity and efficiency.

- **Clear Roles and Responsibilities**: While our team operates without a dedicated Product Owner, the roles and responsibilities within the Scrum framework are clearly defined and understood by all team members. Our team lead, serving as the Scrum Master, takes on the responsibility of facilitating Scrum events, removing impediments, and ensuring the smooth execution of project activities. Each team member understands their role within the Scrum framework and actively contributes to achieving sprint goals, fostering a sense of ownership and accountability within the team.

A significant highlight of this week was the collaborative efforts within the Inspector team. Engaging in discussions with my teammate, Trainee A, regarding her merged request for the Topology views provided an opportunity for constructive feedback and knowledge exchange. Witnessing the application of best practices from the Nokia Design System underscored the importance of adherence to established guidelines in enhancing code quality and readability. Towards the end of the week, I received unexpected news of my Line Manager's departure from Nokia after 24 years of service. This announcement brought about a mix of emotions, including sadness and reflection. Despite the impending transition, it served as a reminder of the impact of strong leadership and mentorship in shaping professional journeys.

In this week, I saw notable progress in various projects, including advancing towards the implementation of the Multi Selection feature for the Alarms view in the Inspector project. Additionally, I successfully completed tasks related to filtering network packages and exploring new functionalities within the Simca project, contributing towards project milestones. I actively participated in meetings, effectively communicated project updates, and collaborated with team members to address challenges and share insights. This emphasis on open communication fostered a supportive and cohesive team environment, conducive to achieving collective goals. This week was characterized by continuous learning and skill development. From refining coding practices to exploring new libraries and functionalities, I embraced opportunities for growth and sought to enhance my proficiency as a Software Engineer Trainee.

The unexpected news of the Line Manager's departure served as a reminder of the unpredictable nature of professional environments. It emphasized the importance of adaptability, resilience, and the impact of strong leadership and mentorship in shaping professional journeys. Engaging in discussions and feedback sessions highlighted the value of constructive feedback in driving improvement and growth, emphasizing the importance of openness to receiving feedback from peers and mentors.

In conclusion, Week 5 was a week of challenges, achievements, and unexpected revelations. As I navigated through project tasks, meetings, and interactions with colleagues, I gained valuable insights, honed my skills, and forged meaningful connections within the team. Looking ahead, I am committed to embracing future challenges, continuing my journey of growth, and contributing towards the success of projects and the team.

Observation week 6 (January 29 – February 2, 2024)

Monday 29th January 2024

The week commenced with the Simca team meetings in the morning, where we deliberated on the team's agenda for the week. During the discussion, I updated the team on the progress of my ongoing task, noting that it is nearing completion. However, I highlighted the remaining challenge of finding a method to display the dropped bytes and received bytes of the captured network data.

Following the Simca meetings, I was assigned to participate in a meeting called Simca Introduction. In this session, my role was to support our team lead in presenting an overview of the Simca product, including its functionalities and our client base, to other trainees within our department. Given that I am the sole trainee working on this product, this responsibility carried significant weight. I shared insights into my contributions to the Simca project, detailing the obstacles encountered and the experiences gained throughout the journey. Fortunately, the presentation was well-received, and some of my fellow trainees provided positive feedback.

During the early afternoon, I attended a farewell gathering for my Line Manager. It was a heartwarming event as colleagues gathered in the main hall for a coffee break. I took the opportunity to express my gratitude to my Line Manager for their guidance and support. Additionally, I had the chance to meet with the Interaction Designer of our department during the meeting. I conveyed my background and expertise in web and user interface design to him. We had a fruitful discussion, leading to plans for a meeting later in the week to assign new tasks on upcoming projects. Furthermore, he offered to support me in advocating for the continuation of my contract with Nokia to our R&D Director, which presents a promising opportunity for me to connect with key stakeholders.

After the gathering, I returned to work on the Simca task. After extensive research, I discovered a solution. By reviewing the documentation of SharpPcap, I learned that we could track received and dropped bytes by monitoring packet length. This method involves counting the packet length whenever the device begins capturing, allowing us to accurately tally the bytes. I believe this solution is viable and plan to present it on Thursday.

```csharp
void OnPacketArrivalHandler(object sender, PacketCapture e)
{
    var packet = e.GetPacket();
    receivedBytes += packet.PacketLength;

    foreach (var streamItem in streamManagement.GetStreams())
    {
        streamItem.Device_OnPacketArrival(sender, e);
    }
}

    device.OnPacketArrival += streamItem.Device_OnPacketArrival;
device.OnPacketArrival += OnPacketArrivalHandler;

device.StartCapture();
```

Figure 23: Count received bytes by packet length.

Tuesday 30th January 2024

Today was packed with two meetings: one with the Inspector team in the morning and another with the Interaction Designer in the afternoon. My primary focus for this week is to finalize the Alarms view and prepare to transition to a new project next week. To achieve this, I aimed to enable the display of multiple impact dialogs when users click on the row buttons of the Multi Selection Toolbar. During the morning meeting with the Inspector team, I communicated this goal, and everyone agreed.

Regarding the implementation of multiple impact dialogs, I began by creating an array to store impacts and assigned this array to the rowData props of the Data Grid component. Since there would be multiple dialogs, it was necessary to implement a unique key for each Data Grid. I accomplished this by using the index of the impacts array. Additionally, I developed the Base Dialog for the warning dialog to facilitate its reuse throughout the application.

In the afternoon meeting with the Interaction Designer, we discussed my upcoming task for February. I will be collaborating with my new teammate, Software Engineer O, on the Simca Dashboard Frontend project. Once I complete the Alarms view in the Inspector project this week, I will transition to working with the new team. However, I assured the team that I will still be available to help with any remaining tasks in the Inspector project if needed.

Wednesday 31st January 2024

Today began with the Inspector meeting in the morning, where we discussed our progress towards completing the Alarms view. We're on the brink of reaching our goal, with all the necessary functionality and features implemented. Now, my focus shifts to ensuring the code is clean and readable before merging it into the main branch. During the process of cleaning and optimizing the code, I noticed that certain useEffect hooks were not essential and weren't functioning as intended. To address this issue, I delved deep into researching the useEffect hook to gain a better understanding of its nuances and improve the codebase accordingly.

useEffect() Hook
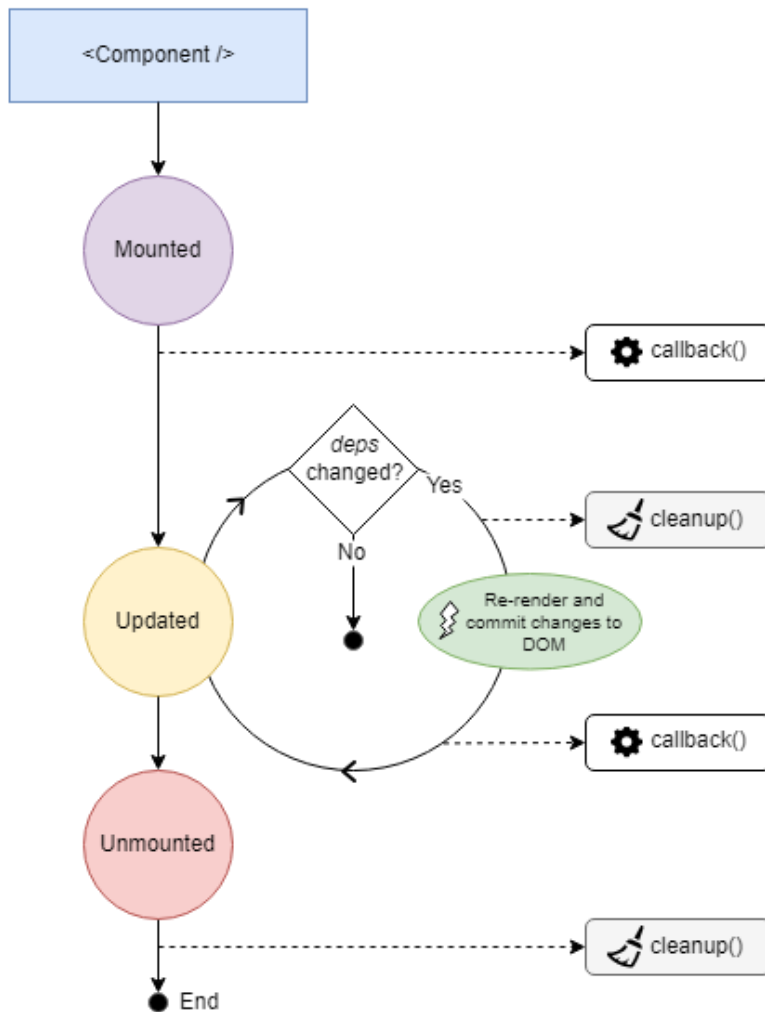
Figure 24: How useEffect work ("A Simple Explanation of useEffect()", 2023)

In summary, the useEffect (callback, dependencies) hook serves as a crucial tool for managing side-effects within functional components. The callback parameter represents the function where side-effect logic is housed, while dependencies consist of a list comprising props or state values upon which the side-effect relies. Invoked after the initial rendering (mounting), useEffect (callback, dependencies) also triggers subsequent renderings if any value within the dependencies list undergoes a change (Pavlutin, 2023). There are some useEffect inside my codebase does not use the right depencies array, therefore, it not showing the best performance for the application.

Additionally, I applied a practice I learned from my team lead by creating base functions to encapsulate logic used in multiple locations. This approach enhances code reusability and maintainability. It consumed the entirety of my afternoon to finalize the Alarms view and submit the

merge request for it. As I left the office at 7 pm, under the darkening sky, I couldn't help but feel content with the productivity of the day.

Thursday 1st February 2024

Today kicked off with the Simca meeting in the morning, during which I presented my solution for printing the statistics of the Network Interface Card program. With my task nearing completion, I will now focus on implementing the final features, which involve printing statistics every minute and subsequently resetting them. Additionally, I'll be drafting documentation for other developers on how to disable the Napatech card and run the Network Interface Card program.

In a discussion with my team lead regarding the solution to reset the statistics, we encountered an initial hurdle: the reset counter function didn't work with SharpPcap statistics. Despite setting it to zero, the statistics remained unchanged. Consequently, we had to explore alternative approaches.

```
uint receivedPackets = stats.ReceivedPackets - receivedPacketsChange;
uint droppedPackets = stats.DroppedPackets - droppedPacketsChange;
uint interfaceDroppedPackets = stats.InterfaceDroppedPackets - interfaceDroppedPacketsChange;

ConsoleTrace.Write(InfoLevel.LiveDebug, $"Packets Received: {receivedPackets}, Packets Droppe

receivedPacketsChange = stats.ReceivedPackets;
droppedPacketsChange = stats.DroppedPackets;
interfaceDroppedPacketsChange = stats.InterfaceDroppedPackets;
```

Figure 25: Logic to reset counter.

After dedicating the afternoon to resolving this issue, I devised a solution and reported it to my team lead. While I couldn't directly reset the statistics, I devised a workaround by implementing a logical function. Essentially, the PrintCaptureStatistics() function will be invoked every minute. As a result, I recalculated the statistics by subtracting the old stats from the new stats, effectively resetting the counter. This approach proved effective in achieving the desired outcome.

Friday 2nd February 2024

Today was marked by two important meetings: one with the Inspector team in the morning and another involving the entire DCAP department later in the day.

During the morning session, we reviewed my merge request, finding the code structure to be well-organized, optimized, and clean. After successful testing of the docker image with the real server, it was deemed ready for merging into the master branch. Subsequently, I was tasked with testing the server, which not only included my docker image but also that of my teammate, Trainee A, specifically the Topology view.

In the afternoon, we convened to welcome the new Line Manager of DCAP Espoo. He introduced himself to the team and outlined the schedule for the upcoming week. I took the opportunity to update him on my status, informing him that I had completed all tasks in both the Simca and Inspector projects and was eagerly awaiting new assignments from the Interaction Designer for the following week. Additionally, I discussed with him my prospects for a new contract with Nokia.

This year, the number of Trainees in the DCAP department is considerably fewer than last year, intensifying the competition. While Trainees typically can extend their contracts after the first contract, this year it seems that Nokia is seeking individuals who can truly deliver results. Despite this heightened competition, I remain confident in my abilities. I have always been proactive in seeking out opportunities, embracing new challenges, and adapting quickly to new environments. The outcome of this week, where I successfully completed tasks in both of my current projects and am poised to transition to new projects next week, is a testament to this.

Week 6 evaluation

This week marked a pivotal stage in my journey as a Software Engineer Trainee at Nokia, characterized by significant accomplishments, critical reflections, and a forward-looking perspective. This comprehensive evaluation encapsulates the achievements, challenges, and growth experienced throughout the week. It delves into the successful completion of tasks in ongoing projects, the attainment of set objectives, and the readiness to embrace new endeavors with a proactive mindset.

Throughout Week 6, I made substantial strides in advancing ongoing projects, namely the Simca and Inspector projects. A major highlight was the successful culmination of tasks in both projects, signifying a significant milestone in my professional development. In the Inspector project, the finalization of the Alarms view was achieved with meticulous attention to detail and a focus on optimizing code structure for enhanced performance and readability. Likewise, in the Simca project, I addressed pending tasks efficiently, ensuring the project's readiness for subsequent phases of development. These accomplishments underscored not only my technical competence but also my ability to effectively manage project timelines and deliverables.

At the outset of the week, I set ambitious goals aimed at driving progress and achieving tangible outcomes. I am pleased to report that these goals were not only met but exceeded. The successful completion of the Alarms view in the Inspector project and the resolution of pending tasks in the Simca project exemplified my commitment to excellence and determination to surpass expectations. This achievement instilled a sense of accomplishment and reinforced my belief in the power of goal setting as a catalyst for personal and professional growth.

In addition to accomplishing tasks in ongoing projects, Week 6 presented an opportunity to embrace new challenges with a proactive attitude. As I prepared to transition to a new project, namely the Simca Dashboard Frontend project, I approached the prospect with enthusiasm, curiosity, and a willingness to learn. Engaging in discussions with my new teammate and outlining plans for upcoming tasks demonstrated my readiness to adapt to change and contribute effectively to the success of the project. This proactive approach not only facilitated a smooth transition but also fostered a sense of excitement and anticipation for the opportunities that lay ahead.

In Week 6, as I transitioned into the Simca Dashboard Frontend project, external sources played a crucial role in augmenting my understanding and proficiency in Typescript, a key technology essential for frontend development. Referencing the Typescript Handbook provided by the official Typescript website served as a fundamental resource for deepening my knowledge of Typescript's syntax, features, and usage. The handbook offered comprehensive explanations, code examples, and practical insights into advanced language concepts such as type inference, interfaces, and generics. By consulting the Typescript Handbook, I gained confidence in writing type-safe code and leveraging Typescript's powerful features to enhance the robustness and scalability of the Simca Dashboard Frontend. Engaging with online tutorials and courses dedicated to Typescript, such as those available on platforms like Udemy, Coursera, and Pluralsight, supplemented my

learning and provided practical guidance on applying Typescript in real-world projects. These resources offered step-by-step instruction, hands-on exercises, and project-based learning experiences tailored to different skill levels, enabling me to deepen my understanding of Typescript fundamentals and advanced topics. By actively participating in Typescript tutorials and courses, I accelerated my learning curve and acquired practical skills applicable to the Simca Dashboard Frontend development. Analyzing articles and blog posts on Typescript best practices provided valuable insights into writing clean, maintainable, and idiomatic Typescript code. External sources, such as "TypeScript Deep Dive" (2020) by Basarat Ali Syed, offered practical guidance on leveraging Typescript's advanced features, optimizing code structure, and enhancing developer productivity. By comparing my Typescript codebase with recommended practices from external sources, I identified opportunities to improve code readability, type safety, and performance in the Simca Dashboard Frontend project, leading to more efficient development workflows and enhanced code quality. Reflecting on my integration and comparison of external sources in Week 6, I recognized the importance of leveraging authoritative documentation and curated learning resources to master complex technologies like Typescript effectively. By immersing myself in the Typescript ecosystem and aligning my development practices with industry standards and best practices, I was able to contribute effectively to the Simca Dashboard Frontend project, delivering a robust and maintainable frontend solution.

This week provided ample opportunities for reflection and learning, enabling me to glean valuable insights and lessons from both successes and challenges. Reflecting on the accomplishments achieved in ongoing projects reaffirmed the importance of perseverance, collaboration, and attention to detail in achieving desired outcomes. Furthermore, the experience of preparing for a new project underscored the significance of adaptability, open-mindedness, and a proactive mindset in navigating transitions and embracing new opportunities for growth. These reflections served as invaluable lessons that will inform my approach to future endeavors and contribute to my continued development as a Software Engineer Trainee.

By successfully completing tasks in ongoing projects, surpassing set objectives, and embracing new challenges with a proactive attitude, I have demonstrated my capabilities as a Software Engineer Trainee. As I reflect on the experiences of the week, I am filled with a sense of pride, gratitude, and optimism for the journey ahead.

Observation week 7 (February 5 – February 9, 2024)

Monday 3rd February 2024

This week kicked off with a fresh lineup of scheduled meetings ahead. Since our team wrapped up all tasks related to refactoring the Inspector project, we collectively decided to ditch the daily meetings for Inspector projects, opting instead to schedule meetings as needed whenever issues arise. While I still have my weekly check-ins with the Simca teams, my focus has now shifted mainly to collaborating with Software Engineer O and Senior Software Engineer R from Tampere. They're leading the charge on the Simca Dashboard Backend and Frontend respectively.

Our go-to channels for collaboration are our weekly meetings and messaging via Teams. During this morning's meeting, I briefed my team lead on my continued involvement in the Simca project, albeit in a different capacity. Now, I'm the go-to person for understanding both the nitty-gritty details of network data handling and transfer, as well as how those statistics translate into the user interface. This new role has been particularly intriguing for me, as it allows me to see the bigger picture of the project.

Tuesday 4th February 2024

Today, I delved deep into the project structure, beginning with an exploration of the Simca Dashboard Frontend project. It's a vital component housed within the larger project known as ts-monorepo. The rationale behind this structure is that ts-monorepo leverages a microservices architecture for frontend development. This concept is both fascinating and novel to me, and I'm genuinely excited about delving into research on microservices architecture within the realm of front-end development. It opens a realm of new knowledge and possibilities that I'm eager to explore further.
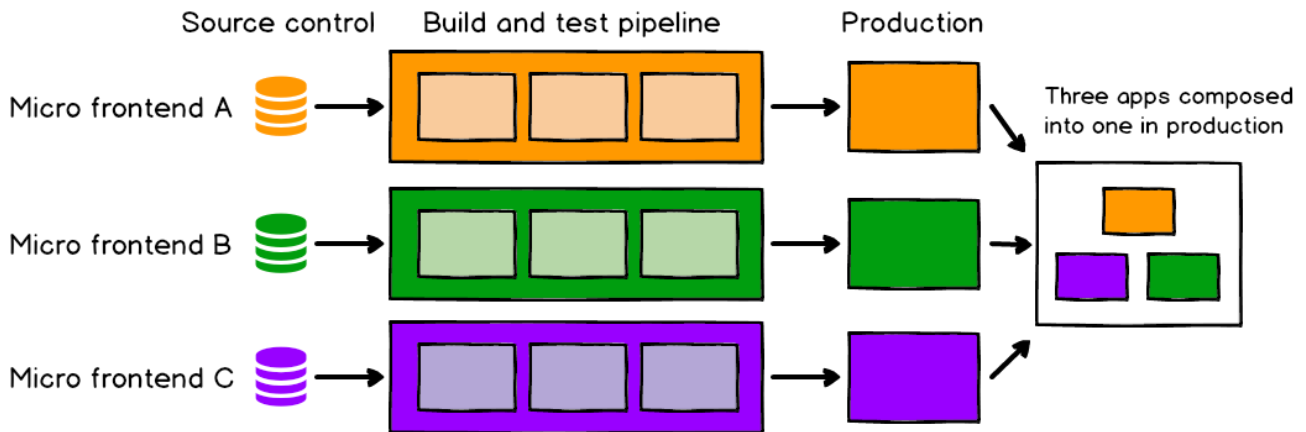
Figure 26: Microservice in Frontend Development ("Micro frontends", 2019)

Key benefits of micro frontends:

- smaller, more cohesive, and maintainable codebases

- more scalable organisations with decoupled, autonomous teams

- the ability to upgrade, update, or even rewrite parts of the frontend in a more incremental fashion than was previously possible (Jackson, 2019).

In our ts-monorepo repository, we've incorporated three projects: Simca Dashboard Frontend, Web Email, and Tracer GUI. This setup helps us avoid duplicating code and components throughout our larger application. Plus, by embracing a micro frontend approach, we're able to streamline our pipeline. This means we can reduce unnecessary steps while still providing specific pipelines tailored to the needs of each individual project. It's a practical strategy that not only boosts efficiency but also allows for more flexibility and adaptability in our development process.

Wednesday 5th February 2024

Today, I had a meeting with the Interaction Designer and Software Engineer O to discuss my upcoming task for the Simca Dashboard Frontend project. The task entails designing a button capable of triggering a Dialog box, which will display snapshot content. Within the Dialog, users will have the options to either cancel or download the content. If the user chooses to cancel, the Dialog will close; however, if they opt to download, a zip file containing snapshot JSON packets will be generated for download.

The Simca Dashboard Frontend project operates on a different codebase and structure compared to the Inspector project. Here, we're utilizing TypeScript and Redux Toolkit. Additionally, we've

implemented unit tests using Jest and end-to-end tests using Cypress to ensure the reliability and functionality of the project.
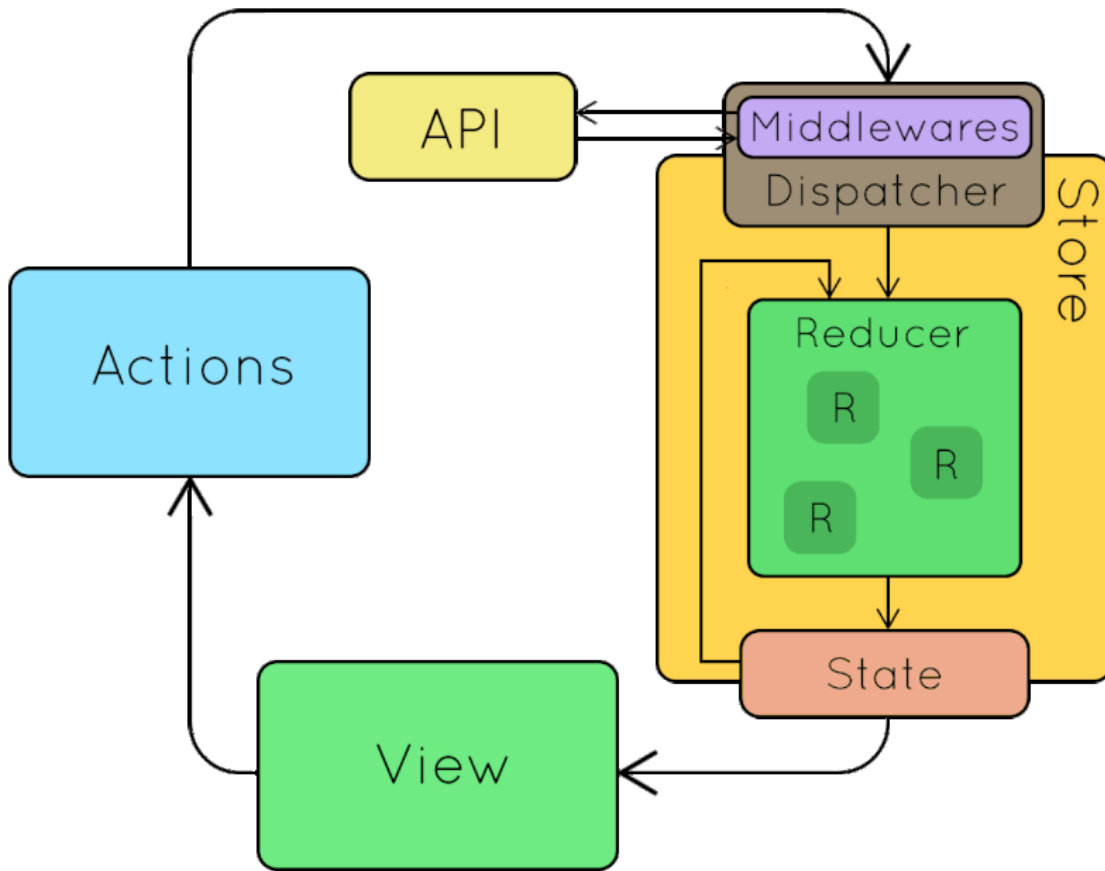
Figure 27: Redux toolkit Flow ("Redux Toolkit", 2023)

I came across an article on the LinkedIn platform that outlined the key features of Redux Toolkit, and I was pleasantly surprised to discover that my project structure incorporates these same key features. It's reassuring to see that the project aligns well with established best practices and frameworks like Redux Toolkit, which are known for enhancing developer productivity and maintaining clean, efficient codebases. This realization underscores the effectiveness of our approach and provides validation for the choices made in structuring the project.

Key Features 🔑:

createSlice: Effortlessly create Redux slices with reducers and actions in one place, promoting modular code.

configureStore: Simplifies store configuration by setting up a Redux store 🏬 with middleware and other configurations in a concise manner.

Immutability Helpers: Built-in utilities allow you to update state immutably without diving deep into complex logic.

Async Logic: createAsyncThunk offers a straightforward approach to handle async operations like fetching data 🗞️, making API calls, etc., complete with built-in loading, success, and error states (Rana, 2023).

Thursday 6th February 2024

Today, I had a meeting with the Simca teams in the morning, during which I updated my team lead on my role transition within the Simca project. I informed them that I've shifted focus to working on the user interface alongside Software Engineer O. Following the meeting, I immediately delved into the task assigned to me yesterday.

After conducting research on the project's structure and codebase, I began strategizing on how to approach the task at hand. I decided to start by designing the feature to download a zip file of snapshot packages. The first step was installing jszip into the project. I utilized the command "pnpm add jszip --save-dev" to achieve this. Notably, our project utilizes pnpm instead of npm. PNPM, as I discovered from an article by Ayoub (2023), is a superior alternative to npm, offering increased speed and efficiency. It's built on top of npm but boasts enhanced disk efficiency and addresses inherent issues present in npm. With jszip successfully integrated into the project, I proceeded to implement it. I created a function named "handleSnapshotDownload," which facilitates the creation and download of a snapshot of data in JSON format, compressed into a ZIP file within the client-side browser environment.

To display the snapshot content in a Dialog with JSON format, I discovered that ReactJson could efficiently fulfill this requirement. Utilizing ReactJson, I can seamlessly present the JSON data format within the dialog by simply providing the snapshot as a prop to ReactJson. To incorporate ReactJson into the project, I followed a similar process as with jszip. Using the command "pnpm add react-json-view --save-dev," I installed react-json-view into my package.json file. This ensures that I have access to the necessary functionality and components to visualize the JSON data format effectively within the dialog. I make my first commit to the new branch today, "Initial commit".

```
{
 1    a: "react-json-view-cool",
 2    b: 666,
 3    c: {
 4       g: {
 5          class: "JsonView",
 6          home: "China ❤️  ShenZhen",
 7          by: "Bytedancer"
 8       }
 9    },
10    d: [
11       1,
12       2,
13       "three",
14       null,
15       [...] (5 items)
22    ],
23    arr: [
24       {...} (1 items),
27       {...} (1 items),
30       {...} (3 items)
35    ],
36    nul: null,
37    undefine: undefined,
38    bool: true,
39    reg: /\w{5,}\d{5,}[a,b,c]?/i,
40    date: "Mon Mar 30 2020 20:38:11 GMT+0800 (中国标准时间)",
41    Nan: NaN,
42    function: ƒ() {...},
43    symbol: "Symbol(hello)",
44    error: "Error: error message",
45    set: "[object Set]",
46    map: "[object Map]",
47    noArg: {
48    }
}
```

Figure 28: ReactJson view ("react-json-viewer", 2022)

Friday 7th February 2024

I began my morning by diving into the snapshot task. First, I logged my hours from yesterday and updated the progress of the Jira ticket to "In Progress." With the administrative tasks taken care of, I seamlessly transitioned back to coding. My next objective was to design a Dialog containing the ReactJson view as its content, along with implementing a button to open and close the Dialog. Drawing from my experience with the Dialog component from the Inspector project, I felt confident in tackling this task without any issues. I create my first merge request for my teammate, Software Engineer O, to review.

In the afternoon, I had a meeting with my new Line Manager, marking a day filled with mixed emotions for me. Firstly, I received the fantastic news that Nokia has chosen to extend my Trainee contract for another six months. This recognition of my hard work and contributions to the company filled me with immense happiness, especially considering the challenging circumstances Nokia has faced this year with numerous layoffs. Knowing that my efforts are valued and appreciated by the company meant a lot to me. However, amidst this joyous news, I also received an offer from another company, Smartly.io, for a position as a Junior Software Engineer. This presented a difficult decision for me as I had to choose between two promising opportunities. After careful consideration, I made the tough choice to accept the offer from Smartly.io, believing it to be the best decision for the growth of my career.

As a result, I will leave Nokia in the coming weeks. It is bittersweet to leave behind the company where I've gained valuable experience and made meaningful connections.

Week 7 evaluation

Throughout Week 7, I've observed significant growth in various competencies, particularly in project management, technical skills, and decision-making. By successfully transitioning to a new project within Nokia and actively contributing to the Simca Dashboard Frontend development, I've demonstrated adaptability, teamwork, and proficiency in utilizing technologies such as TypeScript, Redux Toolkit, and ReactJson. Additionally, my ability to navigate complex tasks, such as designing a Dialog component and integrating ReactJson, showcases an enhanced understanding of software engineering principles and practices.

During the week, I encountered several areas requiring clarification, notably in understanding the project structure, codebase, and specific task requirements within the Simca Dashboard Frontend project. Clarifying these issues involved thorough research, leveraging available resources such as documentation, discussions with team members, and reviewing past project experiences. By seeking clarification proactively, I ensured a clear understanding of project objectives and effectively aligned my efforts with the team's goals.

Several challenges arose during the week, including determining the best approach to implement new features, resolving compatibility issues with dependencies, and balancing conflicting offers from multiple job opportunities. To address these challenges, I utilized problem-solving strategies such as research, consultation with peers and mentors, weighing the pros and cons of different options, and prioritizing based on long-term career goals. Ultimately, I resolved compatibility issues by updating dependencies and opted for a transparent and considerate approach in accepting the job offer from Smartly.io.

During Week 7, I leveraged external sources to enhance my understanding of Redux, microservice frontend architecture, and CI/CD practices. These sources included:
- Redux Documentation: The Redux documentation served as a fundamental resource for deepening my understanding of advanced Redux concepts. By referring to the official documentation, I gained insights into scalable state management solutions, which proved invaluable in the development of the Simca Dashboard Frontend project.
- "Building Micro-Frontends" by Cam Jackson: Exploring concepts from "Building Micro-Frontends" (Jackson, 2020) provided me with a comprehensive understanding of microservice frontend architecture. This resource enriched my knowledge of scalable front-end development practices, allowing me to make informed architectural decisions in my projects.

- "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley: Engaging with the principles outlined in "Continuous Delivery" (Humble & Farley, 2010) helped me adopt best practices in CI/CD pipelines. By understanding the concepts of automated software delivery, I was able to improve project productivity and reliability through efficient deployment processes.

In my pursuit of industry best practices, I meticulously compared my implementation strategies with insights gleaned from external sources. Through a comprehensive analysis of real-world Redux middleware examples alongside the Redux documentation, I adeptly tailored middleware architecture within my projects. This comparison ensured efficient handling of asynchronous data fetching requirements, in strict adherence to established patterns and practices. Delving into various microservice architecture implementations outlined in "Building Micro-Frontends" (Jackson, 2020), I meticulously compared different approaches. This scrutiny empowered me to make informed decisions, fostering modularity and maintainability within my codebase, and aligning with industry standards. Immersing myself in the intricacies of CI/CD pipeline configurations elucidated in "Continuous Delivery" (Humble & Farley, 2010), I meticulously evaluated my deployment processes against the gold standard of software delivery automation. This scrutiny resulted in the optimization of my pipelines for build, test, and deployment, ensuring a consistent and reliable software delivery mechanism. Through these meticulous comparisons and the integration of external insights, I bolstered my proficiency in crafting scalable, maintainable software solutions. My dedication to continuous learning and the application of industry-standard practices in front-end development remains unwavering.

In navigating the decision between continuing with Nokia or transitioning to Smartly.io, I undertook a thorough evaluation of various factors crucial to my career trajectory. While Nokia offered stability and familiarity, Smartly.io presented an enticing prospect of accelerated career growth and exposure to new challenges. The decision to prioritize career growth and accept the offer from Smartly.io was influenced by several key considerations, chief among them being the dynamic and modern work environment offered by Smartly.io. Their innovative approach to work and emphasis on cutting-edge technologies resonated with my aspirations for professional development and alignment with contemporary industry standards. Additionally, the cultural fit at Smartly.io played a pivotal role in my decision-making process. The company's culture, characterized by openness, collaboration, and a forward-thinking mindset, closely aligned with my own values and work ethos. The prospect of being part of a vibrant and progressive work culture where innovation is encouraged was immensely appealing.

However, it's crucial to acknowledge the significance of the offer to extend from Nokia. This offer served as a testament to the value of my contributions at Nokia and underscored the recognition of my performance within the organization. It reaffirmed the importance of the relationships forged and the impactful work accomplished during my tenure at Nokia. While ultimately choosing to embark on a new journey with Smartly.io, I deeply appreciate the opportunities and experiences gained at Nokia. The decision was not taken lightly, and I remain grateful for the support and growth opportunities afforded to me during my time at Nokia.

Observation week 8 (February 12 – January 16, 2024)

Monday 12th February 2024
Following my decision last week, I shared it with my team, Line Manager, and all the individuals

I've collaborated with during my time here. We reminisced about our shared experiences and expressed gratitude for the invaluable lessons learned together.

That's both good and bad news 🙌 Bad news for Nokia, but excellent for you!

I hope you have had good opportunity to learn how this kind of massive organization like Nokia is operating, it's quite different compared to more agile and smaller and perhaps more modern teams.

❤️ 1

Sad to hear that 🥺 It was nice to work with you

❤️ 1

Figure 29, 30: Feedback from my colleagues.

Despite my decision to move on, I've committed to finishing my remaining tasks during my final week at Nokia. I've received an additional task within the same project. My merge request from last week underwent review by Software Engineer O, who provided feedback for improvement. To enhance test coverage, my next step involves implementing a new test case. Additionally, I'll be simplifying the code structure by creating a new file called "Snapshot Dialog." This dialog will be integrated into the Toolbar so that when users click the Download button, it will appear, allowing them to download the snapshot content. The snapshot data will be retrieved using useLazyGetSnapshotQuery directly within the Snapshot Dialog.
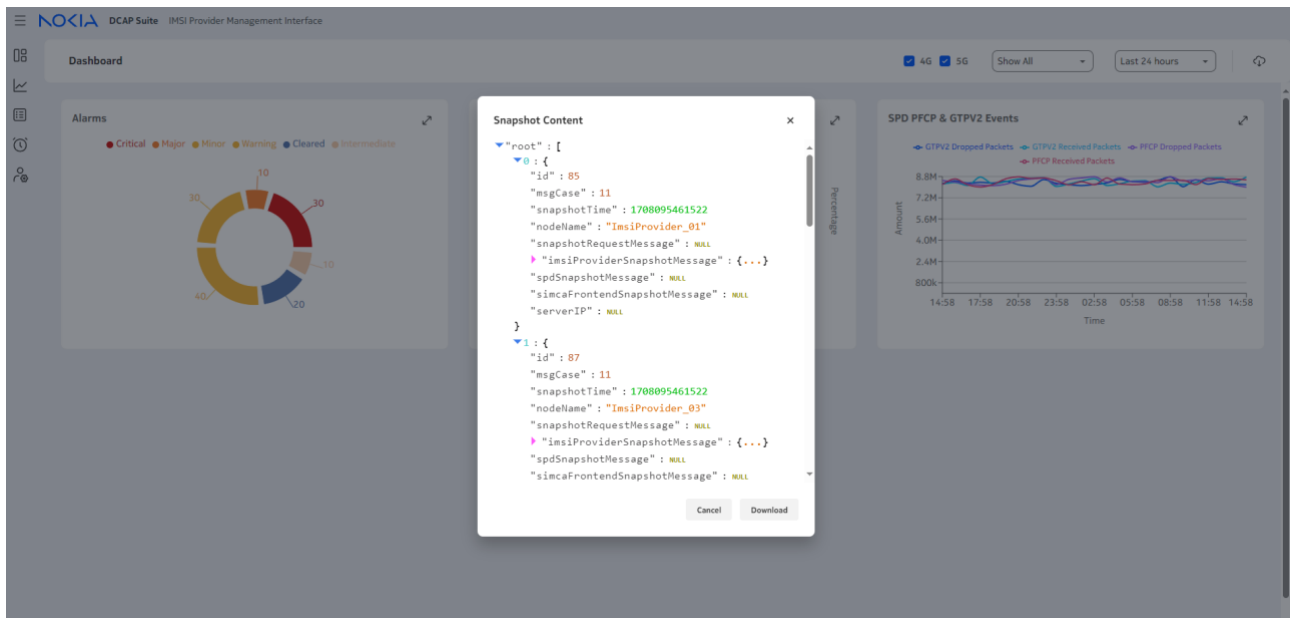
Figure 31: Snapshot Dialog screenshot

Tuesday 13th February 2024

Today, I focused on implementing error and loading handling for the snapshot dialog. Leveraging our base component, StateIndicator, which accepts a type of string, I ensured it displays the appropriate status for each case. The use of Redux Toolkit as our state management tool proved immensely beneficial for this task. Specifically, while I could directly receive isLoading and error

states from useLazyGetSnapshotQuery, I also implemented try and catch error handling for scenarios involving downloading zip files and fetching snapshot data. This comprehensive approach enhances the robustness and reliability of our application.

Figure 32: Error handling screenshot

After completing the error and loading handling, I pushed the changes to the branch and awaited the pipeline to pass. I'm pleased to report that this time, the pipeline passed without encountering any errors. Furthermore, all the test cases have been successfully covered, ensuring the stability and reliability of the codebase.

Wednesday 14th February 2024

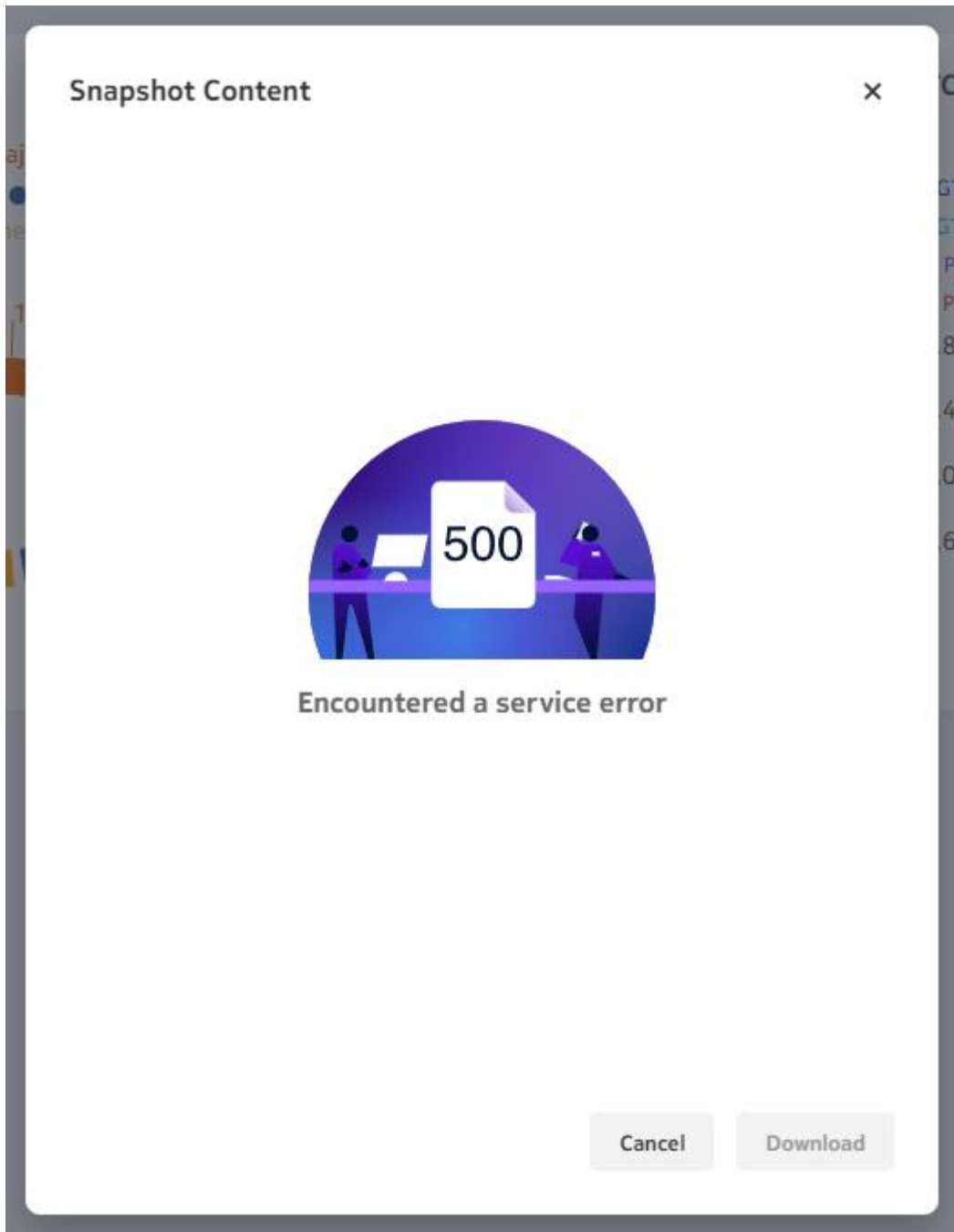Today, I had a meeting with Software Engineer O to discuss the process of pulling and pushing the Docker image to test the branch on the server. The procedure is quite like what we do in the Inspector project. However, since this is a different server, I needed to add the server to the C:\Windows\System32\drivers\etc\hosts file. This required opening the file with Notepad and adding the server's URL link. Once this was done, the URL link worked seamlessly.

The Docker command resembles that of the Inspector project, where we commented and removed the current image and replaced it with my Docker image. Following this, I pushed the updated image to the server for testing purposes.

Additionally, I updated the mock data for the snapshot content to ensure it matches more closely with the real response received from the backend. This involved extracting data from the response in the developer console of the server. Furthermore, I adjusted the width of the dialog to accommodate the expanded snapshot data content.
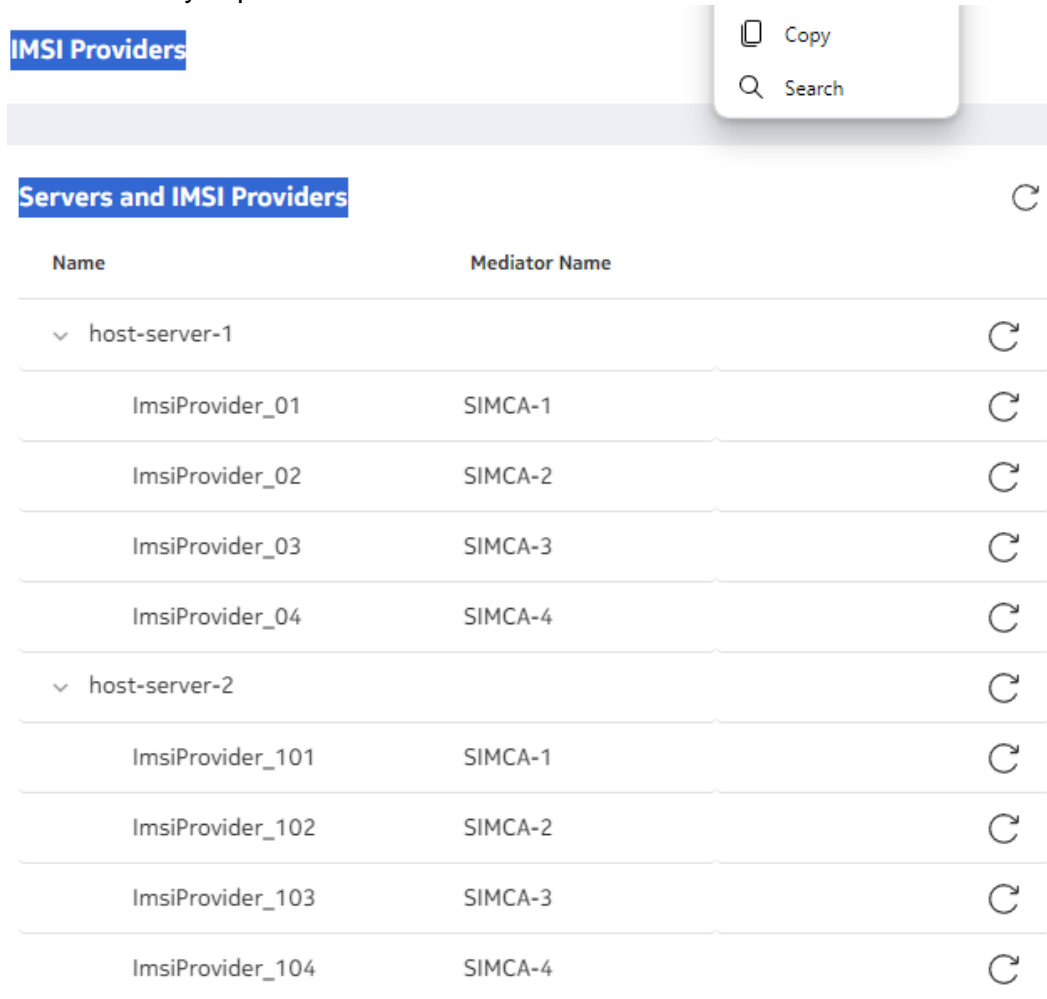
Towards the end of the day, I dedicated time to reviewing the translation support branch from Software Engineer O. I thoroughly looked at all the changes made to the files and provided comments for her merge request. Following the completion of tasks, I updated the status of my Jira ticket to "Done" to reflect the progress made. Additionally, I logged the hours spent working on the tasks for today, ensuring accurate documentation of my contributions.

Thursday 15th February 2024

The snapshot dialog task took up three days of this weeks' time allocation. Despite this, as it was my initial task in this project, I'm confident that I can complete the remaining tasks within the week. Today, I began by receiving the final task assigned to me during my time at Nokia: adding row action buttons to open the Imsi provider detailed view.

Working with row action buttons isn't particularly challenging for me, as I've tackled similar tasks in the Inspector project. However, since this project employs TypeScript, there are bound to be some nuances. One immediate issue I encountered was that the buttons were appearing in every line of the Data Grid table, including the host-server line. Our intention is for the buttons to only be displayed in the line associated with the Imsi provider. Additionally, importing a new icon into the project proved to be more complicated than expected. I encountered errors when attempting to import it from the Nokia CCFK library. This necessitates troubleshooting and finding a workaround

to successfully import the icon.

IMSI Providers

	Copy
	Search

Servers and IMSI Providers

Name	Mediator Name	
⌄ host-server-1		
ImsiProvider_01	SIMCA-1	
ImsiProvider_02	SIMCA-2	
ImsiProvider_03	SIMCA-3	
ImsiProvider_04	SIMCA-4	
⌄ host-server-2		
ImsiProvider_101	SIMCA-1	
ImsiProvider_102	SIMCA-2	
ImsiProvider_103	SIMCA-3	
ImsiProvider_104	SIMCA-4	

Figure 33: Row action buttons screenshot.

After discussing with Software Engineer O, we determined that I needed to add the new icon source to the ccfkMissingDeclarations file to resolve the import issue. Additionally, to address the button display problem, I needed to adjust the logic of the cell renderer. Specifically, I ensured that the button is only shown for parameters with the name of the Imsi provider; otherwise, it would return null.

By the end of the day, I created a merge request for this task, marking it as a productive day with significant progress made.

Friday 16th February 2024

After my merge request passed the pipeline, I encountered a conflict between my branch and the master branch. This conflict arose because Software Engineer O had merged a branch before

mine. Thus, I needed to resolve the conflict. Once resolved, I sought review from Software Engineer O to ensure the changes were aligned with project standards.

Following conflict resolution, I proceeded to test the changes on the real server using my Docker image. To ensure code quality, I utilized SonarQube, a tool that helps in checking the quality gate status, measurement, and coverage on lines. After reviewing the code coverage and removing any duplicated or unnecessary code, I was able to proceed with merging my branch into the master branch.

Successfully completing the task before the upcoming week, I plan to return my laptop and other belongings to the Nokia office in the middle of next week, marking the end of my time here.

Week 8 evaluation

Reflecting on the tasks accomplished throughout Week 8, I perceive significant growth in various competencies, particularly in problem-solving, collaboration, and technical proficiency. By tackling tasks such as implementing error and loading handling for the snapshot dialog, resolving conflicts during merge requests, and adding row action buttons to the Data Grid table, I've demonstrated adaptability, attention to detail, and enhanced technical skills, especially in TypeScript and Docker usage. Throughout the week, I encountered several issues that required clarification. One notable challenge was understanding the logic behind displaying row action buttons only for parameters associated with the Imsi provider. Additionally, resolving conflicts arising from merging branches with Software Engineer O's changes demanded clear communication and collaboration to ensure seamless integration.

Several problems emerged during the week, including conflicts during branch merging and importing icons from the Nokia CCFK library. To address these challenges, I employed strategies such as resolving conflicts through collaboration with team members, seeking guidance from colleagues like Software Engineer O, and troubleshooting import errors through thorough investigation and experimentation.

From literature and industry best practices, several recommendations can enhance work assignments. One such practice is leveraging tools like SonarQube to ensure code quality and identify areas for improvement, such as reducing duplicated or unnecessary code. Additionally, implementing thorough testing procedures, including unit tests and integration tests, can enhance the reliability and robustness of the codebase. Reviewing best practices guides and tutorials from reputable sources provided me with a framework for evaluating my conflict resolution strategies and SonarQube configurations. By comparing my practices with established best practices, I could identify areas for improvement and optimize my workflow to achieve better outcomes. Analyzing case studies and success stories shared by software development communities offered valuable insights into effective conflict resolution strategies and SonarQube implementations. By studying real-world examples, I gained a deeper understanding of the challenges and solutions encountered by other teams, enabling me to make informed decisions and adapt my practices accordingly. In terms of alternative approaches, maintaining a clearer communication channel with team members could have expedited conflict resolution during branch mergers. Additionally, incorporating more extensive documentation for importing icons from external libraries could streamline the process and mitigate errors.

The official Git documentation provided invaluable guidance on conflict resolution strategies during branch merges. By referring to Git's documentation, I gained a comprehensive understanding of Git's merge conflict resolution tools and techniques, enabling me to effectively manage conflicts

that arose during collaborative development. Community-driven platforms like Stack Overflow proved to be valuable resources for troubleshooting specific issues encountered during branch merges. Participating in relevant discussions and seeking advice from experienced developers helped me gain practical insights and alternative approaches to resolve conflicts efficiently. When faced with conflicts during branch merges, I evaluated different options, including resolving conflicts independently and seeking assistance from Software Engineer O. Ultimately, I chose the latter option as it ensured adherence to project standards and facilitated smoother integration of changes. Engaging with discussions on GitHub forums allowed me to explore real-world scenarios and best practices related to conflict resolution in Git repositories. By learning from the experiences and recommendations shared by other developers, I was able to refine my conflict resolution skills and adapt my approach to suit the project's requirements. In the future, I aim to incorporate more structured documentation practices and explore automated conflict resolution tools to expedite the process further. Seeking advice and recommendations from experts in the field through forums, blogs, and professional networks provided me with valuable guidance on navigating complex Git workflows and maximizing the benefits of SonarQube. By incorporating expert insights into my workflow, I could leverage proven techniques and tools to enhance the quality and reliability of my codebase.

In summary, Week 8 marked significant progress in my competence development, addressing various challenges with adaptability and collaboration while incorporating industry best practices to enhance work assignments. Moving forward, I aim to continue leveraging these experiences to further refine my skills and optimize my contributions to the team.

Discussion

In this discussion, we embark on a reflective journey, juxtaposing the initial situation outlined at the start of this thesis with the insights gleaned from diary entries. This comparison goes beyond surface-level analysis; it delves into the nuances of professional growth, the emergence of novel methodologies, and the learning curve experienced throughout this diary-driven exploration. We'll explore the evolution observed over time, the inventive solutions and techniques that emerged, and the multifaceted learnings accrued throughout this journey.

Reflecting on my journey since starting my career nearly three months before commencing this thesis in September 2023, I can discern significant growth within myself. Initially, I found myself grappling with feelings of panic and overwhelm when confronted with the project's codebase, despite my confidence during the job interview. The divergence between my background in web development, focusing on both front-end and back-end, and my first task at Nokia, which involved designing a program to read network data from the Network Interface Card, added to my sense of unease. Each day at work was marked by extreme stress, not necessarily because I couldn't meet task deadlines, but rather due to the high expectations I placed on myself. I harbored aspirations of becoming an invaluable asset to the team from the outset, but I soon realized that achieving such a status wasn't a straightforward endeavor. It dawned on me that it takes years to ascend to senior roles and even more years to assume managerial positions. Moreover, I hesitated to seek assistance from coworkers, fearing that I would come across as a nuisance, and I felt compelled to tackle problems single-handedly. However, this approach didn't always yield optimal results; often, it took me longer to resolve issues, or I ended up needing to seek help anyway. Nevertheless, I've come to appreciate that there's value in attempting to tackle challenges independently, as it fosters personal growth and learning. While there are instances where seeking assistance is necessary, there are also situations where I can derive valuable insights and skills from tackling problems using my own logic and research. This realization has been instrumental in shaping my approach to problem-solving and seeking support within the team.

After a few months of being employed, I've managed to acquire several invaluable skills that have proven immensely beneficial in my role. One of the most significant developments is my increased comfort level with communicating with coworkers to streamline the development process. I've realized that it's perfectly acceptable not to know everything and have become more proactive in seeking assistance when needed. Understanding that asking for help isn't a sign of weakness but rather a means to save time and enhance productivity has been crucial in fostering a collaborative work environment. Throughout my tenure, I've honed essential skills such as the ability to learn quickly through hands-on experience and independent research. I've come to acknowledge that relying solely on theoretical knowledge from academic institutions isn't sufficient for the demands of the workplace. In fact, the practical skills I've gained through on-the-job learning and specialized training programs have far outweighed what I acquired through formal education. Drawing from my experiences as a student, I've leveraged my problem-solving and research skills to navigate through various challenges encountered in my role. One aspect of programming that I've come to realize is that the bulk of the time is often spent pondering over solutions rather than writing code. Debugging is a critical skill that consumes a significant portion of my working hours— approximately 50%. While debugging unfamiliar issues can be challenging, I'm not hesitant to seek assistance from my team lead or fellow developers when necessary. The culture of mutual support within the team fosters increased productivity, which ultimately contributes to the company's overall success and revenue generation.

Having solid fundamental programming skills in various languages has proven to be incredibly advantageous in my role. Being proficient in object-oriented programming languages like JavaScript at an advanced level and C# at a medium level has facilitated my ability to work with and swiftly learn new languages or frameworks. Much of my current knowledge has been acquired through hands-on experience, which has enabled me to adapt quickly to new technologies and tools. For instance, when I first started this job, I had little to no knowledge about SharpPcap or Pcap.NET. However, my familiarity with C# ASP.NET allowed me to integrate myself swiftly into the company's projects and effectively contribute to tasks involving these technologies. Additionally, delving into micro-frontend development within a monorepo has been an intriguing learning experience. It's noteworthy that my current position at Smartly.io also utilizes a similar repository structure for managing frontend packages, which has further facilitated my understanding and adaptation to this approach. Furthermore, exploring the intricacies of CI/CD pipelines in product development has provided me with valuable insights into the full lifecycle of front-end and back-end development. Understanding how each part contributes to the overall product development process has been enlightening, allowing me to grasp the interconnectedness of various components and optimize my workflow accordingly. Overall, my journey of learning and applying programming skills has been instrumental in my professional growth and effectiveness in tackling diverse challenges within the software development domain.

My journey from an entry-level frontend developer to a junior fullstack developer has been swift and rewarding. Through focused learning and hands-on experience, I've significantly enhanced my skills in React.js for front-end development while also gaining valuable insights into back-end technologies and database management. Today, I confidently navigate both frontend and backend tasks, bridging the gap between client-side and server-side operations. While there's more to learn, this transformation underscores the power of continuous growth in web development.

Mastering the art of writing clean code has been a significant aspect of my growth journey. Understanding that clean code fosters clear communication and facilitates collaboration with fellow developers, I've prioritized this skill in my development process. Clean code not only enhances readability and maintainability but also ensures a higher level of security in software applications. By embracing feedback from experienced developers and maintaining an open mind, I've honed my ability to write code that is concise, understandable, and easy to maintain. As the newest member of the team, I've approached this learning process with enthusiasm and dedication, recognizing its value in becoming a better teammate and developer.

My proficiency in Git, acquired through university courses and BootCamp training, has been invaluable in my professional journey. While I initially learned the basics of creating branches, sending pull requests, and merging during personal projects and exercises, I've enhanced my understanding and utilization of these skills within a large development team context. Collaborating with fellow developers on projects with frequent file changes has provided me with practical experience in resolving conflicts in Git. Through hours of research and trial and error, I've developed strategies for managing conflicts effectively and mitigating accidental deletions of others' code. Additionally, I've expanded my Git toolkit by learning to use commands like git merge and git reset, enabling me to navigate complex version control scenarios with confidence and efficiency.

Developing a resilient mentality has been a crucial aspect of my professional growth, transcending the realm of programming or web development. Learning to cultivate patience has proven invaluable in my journey as a developer, especially during debugging, research, and problem-solving tasks that can span hours, days, or even weeks. I've come to understand that impatience can often lead to discouragement and a temptation to give up, despite the proximity of the ultimate

goal. Moreover, honing the ability to work under pressure has been instrumental in my progress. Balancing work responsibilities, study commitments, and thesis writing was undoubtedly challenging, but with determination and motivation, I navigated through it successfully. Ultimately, the resilience and perseverance I've developed have been rewarded with well-deserved achievements.

Managing my time effectively became paramount as I juggled work responsibilities, academic pursuits, and thesis writing alongside seven courses at HaagaHelia, totaling 32 credits, during the challenging Spring 2024 semester. Aware of the significant financial support my parents provided for my education, I felt compelled to optimize my study time to alleviate their burden and advance my own growth. Despite the demanding schedule, I prioritized maintaining my physical and mental well-being. This involved adhering to a healthy diet, regular exercise regimen at the gym to manage stress, and nurturing social connections with friends. Open communication with understanding teachers enabled me to navigate the workload more efficiently, with some allowing flexibility in attendance and assignment deadlines. This disciplined approach ensured that I could handle the semester's demands without major setbacks, reinforcing my belief in the adage, "The harder you work, the luckier you get" by Gary Player.

The journey of crafting this diary-based thesis has been transformative, enriching my comprehension of both professional and personal growth. Cultivating a reflective practice has emerged as a cornerstone of this journey, fostering heightened self-awareness through the daily ritual of documenting thoughts, actions, and responses. This practice facilitated not only a deeper understanding of workplace dynamics but also illuminated recurring patterns and reactions to challenges, guiding my problem-solving strategies. Moreover, maintaining a diary honed my ability to deconstruct intricate problems into manageable components, discern critical variables, and grasp their interconnectedness, vital skills in my field where navigating complex system issues is routine. The diary entries served as a repository of insights, preserving valuable learnings that might otherwise have been overlooked amidst the busyness of daily tasks. A paramount lesson gleaned from this process was the importance of adaptability and agility in a dynamic workplace. The entries underscored the necessity of swift adaptation to evolving circumstances, preparing me for a future where flexibility is indispensable. Furthermore, the thesis process enhanced my capacity to articulate thoughts and ideas logically. Consistent writing not only sharpened my written communication skills but also refined my cognitive processes. This newfound clarity in expression will undoubtedly serve me well in future endeavors, whether in delivering presentations, crafting reports, or collaborating on projects.

Following the conclusion of my thesis, I am committed to further enhancing my professional competencies in project management, programming, and cloud computing. In addition to the invaluable learning opportunities afforded by my work, I have set ambitious goals for additional skill development. One such objective is pursuing certifications like the AWS Cloud Practitioner, which will furnish me with advanced strategies and frameworks essential for cloud development proficiency. Engaging in diverse projects will not only offer practical experience but also insights into various management styles and the complexities of project execution. Remaining abreast of the latest trends and technologies in user experience is imperative for effective usability testing. Therefore, I am dedicated to staying informed about developments in this field to ensure continuous growth and proficiency. Embracing a proactive and lifelong learning ethos will be instrumental in achieving mastery in these pivotal professional domains.

References

Jira (2022).

URL: https://www.productplan.com/glossary/jira/. Accessed 1 February 2024.

Rasmusson, J. (2010). The Agile Samurai. Pragmatic Bookshelf. Accessed 1 February 2024.

Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall. Accessed 1 February 2024.

Anie, Alex. "UseLocation Hook in React Router." Medium, 27 Sept. 2023.

URL: medium.com/@alexanie_/https-ocxigin-hashnode-dev-uselocation-hook-in-react-router-758a0a711308. Accessed 26 December 2023.

"Git Clone Command - Studytonight." Studytonight.

URL: www.studytonight.com/git-guide/git-clone-command. Accessed 27 December 2023.

"Git Merge vs Git Rebase: Which One to Use?" Data Science Simplified, 8 Aug. 2023.

URL: mathdatasimplified.com/git-merge-vs-git-rebase-which-one-to-use/. Accessed 28 December 2023.

"GTPv2 - GPRS Tunnelling Protocol Version 2." Mpirical.

URL: www.mpirical.com/glossary/gtpv2-gprs-tunnelling-protocol-version-2. Accessed 28 December 2023.

"How to Install NPM and Node.js on Windows and Mac Devices?" Radixweb, 18 Nov. 2021.

URL: radixweb.com/blog/installing-npm-and-nodejs-on-windows-and-mac. Accessed 29 December 2023.

"How to View All Logged Time by Users in Jira?" Atlassian Community, 9 Mar. 2021

URL: community.atlassian.com/t5/App-Central/How-to-view-all-logged-time-by-users-in-Jira/ba-p/1633245. Accessed 1 January 2023.

Mishra, Abhishek. "8 Quick Ways to Open Remote Desktop Connection Tool in Windows 11." Guiding Tech, 26 May 2023.

URL www.guidingtech.com/how-to-open-remote-desktop-connection-in-windows/. Accessed 2 January 2024.

Nokia. "PFCP Protocol." Documentation.nokia.com.

URL: documentation.nokia.com/mag-c/23-10-1/books/pfcp/pfcp-protocol.html. Accessed 3 January 2024.

"SharpPcap 6.2.5." Nuget.org,

URL: www.nuget.org/packages/SharpPcap/#readme-body-tab. Accessed 4 January 2024.

"What Is Network Interface Card: Purposes, Functions & Types | FS Community." FS Community, 7 May 2019.

URL: community.fs.com/article/nic-card-guide-for-beginners-functions-types-and-selection-tips.html. Accessed 4 January 2024.

Budd, A. (2006). CSS Mastery: Advanced Web Standards Solutions. Accessed 6 February 2024.

Poulton, N. (2017). Docker Deep Dive. Pearson Education. Accessed 6 February 2024.

Turner, Aaron. "Appneta/Tcpreplay." GitHub, 13 Feb. 2024,

URL: github.com/appneta/tcpreplay. Accessed 11 January 2024.

Docker. "What Is a Container?" Docker, 2023,

URL: www.docker.com/resources/what-container/. Accessed 16 January 2024.

Spurgeon, C. E. (2000). Ethernet: The Definitive Guide. O'Reilly Media. Accessed 13 January 2024.

Docker. "Docker Compose Documentation." Docker,

URL: https://docs.docker.com/compose/. Accessed 16 January 2024.

Docker. "Docker image pull." Docker Documentation,

URL: https://docs.docker.com/engine/reference/commandline/image_pull/ Accessed 16 January 2024.

Morgan, Chris. "Packet.net." GitHub, 13 Sept. 2023,

Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Crown Business. Accessed 20 January 2024.

URL: github.com/dotpcap/packetnet. Accessed 24 January 2024.

Fetisov, Evgeniy. "JavaScript vs. TypeScript: Which Is Better for Your Project in 2023?" JayDevs, 6 Mar. 2023.

URL: jaydevs.com/javascript-vs-typescript/. Accessed 26 January 2024.

Pavlutin, Dmitri. "A Simple Explanation of React.useEffect()." Dmitri Pavlutin Blog, 13 Oct. 2020.

URL: dmitripavlutin.com/react-useeffect-explanation. Accessed 31 January 2024.

Jackson, Cam. "Micro Frontends." Martinfowler.com, 19 June 2019.

URL: martinfowler.com/articles/micro-frontends.html. Accessed 4 February 2024.

Rana, Maham. "Redux Toolkit." Www.linkedin.com, 29 Dec. 2023.

URL: www.linkedin.com/pulse/redux-toolkit-maham-rana-od06f/. Accessed 5 February 2024.

Abidi, Ayoub. "Choosing the Right Package Manager: NPM, Yarn, or PNPM? | HackerNoon." Hackernoon.com, 14 Sept. 2023.

URL: hackernoon.com/choosing-the-right-package-manager-npm-yarn-or-pnpm. Accessed 6 February 2024.

Jackson, C (2020) "Building Micro-Frontends.". Accessed 8 February 2024.

Humble, J, and Farley, D. (2010) "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation.". Accessed 8 February 2024.

"Npm:react-Json-Viewer-Cool | Skypack." 9 Feb. 2022.

URL: https://www.skypack.dev/view/react-json-viewer-cool. Accessed 6 February 2024. Accessed 9 February 2024.

Appendices

Appendix 1. xxx