



Working as a software developer at WeAre Solutions Oy Finland

Duc Ngo

Haaga-Helia University of Applied Sciences

UAS Thesis

2022

Bachelor of Business Information Technology

Abstract

Author(s) Duc Ngo
Degree Bachelor of Business Information Technology
Report/Thesis Title Working as a software developer at WeAre Solutions Oy Finland
Number of pages and appendix pages 63 + 1
<p>In this type of thesis, the big picture of daily responsibility and development progress are depicted in 8 weeks working and writing d from 28th of March 2022 to 20th of May 2022. A diary-related thesis could also assist author in broadening his view and vision of the job and his professional life.</p> <p>In this paper, the learning process are recorded and assessed in order to encourage and reflect on self-development, to support and improve personal competencies, and to identify and address weaknesses. It is also useful for clarifying concepts, issues, and theories. Author sets three goals for himself at the start of the job: to expand his professional knowledge and training, to adapt and get familiar with a new international working environment, and to change his concentration from the front-end to other development aspects. In the end, all of the objectives have been achieved</p> <p>The thesis is made up of four themed chapters that the author wrote over the course of eight weeks while working as a consultant. This thesis is mainly about daily observation and weekly evaluation. The daily will summarize the day's key duties and events, whereas the week evaluation will dig deeper into the author's professional growth progress and theories and research that he learned during the week. The technology will be diverse from Typescript, VueJs, Cypress, Mocha, NodeJs, Docker, MySQL, PostgreSQL.</p> <p>This thesis would be beneficial to undergraduates or anyone interested in learning more about what a junior software engineer performs at work in general. As a result, they can prepare well while still at university and improve the skills needed in the workplace.</p>
Keywords Software development, frontend development, VueJs, Docker, WeAre Solutions, Typescript, web development, Cypress, PostgreSQL

Contents

1	Introduction.....	1
2	Description of the initial situation	5
2.1	Analysis of your current work	5
2.1.1	Projects	7
2.1.2	Evaluation and development plan	8
2.2	Stakeholders	9
2.3	Interaction situations.....	10
3	Diary entries	11
3.1	Observation week 1 (March 28 – April 1, 2022).....	11
3.2	Observation week 2 (April 4 – April 8, 2022).....	18
3.3	Observation week 3 (April 11 – April 15, 2022)	24
3.4	Observation week 4 (April 18 – April 22, 2022)	29
3.5	Observation week 5 (April 25 – April 29, 2022)	34
3.6	Observation week 6 (May 2 – May 6, 2022)	40
3.7	Observation week 7 (May 9 – May 13, 2022)	44
3.8	Observation week 8 (May 16 – May 20, 2022)	50
4	Discussions and conclusions	54
	References.....	58
	Appendices	64
	Appendix 1. Appendix heading	64

1 Introduction

The purpose of this thesis is to document and evaluate the daily responsibilities and development progress of a junior software engineer at WeAre Solutions Oy. The practical part of the thesis was an eight-week work placement, which took place from 28th of March 2022 to 20th of May 2022. The thesis was written in the form of a working diary.

WeAre Solutions Oy was founded in 2019. It is a limited liability company in Helsinki and its main business is Management Consulting and building solutions and platforms for the digital age (WeAre Solutions 2022). The core principles, that form the foundation of every service and project we provide, are automate everything, test comprehensively and monitor rigorously. Building on those pillars results in solutions that can stand the test of time. The 4 main services is Identity and Access Management, Data management, Cloud services, Software development (WeAre Solutions Services 2022). I am a part of the automotive development team building software solutions for digital transform with main customer Bassdone, which is cars, spare, parts and accessories distribution company.

A software development team consists of software developers with different areas of expertise. They are occupied with different tasks to meet the customer's diverse demands. Depending on different technologies' requests, developers work on different tech stacks: TypeScript in the whole applications, Node in the backend, Vue in the frontend, Cypress and Mocha in the testing, MySQL and PostgreSQL in the database, AWS in the Cloud and Docker in delivering software.

My team is called Automotive Development team, consisting of 7 people with 4 developers working interactively on 3 projects. Each of developers usually works on 2 projects at the same time. We hold two monthly meetings to discuss our projects and their progress, as well as one monthly meeting to assess corporate activities such as sales, teams, customers, HR, culture, objectives, and motivation. These enable me to see the big picture of what I am doing through a variety of perspectives and fields inside the organization. Moreover, monthly dialogue is also applied. It is like a small conversation between developers and their team lead about behavior, desire, confusion, and other kind of problems. Our main communication channel is Slack. Contributions of ideas and feedback are always welcome and encouraged, and everyone's help is greatly appreciated.

Employees can work from home or at the office without restriction or limitation, making the working environment very flexible. I like working in an office setting because it exposes me to a working environment and atmosphere that is relatively new and exciting for me at the moment and helps me focus, but I also enjoy working from home since it allows me to be more flexible with my

schedule. However, there are some university classes that require me to be present in class, so I try to visit the office at least twice a week.

The framework used daily;

1. Locally installed software: Visual Studio Code, MySQL Workbench, Docker, 1Password, Git, NVM, Node, Sourcetree, Slack
2. Tools and services: Bitbucket, Jira, docker
3. Technologies and frameworks: VueJs, Cypress, ExpressJs, buefy, bulma, Mocha

The majority of the research and studies I've looked at in my thesis are a combination of diverse sources for varied purposes. The documentation sites (such as <https://vuejs.org/> for VueJs, <https://www.cypress.io/> for Cypress, and <https://buefy.org/> for Buefy), Medium or Hacker Noon blog pieces, and last but not least, Youtube for many tutorials and explanations are the most significant source of theoretical information. When I need to understand a concept thoroughly or know-how of a new framework, I learn from these sources about the general information, best practices, and practical advice. When I faced some bugs, it can be from the npm package, or my mistakes, I often get references from Stack Overflow, which is a professional online community for developers. It is a place where other developers ask questions and get answers from other developers or just some discussions. For me, there are already the answers to most of the questions I asked. It can save a lot of my debugging time and effort.

The most critical skill I need to prepare for this position is a comprehensive understanding of web technologies, including both frontend and backend development. My strong programming skills in JavaScript, TypeScript, REST API, Material UI, NodeJs and a broad understanding of Cloud, Docker container, and Database. These skills also involve the use of related frameworks and libraries, such as JavaScript, in the backend, we utilize ExpressJs with TypeScript, in the frontend, we use VueJs with TypeScript and Buefy for styling, in the database, we use MySQL and PostgreSQL. During my working time, I also studied other testing framework. I've been building and using end-to-end tests, unit tests, integration tests, and system tests using tools like Jest, Mocha, and Cypress.

In addition to programming and testing skills, Git actions is a critical ability that I must understand for my position. The most crucial and challenging aspect of utilizing Git is resolving conflicts. Because Git is a tool for managing a team's workflow, it's critical to figure out which part of code should be selected when two commits edit the same line of code. However, after using git for half of a year, I've gained confidence in using the tool.

Aside from the skills listed above, I am also familiar with Docker due to my work experience. I did not configure the container, but I worked with it to implement the database for local application and learned some important insights as a result. My work tasks are continually changing, thus I have to learn a lot of new abilities over my development period.

Buefy	Buefy is a Vue.js library of lightweight responsive UI components based on the Bulma framework and design. (github.com/buefy , <i>Buefy</i>)
CSS	Cascading Style Sheets (CSS) is a language for describing the appearance of a document written in a markup language like HTML. (W3school 2022, <i>CSS Tutorial</i>)
Cypress	Every developer and QA engineer requires the new developer-friendly standard in front-end testing. (Cypress.io)
Docker	A group of platforms as a service products that offer software in containers using OS-level virtualization. (Aws.amazon.com , <i>What is Docker</i>)
Git	A free and open-source distributed version control system that can handle modest to extremely big projects quickly and efficiently. (Git-scm.com , <i>Git</i>)
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. <i>JavaScript</i>)
Mocha	A powerful JavaScript test framework that runs on Node.js and in the browser, making asynchronous testing easy and enjoyable. (Mochajs.org , Mocha)
MySQL	A relational database management system that is open-source. (Mysql.com , <i>MySQL</i>)
PostgreSQL	Extensible and SQL-compliant relational database management system that is free and open-source. (Postgresqtutorial.com 2022, <i>What is PostgreSQL</i>)
REST API	An application programming interface (API or web API) that follows the REST architectural style's limitations and allows interaction with RESTful online

	services. Roy Fielding, a computer scientist, invented REST, which stands for representational state transfer. (redhat.com 2020, <i>What is a REST API?</i>)
SASS	Scripting language for preprocessors that is interpreted or compiled into Cascading Style Sheets. (Ssas-lang.com, SASS)
TypeScript	A superset of JavaScript. To support a stronger interaction with your editor, TypeScript adds additional syntax to JavaScript. In your editor, catch errors early. TypeScript code is converted to JavaScript, which may be used anywhere JavaScript is supported, including in browsers, Node.js, Deno, and your apps. TypeScript comprehends JavaScript and employs type inference to provide powerful tooling without the need for additional code. (typescriptlang, <i>TypeScript</i>)
VueJs	The Progressive JavaScript Framework is a set of tools for developing progressive web applications. Vue.js is a web-based UI framework that is progressive and incrementally adaptable. (github.com/vuejs, Vuejs)

Table 1: Key professional concepts

2 Description of the initial situation

2.1 Analysis of your current work

Companies looking for an all-encompassing framework to implement the Agile mindset frequently select the Scrum framework. Scrum's popularity is based on its capacity to help teams move and learn more quickly. According to Sandra (2022), although standalone Scrum is used by 56% of teams, when combined with hybrid methodologies, Scrum is used by 83% of companies. However, because of the size of the team, my team lead finds that it's not necessary to execute Scrum method at the moment with a really small team and our customers does not have any planned release schedules. We are using some Agile parts, such as Kanban board and bi-weekly meetings. Two meetings per month are held to review our projects and their progress, and one monthly meeting is held to assess corporate operations such as sales, teams, customers, HR, culture, objectives, and motivation. These allow me to see the overall picture of what I'm doing across the organization from a variety of perspectives and fields. As a one-month junior developer, I didn't receive any high priority tickets that have deadline. Despite of that, I always strive to finish the tasks as soon as possible, and sometimes I can assign myself new task.

My main work tasks for 8 weeks working:

- Set up the local development environments
- Learn, research and implement features
- Set up testing environments and write End-to-End tests

Setting up local development environment for the project always is one of the first thing to do when joining a new product. Since most of the projects that I have joined have been built around 1 year ago. The projects are quite vast and complex, involving numerous layers, security configuration stages, Docker containers and databases. At the beginning, I had so much difficulty setting up local development myself because I never touch any applications with this complexity. Fortunately, these projects have same structure, and my team lead spent a day just for guiding me setting up. Because my current projects might be released in March 2022, I will have the opportunity to tackle a new project with a different structure in the near future. As a result, I'll have to create a new local development environment for future projects. Setting up may appear to be a one-time task throughout the development process, but it takes time and effort. Because projects, requirements, tech stack are different in projects and you are not going to set up very often, so it can be a little bit bewildered every time. My work tasks are continually changing; thus, I need to learn a lot of new

abilities over my development period.

The improvements and specifications of the projects are always evolving in order to satisfy and fulfill the needs of end-users. At the beginning, I needed to read and learn the code base, functions, structure, etc to implement the same features in other components. Moreover, any new features must be added to the present products by the development teams. As a new developer on the team, you will not only learn about the projects, but also about new technologies that can be used to implement new features. I found that the time I spent researching and learning during my development process was greater than the time I spent coding. However, there are times that I have been struggle in one problem for so long even though I strive my best to research but this is reducing the productivity, so I decided to ask for help from my team lead or other developers in the team. With years of experience and kindness, they instructed and explained my problem very deep and detailed, I learned a lot from those conversation and so grateful for that. I still need to be more aggressive in addressing new technologies; I'm still mostly focused on the requirements and instructions of the team lead. Communication and understanding abilities were one of the most critical qualities I needed for my job. To complete my job, I must properly comprehend the ticket description and communicate well with other developers.

Testing plays a very crucial role in software development. The fundamental advantage of testing is that it identifies and eliminates errors. Testing, on the other hand, aids developers and testers in comparing real and predicted outcomes to improve quality. If software is developed without being thoroughly tested, it may be ineffective or even dangerous to clients. End-to-end testing is a software development lifecycle (SDLC) methodology for evaluating an application's functionality and performance under product-like conditions and data to simulate real-world scenarios. The purpose is to recreate a real-world user scenario from beginning to end. (Smartbear.com 2022, *Combine API and UI Testing For Confidence At Every Layer Of Your Application*) Because end-to-end testing is about UI testing and integration testing between the frontend and backend, which I am most familiar with, this is one of the first types of tests I write.

2.1.1 Projects

Vehicle Tax App (CaTa): This project helps our customer Bassadone to handle, check, modify vehicles tax when importing those. When the vehicle is ready they can send the information of the tax to Tax API. Below is the picture of the list of main pages in CaTa

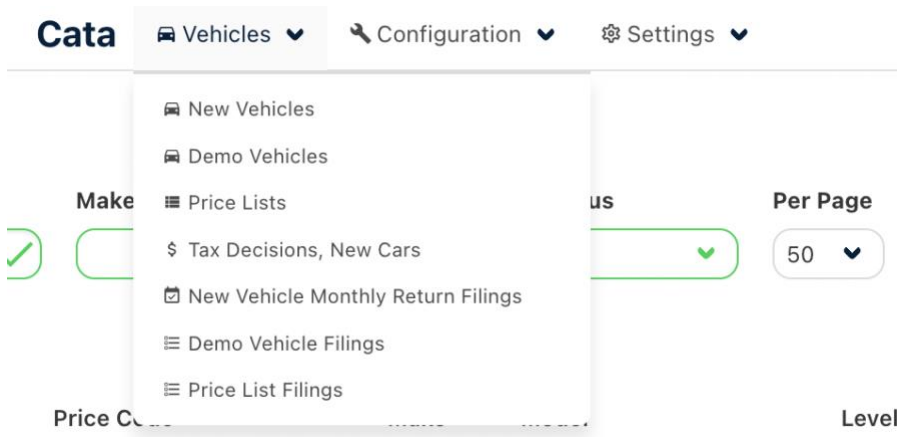


Figure 1: Navigation bar in Vehicle Tax project

Model Reporting (IMAR): This project helps our customer Bassadone to handle, check, modify vehicle models. When the vehicle is ready they can send the information of the tax to Netwheels - Digital solutions for the needs of car dealerships and other automotive operators. Below is the picture of the list of main pages in IMAR

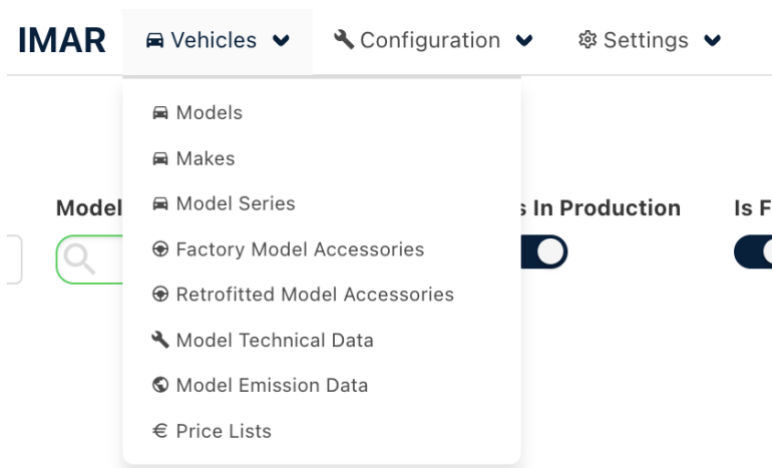


Figure 2: Navigation bar in Model Reporting project

2.1.2 Evaluation and development plan

My colleagues have given me positive feedback on the tasks I've performed. I believe I still need to try more to improve, but as a junior developer, I believe I have done fairly well so far. Most of the tickets are covered in the frontend and testing side, which I was very familiar with now, so I can implement those quickly with few obstacles. The level of my competencies is Novice, which means that I still need some instructions from the co-workers and the team lead. Before joining WeAre Solutions, I had 3 months intensive intern and studying in Integrify Fullstack Academy, so working in pressure while studying is no longer the biggest obstacle. At Haaga-Helia and Integrify, I also gained a lot of relevant programming skills, which makes me stronger with my background. However, I'll need another two to three years to advance to the level of senior or mid-senior developer, as I'll need to expand my understanding of cloud, DevOps, and backend.

- **ReactJs – a JavaScript library for building user interfaces**
This is my strongest skill because most of the time I learn and do frontend is with ReactJs. I'm not working with this framework right now, but our team has plans to transform and upgrade our code base to ReactJs instead of VueJs.
- **Vuejs - The Progressive JavaScript Framework for building user interfaces**
I have started learning Vuejs since I got this position in February. This framework is quite similar with ReactJs – my strongest skill- so there are not many significant difficulties in working with VueJs.
- **Cypress – JavaScript End to End Testing Framework**
I learned Cypress via an Open University course called Fullstack Open and used it to test some of my personal ideas before joining WeAre Solutions as a developer. As a result, there are few challenges in setting up and implementing End-to-End tests in practical applications.
- **ExpressJs - Node.js web application framework**
I learned ExpressJs around 5 months before becoming a developer, and thanks to Integrify, my knowledge of this framework has grown significantly. ExpressJs is used in the backend of two of my largest full-stack personal projects. However, the backend code base in my working applications is extremely complex, and learning and becoming familiar with it takes time.

In the future, I intend to focus more on backend development because I enjoy the challenges in the backend and believe I am more comfortable with logic than UI in the frontend. I also intend to get a Cloud certificate in the near future to complement the backend development plan. Aside from that,

blockchain development is a fascinating and promising topic that I want to learn more about.

2.2 Stakeholders

My personal work can affect the following interest groups:

- My team: Automotive team
- Infrastructure team
- My teammates who work in the same team and projects
- Team lead who organizes and controls the development process of the team
- Project manager who is a bridge between customers and development team
- My developer colleagues who are not directly working with me, but we share experiences and help each other
- My main customer Bassadone Automotive Nordic Oy

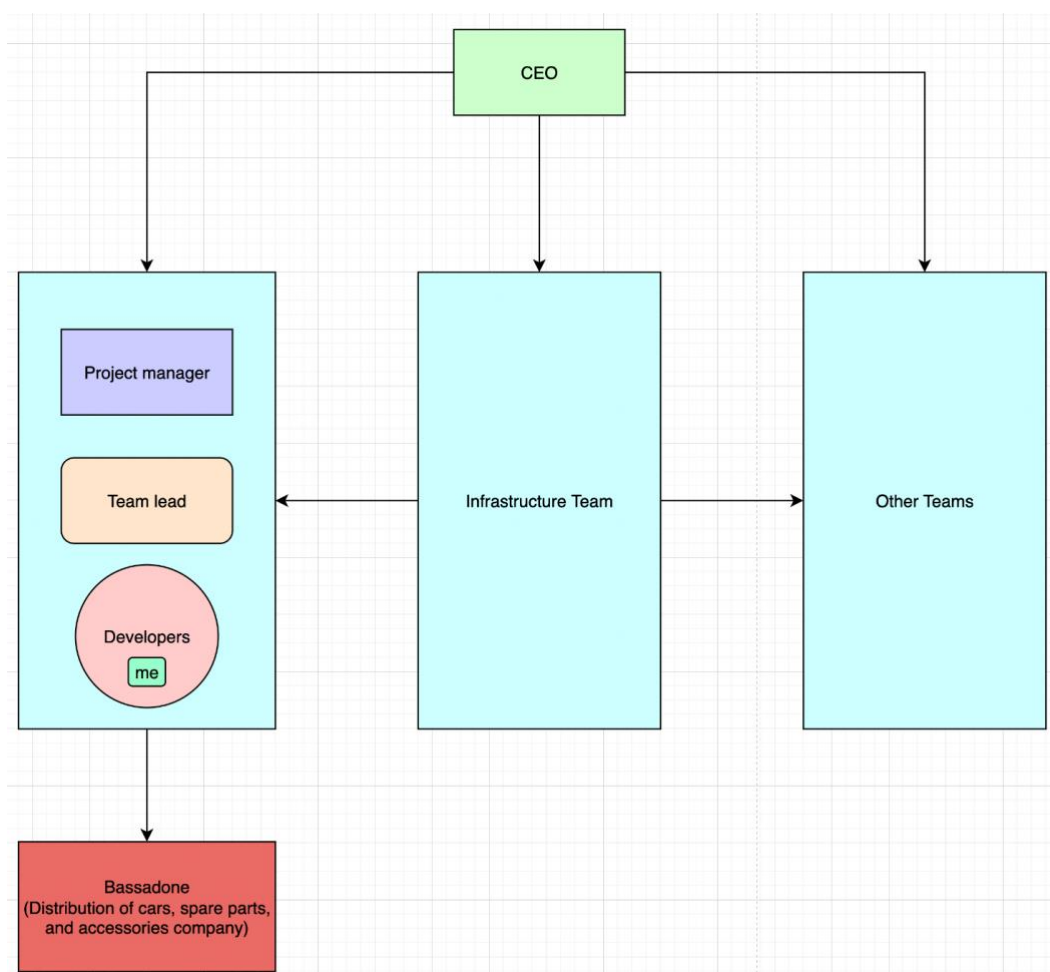


Figure 3: Basic illustration of the stakeholders

2.3 Interaction situations

It was fascinating because one of the most crucial interaction skills at work is the ability to have an online meeting. Because of the current coronavirus pandemic, the majority of my coworkers have not come to work since March. As a result, everyone is familiar with and more at ease with online meetings and calling. Furthermore, I make every effort to visit the office as often as possible since it exposes me to a working environment and mood that is new and exciting for me at the time and helps me concentrate. I try to talk to everyone at lunch in order to better understand and assimilate the people and culture here. Everyone at the company is really kind and willing to talk about anything, including the company's senior authorities, the CEO and CTO, who treat me as if I were a friend at work. There appear to be no gaps, differences, or conflicts between us, which is why I adore, respect, and admire all of my co-workers.

I don't interact with customer service on a regular basis. If an event occurs in the product with which I work, I will usually obtain information from the Project Manager or Team Lead.

Personally, I believe the most evident difficulty I've encountered is the inconvenient nature of online communication. People have greater flexibility in how they start their workdays now that they don't have to go to the office, as long as they complete their task. As a result, it's quite difficult for me to reach people if I need to discuss something because individuals aren't always online at work or respond to messages in time. Especially if you need to demonstrate or explain something to the person, which is always simpler to accomplish in person. Another obstacle in communication at work is my colleagues speak Finnish a lot. Because the majority of my companies are Finns, communicating in Finnish employees is inevitably. No one can criticize it because it is a really natural and comfortable thing to do. However, when they converse and laugh in Finnish, the isolation feeling appears in me. Despite the fact that I am not a person who enjoys learning new languages, I intend to learn Finnish in the future because it will be important when I need to live in the community.

3 Diary entries

3.1 Observation week 1 (March 28 – April 1, 2022)

Monday 28th March 2022

It's nearly 2 months since the day I have started working in WeAre Solutions Oy. Overall, about device I'm using to work: Macbook Pro 13-inch M1, this is the laptop that the company provide to me to improve my performance because my previous laptop is quite old, ineffective, and unsuitable in working in a big project.

As every online working day, after breakfast I started the day by checking emails, messages, and calendar. Today our team had a Developer BI meeting through Google Meet which started at 1pm. To be honest, I'm not a morning person so before the meeting I just checked out the list of tickets and made a to-do list for today and check out comments of my pull requests. Today, I will keep writing End-to-End tests for CaTa project, and do some frontend tasks for model-reporting project.

After lunch, we started our meeting. Today meeting is like every other Developer BI meeting, we focus on going through what we have done in two weeks and next two weeks development priority and requirements. I was assigned a ticket for changing the UI of a table that display the accessories of vehicle models in model-reporting project. I also needed to present about a minor update-package task that the team lead gave me 2 weeks ago because in a software project, we need to keep update and maintain our packages and technologies.

After the meeting I started writing code for the ticket that I was just assigned on Jira. Jira is a piece of software for managing projects and keeping track of issues. Agile development teams frequently use the tool, which was created by the Australian software company Atlassian, to track bugs, stories, epics, and other tasks (Jira 2022). During coding, I needed to ask the team lead to clarify the requirement of the task which I should ask before in the meeting when he assigned me. After that, I needed to created Reset Filter functionality on 4 pages of the application and remove an item from navigation menu.

I finished my first workday of the week with 6 finished tickets. Most of them were tickets that I assigned myself to do, those were not difficult tasks, but it took time to finish

Tuesday 29th March 2022

Today I decided to come to the office in the morning to have a discussion with my team lead because yesterday I expressed that I want to work more in the backend side. At first, I showed and explained to him that I have done yesterday before he gives reviews for my code. Then he introduced and explained about the structure of the backend and how the servers interact with each other. I was assigned a new ticket that is creating login with WeAre launchpad by Microsoft account functionality in CaTa. This functionality was created in IMAR, so my job is read, understand, and apply it into CaTa. The structure is the same with the figure below but with Microsoft account not Google

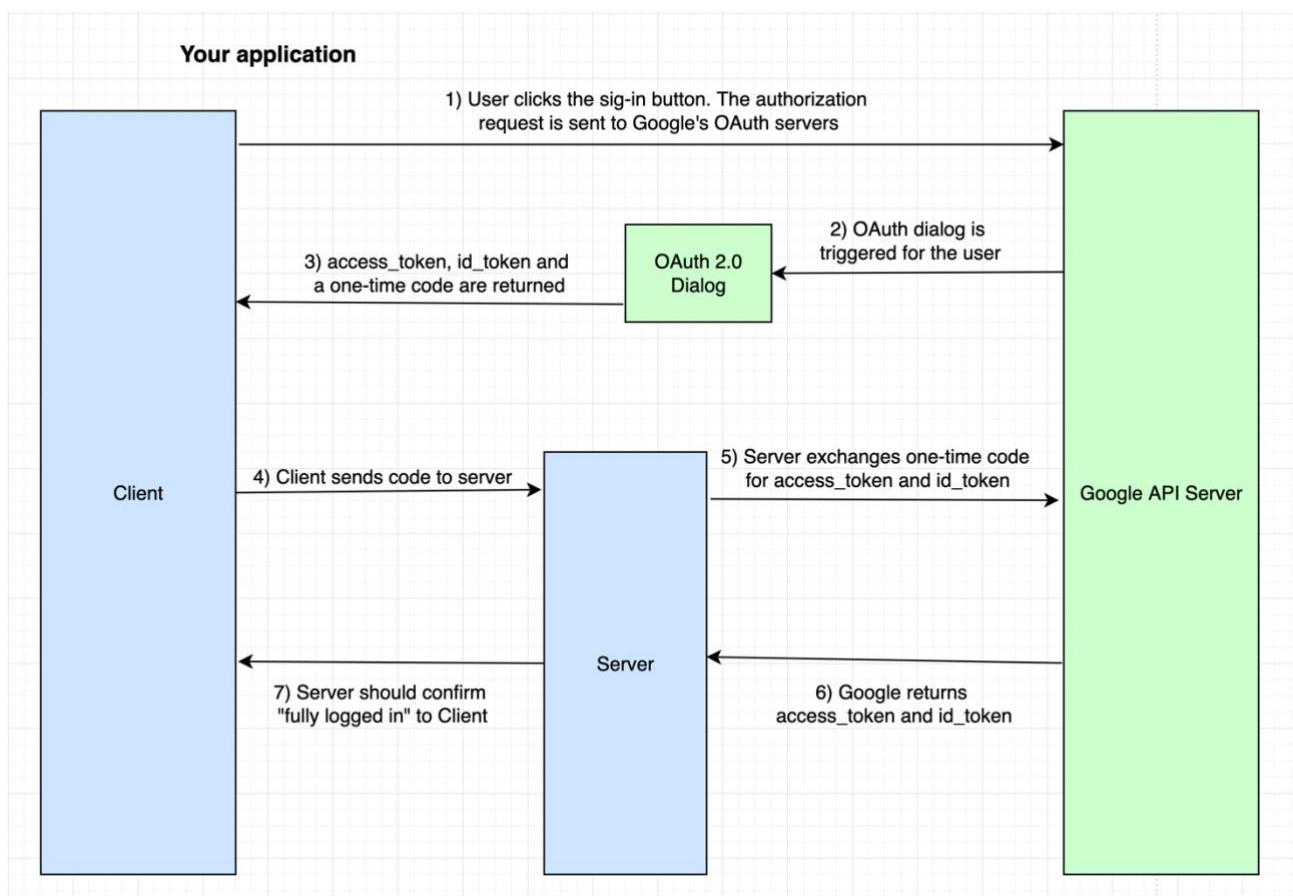


Figure 4: Google sign-in flow for server-side application (adapted version from the original “developers.google.com 2021, *Google Sign-In for server-side apps*”)

After the discussion, I needed to go to Haaga-Helia to attend lectures. When I came back home, I continued my working day online. There was a new frontend ticket in IMAR that required me to modify the Net Price properties to Gross Price because the customer want to pay attention only on the price that include 24% VAT. Then, I modified my End-to-End tests in CaTa that I wrote last week based on the reviews from the team lead yesterday.

Update-packages branch was needed to be rebased by the most updated commits from master branch (main branch of the project). When working in a team, I usually use “git merge” to merge my branch to the master branch. However, the team lead asked me to find out and try to use “git rebase” here. Below is the difference between “git merge” and “git rebase”. Git merge is one of the git merging techniques that preserves the logs of commits on branches. Assume we have a project with three commits on the master branch (commits 1, 2, and 3) and two feature branch commits (commits A and B). If we run git merge, commits A and B will be merged as commit 4 onto the master branch. Git Rebase is similar to git merge, but in this technique, the logs are modified after the merge. Git rebase was created to overcome the limitation of merging, i.e., to make repository history logs appear linear. Assume we have a project with three commits on the master branch (commits 1, 2, and 3) and two feature branch commits (commits A and B). If we use git rebase, commits A and B will be rebased on the master branch as commits 4 and 5, and there will be no logs of the feature branches. (Tanishqua 2022. *Git Rebase vs Git Merge: Which is Better?*)

After all, I started checking out launch pad login in IMAR, I didn't write any code yet because I need to get the whole idea of the functionality.

Wednesday 30th March 2022

I continued trying to implement Single Sign-on with WeAre Launchpad for CaTa. SSO (single sign-on) is an authentication solution that allows users to safely log in to numerous apps and websites with just one set of credentials (onelogin.com, *How Does Single Sign-On Work*). SSO is predicated on the establishment of a trust relationship between a service provider and an identity provider, such as Microsoft as WeAre is using. This trust relationship is frequently established by the exchange of a certificate between the identity supplier and the service provider. This certificate can be used to sign identity information transmitted from the identity provider to the service provider, ensuring that the service provider is receiving it from a reliable source. This identification data is stored in the form of tokens in SSO, which contain identifying information about the user such as an email address or a username.

I want to figure out how to implement it myself, so I will not ask the team lead or other developers till I cannot handle it anymore. The project has a lot of code in different folders and sections that will related and support each other, so I usually approach by reading code in those and try to understand and figure out myself. Therefore, search functionality in VSCode – a source-code editor that I always use - play a very crucial role in my process. After checking most of related files,

I check out the connections between them to understand and try to implement it with the same structure to CaTa.

For example, I can access most of files and codes that related to authentication, login, logout, etc with the key word “authentication”

One disadvantage of this method of learning is that I don't fully get the feature. Because certain files or code may be overlooked due to a search based solely on key words. And because these two applications are nearly identical, copying and pasting without understanding can be unavoidable.

Even though I wrote everything the same as in IMAR, the feature still didn't work in CaTa at the end of the day. So, when I reinstalled Docker after removing all the database containers, I made a really bad decision. Then this error emerged, for which I had no explanation. After an hour of trying to rectify the mistake, I decided to call it a day.

Thursday 31st March 2022

Today I started the day with try to fix the error, but somehow the error was different from yesterday. The error said “Access denied for user ‘tax_app_user’”, so I think the user was not created in the database when I reinstall it in Docker.

Therefore, I opened the MySQL database by a GUI tool for MySQL which is MySQL Workbench (mysqlworkbench.fi) to modify the database and create new user. However, there was a problem here that is MySQL Workbench quit unexpectedly even I reinstall many other versions. I think this because MySQL Workbench is not supported for new version of macOS which is Monterey 12.3 in Mac M1 with Sillicon chip yet.

After hours trying to fix, I decided to not use MySQL Workbench anymore – the GUI tool that I used to manage my SQL data from the beginning. I downloaded and spent time to learn how to use a new tool which is DBeaver (dbeaver.io). The structure and functionalities of the tool are not so different from MySQL Workbench. Then I started setting up the database with DBeaver, but I think I did some wrong steps, so the user error was still there. After a while, I was out of patience, my brain was so stressful, and I did not want to have useless day. Therefore, I switch to IMAR to write some End-to -Endtests. I wrote 4 tests for 4 pages which are Model Technical Data, Model Emission Data, and their edit data pages in IMAR. This did not take much time because these tests are quite similar to the tests that I wrote before in CaTa.

I finished the day by resolving conflicts in my pull requests in IMAR. About git merge conflicts, let's pretend there are two developers: A and B. Both of them attempt to make various changes to the same code file from the remote repository. Developer A updates the file in his local repository and then pushes it back to the remote repository. When Developer B attempts to push that file after making changes on his end, he is unable to do so because the file has already been modified in the remote repository and there is a conflict there. The way I resolve the conflict is that I will merge the master branch – the main branch that I pull from remote repository – to the branch that I want to fix. Then I need to choose which code should be used, the code from my local branch or the code from master branch (from other developers).

Friday 1st April 2022

I started the day with finding way to fix the database error. I was so depressed, so I asked the team lead about the error. After a little discussion, I found what is wrong and set up the database again. I need to allow the user “tax_app_user” to access every port in local environment. Problem solved after 2 days of fixing with many solutions and research. It took time but after this I gained some experiences and knowledges about database and Docker.

I strived to implement SSO with WeAre Launchpad right after fixing the error. However, after hours, I was stuck, and I decided to ask the team lead for hints next week. I continued to write more End-to-End tests for IMAR. The tests were not so different with before tests, so I finished 5 tests.

After work, I also joined an event for Vietnamese developers about DevOps. DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. In the event, I learned the introduction of DevOps and Observability and improve my network by talking and sharing with other experienced developers.

Week 1 evaluation

This is one of the last weeks of IMAR before released for the first time. Basically, there are not so many tasks left, we just wait for the feedback from our customer Bassadone. My main work in this week is fixing and modifying some frontend or UI element in IMAR, writing End-to-End tests for IMAR, implementing SSO with WeAre Launchpad feature to CaTa based on the code in IMAR.

Writing End-to-End tests is not so significant this week because I spent most of last week learning and implementing Cypress in CaTa and the team lead did not require to test so many features yet, but I still tried to write tests for some important ones.

End-to-end testing ensures that the application flow behaves as expected by testing the entire software product from beginning to end. It specifies the product's system dependencies and ensures that all integrated components function properly. End-to-end (E2E) testing is used to test from the perspective of the end user by simulating a real-world scenario and validating the system under test and its components for data integrity and integration.

An E2E test for an ecommerce app, for example, would include actions from the user's perspective. The following are examples of possible steps:

- Logging into the app
- Viewing different product pages
- Selecting an item and placing it in the checkout cart
- Going back to the product page to ensure the right price is displaying
- Placing the order
- Adding certain items to the wish list
- Logging out of the app

Every application is linked to and integrated with a number of external systems and databases. It goes without saying that this complicates the app's workflow. E2E testing checks the accuracy of the operation of an application's various dependencies. Additionally, it determines whether precise information is being transmitted between various system components.

- Backend: E2E testing examines an app's database and backend code. The main functions of the app depend on backend functionality, so this is necessary.
- Multi-tier system: E2E testing is required to verify overall functions as well as the interaction between individual tiers in an application with a complex architecture whose workflow moves through several tiers.
- Distributed Environment: E2E testing is required if an application is built on a SOA (service-oriented architecture) or cloud environment. Additionally, it is essential for apps with multiple parts that must cooperate flawlessly in order for them to function.
- Consistent User Experience: E2E testing, which includes the frontend, also makes sure that the app offers a user experience that is compatible with a variety of devices, platforms, and environments. For instance, cross-browser testing is a crucial component of E2E testing in this regard. (browserstack.com 2021, *End To End Testing: A Detailed Guide*)

E2E testing can be used in a variety of scenarios. Developers and designers, for example, can create a list of the functions, features, and user interface (UI) that are required. They can then follow the flow of data between systems to find inaccuracies, flaws, and dependencies. Teams must also prepare the factors (referred to as conditions) that may change the interface options prior to testing. E2E testing is typically done on completed systems and products, with each review serving as an examination of the final product. You can run a second test if you find a bug or the system's output isn't what you expected. In this scenario, the testing team must track and analyze data to determine the source of the problem, then resolve the issue before re-testing.

The most practical skill that I learn from writing these tests is using the Network tab in Developer tools section of the browser (Microsoft Edge or Google Chrome – which are the ones I use the most). With Network tab, I can find what requests are made when the page is implemented and their orders, the Headers, Payload and Timing of the request. The most significant part is the response of the request where I can know what information or data this request gives to the frontend. There are many tabs and tools in Developer tools of the browser, but Network tab is very special when I can understand how the website runs and built up, how the API called. Therefore, it improved my knowledge about projects and web development. (functionize.com, *Importance of End-to-End Testing 2021*)

This week I also started using a new database client tool for SQL and MySQL which is DBeaver. The form and capabilities of the application are extremely similar to MySQL Workbench, a tool with which I was previously familiar. As a result, I discovered that these database management tools are essentially the same, with similar features and technologies. Any tool like this can be used if the user understands the concept of database. I made a mistake when I spent hours attempting to fix MySQL Workbench, which was not adequate for my CPU at the time, because I was afraid of learning or using something new and messing everything up when I tried things I didn't know for sure. However, after attempting and using DBeaver, I discovered some valuable insights that aren't limited to databases or coding.

It is not a very productive week when I spent so much time doing SSO feature task and fixing errors, some of the errors are very basic and should not be made. However, those problems gave me chance to learn, research the field that I was not familiar with which is Docker and database. There is no deadline for this ticket, but I still really want to force myself learn and implement the feature as soon as possible.

Overall, it was such a challenging week, but in the end, there is no bugs or errors before the weekend. Therefore, I am ready and confident to continue working next week.

3.2 Observation week 2 (April 4 – April 8, 2022)

Monday 4th April 2022

I started the online working day with writing End-to-End tests for IMAR. I wanted to do this first because these tests were not significant and not hard to write but it was different from other that I wrote before, so it also took some time.

After lunch, I got an animadversion from the team lead. “Well, from now on you need to be more thorough. I also wrote to the Jira ticket that it needs database changes. Speed is useless, if the work is not carefully done.” he texted me on Slack. This was the first time I got a criticism with this significant. I apologized and carefully fixed the code and add the database changes as required. After push the code to Bitbucket, I explained for my mistakes. The task required me to change all the properties “netPrice” in the whole application from vehicle accessories and vehicle model colors to “price” because the customer only pay attention about the price that include VAT. I did not understand the requirement of the task, I thought that I will change the properties and then the team lead will change the backend and the database. I just planned to change properties in the frontend at first, but I thought that the team lead will change those in the backend anyway, so I did it for him. Because of that he thought I want to make the whole task. It was still my fault when I did not understand fully the ticket, but this was about communication issues not the negligence. After the discussion, we understood each other more and agreed that if there is anything unclear, we just need to ask.

I continued with testing tasks that I need to modify. I figured out why when I ran unit tests - a way of testing a unit (the smallest piece of code that can be logically isolated in a system) – those were mostly failed. That is because I didn't have “local-unit-test.json” file in the config folder. This file is not pushed to remote repository because we just need these tests in the local development environment.

After all, I tried to fix the error of SSO Launchpad task in CaTa but there is not too much process today. I was stuck for few days. I want to finish this myself but maybe I will need help from the team lead.

Tuesday 5th April 2022

The released date is coming; basically, there aren't many tasks left because we complete most of the application. All we have to do now is wait for feedback from our customer Bassadone. Today my project manager and team lead had a meeting with Bassadone, there are some requirements, but my team lead assigned himself for those tickets.

In IMAR, I began the day by writing some final End-to-End tests. There are some tests that need to be performed, but they are database-related, so I decided to wait until the database is updated to ensure accuracy. While writing tests, I discovered that in order to automatically click the button of the first row in the table to show the detail of the row, I needed to write around 10 lines of code in every test of every page, and most of the pages have tables. It took a long time and resulted in a code that was more complicated, confusing, and unnecessary. As a result, I decided to create my own Cypress Command, which works mostly the same with normal function that we can use to control and execute our tests. Thanks to this, we can now write End-to-End tests in less time and with less effort.

I was still stuck with SSO login in CaTa with status code 500 when I tried to send a request to Bassadone server. I will meet the team lead on Thursday to solve my problem.

Wednesday 6th April 2022

I began the day by asking the team lead for assistance with the SSO launchpad task in CaTa, and he informed me that there was a minor error in the config.json file. For a while, I was perplexed because there is no file named config.json, but then I realized he was asking me to look for an error in one of the json files that configure the application, particularly in relation to integration between the frontend and the backend, or between the application and the identity provider. In the morning, however, I was unable to locate it.

I was heading to the office after lunch at home because I didn't have any lectures at Haaga-Helia today. I was given a ticket on the road that required me to investigate and test the feature of adding colors in a row to vehicle model series. This is the first time I've had a high-priority ticket.

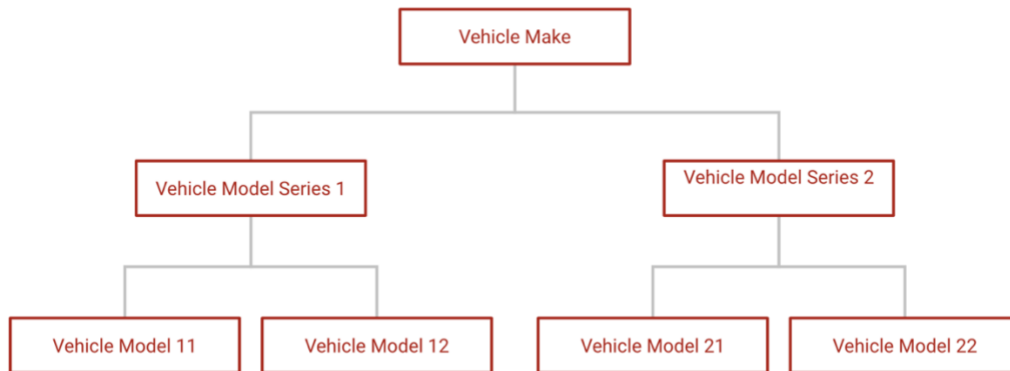


Figure 5: The diagram of relation between main types of data in IMAR which created by my team lead

Before jumping into the ticket, there are 3 main different types of data in IMAR (Figure 5 above). Vehicle Make refers to the brand of the vehicle models, while Vehicle Model Series reflect to groups of vehicle models that have one more many same characteristics. And the Vehicle Model refers to the specific vehicle model. In the ticket, customers have reported that when they try to add multiple colors in a row, this feature only saves the first one, and the colors appear in the model series view but are not sent to the models. As a result of this task, I was able to access the application in its development environment, which is the version that customers can try out and provide feedback on before it is released. After testing with a variety of scenarios, I discovered that there is no error in the application when it comes to adding colors. When adding colors to a vehicle mode series, however, the color will be applied to every single vehicle model in that series, so it will take a few seconds to complete, resulting in a green alert that says "successful". This can lead to a UX issue, which I think the customer got, when new users click add button then they will think it would be added right the way, then change to other pages to check the result. Therefore, there is nothing to fix in here.

However, there is some database errors in my local environment, and I spent a lot of time finding and fixing it because I was still not so familiar with database yet. I created and ran some SQL scripts in a new version of application schema 031.sql in previous ticket (whenever we need to update the PostgreSQL database, we need to create a new sql file in schema folder and set the applicationSchemaVersion in config file to that version which is 31 in this situation)

After I create the file and restart the backend, the scripts in 031.sql run only once locally. However, in another branch, another developer created the 031.sql file and used it to run his SQL scripts locally. As a result, when we merged our code to the main branch, the code did not automatically

execute his scripts as it did in my local environment when I restarted the backend. As a result, the frontend and backend were updated, but the database remained unchanged. It took me a long time to realize what was going on. Then I ran his scripts in PgAdmin, a PostgreSQL management tool, and my database was up to date.

After all, the error in the SSO launchpad task was finally discovered. In the application URL, a small "/" was unnecessary. It's ironic that such a small thing can cause such crash and that it took so long to figure out.

Thursday 7th April 2022

Because I had so many lectures today, I started my working day late. The first thing I did was go over the UI and UX issues that users might have while using the IMAR application; because we didn't have a design team for this project, I had to use the app myself and find the issues. To avoid the UX issue that a customer raised the day before, I created a loading pattern to ensure that the user waits until the process of adding colors to vehicle models is complete. Aside from the customer feedback, I discovered that when a user deletes a color from the table in the vehicle model edit page, the color remains until the user reloads the page. This can lead to confusion, so I decided to improve it.

After that, the team lead informed that there was some error in Cypress test of Resource Update History page. The filter test was failed every time.

The purpose of the test was to ensure that the filter functioned properly. The concept was simple in this case: the total number of rows before filtering must be greater than the total number of rows after filtering. After reviewing the code and running the test, I discovered that the test must wait until the paging request is complete in order to obtain an accurate total number, as the number of rows will be contained in the response as property "rowCount".

Friday 8th April 2022

We did not have any new tickets for today because all the features and problems of IMAR have been completed. I just needed to improve the code of loading pattern that I created yesterday.

I had a meeting with the team head after lunch to talk about the SSO launchpad at CaTa. He demanded that I present what I had done and what had occurred. To be honest, I didn't fully

comprehend the entire process; all I could do was explain or guess what the code accomplished. As a result, the team leaders introduce me to the WeAre launchpad, which was created by the Launchpad team. The launchpad acts as a go-between between the application and Microsoft Azure, the identity provider; it handles all aspects of integration with the identity provider before passing successful or failed login information to the website. From the discussion, I learned a lot about how our application's login system and launchpad work; those concepts are very specific and difficult to find information about on the Internet. As a result, this is extremely valuable to me because I am unable to learn from courses or tutorials.

Finally, because there is no specific task to complete, the team lead told me to update the Vue and Buefy packages in CaTa, just as I had done in IMAR.

Week 2 evaluation

This is one of the final weeks of IMAR before it is released for the first time, just like last week. In general, there aren't many tasks left; all we have to do now is wait for feedback from our customer Bassadone. This week, my main focus is on fixing and modifying some frontend or UI elements, as well as writing End-to-End tests for IMAR. Updating is not a required task, but it is necessary, so I did it whenever I had the opportunity. The best part of this week is that I finally figured out the error in SSO launchpad code that took me a long time to figure out, and my team lead instructed and helped me understand the process behind the login.

The most significant thing that I learn this week is understand how our company Launchpad works. Before I thought the website and the identity provider will works normally like Google Login, but in our company it is not. We have our own Launchpad called WeAre Launchpad - a middleman between the website and the identity provider. It reduces the workload of developers in developing a website that require authentication with Google, Facebook, Microsoft, etc and support SSO when the website only needs to send the request to the launchpad and then receives success/fail order after the process. The WeAre launchpad basic process:

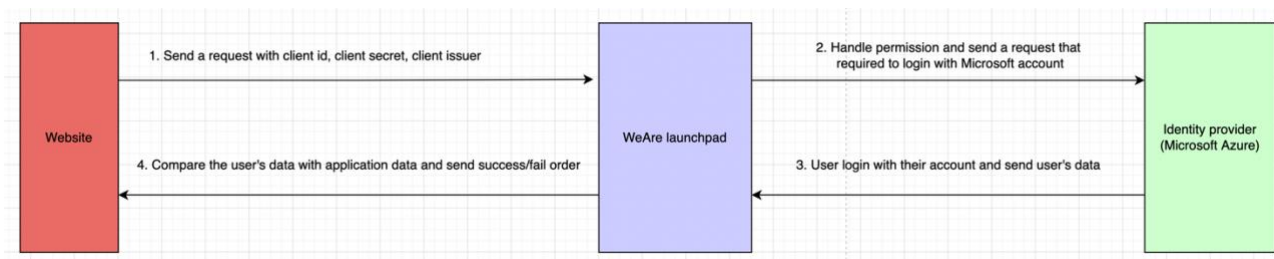


Figure 6: The diagram of WeAre launchpad basic process which created by me

An interesting part of this week is I learned again about using a database management tool – pgAdmin – but this one is for PostgreSQL. I had previously used this technology in the form of an application or a website, but this week I learnt how to use it as a container in Docker - a software platform for developing applications based on containers, which are small, lightweight execution environments that share the operating system kernel but otherwise run in isolation. While containers have long been used in Linux and Unix systems, Docker, an open-source project launched in 2013, made it easier than ever for developers to package their software to "build once and run anywhere". (Scott 2021, *What is Docker? The spark for the container revolution*) I've been writing the code for this container since the first day I started working here, but under the guidance of the team lead. As a result, I didn't know how to use it beyond logging in and viewing the database. I had the opportunity to learn it since I had a conflict in the local database with the application's code as a result of merging my code with that of another developer. As a result, I had to make local changes to the database. This application has features and functionality with MySQL Workbench and DBeaver, two other database administration tools I've used.

While dealing with the database that stored in a Docker container, I also tried to research about Docker to understand deeper into this concept because Docker, in my opinion, is one of the best inventions in technology and development field in 2013. What does the term "build once and run anywhere" above mean? It means that Docker allow developers to package an application with all of its requirements and configurations, such as libraries and other dependencies and deploy it as a single package called *container* - small and light-weight execution environments that run independently from one another while sharing the operating system kernel. Therefore, developers can run this platform-independent application anywhere, any machines with any operating system Linux, macOS, Windows or ARM-based platforms is not so different (for the majority of use cases at least), without mostly any installation for configurations except installing Docker in the machine. Docker containers are portable and have a minimal design. (Scott 2021, *What is Docker? The spark for the container revolution*)

Before Docker, we had to install all the necessary dependencies, such as libraries, configurations, and services like databases or cache, in order to run a project on a different machine. Up until you started working on a single project, it was fine. What happens, however, if you are working on multiple projects at once that require dependencies that cannot be installed and used concurrently? For instance, if one of your projects needs MariaDB and another requires SQL, you'll have to uninstall one to start with the other, which will cause a major mess for you and render your other project useless. Docker is here to come to your aid! It offers a mechanism for managing dependencies that allows each project or application to be isolated, with all of its dependencies contained in a different container. The icing on the cake is the ability to run multiple applications

(containers) concurrently on the same machine. (Santiago 2022, *8 Reasons Why Docker Matter For Devs*)

Docker allows for more granular control and greater portability by isolating applications and their environments, keeping them tidy and minimal. Docker containers make scaling and orchestration easier. Developers can launch numerous containers because they are lightweight, which allows for better service scalability. Because each container contains its own unique set of configurations and dependencies, it is simpler to run multiple instances of the same container at once.

When I spent so much time doing SSO feature tasks and resolving errors, it was not a very productive week; some of the problems were extremely basic and should not have been created. And it also took me quite a lot of time struggling with some very specific concept that I cannot know if the team lead did not explain it to me. Finding the way on my own is very important because it let me use my brains as thorough as possible and requires me to be more patient and more concentrated, but sometimes a little help from the team lead can disembroil the problem and save time. However, those issues provided me with the opportunity to learn and research a sector that I was unfamiliar with, namely the database and project infrastructure.

3.3 Observation week 3 (April 11 – April 15, 2022)

Monday 11th April 2022

I started this week without any specific tickets or tasks, so I simply went through my old pull requests and tried to improve them. Our team had a Developer bi-weekly meeting after lunch. I gave the project manager an update on my work over the previous two weeks at the start of the meeting. We talked about IMAR, a project that will be released next week, on April 18th, 2022. Basically, we've completed all of the features that the customer requested, and we're waiting for them to test it to see if there are any issues or bugs that need to be addressed. New tickets were created in IMAR and CaTa based on some requirements that aren't high on the priority list right now. Adding images to vehicle models and vehicle model series detail pages, as well as creating a new page that lists all anomaly models, are two of IMAR's needs. In CaTa, we must also improve the front end of faulty vehicles. Finally, we talked about writing tests for both the frontend and the backend, rather than just separate tests like we're doing now. However, this type of testing is still on ideas.

After the meeting, I assigned myself 2 new tickets in CaTa because it seemed easier than new tickets in IMAR. The first one is about highlighting as faulty if the vehicles have a registration date

before the price determination date. After researching the application, I figured out that all of the vehicles that have these conflict dates are already marked as faulty. Therefore, the problem here is when the user accidentally modifies the vehicle dates and makes it wrong, the vehicle will not be marked as faulty automatically because we did not build any functionality like that. I decided to solve this issue with a very simple solution, I add a disabled feature to the Save button so the user cannot save the changes if the registration date is before the price determination date.

The next ticket required me to show the VeroAPI error message to the end user in UI, so the user can modify or fix the issue or the data themselves without help from our team. It can reduce the unnecessary workload for us. The first step is I needed to figure out where and how the VeroAPI will send the error information. The application backend already handled the error, showed it in the log, and send it to the frontend with status 500 – Internal Sever Error.

However, there is a problem in handling error in the backend. The information of the error was sent to the front end in the response of the request. But with this situation, the response was just an empty object which is not normal because the error appeared in the backend log.

After some research, I discovered that the express send method cannot send the type Error to the frontend, so I simply needed to access a property called *error.message* instead of *error*, and then the response was sent with the object that converted the error information as I desired. Then all I had to do is to create a notification when the VeroAPI - The Finnish Tax Administration can be contacted in real time using the Vero API integration service. For software developers, it offers simple integration with tax procedures (vero.fi 2022, VeroAPI) - request fails. The problem was that the error content was not very human readable, and the customer might not understand it; additionally, I had no idea what caused the error. Therefore, I'm waiting for the team leader's response.

Tuesday 12th April 2022

I began my working day by attempting to improve the code in the previous ticket, as I was dissatisfied with it. I took a lot of unnecessary steps that made me appear unprofessional. I didn't complete the translation to human-readable error because the team lead appeared to be very busy, and this ticket was not in high demand at the moment.

The project manager and team lead created new tickets in IMAR during the meeting yesterday, so that was the main task for today. The ticket is for the creation of a page that displays all of the

anomalies (issued vehicle model cases) associated with various issue scenarios. As a result, the customer can locate and resolve the model's issue.

We do not have designers for this project, so everything related to UI/UX is all on the developers. At first, I intended to create a dashboard with a variety of cards that represented each case below, similar to how Moodle's cards represent Haaga-Helia courses. However, after some consideration, I've come to the conclusion that this can result in unnecessary efforts and steps for features and pages like Anomalies. As a result, I chose to create a page with a table of issued models and a cases filter, despite the fact that I knew I would learn more if I made it more complex; however, from the customer's perspective, the more complex one would not have a significant impact on UX or convenience.

I wrote the code based on View Vehicle Models page because Anomalies will have the same structure table as this page. This can reduce the workload and make me improve the understandability of the code, features, and structures in the most important page in IMAR which is View Vehicle Models. At the end of the day, I met a problem the paging was not built correctly for Anomalies. For example, if I choose a case in the filter, it only shows the issued models in the first page of the initial list which means if there are 500 models in the database with 5 pages 100 models per page. After filtering, the page only shows the anomalies in the first 100 models.

Wednesday 13th April 2022

Today I continued the IMAR ticket. The first thing I needed to do in solve the problem of paging from yesterday. Before working as a software developer, whenever I met an error or bug, I usually can find the solution on many forums like StackOverflow or Github on the Internet. However, when I started this position, the code, features, and structure are very specific and unique, so there are problems that I cannot find anything similar online. To fix them, I need to be very concentrated and improve my analysis skills to dig deep and find the actual reason that cause the issue.

After hours researching and analyzing the code and other component, I found that this because of the parameter of the request URL – which will contain the information of paging, page size, sorting, filtering, etc. I cannot use the same parameter with the View Vehicle Models page because even though the table still showed the information of models, but this is different situation. The reason why it took me a lot of time to find out because I did not create any parameters before because they are created by the team lead in the initial set up project stage. Another issue with the accessories is that the request I made only said GET maximum 10000 accessories when the actual number was more than 31000. This is due to the team lead's function being limited. I knew

the request needed the parameter to handle more than 10000 items after asking the team lead about it. As a result, I had to create the parameters for these problems myself.

After that, I created Make and resolved – a feature that shows which models are fixed and which are not – filters, and I optimized the code. When attempting to implement those features, I encountered some bugs, which I quickly identified and fixed. I finished my ticket and created a pull request for code review at the end of the day.

Thursday 14th April 2022

I started the working day later in the afternoon because I had some in-class lectures at the Haaga-Helia campus today. To be honest, neither the Backlog nor the Selected for development sections have any tickets left. So, I went through all of my pull requests and tried to make the code more efficient. I received a number of pull requests, but I need the team lead to look over the code and show me how it should be done. But it took him some time to accomplish this.

Today the team lead told me that I should create too many meaningless commits because whenever I changed something even very small like a line of code, to update the code I need to create a new commit to push the new version of code to remote repository. Therefore, the team lead recommended me a new technique that he just learned two weeks ago in Git, which is using `git commit –amend`. The `git commit —amend` command makes it easy to make changes to the most recent commit. Instead of creating a new commit, it allows you to combine staged changes with the previous one. It can also be used to make minor changes to a previous commit message without affecting the snapshot. However, amending does not simply change the most recent commit; it completely replaces it, making the amended commit a new entity with its own ref. This technique is very useful because now I can reduce unnecessary commits that can lead to distraction or confusion in the pipeline.

Friday 15th April 2022

Good Friday holiday

Week 3 evaluation

IMAR is in its final week before being released for the first time. In general, there are no tasks that are requested; however, our customer Bassadone reported that some CaTa features need to be fixed and improved, and they want an Anomalies page that contains issued IMAR models. I assigned myself all of the new tickets because I believe I am capable of doing so and I want to make a significant contribution, as opposed to taking on small tasks previously.

The most significant thing that I learned this week is how to use parameter to send the request to the backend to control the response. This concept was not new for me because there are lectures in Programming 2 – a course at Haaga-Helia – which taught and required me to use parameter. Parameters are the portion of a URL that follows a question mark, also known as query strings or URL variables. Refer to the portion of the URL after the question mark (?) to identify a URL parameter. A key and a value are separated by an equal sign (=) in URL parameters. The ampersand is used to separate multiple parameters (&). Using an ampersand, you can add multiple parameters to a single page.

These query strings allow users to filter a page's content and view only a certain number of items per page. Tracking parameter query strings are also common. They're frequently used by digital marketers to track where traffic comes from so, they can see if their latest social media, ad campaign, or newsletter investment was a success. (*An SEO Guide to URL Parameter Handling 2019*). Example of an URL that have parameters:

https://www.example.com/widgets?sort=newest&color=blue. While “?” is the start of parameters, “color” is the key, “blue” is the value, and “&” is the separator

The most common use cases for parameters are: (botify.com, *What are URL parameters?*)

- Sorting and Filtering: parameters are often used on large ecommerce sites (or similar) to allow users to dynamically generate a page with their desired sorting or filtering applied
 - e.g. */dresses?sort=a-z*, */womens-shoes?color=red*, or */hotels/seattle-wa?rating=5-star*
- Pagination: parameters can be used for identifying multiple pages of archive or search results
 - e.g. */blog/all-articles?page=3*
- Site search: parameters can be used for passing through the search queries that someone used in a site search
 - */search?q=christmas*
- Translation: parameters can be used for language options
 - */home?lang=fr*
- Describing: parameters can be used to pass through details of a product

- /product?sku=12345
- Tracking: parameters can be used for specific advertising campaigns or button clicks to be able to track traffic that came via that campaign or button
 - e.g. /landingpage?utm_campaign=fbid_holidaypromo

When I created a request, I only needed to write some keys and values in the URL, which was a simple exercise in the lecture. In the real project, however, the parameters for each important request were created in the initial setup before I joined, which was different. As a result, I had to figure out how to create a parameter with the appropriate type for the GET request function, which was also created during the initial setup.

Besides that, I also learned how to create a PostgreSQL database as the requirement of the final project in AWS - Amazon subsidiary that provides individuals, businesses, and governments with on-demand cloud computing platforms and APIs – in Server Programming course at Haaga-Helia. I used Amazon RDS as the service - a managed SQL database service provided by Amazon Web Services (AWS). To store and organize data, Amazon RDS supports a variety of database engines. Data migration, backup, recovery, and patching are just a few of the relational database management tasks it can help with. Amazon RDS makes it easier to set up and maintain relational databases in the cloud. To set up, operate, manage, and scale a relational instance of a cloud database, a cloud administrator uses Amazon RDS. Amazon RDS is a service for managing relational databases, not a database in and of itself. (*Amazon RDS (Relational Database Service) 2021*). It is not hard to create a 20GB database in AWS for almost free (1€ fee to create account), but this is one of my first steps in this field.

3.4 Observation week 4 (April 18 – April 22, 2022)

Monday 18th April 2022

Easter Holiday

Tuesday 19th April 2022

There was no new ticket today because there wasn't much work to be done in IMAR and CaTa since IMAR had just been released and CaTa was working well with our customer Bassadone. However, because the tests needed to be consistent, the team lead required that I improve the End-to-End tests in CaTa. In the login test, there was also an error because the user already had a

session, and the clear session function did not always work for some reason. But, in the end, the login test worked perfectly in the local environment, and the team lead and I had no idea why it didn't work in the pipeline, despite the fact that the code was similar.

In addition, the team lead made comments in my old CaTa ticket, necessitating the addition of a disabled feature to the Save button, preventing the user from saving changes if the registration date is prior to the price determination date (more information in Monday 11th April 2022). When merging master, the team lead might remove the disabled button feature, so I had to add it again. When he needs to read and choose which code is correct when resolving the conflict when merging, this error is sometimes unavoidable. Furthermore, he requested that I add a tooltip to the Save button to inform users of the error and how to correct it but adding a tooltip to a disabled button in Buefy was difficult, so I added a small warning line above the button to announce users about the error. The last thing is that I needed to mark the actual fields as incorrect when the dates are conflict.

Wednesday 20th April 2022

My working day started with the technical meeting about Smartfleet – a product company which is my company's subcompany. Smartfleet is a Finnish vehicle management company (website: <https://www.smartfleet.fi/>) which manages business's fleet and minimizes costs when customers buy and exchange cars. The main features of the application that Smartfleet is building is car management system related to fleet management and invoices and total costs cumulatively which handle all the car invoices of the company and then send one checked consolidated invoice to the customer.

The meeting was all about development side of the company. The tech stack was the same as Bassadone projects because it was created by the team lead with VueJs in the frontend, NodeJs in the backend and PostgreSQL in the database. The team lead introduced the application for us and assign us a very urgent task about debugging. There are some errors in the balance invoice feature – a feature that will calculate and balance the invoices when the customer paid more or less than the real price.

After the meeting, the first thing I needed to do is setting up the Smartfleet local environment. Like setting up CaTa or IMAR, I needed to clone the code from Bitbucket and then install and build the application from the code. The next thing will be setting up the database, the team lead sent us 44 SQL scripts file for 44 tables of the database. However, there was a problem when the SQL scripts that the team lead sent us have some conflict with the SQL scripts that already in the code. The

tables did not match with the data in the scripts and the script also create some tables again when they were already existed.

Thursday 21st April 2022

Like every Thursday in the week, today I started the working day quite late because of lectures at Haaga-Helia. The first thing I need to do was importing the database. Fortunately, the team lead understood our situation – the conflict in SQL scripts as yesterday above - when importing data, therefore he splitted the scripts into 2 parts: creating tables and inserting the data. After creating tables, I started inserting data by SQL scripts but there was an error:

ERROR: insert or update on table "customer" violates foreign key constraint "customer_parent_id_fkey"

DETAIL: Key (parent_id)=(d9aa0374-e4ec-4bf7-87ba-4943662c2358) is not present in table "customer".

SQL state: 23503

Thanks to Database Management course at Haaga-Helia, it did not take me too much time to figure out what the error about. This is relation database basics; table customer has relation (column parent_id) to itself. You are trying to insert a row to customer, with parent_id 'd9aa0374-e4ec-4bf7-87ba-4943662c2358' but there is no row with that id in the table. Disabling all the triggers in the database can help me remove the restriction. However, there were 4 tables in total 44 tables that had error lines, so I needed to delete them one by one which took a lot of time. In the end of the day, after setting up almost everything in the database, I started the application, but the admin account could not access because the hash password - a scrambled representation of the password – was not correct in the database. Therefore, I needed to run a function to create another hash password from the actual password then use it in the database. Finally, the application was ready, it took less time than other two projects CaTa and IMAR because this time I understood what I need to do and the application - pgAdmin4 - that I used to setup the database was much faster and more completed than MySQL Workbench in some cases.

Friday 22nd April 2022

Today I decided to come to the office because there was no lecture at Haaga-Helia and the team lead will be there too, so it is convenient for me to ask about Smartfleet – my new project. After setting up the project yesterday, I had some discussions with other developers in the team and it

seemed that they're all using IDE and applications from JetBrains – a platform that provides many essential tools for developers in different fields. All the tools from JetBrains are in the same “ecosystem”, therefore; they connect and support each other which is very convenient. Therefore, the first thing that I did today was to request the licenses for 2 tools from JetBrains which is Webstorm (website: <https://www.jetbrains.com/webstorm/>) – the smartest JavaScript IDE, and Datagrip (website: <https://www.jetbrains.com/datagrip/>) – many databases, one tool.

I was a little bit bewildered in the beginning because, since the first day I coded, I had used VSCode as the IDE and fell in love with it. However, there are reasons that almost developers in my company use Webstorm, and luckily it took me just a while to set up and get familiar with it and I was impressed by their number of features even though the UI was not as good as VSCode.

After that, I and the team lead had a discussion about the bug that we meet in Smartfleet, so I had a clearer about the issue and the application. Now I can start thinking about the solution even though I know I will not be the person who will solve the problem, but I might learn something and have some ideas. I kept researching and thinking about the problem till the end of the day.

Week 4 evaluation

The first day of the week (Monday 18th April) was the released day of the first version of IMAR. Therefore, there was no new tickets from this project, CaTa is still working well with our customer Bassadone. The improvements and pull requests in CaTa that I wrote were not in high priority, so the team lead did not review them yet. The most important thing in this week is that I joined a new project from our customer/child company which called Smartfleet. More information about the company is placed in the daily evaluation section *Wednesday 20th April 2022* above.

Because of those improvement tasks in CaTa, I was able to learn Veevalidate – client-side form validation for VueJs this week. Before sending data to the server, it is important to ensure all required form controls are filled out, in the correct format. This is known as client-side form validation, and it ensures that the data submitted meets the objectives in the various form controls. Client-side validation is an initial check and an important feature of a good user experience; catching invalid data on the client-side allows the user to correct it right away. A round trip to the server and then back to the client-side to inform the user causes a noticeable delay if it gets to the server and is then rejected. Every form in the projects uses Veevalidate for validation. My mission was quite advanced when I need to create a custom rule for the validation (there are a few standards and common rules that Veevalidate already created for users). It required me to learn

how Veevalidate works and dive deep into Rules, it took me a while implementing a new rule that just has a small impact on the application, but it was worth what I have learned.

Client-side input validation is handled by front-end code, while server-side input validation is handled by back-end code (Figure 7). Because back-end code is the most closely related to the database, it acts as a bridge between the front-end code (which the user sees) and the database, which stores the user data. The model/entity class performs server-side validation by attaching attributes above the property declaration to specify the data types that the property can accept. Only refined data is acceptable if the specifications include a large number of parameters. Client-side input validation is faster than server-side input validation. Input validation on the server, on the other hand, is more reliable than input validation on the client. Client-side data validation, on the other hand, improves user experience while server-side input validation improves security. (securecoding.com 2021, *Input Validation: Client-Side & Server-Side Cybersecurity Deterrent*)

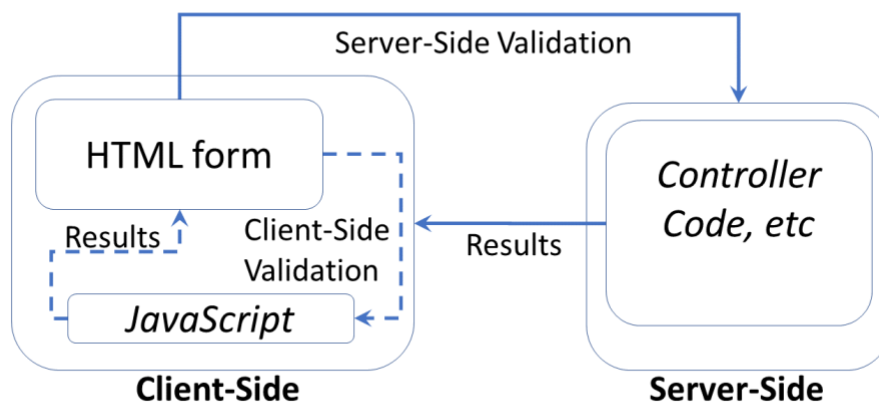


Figure 7: Diagram of validation in an application (wakeupandcode.com 2019, *Validation in ASP .NET Core*)

The most significant thing that I learned this week was setting up projects. The reason is that I joined the Smartfleet team this week and setting up the project in local environment is the first thing when joining or starting a project. This was the third time that I need to set up a project in local environment (the two ones before are CaTa and IMAR), but each project has different tech stack with different sizes of the application and database. Therefore, every time I do it, I learn a lot in many different fields in programming. However, whenever I setup a project, I need to approach many brand-new things and knowledge, because like I said above, projects are different. So, it took me a lot of time to getting familiar and learning about the project. The new project Smartfleet use only PostgreSQL as database, therefore I used PgAdmin4 as database management tool to setup and import data because this was the new tool that I learned which is specified for

PostgreSQL. However, after finish setting up, I received DataGrip license from the company, so I instantly switched to DataGrip as database management tool even though I just used PgAdmin because intelligent features and supported engines like MySQL, PostgreSQL, Microsoft SQL Server, Oracle, and more.

DataGrip is basically the same with other database management tools, but it has many more fancy and useful features or functions that helps developers in many different scenarios. DataGrip provides intelligent query console that allows you to run queries in various modes and provides a local history that records all of your activity and protects you from losing your work. DataGrip supports context-sensitive code completion, which allows you to write SQL code more quickly. Completion is aware of the structure of the tables, foreign keys, and database objects created in the code you're editing. DataGrip detects potential bugs in your code and recommends the best ways to fix them on the fly. It will immediately notify you of unresolved objects by using keywords as identifiers and always offers a solution. There is a feature called Explain plan which gives you a deeper understanding of how your queries work and the database engine's behavior, allowing you to optimize your queries. The best thing is that it has the same ecosystem with Webstorm – the IDE that I am using for editing and writing code. This makes the work that requires using an IDE and a database management tool at the same time is way more convenient.

(jetbrains.com/datagrip, *Why DataGrip*)

3.5 Observation week 5 (April 25 – April 29, 2022)

Monday 25th April 2022

My team and I began the working week with a Developers BI-Weekly meeting. We began by going over what we had accomplished in the previous two weeks, as we do at every meeting. Bassadone appeared to be very pleased with their newly released application IMAR, which included model reporting as a key feature. I showed the team leads my IMAR Anomalies page, and he said he'd review it and give me feedback after he finished fixing the Smartfleet budget issue with balance invoices. Smartfleet has some new tickets, and I "raised my hand" to get one.

I was assigned a new ticket in IMAR which create a feature that gives users the ability to upload documents with PDF type like certificates, contracts, offers, etc, and PNG types like images on 3 different pages customer, vendor contract, and vehicle contract. Before starting, I needed the team leads for more details about the task because I was outside for apartment stuff, so I did not fully pay attention to the whole meeting. This was all my fault when I forgot about the meeting and did some-thing else, but the team lead sympathized and understand me.

Although this is the first time, I have written code in Smartfleet, the structure and codebase are very similar to IMAR and CaTa, as they were all set up by the team lead. As a result, I was able to get started quickly and create the components. At first, I thought I would have to create and use a new library for this upload feature, but after some research, I discovered that Buefy also supports this feature, so everything went smoothly. At the end of the day, I planned, wrote, and divided the ticket into small tasks to keeping it always on track and logical.

Tuesday 26th April 2022

Today I continued implementing add document functionality in Smartfleet. At first, I wrote some code to create a field for category. The special thing of this field is that users can choose default option in the dropdown list, or they can create a new category for the document. This type of field is not that difficult for me because I used to create one in IMAR, so I was very confident. But the problem in here was that the application needs some default categories in the field. I can easily create default categories myself by add it by hands, but they were only added in the local database. In test environment or production environment, the category field was totally empty and that did not meet the requirement of the ticket. Then I had an idea that I need to create a new SQL file in schema – like I did on Wednesday 6th 2022 - that will automatically run when the backend run; therefore, the category field will have those default options whenever the application run.

After that, I was planning to create a new table in the database to save the document, but the thing that I need to save is the whole file, not a string or number like normal. After researching for a while, I was not sure about this because it would be very complicated. However, after asking the team lead for some hints, I knew that I need to save the file to cloud which is AWS S3 and save the file details in a table which is already created before called file_meta.

Wednesday 27th April 2022

Today, I continued working on the vehicle contracts add documents function. Thanks to the assistance of the team lead, I was able to determine the proper method for implementing this functionality. The file information, such as name, description, category, and resource type, will be saved in the PostgreSQL database alongside other data, but the actual file will be saved in AWS S3, an object storage service with industry-leading scalability (mordendatastack.com, AWS S3). The issue was that I had never written anything remotely similar before. However, after spending some time researching the project and the code, I discovered that the team leader had already

created a function for uploading files to the S3 service; now it's up to me to figure out how to use it correctly. Fortunately, there are some components that already execute the function with different concepts.

Another task for today was to test the team lead's solution for the balance invoice issue that was introduced to us last week in order to find a solution, but I didn't really contribute to the process because this issue was out of my range at the time. As a result, I wanted to implement this testing solution task carefully in order to get the correct result as soon as possible. However, because of my local database, my results kept getting wrong, and this took a long time.

At the end of the day, I finished the uploading files feature for Smartfleet but I did not have an AWS account yet, so I could not access to the S3 service to check if my function actually worked or not. It was not a very productive day when I cannot get the accurate result for a testing solution and did not really finish the uploading task.

Thursday 28th April 2022

In the morning, I received an email of activate an AWS account from my team lead, therefore I can access our project cloud storage on AWS. However, I had some troubles with changing to a new password, so I still could not access AWS by company account even the team lead had no idea why that was.

There is a new solution testing task for today, but I did not implement the yesterday task correctly, so I asked for help from another developer. He was super nice when send me his code and also explain it very carefully. After all, I collected an accurate result and confirmed that the solution was precise and thorough.

Because my AWS account was inaccessible, I created a new ticket in IMAR. This ticket is about fixing the UI in a Hyundai accessory, but I was unable to work on it because my local database was too old, and the team lead was unavailable for the remainder of the day. There is no other option but to wait for the team leader to assist me in solving my problem tomorrow.

Friday 29th April 2022

I started the day by trying to fix the changing password issue in AWS service. This time I used the auto password generator of 1Password - password manager, digital vault, form filler, and secure digital wallet – which is an essential application in my company because we store many accounts for many services there. However, it did not solve the problem, so the team lead recommended I

change the setting of the auto generator from 24 characters to 40 characters finally it worked, and I could access AWS S3 now.

After that, I tried to test my feature by adding a document to S3, and the thing that I was afraid of came. The feature worked in adding document information to the database but not adding the file to the S3 service. After a while of debugging, I figured out that it was because the node environment in my local environment is “dev” which should be “local-dev”. The node environment needs to be “local-dev” to get the access key and secret key for AWS S3

And finally, I could add files to S3, the next thing will be making a list of file information and giving the user ability to download the file from S3. There is a page that already implements the listing and downloads feature; therefore, it was not difficult for me to create those for this situation.

At the end of the day, I figured out that the adding document form and the documents list can be created as a generic component – a component that can be reusable and I was creating those specifically for different page customer, vendor contract, and vehicle contract. Therefore, I tried to optimize the code by reducing them from 3 separate components to only one generic component.

Week 5 evaluation

This week, I started to dive in and took my first ticket in Smartfleet. The tech stack was the same as Bassadone projects because it was created by the team lead with VueJs in the frontend, NodeJs in the backend, and PostgreSQL in the database. Therefore, getting familiar with the project was not challenging at all.

The best part of this week was that I had the chance to interact more with the backend, the database, and even our cloud service AWS S3. I feel like I am going to be a more full-stack developer rather than a frontend developer. That is a very good sign of self-development when I can see myself changing and improving in weeks. In my opinion, this is because I work in a small team in a small company; therefore, I can have more chances to access and learn other fields of the industry rather than my specialized skills.

Before joining Smartfleet, I mostly did not touch anything in the database. But now, after spending so much time setting up, debugging, and testing the database, I am so much more confident and understand deeper. I can investigate and predict an issue more precisely and sometimes know

how to solve it. The wasting time decreased, and the productivity went up. This is an experience that is not easy to gain when studying.

The most significant thing that I learn this week is interacting with AWS S3 service. Amazon S3 is an object storage service with industry-leading scalability, data availability, security, and performance. Amazon S3 allows customers of all sizes and industries to store and protect any amount of data for a variety of use cases, including data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. AWS S3 ("Simple Storage Service") allows customers to store and retrieve any quantity of data at any time and from any location, providing developers with highly scalable, reliable, rapid, and low-cost data storage. AWS S3 provides easy management options to organize data for websites, mobile applications, backup and restore, and many other applications. It is designed for 99.999999999 percent durability. You can use Amazon S3's management features to optimize, organize, and configure data access to meet your specific business, organizational, and compliance needs. The figure 8 below will show the basic of S3 service process. (mordendatastack.com, AWS S3)

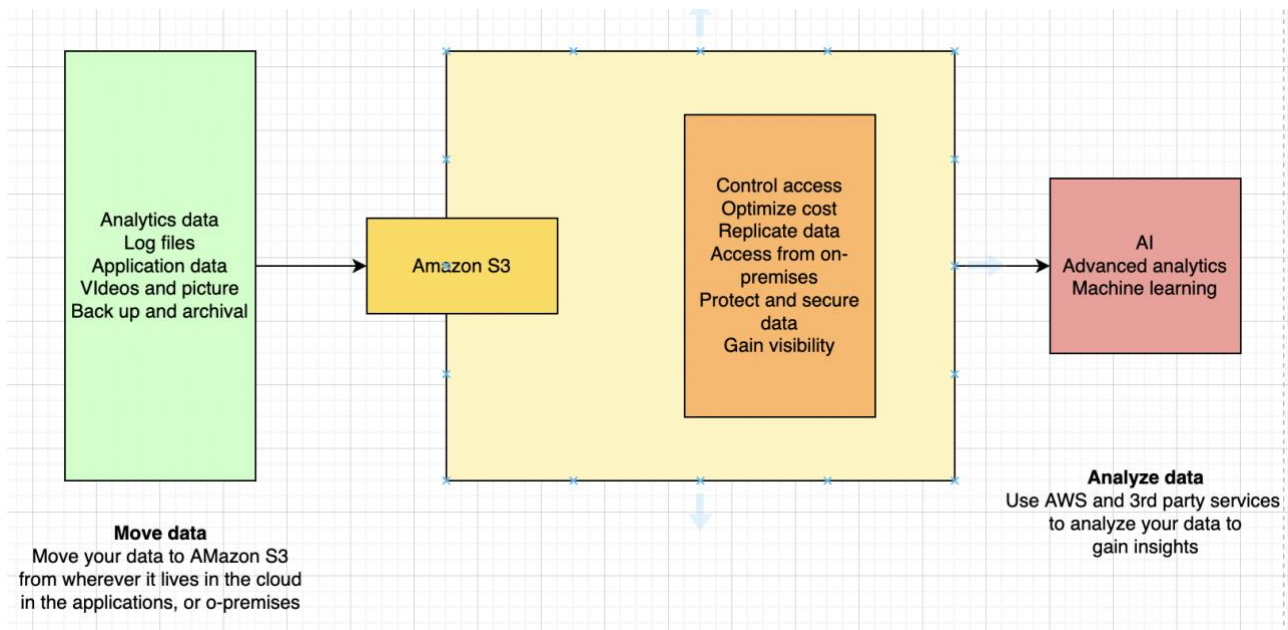


Figure 8: The basic of S3 services process (An adapted version of diagram from the original "aws.amazon.com, Amazon S3")

Users manage the service by buckets, containers for objects stored in Amazon S3. You can store any number of objects in a bucket and can have up to 100 buckets in your account.

Netflix, Dropbox, and Reddit are just a few of the companies that use S3. Dropbox, a popular file storage service, bases its full storage capacity on Amazon S3. S3 is a budget option. In

comparison to other storage options, it's dirt cheap. User only pay for what they use with S3. S3 also has a free tier in addition to its low prices. This plan includes 5GB of storage, 20,000 GET Requests, 2,000 PUT, COPY, POST, or LIST Requests, and 15GB of data transfer. For the first year, the free tier is offered every month. User can avoid paying for space or bandwidth that they don't need with S3. S3 adapts to the needs of your application. There is no limit to the amount of data user can keep with S3 because user only pay for what you use. Users are not required to purchase additional storage space. S3's security proclivity is one of the many reasons why businesses prefer it. S3 is secure by default, unlike custom server setups. This isn't to say user can't use S3 to store information that's open to the public. Unless you specify otherwise, S3 encrypts all of the data to a high level of security. Maintaining multiple copies of a file and tracking changes over time is known as versioning. This is especially useful when working with sensitive data. When users use S3 and enable versioning, they can also recover files that were accidentally deleted. When versioning is enabled, however, multiple copies of the same document are stored. Pricing and read/write requests may be affected as a result of this. When integrating versioning for the application, user should keep this in mind. S3 versioning is disabled by default, but it can be enabled through the AWS Console. S3 also offers 99.999999999% durability (called the 9s durability) and 99.99% availability of objects over a given year.

This week I also learned about the node environment of the application. There are several environments like "unit-test" is for testing, "local-dev" is for developers to write code locally, and "dev" is for user testing and "prod" is for the production version. They are just the names of environments in a project. These environments were setup based on the requirement and the process of the project by the team lead or senior developers who started the project. The difference between these environments is which config file the application tries to use. For example, the "local-dev" environment will use those files below for local config, but the "dev" environment uses other files that I was not allowed to know because it contains some sensitive information about the project like private key, database key, passwords. I will have the chance to approach the other environment like "dev" or "prod", if my technical skills are enough to handle them. Because those are the environment that our user is testing and working on, so it will be a huge responsibility when writing or editing code in those environments.

For example, this piece of code below is working only for the environment "local-dev" or "unit-test" – the 2 environments that I allowed to access. If the environment is not these 2 ones, the application will not run the code inside the "if". Therefore, the code of the application that our customer is using will be a little bit different from the code that I use locally. However, those differences are all about the settings or config of the application, so junior developer like me still actually contribute our work to the project.

```
if (process.env.NODE_ENV === 'local-dev' || process.env.NODE_ENV === 'unit-test') {  
  s3options.credentials = {  
    accessKeyId: getSecret( key: 'S3.awsAccessKeyId'),  
    secretAccessKey: getSecret( key: 'S3.awsSecretAccessKey'),  
  };  
}
```

Metsälä, 20.10.2021, 10.22 • SMT-231 - Altered invoice uploadig credentials

Figure 9: Screenshot of the code that require "local-dev" or "unit-test" environment to run

3.6 Observation week 6 (May 2 – May 6, 2022)

Monday 2nd May 2022

I started the week by continuing the ticket about add documents in Smartfleet. In the morning, I received some comments from the team lead in the repository of this ticket. Those comments were about some improvements in the code and translation need to be fix. The application is in two languages Finnish and English, so beside writing code for features, I also needed to write some translation from English to Finnish. And I'm not an expert in Finnish, so the modification from the team lead is needed but it did not take a lot of time.

The most significant thing that I need to improve is to add the mandatory UI for Category field. But things did not go easily as I thought. I tried to find how to make the Category field mandatory or required, but after doing that I realized that the actual job is creating red outline whenever it is empty to remind or trigger user to fill it. Then I strived to use Veevalidate to make the validation of requirement for the field, but the Vue Multiselect field that I used was not similar to other fields. Therefore, I spent so much time on something did not work. After all, I figured out that I used to write some CSS code for this UI before in IMAR before, and it was so wasted of time when I tried to do it in another way because I forgot I have used to done it before.

Today I did not perform well, because I had some sleeping problem at night. Therefore, I was very sleepy and sober the whole day.

Tuesday 3rd May 2022

Again, today I continued my work on Smartfleet ticket – implement add documents feature. After a little discussion with the team lead, I decided to improve the upload file feature from "Click button to upload" to "Drag and drop the file" and users can drop many files at once. This will be fancier in the

UI and convenient in the usages when users can upload multiple files at the same time with the same category and description. Improving this mean that I need to restructure the whole AddDocument component to meet the requirement. It was not difficult because I wrote the component myself, but it took time because this is one of the most complicated components, I ever wrote with nearly 400 lines of code.

The most significant task was letting the user to change the name of files that they want to upload, because of “strict-mode” of Typescript, I could not directly change the name of the file object. It took me a lot of time researching how to change the name of uploaded file and I only found the answer in Java. Fortunately, the logic was the same is creating a new file and assign the original file to it and set the new name.

At the end of the day, I created a new component and page for editing document in Smartfleet. I was quite late, so I basically finished the day after creating the UI for the page.

Wednesday 4th May 2022

I didn't have any classes at Haaga-Helia today, so I went to the office to be more productive, as the working environment there always makes me more focused. Because the team lead used the user ID and the file name to identify the document that was pushed to S3, there are some duplication errors in S3 storage. It wasn't unique because one user could add a new file with the same name as existing files in the storage, and the S3 service would update the existing file rather than creating a new one. This can lead to some serious errors in document management in the application.

My mission is to innovate new ways of identifying files. I chose to save the file information to the file meta table in the database using the ID, which is unique because it is generated by UUID. A UUID (Universal Unique Identifier) is a 128-bit value used to uniquely identify an object or entity on the internet. By using UUIDs, you ensure that your ID is truly unique in the universe, not just in the context of a single database table or web application. There should be no other ID in existence that is identical to yours. (Lane 2021, *What are UUIDs, and should you use them*). It took me some time to find the way how to pass the ID to the backend, and from there it is combined with the file name and send the unique identification with the file to S3 storage. In the end of the day, I finished the EditDocument component and page.

Thursday 5th May 2022

The last feature I need to implement in this ticket is previewing images in the list of the documents. After a while of researching, I decided to use a presigned URL - a URL that you can provide to your users to grant temporary access to a specific S3 object. Therefore, user can check out the documents without downloading. To implement presigned URL, I needed to use a new library from AWS S3 service which is '@aws-sdk/s3-request-presigner'. This was also the first time I proactively use a new library in the project without the suggestion of the team lead. Once again, I felt a significant self-development when remembering the first days when starting the position. It reminded me that my efforts are not wasted, and I need to keep trying to improve myself. I was struggling in creating UI for the preview because I was planning to create a popup modal window for the preview, but I keep meeting error therefore I decided the preview will be in a new tab in the browser. At the end of the day, I needed to update the cypress test package in the Update branch, but there is something wrong with the database of the cypress test, so I decided to fix this tomorrow.

Friday 6th May 2022

I began the day by receiving new Smartfleet ticket comments. They're mostly UI tweaks that I need to make, such as making the buttons the same size and making the form "tighter" because it was taking up so much space. The most important was setting cate-gory and description fields separately for each file when the user wanted to upload multiple files at once. At first, it appeared to be very simple to fix because the logic was straightforward, and I knew exactly what to do. However, there is a complication in the Vue-Multiselect library – the library that I introduced previously when I implemented the adding colors feature in IMAR did not display the category value in the cate-gory field, despite the fact that the object had the category. It's a strange error, and I believe it's due to the library code. After messing around with this bug for a while, I decided to call it a day because I only had two tasks to complete and they were stuck, and I had a lot of work to finish at Haaga-Helia.

Week 6 evaluation

This was the second week that I dived in the Smartfleet ticket, I have done the requirement of the ticket last week, however the team lead asked me to improve it with documents list, and some features. The most productive day of the week was Wednesday because I had time to come to the

office and the working environment made me more productive and concentrated. My plan for work in this week was finishing the ticket and improving the old task that the team lead review. However, because of the work and exams at school, therefore I could not manage to finish all of the objectives.

The thing that I regret this week is that I did not manage my time well to handle the exam in school and also my own objectives at work. This mostly because of some activities with my friends. We live in the same building so meeting up, having dinner, or chatting a little bit is mostly not avoidable. Moreover, sometimes I cannot handle or control all the meetings like this because of many reasons. The most usual reason is that I let myself dived in the conversation or activity; therefore, when the party finished, there are many things left to do but just little time. Therefore, I need to be more severe and strict if I want things done.

The most significant thing that I learned during the whole week is the process of uploading files, which is a very common and important feature in every application. Users can upload profile pictures and other files to a variety of mobile apps and websites. As a result, when developing a REST API with Node.js and Express, handling file upload is a common requirement. In our project Smartfleet, when user upload a document, the information of the document will be saved in the PostgreSQL database in the table `file_meta` and the file will be uploaded to AWS S3 service.

To upload the file to AWS S3 service or any other services, we need to use `FormData` – and object that help developer to send HTML forms with or without files, with additional fields and so on. The `FormData` interface makes it simple to create a set of key/value pairs that represent form fields and their values, which can subsequently be submitted using `fetch()` or `XMLHttpRequest.send()`. The unique feature of `FormData` is that it may be used as a body in network methods. `Content-Type: multipart/form-data` is encoded and sent out. From the server's perspective, that appears to be a usual form submission. Thanks to `append` method, I can create a `FormData` whenever I want in JavaScript file (or VueJs file). The `append()` method of the `FormData` interface appends a new value onto an existing key inside a `FormData` object, or adds the key if it does not already exist.

Text, graphical controls like checkboxes or radio buttons, and files, such as photographs, movies, and other media, are all sorts of input that web applications receive from users. Each of these inputs is sent to a server, which processes them, utilizes them in some way, possibly saves them somewhere else, and finally returns a success or failure result to the frontend. The server, in our instance Node.js, has less work to complete when submitting forms with text inputs. You can easily grab all the inputs entered in the `req.body` object using Express. However, sending forms containing files is more difficult due to the additional processing required because the type of content it send is quite different. The `enctype` element is present in all forms and indicates how

data should be encoded by the browser before being sent to the server. `Application/x-www-form-urlencoded` is the default value, which supports alphanumeric data. The other type of encoding is `multipart/form-data`, which is used to upload files via forms. To handling Content-Type: *multipart/form-data*, we use a library called Multer - a Node.js middleware for handling `multipart/form-data` that makes the otherwise painstaking process of uploading files in Node.js much easier. Multer populates the request object with a body object and a file or files object. The file or files object contains the files submitted via the form, whereas the body object contains the values of the form's text fields. (Dillion 2022, *Multer: Easily upload files with Node.js and Express*)

This week I also learn about securing AWS S3 uploads using presigned URLs. All objects are private by default, which means that only the bucket account owner has access to them. If we want a user to have access to a certain bucket or object without making it public, I decided to use an IAM policy to grant them the necessary permissions. A presigned URL allows users to interact with items without needing AWS credentials or IAM permissions, in addition to employing an IAM policy. A presigned URL is a URL that you can give your users to grant them temporary access to an S3 object. A user can either READ or WRITE an object by using the URL (or update an existing object). The URL includes parameters that are set by your application. To limit the user's access, a pre-signed URL requires three parameters: bucket, key, and expired time. The user is unable to interact with the given item once the expiry time has passed, as expected. Because the URL can only be correctly signed by the S3 Bucket owner, AWS grants access to the object via the presigned URL. Anyone with a valid pre-signed URL can interact with the objects in the way they were intended when they were created. If a GET (Read) pre-signed URL is provided, for example, a user cannot utilize it as a PUT (Write). (Adian 2018, *Securing AWS S3 uploads using presigned URLs*)

3.7 Observation week 7 (May 9 – May 13, 2022)

Monday 9th May 2022

This is the exam week at Haaga-Helia, so I planned to not spend too much time working because there are not any high priority tickets in both Smartfleet and IMAR. Moreover, I told the team lead that my productivity this week will decrease because of huge number of exams and workload that I need to finish for schoolwork. I continued the day with improving the add documents ticket in Smartfleet, this is the third week I worked on this task. I still have some UI problem related to the library Vue Multiselect that I could not find why is that. Therefore, I needed to take more detours to solve the issue, this made the code longer and more difficult to read and might lead to confusing.

However, I lost my patient on this ticket, I just want to make it works, so I went on the long way. Because I worked on this ticket and the branch for a while now, so there are a lot of ahead commits, therefore I need to merge the master branch (the main branch) to my current working branch to keep the code up to date and resolved the conflict in merging process.

At the end of the day, I realized that I received some new requirements on Anomalies ticket in IMAR 4 days ago, and I did not know about it. I will try to finish both tickets tomorrow.

Tuesday 10th May 2022

I received some comments from the team lead on the add documents ticket in Smartfleet at the start of the day. This is also the ticket on which I received the most comments, with 35. For starters, the preview functionality had an access denied issue, but it only appeared in the environment where the team lead was testing. The feature worked great in my local area. Because it was so long and hacky, the team lead was not pleased with the manner I fixed the category field UI problem I encountered yesterday. However, despite my continued investigation, I was unable to uncover a more effective solution to the problem.

After fixing the Smartfleet ticket as the team lead required, I continued the working day with the old Anomalies ticket, and it seems like I did it so wrong as the team lead expectation. The reason is that I did not get the idea of how to implement the task from the beginning and the team lead said he need to write a better description for this issue. The team lead was so busy with Smartfleet budget issue, so it was a while since the last time I touched this ticket, because I could not know the problem of my code if the team lead does not review it. The new description required me to work more in the database and the backend. I needed to create a new table for Anomalies, unlike before when I used the vehicle_model to fetch and sort out the vehicles that have issues. Now I need to sort the anomalies in the back end by checking every models.

Wednesday 11th May 2022

My plan for today was finishing the Anomalies ticket because at the end of yesterday, I thought I got the idea of the task. In IMAR, the reason that we need to use 2 database MySQL and PostgreSQL is that the MySQL database is for tables which are from our customer Bassadone and the PostgreSQL is the application database. The reason for why we use PostgreSQL instead of MySQL as the application database because it was the decision of the developer who created the

framework for the application. He is not working with us anymore, so we do not know exactly why PostgreSQL is the choice. However, compared to MySQL, Postgres provides a wider range of data types. PostgreSQL might be a better option if the application works with any of the special data types it offers or with unstructured data. PostgreSQL will be more suitable for application that might grow to enterprise scope, with complex queries and frequent write operations. That might be the reason why.

After nearly 3 months working with Bassadone, I just got the idea of it today because I have not worked much in the backend side before. However, this was also the positive sign because I could know that I improve myself day by day. Everytime the backend run, it always run some schedule tasks to copy data from the MySQL table then modify to make them suitable for PostgreSQL tables. My job in this ticket was sorting out the vehicles that lack of information and pass it to the Anomaly table.

The logic was quite simple when I just need to go through every vehicle model with for loop, and check if it has problems or not, each problem is one anomaly. However, the result was so massive with more than 22,000 when the number of models is only 500 which means if every model has 6 anomalies, it is still impossible to get that huge number of anomalies. After checking out the functions, I figure out that the task ran multiples times a day automatically depends on the Cron Expression - a string consisting of six or seven subexpressions (fields) that describe individual details of the schedule ([docs.oracle.com](https://docs.oracle.com/en/technetwork/cloud-coop/articles/146527-cron-expressions-146527.pdf), *A Cron Expression*) - that the team lead setup. Every time the task run, it runs through the vehicle models and inserts the anomalies again with the same data but different id. That is why the application had many duplicated anomalies. After a while thinking and researching, I made a function that handle and update if there are something new existing anomalies in the database. Then I deleted all the old anomalies and run the task again and here we go I got more than 700 anomalies – a very reasonable number.

Thursday 12th May 2022

After finishing two big tasks above, I did not have any new tickets yet. Therefore, I decided to check these two tickets again, because I got some review and feedback that I need to be more thorough and careful when working from the team lead. And the team lead was not wrong at all, when I forgot a small feature in each ticket. In the add documents ticket in Smartfleet, there are many comments from the team lead, and I forgot one. He required me when the user go back from the edit document page, the page will automatically direct to the documents tab to reduce the number of clicks and improve the UX.

At first, I thought it was quite easy task, because the idea is just let the information page know that the previous URL is the edit document page, then with buefy Tabs - Responsive horizontal navigation tabs, switch between contents with ease - I can direct it to the documents tab by passing the index of the tab into the property. But the edit document page is different when user access it through URL not like father and son components, therefore I cannot use props - the method through which data is passed from a parent component to its child components. Normally, I usually use Redux in ReactJs or Vuex in VueJs to manage the state and this type of data. But we do not use such thing in our projects, therefore it was quite challenging. After hours of researching, I finally found the way to get the previous URL with a hook of Vue-Router – a library that helps developers manage routes and URL in VueJs.

The remaining feature in IMAR was not so significant, so it did not take me a lot of time finishing it. For the rest of the day, I took a simple ticket in Smartfleet because I thought I can finish it quickly, but I am still stuck in it.

Friday 13th May 2022

Today I had no lectures at Haaga-Helia, so I decided to come to the office to be more productive. I receive some comments about some small tweaks and improvement in two tickets above: add docuemnts in Smartfleet and Anomlies in IMAR. Today for the first time, the team lead evaluated my work is “very good”, normally I received some feedback that something should be improved. That was a very good sign. For the Anomalies ticket the team lead required me to create an integration test for it.

This is the first time I write the integration test, which is a sort of testing in which software modules are conceptually linked and tested as a unit., with Mocha. However, it was not so different from the cypress tests I used to write, I think all kinds of tests are still with the same structure and logic. After figuring out and starting writing test. I realized that I need to improve the handleAnomaly function – inserting and updating anomalies - that I wrote yesterday. The reason was that the update function did not actually update the anomaly. The only thing that it should be update was the resolved status when a problem has been solved in the vehicle model. Therefore, I need to spend a while rewriting the function. Now for example, if a vehicle model has a problem because lack of technical data, then there will be an anomaly with Not Resolved status here to present that problem. But when user import the technical data to that model in the database, and after the task run it will update the status of that anomaly to Resolved. For the rest of the day, I finished the integration test that tests inserting and updating function.

Week 7 evaluation

Because it was exam week at Haaga-Helia, I worked less than normal this week. I mostly finished everything in two tickets in IMAR and Smartfleet this week; each ticket took me about two weeks because they were the largest tickets I'd ever done, and I needed to balance school exams, assignments, and thesis. Nonetheless, the team leader gave me a positive assessment, which encouraged me much. Through these tickets, I may additionally develop my abilities and understanding in the backend and database.

My job this week were mostly about finishing 2 big tickets that I took in Smartfleet and IMAR. There is nothing much in the ticket in Smartfleet when I needed to fix the last bug related to the UI error of Vue-Multiselect library, however the bug also took me quite long time. I received comments from the team lead about the Anomalies task, the problem was I misunderstand what the team lead required me to do in the beginning; therefore, I basically have to redo the task and it took me more hours to finish. From this, the lesson I learned is that I need to ask clearly everytime I take a ticket to make sure I can fully understand what the team lead and the customer want.

The most significant thing that I learned this week is testing, and mostly integration test. Manual testing entails navigating your product and determining whether it will function as planned. This is, to some extent, a nice thing to do. Using your own application is usually beneficial. Automated tests, on the other hand, speed up your development process and allow you to quickly find flaws, break changes, and side effects.

There are three different common types of tests. Because you may expect particular results for your input, unit tests - sits at the base of the Testing Pyramid in Figure 10 - are the easiest to develop. There are no intricate interactions or dependencies. When you try to test Networking functions, like in LoginAuthenticator - a separate class for validation and authentication logic, you should use Mock Response to write unit test for both positive and negative responses and it only tests the response. Integration tests - sits at the next level after Unit Tests in Figure 10 - are more difficult to write than unit tests since they must account for dependencies. Here you will utilize Real Network Responses to test that the little units you've developed can operate together by accessing the real server endpoint. The main difference between unit testing and integration testing is that unit testing examines individual modules. These modules are merged and tested as a single entity during integration testing to ensure the overall application's functionality. End-to-End tests should test the user journeys of the app utilizing Real Network Response by contacting the Real server endpoints, same as Integration Tests do for Unit Tests. It may also assist new developers in better

understanding the app by allowing them to observe the automated test script running the app and interpret the app's user journey from there. Going back to the example of a Login Feature, End-to-End Tests will: Tap on the Username - Key in "Username" - Tap on Password - Key in "*****" - Tap on Login. (Lawrence 2019, *Unit Tests, UI Tests, Integration Tests & End-To-End Tests*)

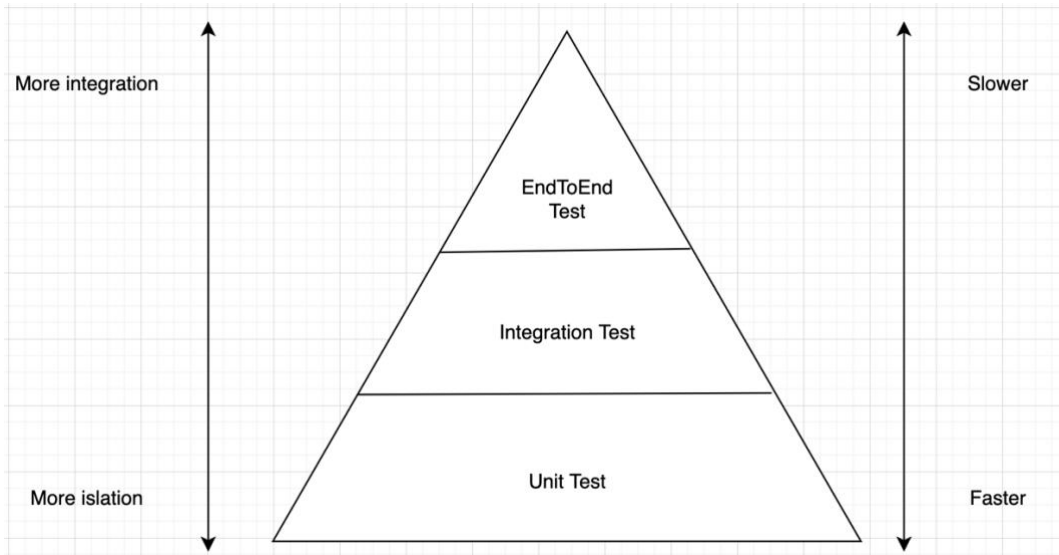


Figure 10: The testing Pyramid (an adapted version from the original “Lawrence 2019, *Unit Tests, UI Tests, Integration Tests & End-To-End Tests*”)

Above is the testing pyramid that concisely show the differences in 3 types of testing. In today's IT and software development environments, integration testing is critical, especially when needs are dynamic, and deadlines are tight. Even though each module of the application has been unit-tested, there may still be mistakes. Integration testing is critical for identifying these issues and ensuring that the modules perform well together after integration. The following are some of the most important reasons to implement integration testing (Rahul 2022, *integration testing or integration and testing (I&T)*):

- **Putting diverse components together into a working application:** Individual developers provide their own understanding and rationale to the development effort when working on different modules. When the modules are integrated, this can result in functional or usability issues. Integration testing can assist guarantee that the integrated units work together as a single unit and meet the criteria. It can also verify that no mistakes occur between the various interfaces of various modules.
- **Ensure that the application incorporates evolving requirements:** Requirements can and do change frequently in many real-time application environments. These additional needs may not always be unit-tested, resulting in flaws or missing product functionalities.

Integration testing can fill these gaps and verify that new needs are included in the final product.

- **Eliminating problems that are frequently overlooked during unit testing: Testing** is required for several modules that communicate with third-party application program interfaces (APIs). Integration testing is essential because this may not be possible during unit testing.
- **Getting rid of other common issues: Integration** testing also aids in the elimination of problems including insufficient exception handling, API response generation, data formatting, wrong external hardware interfaces, inappropriate third-party service interfaces, and error trapping.

3.8 Observation week 8 (May 16 – May 20, 2022)

Monday 16th May 2022

I started the week freshly with a brand-new ticket in Smartfleet. It was a very simple ticket when I only needed to change the select-field filter to multiselect filter to make it more dynamic. We already had a generic component for multiselect filter is `AutoCompleteField`, however after a while trying to use it in the case, I still had struggles on fetching options for the filter. Therefore, I decided to make the filter originally from `Vue-Multiselect` library because I can know that I can do it with the library because generic components sometimes are quite difficult to use. After struggling and researching all day, the ticket was done and waited for review from the teamlead.

Tuesday 17th May 2022

I started two Smartfleet tickets today. One involved minor UI changes to the order of columns on a page, which took only a few minutes to execute. Another work involved filtering Purchase invoice rows, but it was more difficult than the previous one because I needed to not only alter the type of a filter field to multiselect filter, but also develop a new vehicle contract filter for the table. The most important was the table's search functionality, which was based on the product name and description. In most personal projects with small databases, I utilize the JavaScript filter method, which creates a new array with all elements that pass the specified function's tests. But that is not reasonable with real project like this, I need to create parameters, api functions with SQL scripts for this functionality. At the end of the day, I was still stuck with this feature, I don't really want to ask the team lead because I know I can learn from his code.

Wednesday 22nd May 2022

In the beginning of the day, I receive messages from the team lead about the ticket that require me to change the filter in Vehicle Contract page to multiselect field. He said that what I did originally with Vue Multiselect library is good, but it was too unnecessary complicated; therefore, he suggested me to use the generic component AutoCompleteField that he already built when setting up. Even receiving a positive comment from the team lead, I still was not so happy because I tried to use AutoCompleteField before. And just because of a small struggle, I gave up using it and went with longer way by building the filter from scratch. This means I was not patient at all, and a little bit follow the heart because I knew it can be done with AutoCompleteField component. There is nothing wrong at all, but I need to learn to be more patient in the future. This time, it did not take me long to find the bug that I met and fixed and complete the ticket. If I know how to do this ticket from the beginning, it only took about 15 minutes, but in fact it took me around may hours to make it. After that I focused on search functionality in another ticket, I had some progress about still struggling.

Thursday 23rd May 2022

I had a Business Operations exam at Haaga-Helia and was packing my travel suitcase to return to Vietnam, so I didn't commit as much time to work as I should have. Nonetheless, it was a fruitful day because I completed the api function for obtaining purchase invoice rows from the database with extra search capabilities. However, there was a tiny snag while fetching with the WHERE clause. What I meant was (purchase-invoice-row-id AND product name OR description), but it was (purchase-invoice-row-id AND product name) OR description. It was just a grammar issue, but the SQL scripts in the function had a strange format, so I couldn't find it on the internet. As a result, I approached the team leader, but it was late, so he was not online anymore.

Friday 24th May 2022

I did not work today because I had a 20-hour flight to Vietnam.

Week 8 evaluation

Because it was exam week at Haaga-Helia and I also needed to prepare to come back to my home country Vietnam, I worked less than normal this week. I took 3 tickets this week, there were not complicated tasks, but it still took me quite long; therefore, it was not a productive week. Even though those are not highly demanded tickets and I still have positive feedback from the team lead, I still was not satisfied with my performance this week.

My job this week mostly about using generic components to create filters for pages in Smartfleet. At first, I did not choose to use the application generic component because I was struggle in using it for a while. After building the filer from the scratch, the team lead requested me to reproduce it by the generic component. I was not feeling uncomfortable at all because it was totally my fault when I was not patient enough, even though I thought that I can create the filter by the generic component if I spend more time researching and reading code carefully. If I can do it again, I will be more patient. From this, I learned a lesson that patient is one of the most important elements in this field.

The most significant thing I learned this week is the concept of generic component because two out of three tasks that I took are related to this type of component. A generic component is a collection of technologies that enable you to develop reusable custom parts. Developers usually follow specific UX designs when designing any web application for an organization. The organization will not have just one application; as the organization grows, more applications will be required, and all of these applications will be designed in the same way so that users may have a consistent look and feel across the ecosystem. Customers of SaaS companies can choose from a variety of products, and as the company grows, additional products will be required. (Karthik 2021, *Advantages of Reusable Components in Web Application*).

Here at WeAre Solutions, our team customer Bassadone required us totally 10 projects, therefore creating a component to display the data and as a quick solution so we will copy it in all the apps is very convenient. But there's a catch: if the UX design changes in the future, we'll have to update all of the applications, which is a time-consuming task.

Create a library of all those re-usable common components instead of replicating the component in all the applications. It's a one-time investment; construct it once and use it n times. Once the library has been developed, we may utilize the component in any of our current or new applications by importing it. And, in the future, if there is a design change, we simply have to do it once, and it will be mirrored across all applications with minimal work. (Karthik 2021, *Advantages of Reusable Components in Web Application*).

Because the components have been utilized in other apps, the chances of encountering any complications during the construction of a new application are not significant. If we have a problem,

the places to explore for the cause are limited because the issue is more likely to be in the code of a newer program.

Reusing code in different environments, on the other hand, is a common choice among developers because it allows them to make faster progress by duplicating a component and lowering development costs. But doing so is in fact not so simple.

However, this is not an easy task. In most circumstances, recycling a computer component is more of a design challenge than a technological one. “It’s important to keep in mind that a reusable component is at least 3 times more complex to design than a single-use component.” (Robert Glass – American software engineer). It is not just about copy and paste a generic component to a new project. Indeed, an exact duplication of a component in a context that is 100 percent identical to the original context to which the component was applied never occurs. There will always be functional and/or technical variances and diversity between contexts. It's exceedingly difficult to find two separate surroundings in which a single component fits perfectly – there are always discrepancies between them. (Mathieu 2020, *How to develop a good generic component?*)

This brings us to the design issue. A high level of design is required to allow all variants from one setting to be successfully used in another. If we are to produce a component that can be reused in other applications, it is precisely this high degree of design that takes the most effort and care. It's important to remember that developing a reusable component necessitates substantially more effort than developing a component that will only be used once. It will be required to anticipate and plan for all of the variables that the component will encounter during its use while creating a reusable component. It's advisable to wait till the next iterations are finished before designing a good generic and reusable component. By doing so, the development team will have a thorough understanding of all the scenarios in which the future generic component will be used. We can only begin designing the component that will be used in all of the anticipated contexts after they are finalized and ready. (Mathieu 2020, *How to develop a good generic component?*)

4 Discussions and conclusions

At the beginning of the job, I set 3 goals, they were to expand my professional knowledge and training, to adapt and get familiar with a new international working environment, and to change my concentration from the front-end to other development aspects. After more than 12 weeks of working in the new company WeAre Solutions Oy (including 8 weeks of writing a diary thesis), I have accomplished all of my objectives.

I just started my career nearly 2 months before starting this thesis, in February 2022. Compared to myself at that time, I can see a significant growth of myself. At that time, when I saw the code base of the project, I was quite panicked and overwhelmed, even in the job interview, I was so confident. For me, every day at work had been extremely stressful not because of I cannot handle the tasks on time, but the expectation that I put in myself. I wanted to be the excellent and important part of the team, but turns out, it is not that easy. That's why people need years to be senior and more years to be manager. I also was afraid to ask for help from coworkers because I thought I would annoy them, and I felt it was my responsibility to solve the problems on my own time. In the end, this did not lead to a very good result, when it usually took me more time to deal with the problems or I needed to ask anyway; however, it depends on the situation and many perspectives, if I can handle it with my own logic and research, I can gain something from that.

But after a few months of employment, I had equipped myself with a few abilities that are extremely helpful in this position. I feel comfortable talking to other coworkers about accelerating the development process. Because I wasn't supposed to know everything, I don't hesitate to ask for help or feel embarrassed when I do. After all, it can help the company save time and money. I developed several very important skills while working, including the ability to quickly learn by doing and conducting research. If I only learn things from school and assume I can apply them in the workplace, I will never feel prepared. In actuality, what I have learned at work is less than 30% of what I learned in school and less than 60% of what I have learned on my own or at the BootCamp I attend. I have been using my problem-solving and research skills that I developed while I was a student. Every programmer must, in my opinion, concur that after encountering a problem, they spend 90% of their time thinking about how to solve it and only 10% actually writing code. We don't need a lengthy response because coding is not writing an essay. A quick and simple answer is the best. The ability to debug is also crucial. I personally dedicated on average 50% of my working time to debugging. It can be difficult to debug something that is entirely new at times. I occasionally require assistance from the team lead or other developers. They are very eager to assist one another because doing so will boost productivity, which will ultimately boost overall company revenue.

About programming skills, I also realized that having solid fundamental programming skills in any language is a plus. This is due to the fact that I am at an advanced level in the OOP programming language JavaScript, at a medium level in the OOP programming language Java, at an entry level in Golang, and at a beginner's level in Python. That will make working and learning new languages or frameworks incredibly quick. The majority of the knowledge I have now comes from experience; for instance, I didn't know anything about VueJs when I started this job, but thanks to my familiarity with JavaScript and ReactJs, a JavaScript framework used to create Frontend applications similar to VueJs, I was able to integrate myself quickly into the company's projects.

About web development skills, when I started writing this thesis, one of my objectives is learning, coding and working dive into backend side and database. Now I can confidently say that I'm a junior fullstack developer now from an entry-level frontend developer a few months ago. I improved my frontend skills a lot with VueJs, learned and experienced backend and database side. Now I can take fullstack tickets in medium level.

About writing clean code, I also gained a lot of knowledge about writing clean code while working. Because it enables clear communication with the next person who works with what you've written, writing clean code is crucial. It's extremely important, especially in the world of software development, to be able to go back and understand previously written code. The better a piece of code communicates and the simpler it is to fix errors when they occur, the more declarative it is. You will become a better teammate, worker, and developer if you write clean code. Maintainability, readability, and understanding of the code, as well as a high level of security, are therefore of utmost importance. Working with seasoned developers who can review my codes and provide constructive criticism helped me gain those experiences and skills. Being the newest developer and the only junior in the company, I had an open mind and a strong desire to learn, so I put a lot of effort into learning these important facts.

About Git – a version control system, these skills I already learned from university and the BootCamp, I can learn about creating a git branch, send a pull request, and merge, but everything is based on personal projects and exercises. When I am a part of a large development team, I can only fully comprehend and utilize them. Together with four other developers, I worked on a project that we frequently made changes to in the same file. I have learned a lot about solving the conflict in git after many hours researching and many times I mistakenly deleted others' codes. I gained knowledge about using *git rebase* rather than *git merge* at my current employer, and I feel more comfortable using *git reset* when necessary.

About mentality, this is not related to programming or web development; however, mentality is related to everything in life including working. I learned how to be more patient which is very

important in being a developer, because debugging, researching and problem solving sometimes required not only hours, but it can also be even days and weeks. Sometimes, the impatient can lead to the discouragement and giving up even though the beautiful final destination is right there. Working under pressure is also an important part here. I had a chance to improve it while working, studying and writing thesis at the same time. It was not an easy time for me, but with a strong enough spirit and motivation, I handled everything not so bad, and the prize is very well-deserved.

About time management, beside of working and writing this thesis, I also took 7 courses at Haaga-Helia with 32 credits. This semester Spring 2022 was one of the hardest times of my life. I usually attended lectures in the morning, started the working day right after the noon and finished at around 7-8 o'clock. I spent the night for writing this thesis, and weekend for university assignments. The reason and also the motivation for me to working this hard is because of the tuition fees that my parents is helping me to pay every semester. The tuition fees are a large amount of money, and it is even larger in my home country which is much poorer than Finland. Therefore, I want to try my best to reduce the study time as little as possible for helping my parents and improving myself. At that time, I knew this semester will be a very difficult journey in my student life. My physical and mental health needed to be treated as careful as possible. I followed the rules that I need to eat healthy and one time, I went to the gym 4-5 times a week for releasing stress and improving my body. I met my friends a few times a week to keep me social and my mood better. We did not drink or party but talked and ate good food to make sure that we had a good time but not have bad effects to our health. After sharing my situation, some teachers allowed me to not attend the lectures and giving the assignments on time. They helped me a lot and I really appreciate that. This was the time when I disciplined myself the most. Therefore, everything went well as it was planned, there are no big problems happened in that time. I'm planning to graduate after the next semester Autumn 2022, it will be a very difficult semester too because I need to take many courses and credits while still working fulltime. The situation will be basically the same with Spring 2022, but I believed that I can go through it. "The harder you work, the luckier you get" – Gary Player.

Writing my thesis as a diary has proven to be a wonderful experience and has taught me a lot. Typically, I only took down the brief syntax, very quick tips, and very technical documentation in general. However, this is a good opportunity for me to further my analysis of the problem by not only taking notes. From week one to week eight, I had the opportunity to reflect on my mental development with more clarity and analysis. Good lessons have been learned in abundance. For instance, I am aware that my efforts are not always successful, but they do encourage me to advance. Even though I sometimes feel like I wasted my time working and didn't accomplish anything, the experience was priceless. This experience is even more valuable now that it was

gained during the time spent writing the diary thesis. Every time I failed, I discovered something new. I'm becoming an experienced developer and a more mature person as a result of those lessons.

The team's understanding of my expectations and their willingness to give me more opportunities to work on the backend rather than the frontend make me grateful. The team lead is unquestionably one of the best teachers and teammates I've ever had; he never loses patience with me, is always willing to help, and constantly gives me opportunities to get better. It was initially difficult and challenging, but it was how I developed. I'm enrolled in an AWS cloud course on Udemy and may soon have the opportunity to join a project that heavily utilizes cloud technologies. I'll take advantage of this chance to obtain my Cloud certificate, and sooner or later, I'll give being a Cloud developer my full attention. "There's always room for improvement" – Joey Logano.

References

Adian, H. 2018. *Securing AWS S3 uploads using presigned URLs* [Online]

Available at: <https://medium.com/@aidan.hallett/securing-aws-s3-uploads-using-presigned-urls-aa821c13ae8d>

[Accessed May 2022]

atlassian.com *Rewriting history* [Online]

Available at: <https://www.atlassian.com/git/tutorials/rewriting-history#git-commit--amend>

[Accessed April 2022]

Aws.amazon. *What is Docker* [Online]

Available at: <https://aws.amazon.com/docker/>

[Accessed March 2022]

aws.amazon.com *Amazon S3* [Online]

Available at: <https://aws.amazon.com/s3/>

[Accessed April 2022]

Ben, L. 2021. *Amazon RDS (Relational Database Service)* [Online]

Available at: <https://www.techtarget.com/searchaws/definition/Amazon-Relational-Database-Service-RDS>

[Accessed May 2022]

bocasay.com *How to develop a good generic component?* [Online]

Available at: <https://www.bocasay.com/develop-good-generic-component/>

[Accessed May 2022]

botify.com *What are URL parameters?* [Online]

Available at: <https://www.botify.com/learn/basics/what-are-url-parameters>

[Accessed May 2022]

browserstack.com. 2021. *End To End Testing: A Detailed Guide* [Online]

Available at: <https://www.browserstack.com/guide/end-to-end-testing>

[Accessed August 2022]

Cypress.io. *Cypress* [Online]

Available at: <https://www.cypress.io/>

[Accessed March 2022]

Developer.mozilla.org. 2021. *JavaScript* [Online]

Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

[Accessed March 2022]

developers.google.com, 2021. *Google Sign-In for server-side apps* [Online]

Available at: <https://developers.google.com/identity/sign-in/web/server-side-flow>

[Accessed April 2022]

Dillion, M. 2022. *Multer: Easily upload files with Node.js and Express* [Online]

Available at: <https://blog.logrocket.com/multer-nodejs-express-upload-file/>

[Accessed May 2022]

docs.oracle.com *A Cron Expression* [Online]

Available at: https://docs.oracle.com/cd/E12058_01/doc/doc.1014/e12030/cron_expressions.htm

[Accessed May 2022]

finder.fi. *WeAre Solutions OY* [Online]

Available at:

<https://www.finder.fi/Liikkeenjohdon+konsultointi/WeAre+Solutions+Oy/Vantaa/yhteystiedot/3266027>

[Accessed March 2022]

functionize.com. *Importance of End-to-End Testing* [Online]

Available at: <https://www.functionize.com/blog/importance-of-end-to-end-testing>

[Accessed May 2022]

Git-scm.com. *Git* [Online]

Available at: <https://git-scm.com/>

[Accessed March 2022]

github.com/buefy. *Buefy* [Online]

Available at: <https://github.com/buefy/buefy>

[Accessed March 2022]

github.com/vuejs. *Vuejs* [Online]

Available at: <https://github.com/vuejs/>

[Accessed March 2022]

Haydar, A, I. 2017 *Learning How to Git: Merging Branches and Resolving Conflict* [Online]

Available at: <https://haydar-ai.medium.com/learning-how-to-git-merging-branches-and-resolving->

[conflict-61652834d4b0](#)

[Accessed May 2022]

javascript.info/formdata. *FormData* [Online]

Available at: <https://javascript.info/formdata>

[Accessed May 2022]

Jes, S. 2019. *An SEO Guide to URL Parameter Handling* [Online]

Available at: <https://www.searchenginejournal.com/technical-seo/url-parameter-handling/>

[Accessed April 2022]

jetbrains.com/datagrip. *Why DataGrip* [Online]

Available at: <https://www.jetbrains.com/datagrip/>

[Accessed September 2022]

Karthik, K. 2021. *Advantages of Reusable Components in Web Application.*[Online]

Available at: <https://www.linkedin.com/pulse/advantages-reusable-components-web-application-karthik-kumbar/?trackingId=B6sRIbVsjFaMeAlQthciNg==>

[Accessed May 2022]

Lakshay, S. 2022. *Why is Testing Necessary?* [Online]

Available at: <https://www.toolsqa.com/software-testing/istqb/why-is-testing-necessary/>

[Accessed April 2022]

Lane, W. 2021. *What are UUIDs, and should you use them* [Online]

Available at: <https://blog.boot.dev/clean-code/what-are-uuids-and-should-you-use-them/#:~:text=By%20using%20UUIDs%2C%20you%20ensure,be%20the%20same%20as%20you rs.>

[Accessed May 2022]

Lawrence, T. 2019. *Unit Tests, UI Tests, Integration Tests & End-To-End Tests* [Online]

Available at: <https://lawrey.medium.com/unit-tests-ui-tests-integration-tests-end-to-end-tests-c0d98e0218a6>

[Accessed May 2022]

Mathieu. 2020. *How to develop a good generic component?* [Online]

Available at: <https://www.bocasay.com/develop-good-generic-component/> [Accessed May 2022]

Mochajs.org. *Mocha* [Online]

Available at: <https://mochajs.org/>

[Accessed March 2022]

mordendatastack.com. AWS S3 [Online]

Available at: <https://www.moderndatastack.xyz/company/aws-s3>

[Accessed April 2022]

Mysql.com. *MySQL* [Online]

Available at: <https://www.mysql.com/>

[Accessed March 2022]

onelogin.com. *How Does Single Sign-On Work* [Online]

Available at: <https://www.onelogin.com/learn/how-single-sign-on-works>

[Accessed April 2022]

postgresqtutorial.com. 2022. *What is PostgreSQL* [Online]

Available at: <https://www.postgresqtutorial.com/what-is-postgresql/>

[Accessed March 2022]

productplan.com. 2022, *Jira* [Online]

Available at: <https://www.productplan.com/glossary/jira/>

[Accessed March 2022]

Rahul, A. 2022. *integration testing or integration and testing (I&T)* [Online]

Available at: <https://www.techtarget.com/searchsoftwarequality/definition/integration-testing>

[Accessed May 2022]

redhat.com. 2020. *What is a REST API?* [Online]

Available at: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>

[Accessed March 2022]

Ryan, H. 2015. *Rewriting History with Git Rebase* [Online]

Available at: <https://code.tutsplus.com/tutorials/rewriting-history-with-git-rebase--cms-23191>

[Accessed April 2022]

Santiago, C. 2022. *8 Reasons Why Docker Matter For Devs* [Online]

Available at: <https://www.jobcity.com/blog/8-reasons-why-docker-matter-for-devs>

[Accessed Auguts 2022]

vero.fi *VeroAPI* [Online]

Available at: https://www.vero.fi/en/About-us/it_developer/tax-administration-interface-services/vero-api/

[Accessed April 2022]

W3school.com. *CSS Tutorial* [Online]

Available at: <https://www.w3schools.com/css/>

[Accessed March 2022]

weare.fi. *Services* [Online]

Available at: <https://www.weare.fi/en/services/>

[Accessed March 2022]

Appendices

Appendix 1. Appendix heading