

Software engineering diary thesis

Lam Duong



Author(s) Lam Duong	
Degree programme Business Information Technology	
Report/thesis title Software engineer diary thesis	Number of pages and appendix pages 49
<p>The thesis covers 10 weeks of working period starting from 21.01.2019 until 13.04.2019. It is written in the form of a diary.</p> <p>The thesis focuses on a specific description of the daily responsibility, thoughts and feelings of a junior software engineer in 2 consulting companies, as well as difficulties encountered at work. In addition, it gives some explanations for a variety of technologies, frameworks, tools and approaches in the development stages of a project. The goal of the thesis is to assess the author's development and analyze the problems that occur with the accompanying solutions. This thesis can be a material for students who want to become developers or developers working with similar technologies.</p>	
Keywords Rest API, scrum, front-end, back-end, testing, authentication, IoT.	

Table of contents

1	Introduction	1
1.1	Polku Innovation	1
1.1.1	Company overview	1
1.1.2	Working environment	2
1.2	The biomedical company	2
1.2.1	Company overview	2
1.2.2	Working environment	3
1.3	Key concepts and abbreviations	3
2	Starting situation	3
2.1	Analysis of current work at Polku Innovations	3
2.1.1	Company's IoT platform	4
2.1.2	Client's laundry service	4
2.1.3	Client's medical application	4
2.2	Analysis of current work at the biomedical company	4
2.2.1	Client's animal healthcare application	5
2.3	Skills at current work	5
2.4	Interest groups at work	6
2.5	Interaction skills at work	6
2.5.1	Communication at work	6
2.5.2	Challenges	6
3	Diary entries	7
3.1	Observation week 01	7
3.2	Observation week 02	12
3.3	Observation week 03	15
3.4	Observation week 04	19
3.5	Observation week 05	23
3.6	Observation week 06	26
3.7	Observation week 07	30
3.8	Observation week 08	33
3.9	Observation week 09	37
3.10	Observation week 10	43
4	Discussion	46
	References	48

1 Introduction

The thesis was written in the form of a working diary. It described my daily responsibilities as a software engineer. Furthermore, it is followed by a summary after each working week to analyse what I have learnt and found during the week.

Like any other software engineer, my main job is to design, develop and install software solutions. The responsibilities of any software engineer include collecting requests from users, defining system functions and writing code in many different languages, such as JavaScript, Java or .NET programming languages. This position also needs to be familiar with the product development life cycle from initial system analysis to testing and implementation. Finally, the role of a Software Engineer is to create high quality and innovative mobile and web applications that complies with coding standards and design techniques.

I have started to work as a software developer since the very beginning of 2018 and I have worked for 2 different companies. My job includes developing the user interfaces and the databases of web applications. During my work as a programmer, I felt I had learned a lot. There was a big change since the early days when I joined the company until the last days, I worked at Polku Innovation. After working there for a while, I felt I needed a bigger environment to challenge myself as well as myself wanting to work with new technologies, so I decided to switch to a biomedical consulting company. This diary-based thesis will be separated into 2 period. In the first period I will write about the last 5 weeks I worked as a full-stack developer at Polku Innovation and the second one will be the first 5 weeks I work as a frontend developer at the biomedical consulting company.

1.1 Polku Innovation

1.1.1 Company overview

Polku Innovation is an IoT consulting company that is based in Vantaa. They provide technology services and solutions for businesses.

In Polku Innovation, there are 2 different business platforms. One is Polku zone - a software that helps buildings collect information related to humidity, temperature, noise and CO2 through sensors. It helps businesses observe changes related to data in their buildings and provides a solid foundation for developing ecosystems of them. Zone enables businesses to digitize their structures to settle on better choices, increase the execution of structures and improve the condition of clients. Zone's platform is structured only for smart buildings and is extremely simple to utilize. Their administration will enable businesses to choose, convey and oversee from the earliest starting point as far as possible of the IoT arrangement. Figure 1 bellow shows different features of Zone.



Figure 1: Zone platform

Besides Zone, they also help businesses create applications to collect, visualize, analyze, combine and use diverse data of buildings and employees. Polku is master in sensors and detection technology such as building sensors: thermal, light, motion, infrared, and various connectivity solutions. Devices such as sensors (sensors): thermal, light, motion, infrared. Moreover, they also build different applications for wearable devices to measure human's health such as: heart rate, muscle diagnostic, walking/activity bouts, daily steps.

1.1.2 Working environment

In Polku Innovations, there are total 7 developers, depending on the projects, people are divided into small teams of 2-3 people. In each group, there will be 1 back-end developer, 1 or 2 front-end developer. In some few small projects, there will be only one full-stack developer who works independently and take responsibility for the project from the beginning until the product is launched.

1.2 The biomedical company

1.2.1 Company overview

It's a medium-sized company focused on biomedical software solutions. It has extensive knowledge of biomedical sciences. People in the company create cutting-edge software solutions for robust integration of biomedical data combined with intelligent tools to access

data and turn it into executable knowledge. They also provide web-based terminal solutions for the entire biomedical data value chain. Whether you are processing genomic, clinical or multi-data data, their solutions help you manage, manage, analyze and interpret large biomedical data. Their solutions are used worldwide to empower accurate diagnostic and medical services, pharmaceutical and veterinary companies, biobanks and genomic programs. They serve their customers from original ideas to delivery, maintenance and growth.

1.2.2 Working environment

The company focuses on investing in facilities to create an innovative work environment helps employees to best promote their abilities. They provide laptops, computers, keyboards, for employee. People in the company can also take remote working days whenever they want. Besides, it allows people to bring dogs to the office which is very beneficial to dog owners. In the company, there are many different projects and I work with two other developers, 2 testers and a manager in a project, sometimes I also work directly with clients.

1.3 Key concepts and abbreviations

- IoT: Internet of thing
- UI: User interface
- UX: User experience
- HTML: Hyper Text Mark-up Language
- CSS: Cascading Style Sheets
- SQL: Structured Query Language
- API: Application program interface
- HTTP: Hypertext Transfer Protocol
- URL: Universal Resource Locator
- DOM: Document object model
- NPM: node package manager

2 Starting situation

2.1 Analysis of current work at Polku Innovations

As a Front-end developer, I am responsible for the look of the company's web applications and the architecture of user experiences. To accomplish these goals. A front-end developer must master three major programming skills: HTML, SCSS, and JavaScript programming languages. In addition to mastering those languages, I need to familiarize myself with frameworks such as VueJs, jQuery, Nodejs, NoSQL to ensure the content is always displayed well on every device, and libraries like jQuery and SCSS, pack code into a form that saves time and is more useful. A lot of work for me also requires experience with Ajax, a technique widely used by JavaScript to enable pages to load automatically by downloading server data at background part.

In the early stages of a project, I had to work closely with designers and clients to turn mock-up, or wireframe, from development to actual product. I can also pinpoint specific issues in the user's experience, providing systematic recommendations and solutions to influence that design. It is important that they can collaborate with other groups in the company to understand the specific purpose, needs and opportunities, and then follow those instructions.

At the company, I participated in many different projects. At the present time, the company has 5 projects and I participated in 3 of them, including the company's production, a booking application and a medical application.

2.1.1 Company's IoT platform

In the company's product, I work with 2 other people. They are full-stack developers that have many years of experience in this industry. I work mainly on the user interface. because the product has been completed up to 60%, so my job includes: developing unfinished functions, changing the interface yet beautiful and creating new components as well as new functionalities that the company requires. Besides, the product has many errors, so I am also responsible for fixing those errors.

2.1.2 Client's laundry service

This is one of the projects that has been launched and it is in the maintenance stage. I took over the project from a developer who just left the company 1 month before I started. My main responsibility in this project is to fix the application's error. The project was taken into use and after a period of using the product, customers found some errors in it and asked me to maintain it.

2.1.3 Client's medical application

This is a project that I have learned a lot from. I made it from scratch, taking care of all the work from the back-end to the front-end development. At first, I worked alone, but when the project was completed 70%, there were 2 other people involved to help me speed up and bring the product to market. My daily work includes developing Api based on a company template and develop frontend from the first functions and features.

2.2 Analysis of current work at the biomedical company

In the company, I'm also a frontend developer and my main responsibilities are creating new UIs and make sure they work well on different devices especially with mobile devices. I use Reactjs for the project as the main JavaScript library and material UI for component

styling. Besides, I use assemble to track working status, for example: when someone assigns a task for me, he will use assemble to assign the issue to me and the application will send me a notification or when I get some tasks done, I will move the task to testing column and the application will send a message to testers and inform them that my tasks are ready to be tested.

2.2.1 Client's animal healthcare application

This is the project that I'm participating in at the biomedical company. I work with 2 other full-stack developer. The project has been launched and my responsibilities is developing new UI features or maintain old functionalities.

2.3 Skills at current work

HTML and CSS are the most basic building blocks in web coding. I must study and master at these 2 tools to manage daily styling tasks.

JavaScript - allows adding tons of functionality to the site. I use a lot of JavaScript from the most basic level to the most complicated one. JavaScript supports a lot of interactive elements for web pages - elements used to create interactive things.

jQuery - a JavaScript library: a set of plugins and extensions to help develop with JavaScript faster and easier. Instead of coding everything from scratch, thanks to jQuery, I have additional elements available to the project and customize as needed.

VueJs - A JavaScript framework that provides the structure available for JavaScript code. It's fast and easy to maintain.

Responsive Design - One of my main responsibilities is to make sure that users can view the applications from different web browser and devices. When a website is accessed from the desktop with a large screen, users will receive multiple columns, large and interactive graphics created specifically for the mouse and keyboard used. On mobile devices, the same web page will appear with a single column optimized for touch interaction but using the same base files.

Mobile designs can include responsive designs, as well as specific mobile designs. Sometimes the user experience you want when they visit the site on the desktop will be completely different from the experiences that occur on mobile. For example, a banking website with online banking will benefit from a separate mobile site that allows users to view things like the nearest bank location and check accounts.

2.4 Interest groups at work

Internal

- Product owners
- Project managers
- Developer teams
- Testers/QA

External

- Company clients
- Partners
- Private customers

2.5 Interaction skills at work

2.5.1 Communication at work

At the company, we use slack as the main communication tool at work. At each project, we created a separate channel for that project on slack, and all the information related to that project from important documents, to work progress was shared there. We also created a common channel for everyone to say what they think, or they want to take a break or work remotely. In this channel, we also report the progress of the work every 1 day.

In addition to slack, we also have a meeting for everyone once every 1 to 2 weeks. to report what we did last week, what we are doing now and what to do in the future.

For customers, because most customers are far away so most discussions are via Skype. If it is not a formal meeting, but simply that the customer wants to add a function or change a little interface, then we will directly call the customer or vice versa, to agree on what they want us to do. In addition, we use WhatsApp, email to send information about changes customers have requested.

2.5.2 Challenges

The biggest difficulty is the language barrier. We work with a lot of customers Finland, so most of the discussion related to customer projects were conducted in Finnish. And of course, I will not understand. In this case my colleague will help me talk to the customer and convey their wishes to me.

Secondly, because I have studied and worked with computers for so long, I became afraid of communicating and not confident enough in communication. Not to mention that I am the only female and the youngest employee in the company, which makes me shy whenever I talk to colleagues.

The last barrier, that is culture differs from Western and Asian culture. Many times, I feel lost and not see the good point of the story from colleagues because the cultural difference is too big. and when we discuss together, conflict will sometimes arise.

3 Diary entries

3.1 Observation week 01

Monday 21.01.2019

During the previous 2 weeks, I spent a lot of time reading and learning more about Nodejs, MongoDB and connections between the backend and frontend to prepare for this project. If in the previous projects I only focused on the user interface, in this project I also have to be responsible for both the server and the database.

That was the first day I started implementing the medical project for a client company from Estonia. Started with creating the database. I started by creating an account on MLab to store the database. After that, I work with the software authentication. Like any other software, using this software, users need to log in to be able to see the information it provides. It took me a lot of time to make this authentication step, it requires me to have knowledge about authentication, authorization, session, token, and cookie.... I have read and followed quite a few instructions on the net and checked which one works best. After trying a few ways, I decided to use Oauth 2 for this project.

All the work of today was just to stay at the project creation. Prepare all available services that the project will use and select the framework that best suits it.

Tuesday 22.01.2019

After having set up the project with the accompanying services. Today promised to be a hard-working day to implement theory into code

I started installing libraries and packages for the project. Because I used nodes to make the backend for the project, all the installation tasks are very simple with the terminal of the computer. the syntax is 'npm + package name'.

I installed packages such as

- body-parser (parse requests to the server)
- express (make the application run)
- no demon (restart when having some changes)
- mongoose (object data modelling to simplify interaction with MongoDB)
- bcrypt (hashing and salting passwords)
- express session (handling sessions)

- connect-mongo (store session in MongoDB)

After that, I learned how to create schemas with MongoDB. Figure 2 describes how a schema model looks like.

I then processed the data with the `find`, `findOne`, `findOneAndUpdate`, `findOneAndDelete` commands. These are the commands used by the computer to add, edit, find and delete data in the database.

```
1  var mongoose = require('mongoose');
2  var AuthorSchema = new mongoose.Schema({
3    email: {
4      type: String,
5      unique: true,
6      required: true,
7      trim: true
8    },
9    username: {
10     type: String,
11     unique: true,
12     required: true,
13     trim: true
14   },
15   password: {
16     type: String,
17     required: true,
18   },
19   passwordConf: {
20     type: String,
21     required: true,
22   }
23 });
24 var Author = mongoose.model('Author', AuthorSchema);
25 module.exports = Author;
```

Figure 2: Mongoose schema

At the end of the working day, I have created a few schemas and worked with data model but still haven't had time to test them. I will read more about it tonight and will test everything I tomorrow.

Wednesday 23.01.2019

As I mentioned, today I would test yesterday's work. But first, to check if the schemas and changes in the database work, I have to create routes for them and test them. For API's testing, I mostly use Postman. Postman is an amazing tool that allows us to work with APIs, especially REST. With Postman, we could call the Rest API without writing any code. Owing to Postman, you would find that it was not complicated to call the Rest APIs (like Facebook, Google, YouTube). In here I used an author Router to setup 1 route to a

page and sub-pages of its example of author posts. The purpose was when the user accesses localhost: 3000 / author, it will display "Here is the author page" and when accessing localhost: 3000 / author / post, it would show "every post of author here".

At this step, I had some minor problems. That was the postman always says '500 Internal Server Error'. I tried many ways but could not get it work. I asked one of our senior developer to help me. He found out that I used the wrong name for the route, and postman could not find the name. In other words, the name is used to test did not match with the name i wrote initially. After revising the router's name, everything ran perfectly, I can now send the request to the server and get the results I want back.

After making the router work, I started to make the authentication by using the PassportJs library and Oauth2 library. To do that created the user model, configure the passport for local authentication and use the passport configuration for processing login and registration. I created a user model used for this entire series. Users will be able to connect to his or her accounts. In the case of local, I used email and password. For social networking accounts, I used ids and tokens and some user information. All passport configurations will be written in passport.js file.

So far, I created a passport object inside server.js, and then push it into passport.js. After that I configured Strategy for local, Facebook, and google cases. This is also the file I displayed functions *serializeUser* and *deserializeUser* to store user information in the session.

Thursday 24.01.2019

Yesterday I configured the passport for the local account. Today I handled the login and registration in the passport.js file. Figure 3 shows how login function is implemented.

```

var LocalStrategy = require ( 'passport-local' ) . Strategy ;
var User = require ( '../app/models/user' ) ;
module . exports = function ( passport ) {
  passport . serializeUser ( function ( user , done ) {
    done ( null , user . id ) ;
  } ) ;
  passport . deserializeUser ( function ( id , done ) {
    User . findById ( id , function ( err , user ) {
      done ( err , user ) ;
    } ) ;
  } ) ;
  passport . use ( 'local-signup' , new LocalStrategy ( {
    usernameField : ' ' ,
    passwordField : ' ' ,
    passReqToCallback : true
  } ,
  function ( req , email , password , done ) {
    process . nextTick ( function ( ) {
      User . findOne ( { 'local.email' : email } , function ( err

```

Figure 3: Login authentication

Now I have provided a Strategy to the passport with the name local-signup. We will use this Strategy for processing registration forms. After I added the handle to request POST / login. Figure 4 indicates how a request is handled with passport.

```

app.post('/login', function(req, res, next) {
  passport.authenticate('local', function(err, user, info) {
    if (err) {
      return next(err); // will generate a 500 error
    }
    // Generate a JSON response reflecting authentication status
    if (! user) {
      return res.send(401, { success : false, message : 'authentication failed' });
    }
    req.login(user, function(err){
      if(err){
        return next(err);
      }
      return res.send({ success : true, message : 'authentication succeeded' });
    });
  })(req, res, next);

```

Figure 4: How a post request is use

With the help of a senior backend developer, I was able to create authentication successfully. I got a passport, routes work. On my browser, I accessed to <http://localhost:3000/login> and test what I have done. Everything worked and the logged in user information will be saved in session.

Friday 25.01.2019

After the authentication has been completed, I continue to add more schemas and models to the database, almost everything is just copy and paste from the schemas and the models I created earlier and just change the name, so this task It's quite simple but it takes a lot of time, because I need to create 14-15 different schemas and models and test them. Therefore, I spent the whole day today working with that.

Week 1 analysis

Generally, this week was my first experience with backend development. I have learned how to work with database and server.

It helped me to understand better the difference between NoSQL and SQL which I learned earlier at school. Talking about SQL database, the basic concept is so; It has a relational database. SQL is a domain-specific programming language that is used to communicate with a database. In other words, it is a relational database. Relational databases use strict relationships to store data. Relational database matches data by using common features found in data sets. And the result group is called Schema. On the contrary, NoSQL means Non-Relational. However, the term is can also be understood as Not Only SQL. Not just SQL This is the general term for database systems that do not use relational data models. NoSQL specifically emphasizes the key-value pair storage model and distributed storage system. (Bytescout 2018).

Based on what I have studied and worked with, I feel that working with NoSQL schemas and models is quite simple and it's simpler than SQL that I have practiced at school due to several reasons.

- Data in NoSQL DB is saved as document, object. Query is easier and faster than many RDBMS.
- NoSQL can work completely ok with unstructured form data.
- Changing data structures (Adding, deleting fields or tables) is easy and quick in NoSQL.
- Since the ACID of transactions and data consistency are not heavily enforced, NoSQL DB can be expanded and run on multiple machines with ease.

For creating authentication, it was difficult and required me to have a lot of knowledge about the behavior of the browser and the server. Meanwhile, I have not had any experience in this field before. So, all I could do was follow the documents online and asked colleagues when I had difficulties.

However, I still feel that I have not really understood some of the technologies in this backend development such as cookie and token.

Moreover, I had met some problems with errors handling. Which means that when I do something with the database, nor specific success or failure messages are showed. So, I would not know where the error was in the code in order to quickly find and fix it.

3.2 Observation week 02

Monday 28.01.2019

This week I would continue the unfinished work from week 1 with the database and server. Specifically, adding errors handling and completing routes, schemas models. After referencing source code from previous projects, I realized that errors handling functions are included in all functions in models. Here I found out that I was too rushed to write the schema and the model before errors handling. This will take a long time to open each file and folder of each function to add error handling. But I have to do it because there is no other option.

I started a separate file for errors handling. I defined different error types based on what I have consulted from senior developer and I also based on the list of HTTP status codes. for example:

- 1xx: Information: When receiving such codes, the request has been received by the server and the processing of the request is continuing.
- 2xx: Success: When receiving such codes, the request has been received, understood and handled successfully by the server.
- 3xx: Redirection: This status code indicates the client needs to have an additional action to complete the request
- 4xx: Client Error: It means that the request contains an incorrect syntax or is not executed.
- 5xx: Server Error (Server Error): It means that the Server failed to make a request that looks like it is feasible.

I divided them into many groups to show appropriate messages.

After implementing errors handling, I added it too every function in the database model.

Tuesday 29.01.2019

Backend development work seemed to be completed 70%. And today it was used to connect to the front-end to test. I was back to my strength that was working with the user interface. The framework that the company chose to use for the front-end is VueJs - a new JavaScript UI library that is quite hot recently and Vuex for state management.

I started project by creating Vue app. This is quite simple, and everything has been built by the library and all I have to do is *Vue create project name* in terminal. Then I moved to the connection between frontend and the Api that I have almost completed yesterday. Because the backend has a lot of security information, so the frontend must also hide that information. Therefore; based on previous projects, I created a plan for the development tasks that need to be configured before doing the code. I spent the whole morning to make plan of configuration files and divide folders for the project.

The very first thing I tried to set up is to create a secret place to hide important information of the project such as credentials and private keys. When I first started working as a developer, I was always worried about pushing code on GitHub or GitLab that would contain information about Database or API keys and URLs. Through research, I have found an amazing tool, a rather small and convenient NPM package called *dotenv* - a module that loads variables from the *.env* file into *process.env*. Therefore, it is the ideal place to store usernames, passwords, URLs and other sensitive data. I keep all the security information in this file and to avoid having important information leaked out, I have to create a file called *gitignore* and put the *.env* file in it. Thanks to that, when I put the code on GitHub, it ignores the *.env* file and so the important information is kept confidential.

After working with *.env* file and trying testing on it, the application still got no error. And it can take the variables from *.env* file without exposing any important information after being pushed to GitHub.

Wednesday 30.01.2019

After the configuration of *.env* file completed. I moved on working with the connection between the front-end and the API. I was hesitant about this for a long time because there are many solutions for connecting the frontend and the backend on the internet. And most of them are quite inconvenient and can be used only for small hobby projects.

In this case, I had to discuss with one of our senior full-stack developers. He instructed me to use Axios and save the methods of Axios in a single file. Axios is an HTTP client based on JavaScript Promise object, Beside Vuejs it can be used in many other font-end applications such as React, Angular Using Axios easily sends asynchronous HTTP requests to REST endpoint.

After the discussion, I implemented the file to connect two sides of the application. I declared all request options of Axios in that file including post, get, put, delete.

Thursday 31.01.2019

After I finished the API connection file, I started working on the login and register page of the user interface. Some main interfaces of the project include login and register page was pre-designed by our designer. So, all I need to do is follow the design as accurately as possible. The main technology I used is just SCSS and HTML template. The task was quite easy, but because it had to be the same with the design, it took me a lot of time to do the styling with SCSS.

Friday 01.02.2019

After completing the styling for login and register pages. I conducted the functional programming part in these 2 pages to help users log in and register like any other applications.

Like the backend, the frontend also has a lot of supported tools and packages to make the performance of the software better. There are a bunch of packages I have install for the frontend, including:

- vuex (state management)
- vue-router (routing)
- lodash (a library providing functions)
- vue-i18n (for using many languages)
- vee-validate (validation inputs)

To set up the login process, I had to handle the v-model event and keep the values of the text box in a state variable. When the user clicks the button login, you made the function to read the email values and passwords in the text box, same with the register page, when the user fills in his or her information to register, a function was made to read all the text fields. When user click register button and all fields are validated and have no problems. The user will register successfully.

While executing login method. I had encountered a small error that is instead of connecting to the backend localhost URL. it connected to its own URL. And browser reported '500' error This problem occurred because the Api URL initialization method had not been called when the software ran initially. I was quite confused and needed help from my colleagues to fix this error.

To fix this error, I just needed to call the URL declaration function in the App.js component – the component that wrapper all other components in the application.

Week 2 analysis

This week was a relatively good working week for me. I have learned how to use `.env` file to store information. Before knowing about `.env` file and how it works. I have had several problems with the deployment of the software. For example: When I need to deploy a Node.js application to Heroku or any other hosting services, and the app connects to the database of a database hosting service like MLab. To connect to the database of MLab, I need to access to the database URL that mLab provides in the application. This url has a form: `mongodb://<database_username>:<database_password>@code.mlab.com:code/project_name` contain the username and password of the database Users. If the username and password are always set in database URL, then if I push the source code to GitHub, the whole world can access my database. For an academic project, of course, no one will care, but a commercial project, is not allowed to reveal database's username and password.

Development has been much easier since the invention of the `.env` file. You could easily set your environment variables and values with the syntax `ENV_VARIABLE=VALUE` and boom! These variables got loaded as your environment variables, making it possible to get quick access to them' (Erisan Olasheni – 5.10.2018). Therefore, by using `.env` file, I can shoot an arrow that hits 2 targets:

- Hide confidential information (username, password, private key) in the file `.env`, not committed via regular git.
- Take advantage to switch back and forth between environments for different purposes.

Moreover, I've also learned how to connect the frontend with the API very conveniently and make it reusable in many different places in the application. Besides, I was also taught a lot about how to write clean, easy-to-follow and easy-to-maintain code from colleagues.

I felt clearly that my knowledge and ability in programming are significantly improved. Because this is the project I made from the scratch. From the initial setup, prototyping to project implementation with code.

3.3 Observation week 03

Monday 04.02.2019

This week I seemed to be working with the frontend even more. After completing the login function. According to my original plan for the project. I implemented router for this software. Vue router is used to help guide the UI in sync with the URL. To make it easier to

understand the react router, it helped components join. In order to test the routes, I created a few components and assigned each component a specific route. Implementing routes was not difficult and didn't take time at all. I finished the routes very fast and continued with components developing. Today, the main page called 'Dashboard' was the place where I worked mostly at. This page UI was not designed before so I could be free to style if I follow main colors that the UI designer has chosen.

Although the UI was not designed in advance, it is one of the most content-rich pages in the application. At first, I was confused because I did not know where I should start with this dashboard. I was thinking: 'Should I fetch the Api first? Or make the UI? Or move to something easier?' Finally, I decided to make the UI first.

Since there were so many components that needed to be displayed and there are too many functions, I decided to use an external component library called Element UI and a library for styling CSS called Materialize CSS, then customize them again into mainstream colors decided by designer . Styling the UI always takes a lot of time. Besides having to make it beautiful and good at all kinds of devices including computers, tablets, mobile phones. I have to make sure it works well on different types of browsers such as safari, internet explorer, Firefox, google chrome.

Moreover, because it was convenient for reusing components. I had to split the dashboard UI into many smaller components and then merged them in the same view. At the end of the working day, I have completed the entire dashboard page and made it suitable on all devices, and every screen resolution.

Tuesday 05.02.2019

It's time to implement functions for the dashboard page. Today, I worked with Vuex a lot. As I understood the application would grow bigger and bigger. Vuex is an excellent choice for storing states. Because Vuex is a library that helps manage the state of components in Vue.js, which is a centralized repository for all components in an application, with the status principle can only be changed. in predictable style.

Vuex operates under the "One-way data flow" model with the following components:

- State: The state, which is where the application originates.
- View: View , which is the mapping that mapped to the state.
- Action: Action, which causes the state of change to respond to user input from the View.

Vuex made our project management effective. It balanced the speed and performance of the project.

In Vuex, I fetched the Api and store the database in a form a states there, and then display them in the dashboard UI. I also wrote functions in Vuex and called them in the UI, that help the interface interact with the database including updating information or removing information from the database.

While writing those functions, I met two major obstacles, that is, the state was mutated when there was a change outside Vuex. In other words, states can only be changed in vuex and cannot be modified in Vue files. So I have to use lodash- a powerful JavaScript library used to handle Array, Object, Function, Collection, etc. and the deep clone function from it to avoid that mutation errors.

The second error I encountered was when fetching data and the returned object was too big or too deep. The UI interface could not read it, which caused the software crash. In order to fix this error, I had to do a lot of researches and also discuss with other developers. We solved this problem together by checking if the turned data is totally loaded and if it was, we loaded the UI.

Wednesday 06.02.2019

Today I participated in our scrum meeting - a daily group meeting to help team members update their status. I had to demo the application with the product owner and other developers. We had a three-hour meeting. Each person talked about what he did last week, what's in progress at the present and plans in the near future.

I talked about my project and its current situation. After the meeting, I showed the product owner the current version of the application. It was a successful demo because the application was working well with what it had at that time.

After the meeting, I continued with my development tasks. I continued with the dashboard functions. There were some smaller functions that I spent the rest of the afternoon to finished. When I was in the process of finishing the dashboard page. Suddenly some changes that have been required by the customer and the product owner for dashboard pages related to functions. They wanted me to change partly of the UI and add one more function to the page. Besides that, a function that I have just finished had to be removed from the page. Then I had to call directly to the client and discuss with him more about the changes.

As a consulting company, changes are not avoidable during the development process. However, that also bring a good experience for me in term of communicating with clients.

Thursday 07.02.2019

After the agreements were approved, I continued to work with the dashboard page, changing it according to customers' opinions. This also took a lot of time to implement. After that, I returned to the route. For any web application, there must be pages or parts of the application that you do not want users who have not logged in to have access to. Although Vue Router does not provide us with a specific function to be able to do this job easily, by using a combination of basic features we can fully implement this checking function. In this function, I created an object and that object includes: a state called *isAuthenticated* to save the state of checking whether the user is logged in or not, a function *authenticate* to set for the state of the logged-in state, and a function *sign-out* to reset the status to un-logged.

To implement this protecting route for the application, I used 'Global Before Guards' and it is described in figure 5.

```
const router = new VueRouter({ ... })

router.beforeEach((to, from, next) => {
  // ...
})
```

Figure 5: Global before guard in Vuejs

This is a method recommended by official Vuejs document.

Friday 08.02.2019

After the dashboard page is completed, I switched to working on the user's personal page. Here, I have to go back to develop more database. In this page, I had to use API from a third-party company, which was a very tricky task. I had spent hours on researching and trying but could not get it work. I finally came to a senior developer and ask for help. This time, he knew this is an advance topic for me, therefore, instead of giving me short instructions, he decided to do pair - programming with me. Meaning that while one is typing the code, the other person just looks. There are many approaches as well as best practices for how to get the best results from this cooperation.

Pair-programming is a great combination to increase the learning speed for me - a junior developer. There can be open mind and ask questions about many things that only senior level can achieve. This can be a great way to share experiences across the team.

Thanks to pair-programming, we finished the task quickly and after that I realized that I learned much faster when I could code with senior, I understand more about the code, I could save time and money for the company.

Week 3 analysis

In contrast to the second week, the third week I met quite a lot of difficulties at work. There are many new technologies that I totally do not know. as well as errors encountered when writing code.

However, despite many difficulties, thanks to the help of my colleagues, I still got the job done and gather more knowledge in programming from them. They showed me how to write code that only people with long experience can make. They even did pair-programming with me to teach me new technologies. As we talked more, we understand each other more, which made me feel very grateful and felt that I was very lucky to work with extraordinary developers like them.

After working 3 weeks with this project, I feel I understand the project more and more and I am confident that I will complete it very well.

3.4 Observation week 04

Monday 11.02.2019

My goal today was to finalize everything related to the routes, authentication and protection for pages from the users who are not logged in.

I could say that, if I had taken the position of a developer in week 3, my goal today would be to become a tester. In the development phase: the point of view is to focus on building functions. In other words, developers are the ones who think about how to solve the requirements that customers have made. But for testers, they apply all their creativity to checking functions with scripts and cases that make the application generate errors.

Therefore, today, was the one who tried to make the application crash. I spent 1 2 hours to test the endpoints related to authentication in postman. This includes retrieving the current token and specifying the user information who was just logged in.

If the API side needs testing, the UI side is similar. I've tested authentication-related functions in many different ways.

- I tried logging in with a random account that was not saved in the database.
- I tried logging in with a valid account to see which page it led me to. and check if the pages are displayed
- I try to access pages when I haven't logged in to see if they have been protected
- Once I have logged in, I try to go to the login page to make sure it's not there anymore and instead the browser would redirect me to the current page.

During the testing. I found some bugs in both back-end and front-end and I spent the whole day fixing those bugs and testing again. At the end of the day. I achieved my goal that was completing the authentication and the routes for login page and some other protected pages. Everything seemed to work well without any other bugs.

Tuesday 12.02.2019

After finishing the authentication, my goal today was to create a new page for user. It's the place where user can see his or her own data and can make changes related to their account and personal information.

Creating components, classes, defining functionalities didn't take so much time. Everything went in a certain form like the way I handled the previous components and functions. So, everything was quite simple in terms of the application's architecture. All I need to change were the functionalities inside the component. Before writing those functionalities, I had to go back again to Postman to do the testing for all API's endpoints that I would fetch and display in the UI.

After the test has finished and everything looked good. I continued working with the UI. I still chose to use Element-UI for rendering UI's template because It was easy to use and understand. I can just go to Element-UI website and find the component that I want to use. It can be a small button, or it can be the whole page template. All I need to do is just import it into my application and use it there. Thanks to Element-UI, I made the UI for the application very quickly which meant I had more time focus on the functionalities that displayed information from the database.

I was not able to complete the user's personal page today. But there were not so many functionalities left and that would be my goal for tomorrow.

Wednesday 13.02.2019

Besides finalizing the user's personal page, I also wanted to work a bit in the laundry application today as the client reported that they found some strange bugs when using the

application. And they also wanted to improve the performance of it because it's been getting slowly recently.

There was not so much on left the personal page. I tried to quickly do all the remaining tasks including the functions to help users change their passwords, personal information such as name, age, address. Then I made the responsive to make sure that users could view their profile and edit information even when they use the software with a mobile phone or tablet. I also took a moment to check if everything was working and there were no bugs.

After finalizing the user's personal page. I worked on the laundry booking application. It was made with jQuery – a JavaScript library which is quite old and different to what I have been working with over the last 3 weeks. The problem the client mentioned was about some bugs in the performance of the application. Because it was a completed application and it was launched a long time ago. Many users are using it and the more users use it, the slower it becomes. What I have to do is fix those small bugs and improve the performance of the software. Fixing those small bugs was not a big problem but I did struggle with the slowness of it. I had to do a lot of research on this problem but unluckily I could not solve it today.

Thursday 14.02.2019

Yesterday's problem with the laundry app had not been solved yet. Therefore, today's goal was to solve that bugs and deploy the new version of the software. After that, if there was no problem with it, I would go back to the medical app that I work with every day.

I tried to do more researches and watch some YouTube tutorial videos about how to increase the performance of an application that's using real DOM (Document object model). Thanks to that, I found some interesting solutions that finally helped me to make the software run much faster. I finalized it and deployed for the client to test and review.

Back to the medical application, it's time to work on the most important page of the application, where users can see the important analysis about their health that they have bought. This is also a complicated page with many small components and functionalities inside. Before starting to implement it, I had a discussion with the product owner and other developers to make sure I have understood everything correctly about what I need to do with this page. When everything was clear. I made a plan of tasks for this page and I would follow my plan to finalize this page before the second demo.

Friday 15.02.2019

As the plan for work flow of this page was made yesterday. My goal today was creating the UI for the whole page with Element-UI and testing all related API's endpoint which would be used in this page.

By the end of the day, I was able to complete the UI, but the API arose some problems that I need more time to investigate and fix.

Week 4 analysis

This week I had a completely new experience, that was to become a software tester. I have learnt that in order to be a good tester, I must always put myself in the position of the end user. People who want to get the perfect product and the least error. The motivation for me to find the bugs will be more. I must have an extensive view of the product so I can find bugs or holes that will appear bug. It was also way to help me increase my work efficiency in the best way.

When I tried to make the application crash to find bugs, I have started to understand that detecting an error is not easy. If they are too easy to detect, they have been fixed when I implemented those functionalities or by other programmers who wrote those codes. So, I need to be careful in testing to avoid ignoring possible bugs. Once I find the error, I should analyze it in detail because of the bugs, it may generate more bugs. And finally, I need to be persistent with work, and try not to be discouraged if cannot find the error. Certainly, during programming, there will be bugs that cannot be completed as long as you are persistent enough to find it.

Week 4 has passed quite well with another library: Query and especially I have gathered new knowledge about the difference between applications using real DOM and virtual DOM. DOM (Document object model) is an Abstraction of an HTML page structure. It retrieves HTML elements and repacks in an object in a tree-like structure. It also maintains links between parent and child HTML elements that are being nested, provide the API for us to go through nodes (HTML elements) and control them with several ways such as adding nodes, removing nodes, updating node content, etc. (Bartosz Krajka 2015) Figure 6 indicates the HTML DOM model as a tree of object. Virtual DOM is a lightweight DOM Abstraction. You can treat it as a copy of the DOM, which updates do not affect the real DOM. It has all the same properties as the real DOM object, but it doesn't have the ability to write on the screen like the real DOM. Virtual DOM is efficient and speeds up from the fact that it's lightweight. In fact, the new Virtual DOM will be created after rendering again.

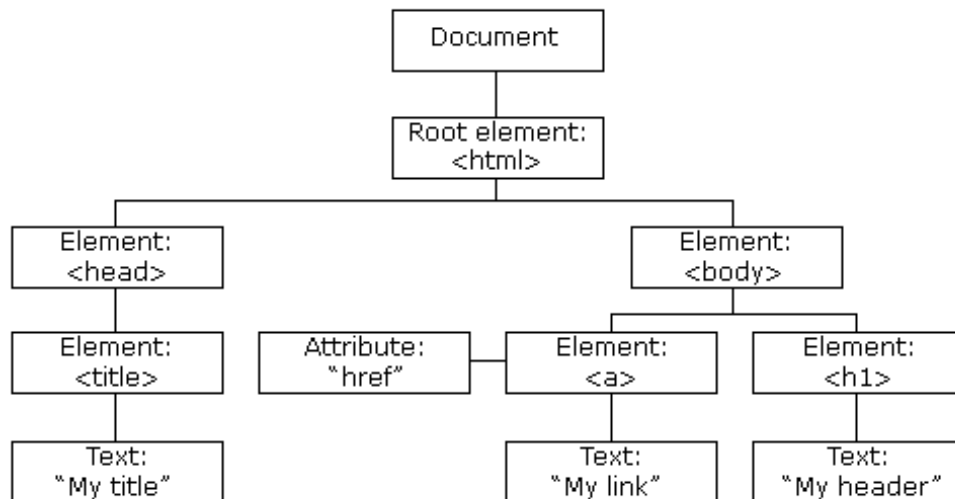


Figure 6: DOM tree structure

By doing researches about how to improve the jQuery application's performance. I have understood how new technologies like Vuejs and Reactjs surpassed the old technology like jQuery.

3.5 Observation week 05

Monday 18.02.2019

My goal today was to correct API errors to be able to display and process information related to that on the UI.

There were a few problems with the database including

- UI cannot read nested objects, which leads to software crash
- Send incorrect commands to the server
- There is no validation for forms

It took me a long time to fix all these errors. To make sure that on the UI, the user or admin did not miss any important information to fill out. Validation is required. I started by writing the validation functionalities for forms on the UI, there were different forms that are used in the interface I am working on. Then I also had to write validation for the frontend to display error message and inform users that there were some required fields that they missed to fill in.

After completing the validation. I continued with the nested object. This meant that some conditions had to be checked before an object was called and displayed. By the end of the day, I could display data and process information with data on the UI and fixed all bugs in the backend related to this view.

Tuesday 19.02.2019

All the most important and necessary jobs for the homepage have been completed. So, today's goal was to make everything perfect and test them before deploying and make them ready for the next demo.

The first change needed to be done was the replacement of icons instead of texts. This was quite easy because icons that needed for the application were all available on the library Element-UI, they were just needed to be import and could be used anywhere in the application.

The second task was more difficult, it was about making the pagination for the page. Because the number of items on this page could be up to hundreds of thousands of items. So, the customer wanted that each view to only display 10 items and items to be divided into many such pages. For this pagination, I decided to use an open source library called vue-paginate. There were several configurations that need to be modified but I could finalize them and test them all by the end of the day.

Wednesday 20.02.2019

After having completed the most important interface of the software. I was asked to create a new page displaying welcome messages and instructions for users who first logged in. This is not a difficult task because it is merely an interface created by html and SCSS. But it will take a lot of time because I have to make it look exactly the same from the partner company's website. In other words, I have to clone this interface from another available interface, therefore; my goal today is to make the welcome UI look good at least on web browser.

HTML and SCSS were my strength because I learned them very well since the day I went to college. Although creating this new interface takes a lot of time because this is a page with lots of content. Includes introduction of software, user guides, information related to services, contacts, companies. But I did not encounter any difficulties in the process of implementing it.

I finished the UI for the browser but had not made it responsive. Because this would be the first page users see when they first used the software and according to the survey from the client company. Customers tended to use mobile phones to access software, so this interface needed to be displayed well on different devices especially on mobile phone screen. Moreover, there were many users use internet explorer, some elements in SCSS

were not supported by that browser, so I had to make sure everything works even on the oldest browser.

Thursday 21.02.2019

The goal for day was to make responsive for the welcome page. I decided to write a total of 4 screen sizes for this interface, including computers, tablets, big and small smart phones. Here, I realized that a rule called 'mobile first' must always be followed, which mean, whenever a website is made, programmers must prioritize the phone interface first, but I have made the interface for desktop first. I quickly fixed it and I also find another way to make the responsive faster and save more codes.

In order to do that I would used flexible combination of techniques including flexible grid, responsive image and SCSS media query. When users switched from their laptop to tablets or phones, the site automatically switched to fit the screen size and processing scenario. In other words, I made technology that automatically responds to the user's device. By that way, I have completed different views in different devices for the welcome page. Not stopping there, I have to check if it works well in all kinds of web browsers.

I have tested the UI on different browser including safari, google chrome and Firefox, everything was perfect, but I still was not able to check it on internet explorer because in my company nobody uses windows operating system to help me test it.

Friday 22.02.2019

Everything has gone very well so far. And today was my last day at the company. The software has worked relatively well and only needs a few small developments then it can be launched. Since today was my last working day, most of my work was just writing documents related to software installation, taking note of the login information for the database and the UI.

Week 5 analysis

This week was an easy week of work, One of the things that I learned was to always prioritize the mobile phone view. Because nowadays, smart mobile devices exploded and developed. Consumers who have been using their smartphone device to access the website have a higher proportion, compared to the proportion of people accessing with the desktop computer. Getting a small screen from mobile devices as a web design base will help

the website to have a high level of adaptability and better interaction with users. That's why Mobile First needs to be concerned and prioritized in current web design trends.

Besides, I also learned how to write validation for forms in the most optimal way. Because form validation is important for a secure website as well as its availability. The validation process evaluates whether the input values are properly formatted before sending it. For example, if we have an input field for an email address, the value must have a valid email address; It should start with a character or a number, followed by a symbol @, then end with a domain name. And it must be written in both the backend and the frontend.

Front-end validation in forms is able to highlight wrong user input and disable the submit button. This is useful because it gives the user some immediate messages, alert or feedbacks when user fill some values in the form. However, Front-end validation does not prevent the user from submitting a form with incorrect values.

Meanwhile, in backend validation. It checks the sent values with validators and refuses to submit forms if the value does not match the validator. There will be no messages to inform users about incorrect values in the UI. Developers will have to use JavaScript logic to implement it on the UI.

After nearly 10 months working at the company, I have learned a lot in general. However, I always wanted to learn new languages and libraries and want to try different working environment to know what I really fit into. So, I decided to join another company where I can work with most popular programming languages and JavaScript framework.

3.6 Observation week 06

Monday 11.03.2019

Every technology at the new company is completely different from what I have worked with in my previous company. As the company provided me with a new laptop and keyboard set so Today's goal is to complete the computer installation, the programming environment of all projects. A lot of people came to help me with the installation from hardware to software. Hardware installation was quite simple because I only need 1 laptop with 1 monitor. but installing the software was quite complicated.

I started by downloading all work-related software and packages including: Visual Studio, Node, NPM, Brew, Dropbox and set up web applications that support my work include: assembla - a tool tracking development issues and processes, Storybook – a development tool to track UI components.

I spent a lot of time setting up the working environment for my project. Everything was specifically instructed in the guide attached to the project. But it was still very difficult for me to install the project myself without any help from my colleagues. There were many new commands that I have never used before to run the project. And my colleagues had to explain me step by step to make sure I could run the project successfully. I seem to have spent the entire morning on the installation. After that, we had a small meeting between everyone in our team to report about our work situation.

Everything went smoothly, my supervisor then explained more to me about the project, such as its purpose, functionalities, and how it works. The best way to get used to the environment is to start developing it. Therefore; he started to give me some small tasks so I could get acquainted with the project.

By the end of the day, I finished all the installation and I started to play around with the project to get used to the architecture of it.

Tuesday 12.03.2019

My goal today was to complete half of assigned tasks. To become more familiar with the structure and programming styles of the project. I carefully checked it from how it was splitted by folder, data, files to syntax that is used throughout the project. Besides, I also had to inspect the software in the web browser. This was done with the google chrome - react developer tool and another tool that every developer uses which was the inspect element tool in any chrome browser.

I searched for the component that I was asked to change in the assigned tasks. I thought just a few small changes to the SCSS would be enough to complete these tasks but everything got more complicated than I thought when I actually implemented the codes. The first difficulty I encountered was working with data. What I needed to do is to fix the interface according to the client's request and I needed to retrieve the data from the backend. However, the backend is written by a language I have never used before and never learned before. This made my job become difficult. I had to sit down with two colleagues to come up with a solution to this problem.

We decided to separate the backend and the frontend to make things easier. Before that, parts of the UI were render from the backend it was not a good idea making like that. After that, I could change the UI very easily. I quickly finished the first tasks and moved to the second one. I started by inspecting the component related to this task. That was a hard

job. because it required me to know how to customize the UI library that we are using for the project. A lot of customization were needed there. I spend a lot of time researching that library and how my colleagues customized it before.

By the end of the day, I was able to finish 1 task completely and in the completion phase of task 2.

Wednesday 13.03.2019

My goal today was to complete the assigned tasks and make them ready for the code review and testing.

I have gradually got used to the new working environment. Since this product has been launched, every change needed to be perfect without any minor errors before it is launched next time as an updated version. Therefore; I had to be very careful in customizing the UI library to avoid it being overridden on the wrong components. The codebase was written very neatly, and each component is reused in many places. Sometimes, I tried to change the SCSS of one element but the styles that I changed also affected other elements which made me really confused and I had to try many other ways to apply those SCSS styles to it.

I have used all day to test out the approaches to this issue. And I found the best way which was passing a property for the reusable component and use it in each specific element that I want to change. At the end of the working day, I finished all tasks and pushed all the changes to GitLab for the review session.

I was quite excited to be reviewed because I had never done this before. I thought that when a senior developer reviewed my codes and made comment on it, I would learn a lot and would improve very quickly. While waiting for the review, I was given a new task which was bigger and harder to start investigating and prepare for the code implementation tomorrow.

Thursday 14.03.2019

A senior developer in my team helped me review my codes. Generally, it was good, and he made some details comments on what I need to improve and fix such as naming variable, functionalities, importing file methods. After modifying changes based on his comments, I can merge my parts to master branch and get them ready for testing. My goal today is to correct the codes based on my colleague' comments and then merge them into the master branch for testing. After that I will start with the new task.

Editing codes did not take too much time, as it was simply just naming variables, and writing reusable styles. I quickly merge my work and switch to the new task. The new task was quite difficult because I had to change the interface for a very big component and that component not only contained a lot of sub-interfaces, but it was also used in many places. That made development takes a lot of time and effort. Not only did it stop there, it required a very precise solution so that it did not affect other important interfaces. Fearing that my approach to the problem would not be the best way, I discussed with colleagues who had more experienced.

Now that testing for my previous task was done, they reported that everything worked fine but there was a small error with my styling, which was one of my styling override another component and I had to fix it. This brought me a new issue and I again spent a lot of time solving it. At the same time, my colleague introduced by a new tool that help me to find and inspect component easier. It was Storybook. Thanks to that, I solved the error but because of that, I could not finish the new task today.

Friday 14.03.2019

My goal today was to complete the task that's still unfinished yesterday. I started fixing the parent component first, then went looking for all the places where it was called and modified the child component of it.

It took me the whole day to finish this task because there were about 10 different components using that component that needed to be changed ad modified.

Week 6 analysis

The first week of working at the new company took place quite well and in a short time I have learned a lot from the members of my team.

Today when developing a project, we spend a lot of time building UI Components. Above all, we needed to focus on understanding the properties of the components as well as showing all the props and states of these components. Right from the starting point, it was undeniable that difficult in the frontend of a project. Based on the design I could complete a component, or even complete a page. But I'm not sure if the quality of the completed code covers all cases? Therefore, one of the things I learned and found most useful was the Storybook.

Storybook was an environment where I could easily develop and test the UI for my application. With it, I could test UI with virtual states and tested how the UI would work. Story-

book ran independently of the app and did not need to worry about dependencies and requirements. It provided me with an add-on repository as well as an API so that I could match the UI best. Currently I saw that it was supporting React, React Native, Vue and Angular.

The second thing I felt very excited about was the code review. Code review was one of the methods to keep code clean. Reviewing codes for each other would discover many things such as: codes in this place could be written shorter and get higher performance, or the other place could apply a specific pattern design, or function A, module B was being repeated code too much all the problems that everyone could make, including experienced programmers but it could not escape under a dozen eyes of the development team members. It helped a developer to improve programming skills. A developer could learn the beauty of others' codes and others could point out your bad points. In any case, his or her skills would increase dramatically.

3.7 Observation week 07

Monday 18.03.2019

After changing all the week 6 task 's related interfaces, I moved on to the next part of the task, which is working with the routers behavior as the customer's request. My goal today is to complete this whole task and make them all ready for the 2nd code review.

Because I myself have not been so familiar with the structure of the project. So, it took the time for me to do researches about the best approach to the router problem. I searched all the files that used the router written earlier by other members of my team and learnt from those lines of code. After I have implemented the routers for this task, I started testing all the pages associated with it. In some cases, it has failed and I had to spend a little more time writing some more conditions to make sure it worked well in all cases.

I finished the job and before pushing all changes to GitLab, I pulled all the changes from other developers to make sure there were no conflicts. As I expected, there were many changes which had been merged into master branch from other team members before, so when I pulled the codes from there, I met some conflicts needed to be fixed. Fixing the conflict is not too difficult, but it requires me to be very careful in reading the changes that come from me and other programmers to retain the right part and eliminate the wrong part. At the end of the day, I was able to fix all the conflicts and push my codes back to the master branch for the code review tomorrow.

Tuesday 19.03.2019

While waiting for the code review, I moved to another task. That was a new feature and we have not yet written the API for it, my plan was to use a fake data to generate the UI and then, when my colleagues finished the backend, I would replace the fake data with the real one. My goal for today is to finish the UI for the new task and fix the codes if there's something wrong after the code review.

I continued to use the UI component library for this task. it was about making a nested list and it required me to use logical thinking a lot to handle nested objects. It took me a while to think, try and implement the solution, however, I was able to complete it because I have worked with the subject quite often. After I completed that job, it was also time for my colleagues to complete the review code for me.

I returned to the old task because I found that my colleague had completed the code review and he left some comments on my codes for me so I could change them and merge into the master branch after that. There were a few issues with UI testing that made my code get errors, but with the help of my colleagues, I have edited them and some other modifications have been made to make the codes neat and easy to maintain.

Wednesday 20.03.2019

I went back to working with nested objects and nested lists today. And my goal today was to complete it as much as possible. Because this work included a lot of sub-task inside. Besides, today was also my first sprint meeting with the customer. The meeting took a long time because we discussed a lot about the project's upcoming tasks needed to be done and make plans in the near and far future.

After meeting with customers, my team had to sit down together to discuss about the customer 's new requirements and the best approaches to implement them. We all agreed that there were some unreasonable requirements, and they related to the work I was doing, so to be sure we would implement the best features for the project, our project manager would have to talk to the customer again.

After the meetings, I returned to what I had been doing before, I met a few difficulties with the strange behaviors of some functionalities that I could not finalize the task today.

Thursday 21.03.2019

I continued my work from yesterday with the nested list. My goal for today is to complete this task and get it ready for the next code review and deployment.

Because the work has been completed to 80%, I quickly finished it. However, as soon as I finished it, the customer wanted to change its interface again. In other words, I would have to modify the behavior of the previous functionalities as well as the interface of them. Moreover, our tester found some minor errors about styles from previous tasks I have done.

The reason for these errors is because I didn't understand all the interfaces and functionalities of the software. So, I have overwritten the parts that I didn't see. I had to ask our application specialist for help in this problem. He explained all the functionalities and the existing interface of the software for me and guided me on how to enter each interface as an administrator and a normal user. After that, I could fix all the bugs that our tester report and push them back to master branch.

Before making changes that customers requested, we had an internal discussion about changes. We all thought that the request of customers is not reasonable in terms of user experience. It will make the application confusing to new users and users who have little knowledge of zoology. Therefore, these changes have been delayed waiting until tomorrow's meeting with the clients.

Friday 22.03.2019

Today I had another meeting with the client and discuss about the filter for the nested list, I was asked to make another interface for the nested list for customers to choose and to make it easier for us to convince them to accept our idea. By comparing the pros and cons of each interface, customers would have a more general view on them. I finished the other new interface before noon, it required a lot of logical thinking to complete it but I made it work and prepared to show it to the customer before the meeting. By my many reasons, my colleague was able to completely convince customers to choose the interface that we wanted to implement.

Besides, we also discussed another interface that I was working on week 6. More specifically, it was a navigation and was used in some pages in the software. Customers wanted me to keep what I did but there were 2 specific pages they didn't want to use it. Instead of that, they wanted a completely new interface for the navigation of these 2 pages.

After customers gave conclusions to all the issues that we were still wondering, I returned to work with the decisions that were made about the interfaces. By the end of the day, I finalized the nested list and its filter with new changes and requirements from customer.

Week 07 analysis

In the past week, I have experienced the first print planning meeting and I have learnt many things from that. As I understood, in the method of Scrum, sprint planning meeting is the meeting between product owner, Product manager, developers, designers,... In the sprint planning meeting, the product owner described the team with the highest priority features. Everyone on the team asked questions to understand the problem, then we transfer user stories of product backlog into more detailed tasks in the sprint backlog. Figure 7 describes a sprint planning meeting.

There are two things that result from each sprint planning meeting:

- The goal of sprint
- Sprint backlog

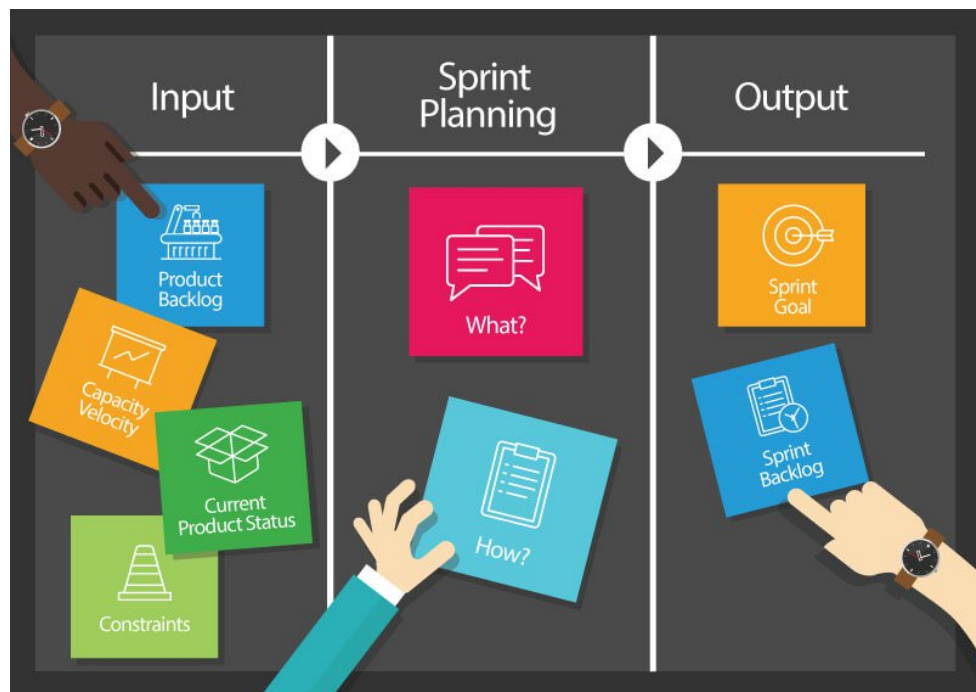


Figure 7: Sprint planning

The goal of sprint is describing what the team is trying to achieve in that sprint. Sprint backlog is the result of planning for sprint. A sprint backlog is a list of items in the product backlog that the team has committed to complete plus a list of tasks to do to complete those items. Each task in the backlog sprint is also used to estimate the workload.

3.8 Observation week 08

Monday 25.03.2019

In week 7, I completed the nested list and my goal for today is to deploy it for testing. I also want to complete another issue related to a new navigation that the clients have recently wanted to have.

After pushing all changes to GitLab for the code review, I started with the new navigation. I created the navigation with material UI and as it was a totally new UI, I also had to write UI testing for it. I quickly finished this task and push changes back to GitLab, while waiting for the next review, I had to fix some bug related to the nested list as the review was done and my colleague had left some comment for me. There were some small variable names that were too general when I put them in the component's context. My colleague realized and he wanted me to change those variable name as well as clean the codes. After that, I merged the list to master branch and they were available for testing.

Regarding to the navigation, after the review was done, there were some errors in the UI testing that made me confused, I spent some time trying and running the code again and again but could not fix the auto testing bugs. I had to ask my team mates and luckily, they understood very quickly my problem and helped me with that.

Tuesday 26.03.2019

Today I ran a strange problem when pushing codes to GitLab, there were some style guide warning that made my codes could not be merged into master branch. I spent almost the whole morning but could not get it work and made it ready for the review. One of my colleagues also helped me to investigate and after many failed attempts, he figured out that because of the new version for the library we were using has been update last night by the library's author, our library's version was outdated and it led to the errors. He helped me to fix that error and did the code review for me.

My colleagues made some comments on my codes in the code review. That was a big task, therefore, there were many changes needed to be reviewed. Besides some variables' names needed to be changed, as the file was too big, I was suggested to split them into smaller parts according to the review's result. I concentrated on fixing the codes based on the review's comments, splitting components was a tough task because it took time and I also had to test again and again to make sure the UI still worked well. After finishing this task and having the daily meeting with the project manager and other team member, I was assigned a new task and it seemed to be a tricky one.

By the end of the day, I was able to get the nested list to be merged into master branch and got myself familiar with the new task.

Wednesday 27.03.2019

In the morning, I worked on the new task, it was about the validation in different forms in the application. I thought it was an easy task but the more I investigated in the task, the more I found it difficult and it might require a lot of time to be done. There were many input fields in the application and just some of them had the validation, my responsibilities were to make the validation for all kind of user's input including uploading images, data files, integer, float number inputs. I was able to make changes for one of input groups by noon and today we also have a sprint planning meeting with our client.

In the meeting, I showed them what I have done for the next production deployment and after that, we realized that we misunderstood each other several days ago about the nested list I made. They wanted it to be done in another way and once again I have to do it again after at least 2 times. This time, I tried to quickly fix it during the meeting and showed them the new UI for it, I made it very fast that they could see changes in just about 1 hour and this time, we could not misunderstand each other anymore. We promised to our client that we would make the production deployment within this week, therefore, I had to deploy this UI to development environment as fast as possible for testers can test it through before Friday.

I finished the new UI for the nested list and went back to the validation task, it was getting more and more difficult because a problem arose when I tried to test if it worked in the browser. Some of the input types are not used yet in the UI, but I still had to get them work for the future using purposes. Even though I tried so hard, but I was just able to complete 2 more input groups.

Thursday 28.03.2019

I continued with the validation, today, I got some instructions on how to see and test them all from my colleagues. They guided me to use admin panel to display what I needed to test on browser. By that way, I could test what I have implemented with the validation very fast, and I could so try with many testing cases including mandatory fields, ratio fields, selection fields, ... Besides, I also got feedbacks for the nested list from our testers. They wanted me to fix a browser behavior about the scroll view that looked not good in term of user experience, otherwise everything worked perfectly. This scroll problem is one of the trickiest tasks that I had have to solve. It required me to have a good knowledge about plain JavaScript, real DOM and virtual DOM relationship.

I spent the whole afternoon just to solve this scroll bar problem, I tried many ways and read many articles about react ref, how to access real DOM when working with virtual DOM and how *forwardRef* in react worked. That was a tough task, but I have gained new experience today.

Friday 29.03.2019

My goal for today was to finalize everything needed to be in the next production update. Including the nested list, the navigation and the main menu bar. I worked very hard on them as there were many small issues arose after the testing.

By the end of the day, I was able to deploy everything to staging environment where we use to do the final integration testing.

Week 8 analysis

In this week, I learned 2 new things, they were using 'ref' in React and integration test. For me, who have just learnt React and started to work with it for 3 weeks and for those who just came into contact with React, there must be quite difficult and complex concepts and syntax, not easy to access, including "refs", figure 8 present how a 'refs' in React was actually used. According to React's documentation, refs were used to get references to a DOM node (Document Object Model) or could be an instance of a component in a React application. i.e. refs would return to a node that we refer to.

```
1 | class MyComponent extends React.Component {
2 |   constructor(props) {
3 |     super(props);
4 |
5 |     this.myRef = React.createRef();
6 |   }
7 |
8 |   render() {
9 |     return <div ref={this.myRef} />;
10 |   }
11 | }
```

Figure 8: Using ref in React

Usually, React code would rarely have to touch refs, but if you needed to interact directly with the API of the Element DOM, especially if you used React with traditional JS DOM li-

braries the use of refs was inevitable. In general, you should still avoid using refs a lot, except in case of force major because the possibility of this API might also continue to be updated.

Besides, I could understand how integration testing was different from unit testing as many errors that I could not see were detected by testers after the integration testing. Integration testing is defined as a type of test performed in an environment that has integrated individual units of a software to verify the behavior of the complete system. This is an experiment conducted on a fully integrated system to evaluate the compliance of the system with its specific requirements.

In the integration testing, the desired status would be checked, for example, can users log in by clicking on the login button after entering a valid value name and password? Welcome message appears after successful login? Error message will appear when login is invalid? Are there any website cookies stored for the login field? Can an unsigned user log in successfully? Is there any forgotten password link for users who have forgotten their password? I have done unit testing before and I thought it was enough for a software, but after I knew about integration tests, I figured out that both of them are necessary and if we want to make a high quantity application, we should use both kinds of mentioned testing methods.

3.9 Observation week 09

Monday 01.04.2019

Today I ran out of tasks, so I helped my colleague with his task because there were two more bugs needed to be fixed for the upcoming production update and he has not completed them yet. Therefore, he shared some of those bugs for me to fix. My goal for today was fixing all assigned bugs and get them ready for the production update.

There was a problem about the material UI library that we are using. It did not support the features that we needed and made an important functionality broken because of its default styling solution. I spent a lot of time on this problem and could not fix it totally and my colleague couldn't solve it either. Then I realized that it was impossible to apply material UI default theme with our functionality, we had to choose one: keeping material UI them and rewrite the function or keeping the function and remove the material UI part then rewrite the styles in a normal way. And we chose the second solution. After that I removed default themes from material UI from the component creating bugs and wrote a new CSS for it. However, there was another problem with the CSS as some of inline CSS elements could not be implemented with material UI components and in this situation, it was about the

media queries. In other words, I styled the component with inline CSS for it to be consistent with other components, but the style could not be specified for small devices. Therefore, I had to get rid of the media query element in this component and luckily it did not make the component look bad.

After fixing all bugs related to the conflicts between material UI and our implementation, I went through the application to make sure what I made didn't break any other component as it was a reusable component and used in many places in the application. Luckily it did not break anywhere and I could push all changes to GitLab for the code review. Everything seemed to work perfectly and they were all ready for the deployment.

Tuesday 02.04.2019

All new features for the coming production update were ready to be deployed so I once again ran out of programming task. Instead, I started to investigate in new features coming after this deployment and did the estimation as well as the implementation plans for them.

After doing the estimation for tasks, I started to work with the components in Storybook and waited for the databases from backends. As the clients already made the prototyping for the new UI and many of small components were reusable and what I needed to do is finding components suitable for the new UI and create the new UI with them. Databases from the backend was not ready yet so I just played around with those components and tried to create fake information to display on the UI.

Wednesday 03.04.2019

My goal for today was to complete the new UI for organization panel with fake data. There were a lot of things need to be done to create this page. I needed to split this page into at least 6 different small components and wrote testing for each of them.

I started to find reusable components that I could use for this new UI and tried to implement the codes following the project's architecture. According to the architecture, each component had to have a separate UI testing and when they were combined, we could see how the application looked like in Storybook without running the localhost. In each component, I also created a fake data object to see how it look like with information. There were 2 components that I could reuse in the page but most of components needed for the page had to be created from scratch. It took me quite long time to create folders, writing codes for new components, making the UI testing and customize styles from material UI library.

In the afternoon, after my colleagues had a sprint planning meeting with the clients and showed them new features, we were about to update in the coming production deployment.

One of the features was about the nested list and this time, they once again wanted to change it. Before that they had have ideas about how it should be implemented and after I completed the implementation, they changed their mind and I have to remake the nested list. The nested list is a big feature and this time, in order to remake it, I had to delete many logical functionalities and write new ones as the new list they wanted then is totally different from the old one. I committed all changes related to what I was doing for the organization's panel and switched to the nested list again because it was urgent to be completed for the deployment.

By the end of the day, I was able for complete the new UI for the nested list, but I could not apply any logics on it. And I also need more time to investigate how should I implement the logic on it as it was a difficult task.

Thursday 04.04.2019

Today I continued with the nested list. At first It was tricky because I used material UI for writing components and some theme styles from material UI could not be overwritten which made the list looked strange. Then I had to change my approach of implementing it. Instead of trying to customize material UI styles, I created CSS styles matched well with default material UI. I finished the new list quite soon after changing the approach and pushed changes to GitLab for the code review. The code review took place very fast as there was no problem with the codes. I could merge changes to master branch for our tester to have a look before being deploy to the production

I was very careful with the new list and I tested it myself during every time I made some changes. Therefore, testers could not find any bugs and after that the list was ready for being deployed. By the end of the day, we were able to update new version for the application. And now, it's time to start with another production update.

Friday 03.05.2019

Production was updated successfully by our DevOps yesterday, and today I was assigned a new task. According to our client's requirement, it needed to be done and deployed to the production by next Monday with some other small changes. Therefore, my goal for today was completing the new task.

Th task was about changing logics of an existing feature to make sure the feature would have 2 different UI and the UI for each logic is determined by the data length used. That was a tricky task because It was the combination between react and formik – a package to manage forms. There was some problem with the default settings from formik and they could not be integrated with react by the ways that I our client wanted it to be. Therefore, I

spent sometimes to read documents and instructions on the internet to find solutions for these problems. One of these problems was I could not change value of input field with formik and I found a solution on stackoverflow.com that actually help me to get my codes work. Figure 9 indicates how formik change value of text input in a react selection field

```
<Select
  {...field}
  onChange={(value) => setFieldValue('fruitName', value)}
  onBlur={() => setFieldTouched('fruitName', true)}
  value={values.fruitName}
  ...
>
...
</Select>
```

Figure 9: Changing value of react-select in formik

After spending a significant amount of time on this task, I could finally finish it and once again sent changes to the code review session and hopefully I could deploy these changes to production on Monday.

Week 09 analysis

In week 9, I was more familiar with the new job as well as the project that I has been working on. In this week, there were 2 main difficulties that I have faced to: one of them was the customization with material UI library and the other one was about the integration between formik and react-select. In term of material UI customization, it gives us several ways to get customization including Individual override component styles via the style prop. This is used when we need to override the default theme of Material-UI according to the styles of each project. In the figure 10, default theme was replaced with new colors. In other words, new colors replaced the default theme values of material UI.

```
1 | import React from 'react';
2 | import Button from '@material-ui/core/Button';
3 | import { createMuiTheme, MuiThemeProvider } from '@material-ui/core/styles';
4 |
5 | const theme = createMuiTheme({
6 |   palette: {
7 |     primary: { main: 'black' },
8 |     secondary: { main: 'pink' },
9 |   }
10 | });
11 |
12 | function paletteCustomize() {
13 |   return (
14 |     <MuiThemeProvider theme={theme}>
15 |       <Button color="primary">Primary</Button>
16 |       <Button color="secondary">Secondary</Button>
17 |     </MuiThemeProvider>
18 |   );
19 | }
20 |
21 | export default paletteCustomize;
```

Figure 10: Customize palette theme from material UI with new colors

Another way to override material UI theme is using *withStyle* or inline style. They were used when people want to customize single or several specific elements in their application without affecting other components. This customization method was tricky and people need to remember which classes the component has or whenever they want to change style of a specific component, they have to read related documentation from material UI official website. For example, component *Chip* in material UI had many classes that we could modify such as *root*, *colorPrimary*, *selectable*,...whereas, component *Card* only had class *root* that we can modify. Figure 11 shows how *Card* component was customizing with class *root* with *withStyle*.

```

1   const styles = {
2     root: {
3       background: 'pink',
4       padding: 20,
5     },
6   };
7
8   function SimpleCard(props) {
9     const { classes } = props;
10
11    return (
12      <Card classes={{root: classes.root}}>
13        <CardContent>
14          lorem istum
15        </CardContent>
16        <CardActions>
17          <Button size="small">Learn More</Button>
18        </CardActions>
19      </Card>
20    );
21  }
22
23  export default withStyles(styles)(SimpleCard);

```

Figure 11: Customize palette theme from material UI with new colors

Besides material UI customization. Another obstacle I mentioned before was about react-select and formik. Basically Formik helps us perform 3 frequently used tasks when manipulating the internal form Reactis:

- Put data in or get input data in form as usual props
- Check the validity of the data and display the message if the error
- Support sending forms to server

The value of the fields in our form, if previously stored in the state of that form component, now, when used Formik, the value of these forms will be managed by formik and passed into our component form as the props. The problem I met with formik was *Cannot read property 'type' of undefined" with react-select*. According to a stackoverflow's user. The problem was supposed to happen because of its calling *handleChange* with something different than native DOM event. And in this case, we need to use *setFieldValue* to handle value changes between react and formic.

3.10 Observation week 10

Monday 08.04.2019

Today, after getting permission to merge changes from week 9 to master branch, I take a new task while waiting for the testing. When I was reading the description of the new task, our tester reported that they found a bug related to my changes, so I had to fix it. I have tested very carefully what I have done but I was not able to understand the application and its functionalities as much as our application specialist did. However, the bug he found did not take so long time to fix so I quickly finished it. This time, there was no longer errors. Besides fixing the bug, I also made a quick change to some of existing icons in the application. In other words, our clients wanted us to change some icons in the application because they had come up with new icon designs. I finished everything needed for today's production update and went back to the new task.

The new task was about refactoring some duplicated functions used throughout the software. This one is quite tricky because the file was written by one of my colleagues and I found it hard to understand and implement with the component I was required to refactor. I have spent sometimes to read and understand how to use this file in other components. There was an example that my colleague used it in a component, but I thought it made the component even more complicated than it was then, so I was still confused and decided to leave the task for tomorrow.

Tuesday 09.04.2019

Today I have worked mostly on the codes refactoring. I read the reusable component that my colleague made carefully, and I used it to rewrite a component I had written before. Before the daily meeting, I was able to finalize it and sent changes to the review session. There was nothing needed to be changed after the code review, therefore; the new changes were merged to master branch and ready for the testing and deployment.

Like every day, we had a daily meeting at noon to report about our work process. Today, our project manager informed us about some minor bugs arose on the production updated. Most of them are related to the responsive design, in other words, our testers have tested the software very carefully in different devices, however, they did not have chance to test it on very narrow screens such as iPhone 4 and one of our clients using iPhone 4 reported to see some elements that looked bad on her narrow screen in term of UX design. Therefore, I spent the whole afternoon to fix those responsive bugs.

I understood what problems were about, so I focused more on mobile view of the software. I rewrote styles to make sure the software would be compatible with any kind of devices using *maxHeight* and *maxWidth*. With these two elements, the UI would take as much height and width as possible, which could solve the responsive problems mentioned before. Those changes were quickly taken into the review session and once again there was nothing needed to be modified so they went to the testing session after that.

Wednesday 10.04.2019

Today I worked on another bug reported by clients, which was about some CSS styles and the software responsive design for some icons. There were some icons in the same page were not aligned and did not look good in small browser, and a small functionality needed to be removed from several views and only kept on only 1 specific view. The two tasks were easy, but it took time to test the UI on different screen sizes to make sure the responsive bug would never arise again. After fixing those errors get got code changes merged to master branch, I had a discuss with my colleagues about my next task.

I was assigned to create a new reusable component for our next feature. It was about the terms of service that we could see in any website. According to the task's description, the terms of service component was required to be reusable and had to have some functionalities that interact with users. It could be used as a normal view or a pop-up modal in the application. Like any other components, I used Material UI to write this new one and used Storybook for the UI testing. By the end of the day, I was able to finish haft of the UI but there was still no functionality yet. I would continue tomorrow.

Thursday 11.04.2019

Today I worked on the unfinished UI from yesterday, as we didn't have the database for this part yet, so I worked mainly with the mock data and tested it on Storybook. I completed the task before noon and checked it with my colleagues to make sure what I have done met the requirements. After that, we had the daily meeting and we found out some issues related to narrow screen again. In Finnish language, there were many long words that take so much spaces on the UI, which made the UI look bad, and those long words were not able to automatically splited and started a new line. To solve this problem, I used *word-wrap: break-word* to make long words start a new line when it reaches the end of the page's width.

In the afternoon, some new database models for our new feature as well as mock data for its testing purposes were ready. At the same time, I was assigned a new task related to

those data. My task was about using the mock data to render the new UI's information on organization page I have made on week 9. I also needed to think about how I should organize the information and make sure they all look good on any kind of screens. By the end of the day, I was able to understand the data's structure and figured out how to bring them to the Storybook's UI.

Friday 12.04.2019

Today I mainly focused on the organization UI, I have handled 2 different types of data, the first one was a list of information needed to be displayed on the UI and we decided to display them in a form so that users can edit their information easily. Besides, there were also some information editable by users, but they needed to have permission from the application's admin as that information are related to some important credentials. By the end of the day, I was able to finalize the new UI for organization page and get it ready for the code review.

Week 10 analysis

This week was a relatively good working week for me, I have learned how to use a rather advanced concept in that React which was render props. According to react official document, The render props (also known as render call-backs) are a very popular technique to re-use code in React and are used extensively in libraries and large projects. However, there must be many people who don't know it yet, so I'll introduce it a bit. The render props technique using 1 props as the function returns React Element, and the component renders the return value of that function instead of directly rendering the React Element as usual. In simple words, Render Props is a trick that helps us reuse the logic of a Component for another Component. The render props are used as a props with a function value.

Moreover, I realized that I became familiar with the software so that it didn't take too much time for me to do research when I was assigned a new task. Therefore, my work this week did not encounter any serious difficulties.

4 Discussion

When I started my career in programming more than 1 year ago, it was difficult and frustrating. Sometimes I felt that my depth of knowledge was not enough to meet the job, not sure how people could write such large and complex applications. They made my job become very stressful. On the contrary, as time went on, I trained myself. With help from my colleagues, I felt I have been working more effectively, solving problems faster whether it was a new or old technology, I feel more comfortable at work. Looking into the past 10 weeks, I can see how myself developer as a software engineer and see the differences in 2 different working environments and 2 technology stacks I have used. In the start-up environment, I looked back the last 5 weeks I worked there to see how maturity I became after months of work and when all tasks were quite familiar with me. Besides, the first 5 weeks in the new company brought me the feeling of unfamiliarity, nervousness and difficulties when working in a professional environment with a completely new technologies and they were totally different from the previous ones that I had worked with.

Besides technical skills, I realized some approaches that made my work more effective. The first approach is the communication between team members. We conducted a fifteen-minute daily meeting every day to share our work progresses and to discuss about each other's difficulties. Furthermore, we help each other review codes to make sure the final codes version is clean and easy to understand before taking them to the production.

By writing this thesis, I saw my growth at work is considerable. I also realized the big difference in the process of working between a small start-up company and a company that has have 10 years of operation. In the start-up environment, I could write any block of codes on the condition that it worked well regardless of whether it was neat and easy to maintain. The testing was not highly recommended because we did not have much time to write test programs. Instead, manual testing was done but it could not cover all testing cases. However, in the bigger company, everything had to follow the process: Starting with writing codes, then code review sessions, if the code is not clean and confusing it needs to be implemented again, after being allowed to be merged, the codes would be given to the robot testing automatically, and finally the testers would check the last time to make sure the software works well with new changes.

The more I work, the more I realize that everything I have learned at school is helpful for my work. Starting with basic programming courses in the first semester with HTML, CSS and JavaScript, Java, SQL. The instructor's exercises were helpful for my problem-solving skills. Besides, the courses in the 2nd and 3rd semesters including React, React native,

Nodejs have been innovated by the teachers to suit the working market was also something that I felt very grateful for.

References

ALEKSANDROV 2013. RESTful API With Node.js + MongoDB. by Evgeny Aleksandrov. URL: <https://aleksandrov.ws/2013/09/12/restful-api-with-nodejs-plus-mongodb/>. Accessed: 08.02.2019

Ankit Singh 2017. Refs in React : All you need to know ! Ankit Singh. <https://hackernoon.com/refs-in-react-all-you-need-to-know-fb9c9e2aeb81>. Accessed: 29.03.2019

Arunoda Susiripala 2016. Introducing React Storybook. URL: <https://medium.com/kadira-voice/introducing-react-storybook-ec27f28de1e2>. Accessed: 14.03.2019

Bartosz Krajka 2015. The difference between Virtual DOM and DOM-by Bartosz Krajka. URL:<https://reactkungfu.com/2015/10/the-difference-between-virtual-dom-and-dom/>. Accessed: 15.02.2019

ByteScout Team of Writers 2018. URL: WHAT IS SQL AND WHAT IS IT USED FOR <https://bytescout.com/blog/what-is-sql-and-what-is-it-used-for.html>. Accessed: 06.02.2019

Jose Luis Dominguez 2018. Lodash Library: What is that and what does it consist of? URL: <https://www.agiliacenter.com/lodash-library/?lang=en>. Accessed: 01.02.2019

JSCAMBLER blog 2018. Vue.js Authentication System with Node.js Backend by Lamin Sanneh URL: <https://blog.jscrambler.com/vue-js-authentication-system-with-node-js-backend>. Accessed; 11.02.2019

Material-ui 2018 URL: <https://material-ui.com/customization/overrides/> . Accessed: 06.04.2019

Mountain goat software 2018. Sprint Planning Meeting. URL: <http://www.mountaingoatsoftware.com/agile/scrum/meetings/sprint-planning-meeting>. Accessed 20.03.1019

Nilson Jacques 2018. Introducing Axios, a Popular, Promise-based HTTP Client by Nilson Jacques. URL: <https://www.sitepoint.com/axios-beginner-guide/>. Accessed: 30.01.2019

PASSPORT JS 2018. Passport document. URL: <http://www.passportjs.org/docs/>. Accessed: 23.01.2019.

ReactJs 2019 <https://reactjs.org/docs/render-props.html>. Accessed: 14.04.2019
RESTFUL API 2018. HTTP Status Codes. URL: <https://restfulapi.net/http-status-codes/>.
Accessed: 28.01.2019.

Sanath Kumar 2018. Mastering Vuex—Zero to Hero. URL: <https://medium.com/dailyjs/mastering-vuex-zero-to-hero-e0ca1f421d45>. Accessed: 05.02.1019

Siriwat Kunaporn 2018. How to Customize Material-up them. URL:
<https://blog.bitsrc.io/how-to-customize-material-ui-theme-v3-2-0-part-3-750db6981a33> .
accessed 06.09.2019

STACKOVERFLOW 2013. Error handling. URL: <https://stackoverflow.com/questions/15711127/express-passport-node-js-error-handling>. Accessed: 28.01.2019

STACKOVERFLOW 2018. How to receive select value in handlesubmit function with formik react js? 2018. URL: <https://stackoverflow.com/questions/50343914/how-to-receive-select-value-in-handlesubmit-function-with-formik-react-js/51029125> . Accessed: 05.04.2019

Stackoverflow 2018. URL: <https://github.com/jaredpalmer/formik/issues/789>. Accessed: 05.04.2019

VueJS 2017. Navigation Guards. URL: <https://router.vuejs.org/guide/advanced/navigation-guards.html#global-before-guards>. Accessed: 07.02.2019