Haaga-Helia
University of Applied Sciences

Growth hacking as a software engineer

Jose Andres Zapata Guaman

Bachelor's Thesis
Degree Program in Business Information Technology
2019

Haaga-Helia
University of Applied Sciences

Abstract

29.11.2019

Author
Jose Andres Zapata Guaman

Degree programme
Business Information Technology

Report/thesis title
Growth hacking as a software engineer

Number of pages and appendix pages
70 + 5

The following bachelor's thesis illustrates the work and professional growth of a software engineer in a digital consultancy agency. The purpose of this diary is to showcase the daily challenges faced as professional in the IT field. Throughout an observational period lasting a total of 13 weeks.

The beginning of the document focuses on introducing the reader to the current work environment. Stating the role of the author, experience in the company and starting skill level. Stablishing a framework which focuses on the ways of working in the current project, the client's organization and interest groups. Presenting the skills required in the project and literature material sources.

The dairy entries are written in a daily format showcasing the main goal of the day. Including details on what has been done and if any problems were faced. At the end of the day there is a status report on the daily progress made. Every week entry includes an in-depth analysis. Containing key learnings from both technical and personal perspective, explaining major problems faced and how they are solved. Introducing best practices based on literature sources for current and future scenarios.

The study concludes that communication is a key when working in a team and in software development. New ways of asking questions and expressing ideas have been found. Proper planning and requirements play an important role in the performance of a project's development. Requirement validation proved to be efficient by providing faster results and bringing value to project. At the end of the observation period the authors skills have improved in multiple aspects, meeting the target skill level. Obtaining proficiency in the project's tools, technologies and expanding soft skills.

Keywords
Software development, full stack development, A/B testing, sales growth, web development, React JS, Conductrics, Google Analytics, JavaScript, Communication

Table of contents

Terms & Abbreviations

A/B Test	A way of validating a hypothesis by creating a controlled experiment with two different variations known as A and B.
Conductrics	Platform based service which specializes in A/B testing and machine learning.
DOM	Document Object Model is the document the browser creates when a website is loaded.
Element Object	HTML element inside of the DOM. Example: div, span, anchor, etc.
Tech Stack	Collection of software technologies, frameworks and programming languages combined to build a software product.
CRUD	Data manipulation functions - *Create, Read, Update and Delete*.
React Component	Reusable and independent UI element
Prop	Input parameter used in a React component
REST	Representational State Transfer
API	Application Programming Interface
SQL	Structured Query Language
MVC	Model View Controller
OOP	Object Oriented Programming
GA	Google Analytics
CMS	Content Management System
GIT	Version Control System

1 Introduction

The following document showcases and describes in detail the daily tasks and responsibilities of a full stack software engineer in company X. During the period of 19.08.2019 – 22.11.2019, for a total of 13 weeks. In the first week company X will be introduced and described, workplace environment, ways of working and starting skill level. The next 10 weeks will be a set of collection of daily entries of what is done at work along with an analysis for each week. The 13th final week will illustrate a professional growth analysis, conclusion and future plans.

1.1 Company X

Company X is an international digital consultancy company. The company was stablished in early 2016 with a core team made up of five people. Their main goal for starting the company was to offer something different than the rest of consultancy companies. Their headquarters are located in Helsinki, Finland with a team of 80 employees. Company X is part of a larger international group of over 650 specialists. What sets company X apart from other firms is that they're data driven with a focus on growing the client's sales by any means necessary.

Bringing impact and change in the following areas :

- Culture
- Marketing
- IT Platform
- Digital Sales
- Physical Sales

Company X grows a clients' business and introduces the inhouse team in new ways of working. The main goal for this is for the client to learn how to achieve their full potential as a business and continue growing. Using an expeditious iterative process called *growth hacking*. This term was established as a collection of actions which all lead towards potential impact and scaling growth by (Ellis, 2017). The core business value of this methodology is to create fast paced experiments.

The experiments are a team effort which combine a mix of experts from different fields in the organization. Involving all aspects of the business such as marketing, content management, web development and business strategy. The success and validity of an experiment is data-driven, combining both sales and analytical data. Providing a conclusion based on the data found, it's possible to learn what works and what could be improved.

Taking in account of what worked, it's possible to build more experiments creating an iterative process of constant growth.

1.2 Work environment

I joined Company X on April 2019 as an associate consultant. My area of expertise is software development and customer consulting. I work in the headquarters of the company which features a rich work culture with a flat organizational mindset. The company provides its clients consulting services. Meaning most of my time and work is done at the client's premises.

My weekly schedule is the following, on Monday to Thursday I focus on client projects, working closely with them in their premises. This makes reaching common goals and communication a lot easier and efficient. The client premise is an open office were different teams can easily interact and work with each other. There is a vast amount of freedom in how the work is done. What truly matters to the client are the end results based on the user experience and in their revenue.

On Fridays I work from Company Xs' office, focusing in internal actives and working on client projects. On that day the whole team is invited to join the complimentary company breakfast. After energizing for the day, there is a chance for everyone to socialize and share what they've done during the week. Then there is a company presentation which shares important announcements, current financial situation, sales leads, trending topics and events happening in town. There is also a developer biweekly were technology topics are discussed and learnings are shared between peers.

1.3 Client Projects

For the last four months I've been able to take part in three different client projects. Each using different technology stacks based on their business and technical requirements. My contributions on all of these projects have been in the technical scope of full-stack software development. The largest project being a cross platform mobile application with cloud based micro services. All company X teams present in each project share one common goal. Using technology to create impact and increase revenue in the client's organization.

During the duration of this thesis I'll will be continuing to work on my third client project. I've joined this project two weeks prior to the start of the thesis. Throughout the diary this project will be referred as *Project Y*. The project revolves around a web-based e-commerce platform.

1.4 Personal development

After completing all the required university courses, I enrolled as an assistant teacher in the Mobile programming course. I decided to help in this course because I had a genuine interest in building mobile apps with React Native and JavaScript. This was a great opportunity for me which let me help fellow students understand these technologies. It was also a great way to review these skills for myself. Explaining a complex technical term in an easy to understand way was my goal as a teacher in the course.

Each time I was asked a question and I explained a concept, I felt great that I was able to help someone and at the same time it re-enforced my own knowledge. I learned to explain my thoughts and ideas in a concise and clear matter. Both of these points helped greatly once I started to work as a professional. During my work placement in Company X, I had the opportunity to use all these skills that I learned by putting them in practice. The first project I took part in was a React Native application. In the first month I learned so much from my mentor who was in charge of the project. This included the technical architecture of the project and from the existing codebase.

I got good habits of writing re-usable code and using best practices from my peer. The codebase was quite extensive and advanced for a newcomer. Despite being new in the project, I was familiar with the technologies used and in a matter of weeks I was able to contribute and build production grade features to the project. In the second month my mentor went on holiday and I was appointed as lead developer for one month. During that time, I was responsible for building features, publishing new releases in production environments and maintaining the project.

Creating both client and back-end features I was a real full-stack software engineer. Not only did I increase my technical skills, I also learned much about myself. During this challenging period, I was able to keep up the development of the app and meet all the requests and requirements from the client. I learned that when it’s needed, I can step up and take charge as a leader in a project.

1.5 Skill level

I'm a third-year bachelors' student in Haaga-Helia University of Applied Sciences. Taking part in the Business Information Technology (BIT) programme, majoring in software engineering. I've concluded all of the required degree courses and my work placement has been completed in Company X. The knowledge and guidance I've received during my studies have provided a solid foundation of software development practices. I've gained an overall understanding of what a real-world IT project contains, what are the major areas of technologies and how different technologies can work together to achieve a greater goal.

Given the choice to choose the order of my specialization courses, I decided to start from the bottom of the tech stack, the database layer. By taking an introductory and a summer course about the subject, I understood what a database is and how it worked. After completing these two courses I was able to build, manipulate and maintain a relational database using SQL. After understanding how data is handled in a system, I wanted to expand that knowledge. By taking the database developer course it was possible to build on top of these concepts, learning how to implement business logic at the database level and automate build and maintenance processes.

Moving on towards the server programming course which focused on building a *REST API* with full *CRUD* functionality. This was based on a Java framework known as Spring *MVC*. In a matter of weeks, I was able to understand how a modern web app accessed and modified data in real time. This new set of tools act as a doorway into the contents of a database. Obtaining good insights on best practices, safe data manipulation via the web and how to secure a server application.

Taking the front-end course taught me how to render, view and manipulate the data within a web browser. Utilizing modern web technologies such as HTML 5, CSS3, JavaScript and the React framework. I developed a keen interest in how all these different technologies came together to build something real. I've had always been interested in how things work inside and out. I'm fascinated by the idea of being able to reverse engineer things, see how they work and possibly improve them.

Taking it a step further I decided to join the mobile programming course. It was intriguing to be able to build a real-life mobile app for both Android and iOS using the same codebase with the React Native framework. By taking the learnt knowledge on the concepts of CSS, JavaScript and React, the transition to the React Native framework was straightforward. Learning new techniques on how to combine and use the mentioned concepts was the key to master this framework. By the end of the course I was proficient in the workflow of a mobile app and React Native as a framework at a basic level. By developing a personal project with the learned skills, it was able to expand beyond the basic scope of the course.

I decided to build my own e-commerce platform from scratch. Brainstorming how everything should connect and work together. I obtained product data from a real e-commerce store and build a rich featured back-end server which could be consumed by a front-end client of my choosing, in this particular case I wanted to build a mobile app with React native. I worked hard on the project for about two months and the end result was a fully working full-stack mobile application, the project was a total success. By planning and investigating both technical and business requirements of the projects scope, it was possible to break complex stories down into smaller and easier tasks. Using the learned technology and practices I learned to overcome blockers in the project and achieved my main goal.

Out of all the courses I took during my studies, this is the one that I enjoyed the most. it pushed me to investigate and learn much more about programming and related technologies by letting me build something I had real interest in. In my final semesters I wanted to learn and practice working in teams in a real IT project. I took part in both instances of the Software Project course. These courses helped me practice and expand my skills. This included not only the technical side but also it helped me improve my people and communication skills.

At the start of this diary I have a general understanding of how technologies can work together and creating a full-stack application. I know how to build one and understand how it works. However, I do feel I lack knowledge and experience in working in teams and deeper understanding of web technologies. I would like to learn more about the topics and understand how a professional project is developed and maintained.

2 Framework

This chapter is dedicated on showcasing my main responsibilities at work and explaining in detail what is needed to accomplish each task. Stating my skill level and the level required to perform my assignments. Introducing the reader in the main entities groups and the people skills needed for daily interactions in the workplace.

2.1 Analysis of current work

There are many different teams inside the client's organization who are using and developing different sections of the platform. My team is known as the *growth hacking team*, with one clear mission. To increase sales revenue on all parts of the platform for the client. We are a cross functional team working together towards our main goal.

The growth hacking team consists of a combination of :

- Sales managers
- Designers
- Developers
- Growth owner.

Each proficient in their own area, combining all their skills into one growth machine. As a growth developer, my tasks focus on utilizing technology as a tool to enable and turn the team's ideas into reality.

My main tasks in *Project Y* include:

- A/B testing
- Analytical tracking
- Data analysis
- Performing JavaScript injections throughout the platform
- Developing new features
- Publishing releases to production
- Code refactoring
- Bug fixes

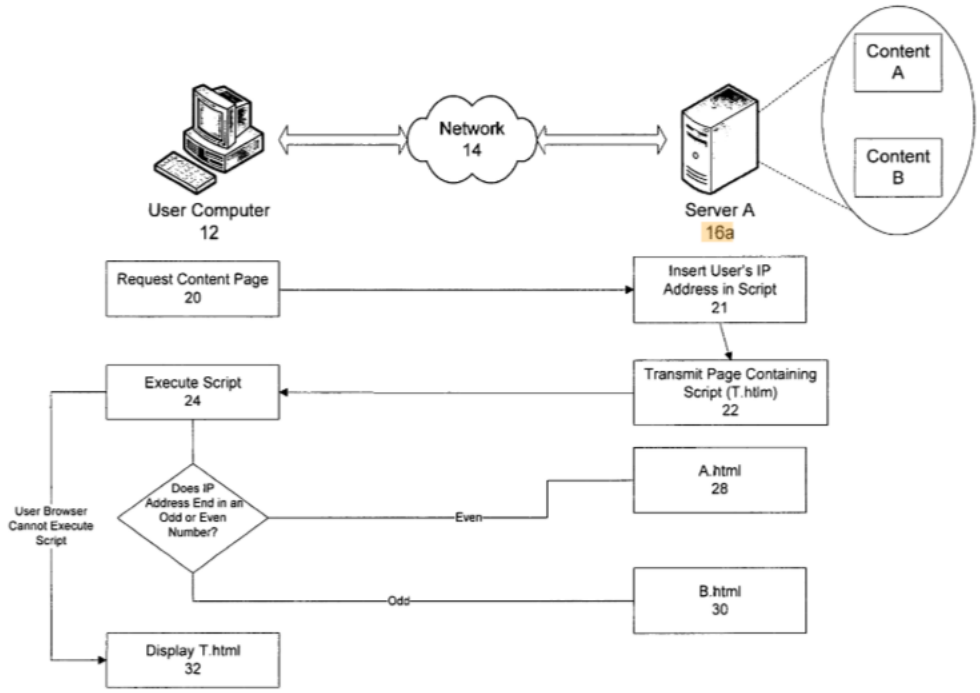

Figure 1 This illustration showcases how an A/B test is technically executed (Dixon, Enos, & Brodmerkle, 2011)

According to (Dixon, Enos, & Brodmerkle, 2011) A/B testing can be used to isolate and test specifics aspects of a website and how this impact its effectiveness. A real-world example on an A/B test would be if a flashing banner would attract the users in entering that part of the site. In order to start a test, two different variants must be stablished. Version A would be the original version of the site, while the B version would be the modified instance.

Conductrics is a platform used to create and deploy A/B tests in any part of the site. When a visitor enters the site, the agent will serve a random version of the site, being either the original A version or the modified B version. This tool keeps track of the visitors in each variant and their interaction inside of the page. Conductrics provides a direct integration with Google Analytics, making it easy to share data between services. Further data manipulation and observation can be continued in the Google Analytics portal.

In addition to the feature of serving different variants of a page, Conductrics lets us add or over-ride the content of a specific elements on the target page. This will be coded using Vanilla JavaScript, which is a *plain* version of JavaScript without any additional libraries (Koen, 2013) This is done by injecting JavaScript, according to (Elrod, 2012) a JavaScript injection executes a JavaScript function on the client-side of the application invoked by the client. In practice this means that once all the elements of the DOM are rendered, the self-invoking function from Conductrics runs and in our use case, modifies the DOM.

This lets us modify different parts of the platform without having the need of accessing or changing their source code which is mostly handled by other teams in the client's organization.

Each test is based on an initial hypothesis, which can be proved by using analytical data tracking the user's behavior in the test. Each instance of a test has a set lifetime cycle, once the time is completed the test stops running. The result of the test is data driven, which validates the hypothesis based on the collected analytical data. The dataset is then analyzed and if the results prove successful permanent solutions are implemented and if not, the original version stays.

Most of the feature driven development I'll be building, will be using the React.JS which is a front-end framework made by the Facebook team and along with modern ECMAScript 6 syntax. There is also a heavy use of Cascading Style Sheets (CSS) which is a stylesheet language used to describe the presentation of a document written in HTML or XML (MDN Web Docs, 2019a). There's an inhouse CSM used in by the client which I will be using in order to publish new pages on the platform and also adding images and other assets. Back-end development uses an existing REST API based on Node.JS and JavaScript and for the repository source control an intermediate command of GIT is required.

2.1.1 Evaluation

At the start of this diary, I consider my skill level is of a *novice actor* and a *skillful performer*. I feel my skill set covers most of the requirements for this project. I'm proficient in React and JavaScript as I've learned them in school, practicing these in both personal and previous client projects. For the tools used in source control, GIT has always been my tool of choice. I do feel I will need guidance in the analytical set of tools and workflow of conducting and implementing A/B testing.

The tools that I have never used before are Google Analytics, Conductrics, JavaScript injections and their custom in house CSM tool which will take time for me to learn and get comfortable. I've always had a keen interest in manipulating data and using functions, that has been my main focus ever since I started programming. I feel I have a basic understanding of CSS and would need to improve this skill in order to achieve the visual styling part of the tasks. Overall, I'm certain I can meet the deadlines of the tasks, but I will need heavy assistance on getting up and running in the workflow of this project. as its focus is orientated on creating test variants and using analytical data which is new to me.

During the period of the learning diary I'll be using the following sources as a support basis of knowledge and guidance in the professional field. The scope of the work done in the project is focused on hands on technical implementations.

- Literature related to software development, A/B Testing and communication
- Scientific articles related to communication & software development best practices
- Official technical documentation from sources such as Mozilla Developer Network, W3 Schools, React JS documentation, Lodash portal and Google Developers
- Technical expert forums such as Stack Overflow

2.1.2 Development goals

Even though sometimes things get hard, I like challenges and enjoy problem solving. It's something I am faced with every day as a software engineer. I feel I there is much to learn and I strive to do so one day at a time. From a professional standpoint, I've been focusing on developing mobile applications and progressively learning and implementing more web-based solutions. I will certainly continue to expand my knowledge in the area of software engineering and customer consulting.

From a technical standpoint, my focus so far has been on data manipulation, building mobile applications and back-end servers. During this project and the future, I plan on learning more about front-end technologies, responsive web design principles and mastering CSS styling. I also have a keen interest in learning a new emerging back-end technology known as GraphQL. This technology is a query API language which is developed by Facebook and was released publicly in 2015.

"GraphQL provides an alternative to REST-based architectures with the purpose of increasing developer productivity and minimizing amounts of data transferred." (GraphQL Foundation, 2019)

Being a growth software engineer in Company X, I would like to learn more about customer consulting and how growth hacking is done in practice. Understanding the needs and wants of the client and evaluating these based on their business and technological requirements.

By being in constant interaction with the client I feel I will get to improve my communicational skills and excel in providing value to the customer. At the start of this diary, this topic is quite new to me. I see this as an excellent opportunity to understand how continues sale optimization to the business is done by using technology as a tool to achieve results.

2.2 Interest groups at work

The external stakeholders in *Project Ys* organization are shareholders, customers, governmental and competitor entities in the equivalent field. All these entities have a major influence in the project and can improve or affect the project. In practice this means if support from the stake holders is dropped, it will bring a negative effect. In the other hand, if their competitor groups are start offering inferior services or products, customers are most likely to continue with Project Y thus bringing a positive effect.

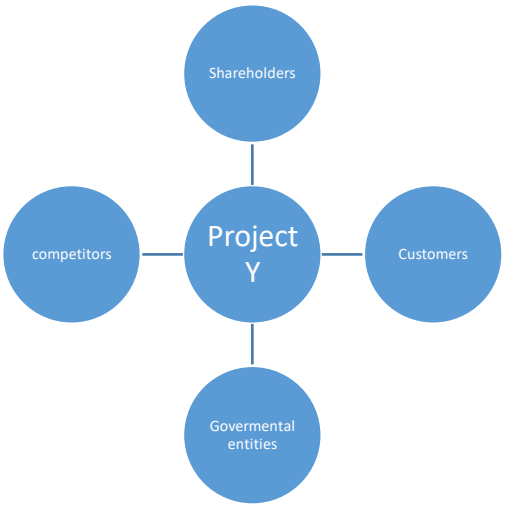

Figure 2 External groups in Project Y

For the scope of this project, the internal groups focus revolves around my work team. The internal interest groups start with the product owner who defines the main stories and prioritizes tasks for the teams. Sales managers cover the financial side, making sure their goals are meet. The data Ops team provide us with formatted data based on requested our requirements and the web shop team builds and maintain critical parts of the platform.

Figure 3 Internal interest groups inside the Growth Hacking Team

2.3 Interaction skills at work

The entire growth hacking team is physically present in the client's premises. We are all located in the same area and floor as the sales managers and web shop team. These two entities are the ones we interact with the most. This location makes it easy and efficient to communicate with each other. Ninety percent of the team's communication and interaction is done in a face to face matter.

A study conducted, compared face to face communication versus computerized interaction, suggests that even though both methods of communication can be used interact and solve problems. Those who communicate via face to face are more likely to reach an agreement as compared to the latter. (Starr Roxanne, Kenneth, & Murray, 1986)

I agree with these results as I feel it's easier just to go and talk with the person instead of writing to them, especially if they're a couple of meters away from my desk. This also provides value to client as its easy for them to express their needs and easier for us to understand their environment and pinpoint possible improvement areas. We also use Slack and e-mail as a means of internal communication. Every day I talk with the growth owner to understand what I need to do and ask clarification questions.

Once I'm working on a task, I talk with the designers to get a visual prototype based on the specifications I have in the current task. I always ask and interact with fellow engineer whenever I have a question or they are trying to solve something, I try to help them as well.

Figure 4 Team communication is a cycle that must be clear and easy to understand for all parties to achieve our success as a team.

In the office wall our team has a physical project board used to manage and keep track of the projects progress. There one can visually see the status of each task, which are divided by segments. Each task is written in a stinky note which is color coded based on the segment or track it belongs to. Each member of the team has a dedicated lane making it easy for everyone can see what they're working on now, what they'll be working on later and any tasks that are blocked or postponed. This method of project management is quite visual, making easy to track and see the big picture of the project.

Each morning at 10 AM the whole team has a daily meeting where each member gives a descriptive update on the following:

- What was done yesterday
- What will be done today
- Announce any possible blockers

throughout the meeting, the project board is updated to reflect the changes highlighted by the team and new tasks are given by the growth owner. Important events and information regarding are also shared. This is also a good time to coordinate collaboration needed from other teams to accomplish the task at hand.

3 Diary entries (26.08.2019 – 22.11.2019)

The following entries are divided into weekly segments which include daily entries containing daily goals, skill and progress development. At the end of each segment there's a reflectional analysis which showcases how the professional skills have developed in the week. Explaining in detail the main tasks, problems faced and how they were solved. Highlighting literature used to fortify the development of the tasks and obtaining best practical approaches which will be used in the future.

3.1 Week 1

This week focuses on getting familiar with the projects workflow and tools set by exploring the codebase and developing features which touch multiple sides of the platform.

Monday

I will start the week by creating different variants of a specific page of the site and set up and execute an A/B test in production.

In the previous week the team offered me a great opportunity of creating an A/B test of my own and today was the day to put that in practice. The end goal of the test is to make a certain button element on the website more visible and attractive to the end user, exposing a hidden feature that was being missed by most users. Before starting the test, a hypothesis was defined. The usage of the filtering feature of the site was quite low, its being ignored by the target audience and the cause of this can be because of its color and form making it easily dismissed as a block of text by the user.

It's believed that by adding a bold color to the element it would attract the eyes of the user and usage of the filter button would increase. I obtained the correct shade of blue from one of the designers and proceeded to code the solution. Once it was ready, I built a minimized version of the code and proceed to upload that into a Conductrics agent and started the A/B test successfully. I got to interreact with new tools and understood how these work in general by completing this task.

Tuesday

The main task for today is to complete an existing feature, an outlet product section. To achieve the main task, I will first need to obtain a specific selection of product data. Once the data is obtained, manipulate and parse the data in the correct format the front-end user interface accepts. Fetch the data from a React component and showcase the product specs and the available stock quantity of each item in the selection.

The second task is to add an anchor button to a specific banner in the website. To be able to add the button I need to investigate and understand where these elements are being generated. Once the code block is identified I could add a new anchor button pointing to the banner section of the website.

I started by continuing my main task regarding the outlet section of products. When I started the outlet feature, I created my own branch from the codebase. A colleague developer had published changes to the codebase that enabled the item availability count for each of the outlet products. I had to figure out a way to get his changes into my own branch. In the project we use Git as a source control tool.

There were two ways to get his changes together with mine, one takes the incoming changes and merges them into what I've done, this is known as git merge. The second option is to apply all incoming changes first and then on top of that set all the changes that I've done, this is known as git rebase. I proceeded to use git rebase because I noticed that there were far more incoming changes made by the rest of the team. It was best to apply my own work on top of all the incoming changes than having mine set in different places.
I started by writing a function that would first fetch a list of items from our content management system, then pass list as a parameter to a second function that fetches the full product data.

I created a log of the data being received and noticed it was the correct data, but it was quite nested. The response data had a complex structure of an array instance containing separate arrays and inside of each inner array there was one JSON object of data. Meaning each product object was inside in its own array, accessing data in that format is not optimal for the front-end, so I used a helper function from a the Lodash library called flattenDeep. What this does is take each of the items inside all the array and puts them all in one array, that way its much simpler to loop though the data when generating the user-interface components. I already had a tester React component for the outlet data, I passed the fetched data to the said component and the product data was rendered in the browser.

Thanks to the changes made by my colleague it was also possible to see the quantity of items available in this section. After lunch I was approached by the growth team owner and was asked if it's possible to add an extra button to the navigation bar on a certain page in the site, this button should scroll the user down to a certain position on the page. This had to be done before Thursday, so time was of the essence. I decided to take a pause from the outlet products and focus on this task instead. Investigating where those buttons came from, I found out that the element where the button should anchor to, was being injected via JavaScript on the site which meant it was not all generated from the original source code of the site.

This made this simple task a bit more challenging, I found where the injection was taking place and tried injecting the button dynamically. At first it worked but it only executed after a couple seconds that the site loaded. This would not be a good user experience, so I went back to the drawing board and explored how the actual buttons were being generated. These elements were being created in React, adding the extra button would work but the header element was used in many parts of the site so that wasn't a good option either.

Wednesday

The main task for today is to complete the extra button in the header and outlet feature.

I was able to complete the header extra button by using a react feature known as props, this caused the expected behavior of adding the extra button in one place safely. I had been working in this feature before starting this diary, in summary this would be a limited section of product items. The goal is to show the list of items and the real time stock count. Once the stock quantity is consumed the item should be hidden from the selection. I made a connection between the CMS where the client could add and modify the list of items. The front-end will then fetch the product data from there and generate the UI.

Once everything was connected and working, I decided to test what happens when the stock of an item would reach a value of zero. I was faced with an empty screen, once one of the items quantities ended the list of products became an empty array. I debugged the codebase and identified that the function which fetches data in the front-end used Promises to fetch all items asynchronously using a JavaScript Promise. I was able to overcome this problem by implementing error handling in function which fetched the items.

Thursday

The main task for today is learn and upload images into the CSM and update a sales banner content.

At first, I thought that uploading images in to the CSM was something done in the systems dashboard. It turned out that it was a bit more technically and surprisingly simple. The clients project has a separate codebase in which all the assets are kept, I cloned the repository, copied the image into a specific folder and committed these changes with Git. Once the repository was updated, in the CSM I had to manually trigger a refresh and all the assets were updated instantly.

The sales banner that I had to update was located in the same page where I had developed before. I didn't find the code that corresponded to that page in the original source, after a deeper investigation I later found out that that specific banner was being inserted via a JavaScript injection as a Conductrics agent. Once that was stablished, I was located the injection and updated the content and image for that banner. In the afternoon there was a big meeting with all the teams in the project and I got to learn more about what other teams do and were the project is heading. Today was a very productive day and I got to interact with different tools in the project. I felt I learned a lot about how things are done in the team and in the project.

Friday

The main task for today is learn about the React hooks API.

Being a Friday I was at Company X headquarters, the day started by having breakfast and attending the weekly company meeting. After that I took part in a learning workshop where one of my colleagues gave a presentation about a new feature in the React framework. It was a really interesting topic which I had genuine interest in, before the presentation I had read official React hooks documentation. This new API adds the ability to handle state and life cycle events inside of functional components (React JS Documentation, 2019a)

Before the 16.8 release of the React JS framework, functional components would could only take in parameters and output generated HTML elements. A functional component could not fetch data or keep track of state or render dynamic content. This all changed with the new React hooks API, from a programmer respective the new hooks API made life easier, in practice it meant avoiding code repetition thus less code.

Weekly analysis

This week I I've been focused on learning; getting familiar with the projects scope, tools, workflow and how data analytics is implemented. I was able to create and publish my very own first A/B test live on the site which is a major part of my responsibilities as a software engineer in this project.

I learned to identify and differentiate Conductrics agents, each one is a separate JavaScript injection in the codebase. In total there is over 70 instances. This was quite overwhelming since it was hard to pinpoint what each instance does and how to find the correct files inside the codebase. I came to learn that there is an established naming convention between each agent and their corresponding script in the codebase. As (Mohan, 2019) points out in an naming conventions article, There is no right or wrong way to set a naming convention in a project, rather each team agrees on one single structure which is followed making it easier to read and create new instances for each member of the team. From now on I will follow our projects naming convention so my work can be easily identified by other team members and myself.

I got to learn a new way of merging changes of other engineers inside my own git branch. In the past I've always used the command *git merge* but, in this case, it caused a lot of merge conflicts. There were many files that had been edited by other colleagues and me. What I wanted was to obtain the recent changes made by another colleague and apply my own work on top. I asked a senior engineer about this topic and he pointed out the best way to achieve this was to use *git rebase.*

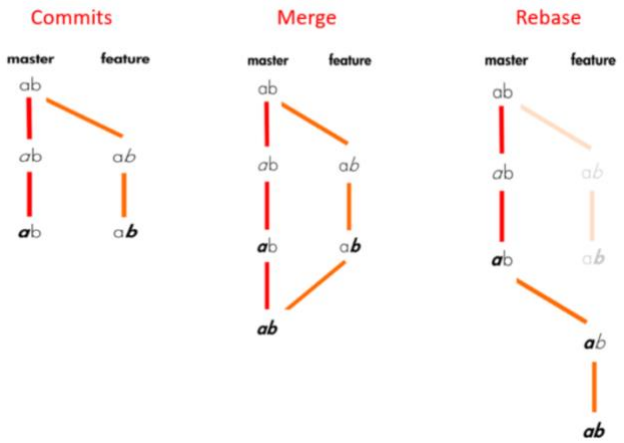

Figure 5 Visual representation of the difference between Git merge and Git Rebase. - https://hackernoon.com/git-merge-vs-rebase-whats-the-diff-76413c117333

Based on this image I was able to understand how these two differ from each other and as (Di Marco, 2009) points out, both have their own use. Git merge combines master with the feature branch, essentially merging both. While Git rebase updates, all changes from master retrospectively and applies the changes from the feature branch on top. Not only did this new feature accomplish exactly what I wanted, it also kept the git commit history clean and reduced the amount of merge conflicts when integrating the incoming changes into my git branch.

While fetching some data from a back-end server with the existing solutions I noticed the data structure being fetched was nested in two levels or arrays.

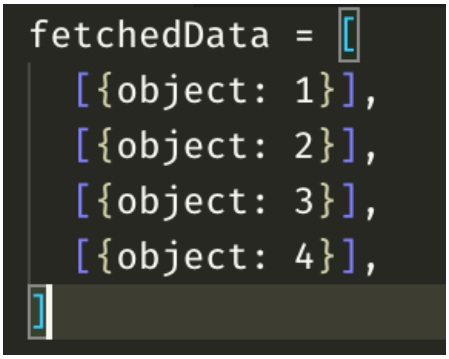

Figure 6 Data structure, simulating the response from the server – Zapata, 2019

Data in this type of format is not in an ideal structure as each nested array only contained one single object, the best way to present this format is to eliminate the nested arrays and have all the object entries in one single level of array. I could've used this data as it was received it, but I chose not to as it would've required changes in the UI components to accept the nested data structure. Changing them would have worked for now but it wouldn't be a proper solution. The nested data coming from the server could have a structural change in the future, this would've potentially caused bugs and break functionally in the platform. Another way of solving this was to create my own function which would get rid of the nested array structure, but this would've taken extra time and resources to complete.

Fortunately, I was introduced to a new tool inside the platform known as Lodash.

"A modern JavaScript utility library delivering modularity, performance & extras." - (Lodash Official Documentation, 2019)

By using one their helper functions called *flattenDeep( )* I was able to parse the nested data into a single nested level data structure.

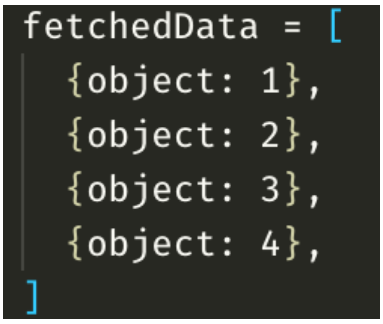

Figure 7 Result of the flattenDeep helper function – Zapata, 2019

This helper function helped me maintain a clean code structure and save time while doing so. In the future when I need to modify or parse data I will check if this library can be used. After testing the UI part of this code, I came across an interesting bug in the existing function used to make the data request. What that function did was make multiple fetches to an API and wait for the server to respond. It would wait for all of the requests and only return these if all of the requests resolved correctly.

In the outlet case, once an items stock was done the server responded with a 404 error. The fetching function retrieved many items at once, and if one or more was failing it would not return anything at all. This behavior was happening because the function was using a Promise.all( ) method.

"The Promise.all () method returns a single Promise that resolves when all of the promises passed as an iterable have resolved or when the iterable contains no promises. It rejects with the reason of the first promise that rejects." (MDN Web Docs, 2019b)

In practice this meant that if for any reason, if any of the inner Promises being executed failed for, the result will be an error for all the inner promises instances. To tackle this, I implemented an error catch handling strategy, so that if any of the Promises in the list would fail for any reason it would return an empty array entry and continue with the next item in the list. This not only solved my problem but also improve the whole codebase overall and prevent bugs that would result in an empty page if any of the promises requested in the page were to fail being while being fetched from the server.

During this week I spent two days trying to figure out how to make a React component behave differently in only one place; this component was being used in several places so changing its structure would have affected all the places it was used in. I tried using an existing JavaScript injection to make the changes only in one place but that didn't work out as expected, it showed the button, but it took a few seconds to render.

After trying multiple different approaches and investigation, I came to realize that my first approach was incorrect. The DOM elements of which I was targeting were being created by React, not manually and what my function was doing is modifying an element manually, which goes against one of the frameworks core rules. It's considered a bad practice to manually modify the DOM elements generated by React. An element created by React is *immutable* (React JS Documentation, 2019c) . This meant that an elements property shouldn't change, attributes or children elements because the framework takes care of changing the elements if needed when the framework triggers a re-render of the component.

I came to the conclusion that in order to add the extra button in the header component only in one specific instance. The best way to do this was by passing an optional *prop* to the component. By doing this, its guaranteed that the rest of the places were the header component is used won't show the extra button. as I activated the use of the prop in the instance that should show the extra button and it worked as expected.

3.2 Week 2

This week focuses on analytical user tracking with Google Analytics. Understanding its use in the project, learning the technical implementation and viewing the results in the portal. Building new features and getting familiar with the deployment workflow.

Monday

The main goal the day is to take the prototype navigation bar and develop it into a working solution which will be used for an A/B test. By doing this task I expect to get a deeper understanding of how tracking can be implemented in the site. Once the development of the feature is completed, the test must be created and published to the production version of the site.

I created a JavaScript injection that uses the DOM API, it first takes a copy of an existing button, modifies the contents of the element and inserts the custom button into the existing navigation bar. I published this test to the production site, and it was available instantly to half of the incoming traffic. After completing the task, I was given a new task by the growth owner along with an image. This involved creating a new content block and adding it to an existing page. I took the provided image and published it in the CMS platform and proceeded to build the content block, which was on an existing page I've worked on before.

I'm still not familiar with how the analytical tracking works on the website from the technical perspective and I spent the rest of the day asking and learning from my teammates. They showed me clear examples on how the system is in place and where its used.

Tuesday

I'll be focusing on learning and implementing analytical functionality to the task I created yesterday and publish a new promotional content block in an existing page.

Taking what I was shown the day before, I added the GA function which starts analytical tracking. I deployed the update to the production site and moved to the GA portal to see if the tracking worked. After exploring the portal, I was able to find the correct place where all these events were being displayed.

I received an updated image from one of the designers for the pronominal content block. This image had to manually be added to the CSM repo and then deployed.

Once the image was accessible with an URL, the next step was to deploy a new version of the page with the added content block. Once deployed while testing the production build, I noticed that the action call button on the banner was directing to the correct page, but it was doing so by opening a new window. The expected behavior is that when the action button is clicked it re-directs the user in the same browser instance. I spent quite a while investigating and researching this topic and I learned that by removing the original path of the URL and just placing the parameter name it corrected the behavior.

Wednesday

Today's goals are to deploy an updated version of the site with the correct URL redirect & ideate a new product teaser landing page.

Regarding the task started yesterday, I got a working version in my local environment which redirected to the correct page. I proceeded to deploy the changes in the live site, using clients CMS. I noticed that the action to call button indeed made the correct React content render, but the URL path didn't change. Using the browsers developer tools, specifically the network set of tools. I pinpointed the problem and learned that the cause wasn't related to my code.

The cause of the bug was the content being received when entering the redirected URL path. I confirmed this with my team, and I was informed that this was handled by the web shop team. I had a meeting with them and showcased the problem and they acknowledge the bug and added this task to their backlog. I spent most of my day investigating the issue and started gathering leads for the teaser product specifications for the new page at the end of the day.

Thursday

The main goal for today is focused on the new product teaser page. I'll be gathering specifications and creating a base framework for the page. This task requires me to create a new page in the CMS and code the front-end and connect a form with an API.

Based on the manufactures request, none of the technical specifications of the product could be released before the official release date of the product. I meet with a designer and we agreed on a design which will have a short description about the teaser and a specified the contents of the form for the user to fill and get contacted once the product is available. I learned how to create a new page in the website using the CSM.

In order to test if the new page was created, I was able to put in practice a skill I learned a couple of days before, publishing content to a page with the CMS.

I published a small placeholder text and when I visited the path to the page I had just created, I confirmed that I had created and published the page correctly. Interacting with the CMS and its tools every day paid off. Each time the process gets faster and easier for me. I spent the rest of the day writing React and CSS for the teaser page.

Friday

The main goal for today is to develop and complete the new product teaser landing page.

After obtaining all the specifications and content for the teaser page I had a clear picture of what functionality and content the page would offer. I was able to create the react UI code without a hassle. The major challenge was understanding what would happen after the form was submitted and how the data the form sent would be processed. With the help of an example my college showed me, I learned where to send the data.

I also learned how generate a unique campaign code for this form so that it would be possible to target the segment of people who signed up. Overall this task was great learning opportunity and I got to interact with the applications back-end server and front-end to accomplish it.

Weekly analysis

At the start of the week, I was given a task which required me to work on the header component which I had worked on the previous week. I had to add more temporary anchor buttons to the header which should be shown only in one specific place. By utilizing a technique known as conditional rendering it's possible to encapsulate certain behavior inside of a React component and only trigger a render of the encapsulated elements when the set condition is true (React JS Documentation, 2019e) .By passing a prop as the condition, the extra button would be present in this instance but with this task there would be three extra button props. This started to make the code chaotic, so I decided to improve the code I've written earlier by using a different approach. I removed the props in the component and instead created an array of three objects, each containing the respective data to generate an anchor button.

The goal was to combine the new array with the existing one, thus having only one place of data to generate all the elements required in the header. In functional programming it is considered bad practice to modify existing data elements such as an array. The best approach would be to take the original entries and the extra buttons array, combining these two and making a total new array element. According to (W3 Schools, 2019e) the mentioned method does not change the original arrays in question. Instead it creates a new one by taking the values of the passed arrays.

I achieved this by using a built in JavaScript array method known as concat( ).

```
const original = [{ object1: data }, { object2: data }, { object3: data }];
const addOnEntries = [{ object4: data }, { object4: data }];
var allEntries = original.concat(addOnEntries);
//!OUTPUT
[{ object1: data }, { object2: data }, { object3: data }, { object4: data }, { object4: data }]
```

Figure 8 Example usage of the concat array method – Zapata, 2019

Using this approach made the making the application more efficient and the code easier to read for further development.

I've gotten to practice my debugging skills in the task where a URL redirect didn't go to the correct page. Using the browsers network inspections tools documentation (Basques, 2019) and what I learned from the server programming course in school. I was able to understand the network requests and understood the difference between the multiple different requests. An HTTP request is a *Hypertext Transfer Protocol* used between servers and clients, in this case between the web-shop server and the front-end client in the browser. According to (W3 Schools, 2019d) when an HTTP request is made to the server, the response contains status codes. Information and content requested. I identified the network request to be of type of GET. According to (MDN Web Docs, 2019j) an HTTP GET request is only used to retrieve data. In this case once the page redirected the data obtained from the request was corrected. I was able to correctly find the cause of an existing bug which was happening on the server side. I took this forward with the corresponding team and it will be fixed in the future.

During the week I had a lot of interaction with Google Analytics. Although the tool is quite immersive and offers a lot of functionality, for a beginner like me it felt overwhelming. The dashboard of GA has a lot going on but the more I see it and interact with it I'm starting to see a pattern of the most common areas of usage of the tool.

I successfully implemented analytical tracking to specific events and learned how to view the analytical data inside the GA portal. I also got lots of interaction with the in-house CSM tool which I used to add content and publish new pages in the site. I feel that this week I've increased my productivity level, as I'm getting more familiar with the codebase and existing tools everything seems to fall in place.

3.3 Week 3

This week focuses on getting familiar with the projects business logic and learning how the purchase funnel works by adding new functionality.

Monday

Today's main task is to start an in-house training course & create a new feature. The main feature for today is to create an A/B test in one of the product pages, the modified variant will have an extra add-on purchase option.

In the morning I started doing an online training on X topic. After having the daily team meeting, I was given a time sensitive task by the product owner. This task has a deadline of going live tomorrow because it's part of an ongoing sales campaign. I obtained the requirements and requested a design prototype from the designer. While the design was in progress, I started to investigate the codebase.

The task seemed straight forward, fetch the required data, parse the data in the correct format and create a new React component which should render an add-on purchase element to an existing product sales page. The code was made using React but the way the state was created and handled was something completely new to me. It was using JavaScript classes to generate the initial state of the whole page and manage it. I spent the whole day going through the code together with a colleague as both of us didn't comprehend how it worked. At the end of the day I got a basic understanding of how the external class worked and made a small amount of progress in the overall scope of the task.

Tuesday

Today's main focus is to update the content of the teaser product page and deploy it in production. Continue working on the building the add-on feature which I started yesterday.

During the morning I updated the content and deployed the teaser product page successfully. Through the day one of the sales managers requested I also updated content on another part of the site which I completed successfully. For the rest of the day I continued working on the add-on feature which revolves around using separate instances of classes to manage the state. Even though this way of handling state is new to me, I was able to understand it more, the way the code is structured is based on an object-oriented programming (OOB) principles.

The code I touched today reminded me of Java, thanks to the server programming course I was familiar with the workflow and could relate how the class is made and used. Even though I missed the deadline for this assignment, I felt that I learned a lot about this new methodology, and I was able to fetch the required data and display it in an a React component.

Wednesday

Today's main goal is to continue the add-on feature. This task includes updating the purchase funnel for the add one and obtaining the remaining data needed for the UI.

I got to know and interact with the sale manager in charge of this particular product. After out discussion I got a deeper understanding of what data and content should be displayed to the targeted user segment. It was a really nice experience to understand their point of view on the value this would bring and why we're doing this feature. After getting all the specifications correctly it was time to get back in the code. It took a lot of time and reverse engineering in order to understand how the purchase funnel worked and what external parts connect to it.

The time invested in this paid off, I was able to get the main technical functionality working, this included fetching the correct product data and setting up logic to only show this new add-on to a specified visibility group. After setting up the data fork flow I proceed to understand how each item was added to the cart using existing functionalities. With this as a starting point I was able to item that included the new add-on option to the cart order.

Thursday

Today's main goal is to finalize the add-on feature and publish it to production. Setting up an A/B test programmatically for the add-on feature and update content in other parts of the site.

After an interactive checking process of the UI and the contents with the sales manager in charge. Minor updates of the content were defined and made in the add-on feature. Once the UI was ready, it was time to set up an A/B test in for this feature using a different approach. Together with a senior engineer I learned how to conditionally render content directly from the React code using Conductrics without a JavaScript injection.

While implementing this I had to code a JavaScript promise to make use of the Conductrics API and set the data only in the modified B variant. This was a big learning experience as I'm not fully proficient in JS promises but after implementing my own, I got a deeper understanding of how a promise works. Then I proceed to test if both variants were working as expected, once I was sure both were working in all browsers, I committed my changes to the GIT repository and made a release to the production environment. I moved to the second task by gathering all the content that needed to be updated from one of the sales managers.

The page that needed to be updated was one I worked on earlier this week. Being the author of that section of the code made the update fast and straightforward. I release the update to production and the content worked as expected. I feel that today was a good day, I learned a lot of new things by doing and was able to achieve my set goals.

Friday

Today's main goal is to focus on internal Company X activities and learning.

I spent most of my time today being active in internal company actives. In a couple of months there will be a Hackathon in Helsinki, in which Company X is taking part as a partner. I was given the opportunity of being part of the team representing Company X. In the past years I was a participant in that hackathon event, and I enjoyed my time there. It's a unique experience where you take tackle a problem and build a concept in a short time span.

We formed a team of 8 people and attended a meeting with Together with organizers of the hackathon where we were introduced in what the event is and what our role would be. Brainstorming of ideas and possible challenges were set, which will continue to be developed further. In the afternoon I was informed by the growth owner that next week I will be working on expanding functionality of an existing page, I spent the rest of the day investigating the codebase pertaining this page.

Weekly analysis

I spent most of the week working on the add-on feature which required direct interaction with the purchase funnel. This was one of the core parts of the business logic of the application. Most of my time was spent reverse engineering the code to get a clear understanding of how it worked. From this I learned that the business logic was being isolated from the UI as a separate JavaScript Class.

At first sight it seemed like an over engineered solution which I questioned all week. After investigating this topic, I learned that there are crucial benefits from doing this separation of logic. Based on (Kraft, 2016) technical article, the separation of business logic from the UI actually made since, in the long run this separation brings benefits to the application and engineer.

In *Project Y* 's use case the major benefits are the following:

- State management in one single place
- Ability to access and view business logic anywhere in the application
- Encapsulation
- Reusability

This was all possible by using an OOP methodology and defining a class, this can be considered a predefined template which is used for creating new instances of objects from a particular class (Moore, Njeru, & Pocevicius, 2019) . Inside a class it's possible to set an initial state and methods which can be used to manipulate the class. Using a defined class gives the engineer the ability to reuse code.

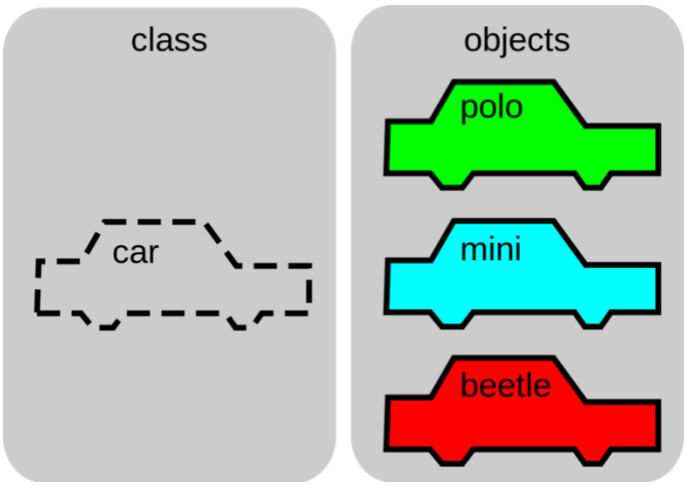

Figure 9 A visual representation of a class and object instances -CPT-OOP-objects and classes - https://en.wikibooks.org/wiki/File:CPT-OOP-objects_and_classes.svg

As seen in the image, the class provides the starting building block template which then can be used to create multiple unique instances. Each instance is isolated and can contain unique data, with the benefit of having the same initial state and capable of using all the methods defined in the original class.

Coming back to what was done in the previous weeks, setting up A/B testing using an approach of injecting JavaScript. This approach loads the original site for the A variant and on the B variant the modified version. During this week I've learned a new approach of handling A/B test variants programmatically. This uses a function which connects with the Conductrics API and returns a random variant type. The best place to execute this function would be when the page is requested. In the tasks I've been working on, the pages are made in using the React framework.

That meant that the ideal place to use this would be when the React component is created. Investigating the frameworks documentation, I learned that the best approach possible would be to use one of the provided lifecycle methods from React.

"*The componentDidMount( ) method runs after the component output has been rendered to the DOM*" (React JS Documentation, 2019d)

This would be the best place to execute this function. Once the component is rendered, meaning when the element is added to the DOM. The method executes and calls the Conductrics function which returns the variant result. The returned value is guaranteed to be a string containing either A or B. With this information we can achieve conditionally rendering content without having the need of injecting JavaScript or modifying the DOM manually. Personally, I liked this approach more than injecting JavaScript as it keeps the code all in one place.

It also follows the approach of not modifying the React element directly thus getting better performance from the React framework by letting it handle when elements should be re-rendered or modified from the DOM. throughout the codebase there is a lot of functions which rely on the concept of JavaScript Promise. Based on (MDN Web Docs, 2019e) a *Promise* is considered a JavaScript object which has two call-back functions which complete and asynchronous operation.

I've used them before in other places, but I hadn't quite gotten a full grasp on the concept. During the week I interacted with them and even wrote my own Promises to understand the concept.

```
const promiseExample = parameter => {
  return new Promise((resolve, reject) => {
    //? Asyncronous operation begins
    fetch(parameter).then(fetchResult => {
      if (fetchResult) {
        resolve(fetchResult);
      } else {
        reject(() => new Error());
      }
    });
  });
};

const executeManyPromises = () => {
  //! Execute multiple Promises together
  Promise.all([
    promiseExample('https://api.chucknorris.io/jokes/random'),
    promiseExample('https://thesimpsonsquoteapi.glitch.me/quotes'),
  ]).then(([result1, result2]) => {
    console.log(result1);
    console.log(result2);
  }).catch(
    (error) => {
      console.log('this Promise was rejected' + error);
    }
  )
};

executeManyPromises();
```

Figure 10 A code snippet showcasing how to use Promises - Zapata, 2019

Here we can see that the function promise Example returns a Promise object which make an HTTP request in an asynchronous matter. Meaning once the fetch function is called, the result from it can then be used, this is done by checking if the value of fetchResult. If it's true, the promise will resolve and return the result and if false it will be rejected and return an error object.

“The Promise.all() method returns a single Promise that resolves when all of the promises passed as an iterable have resolved or when the iterable contains no promises.” - (MDN Web Docs, 2019h)

With the Promise.all() method it’s possible to combine multiple promises into one parent Promise. The returned value is an asynchronously resolved Promise containing the values of all the inner Promises. One thing to keep in mind is that if any of the inner Promises fail, the returned value will be a rejected Promise stating the reason the Promise failed. This is something I’ll keep in mind when using this learnt knowledge.

I’ve felt that this week has been quite intense, full of new challenges and learnings. The requirements and complexity of the given tasks have definitely increased in complexity compared to the previous weeks. Overall, I’ve managed to complete the requested tasks by putting in time and effort to learn the current implementation, feeling this week was hard but also quite beneficial. Getting familiar with the order process and business logic in the project will make things easier in the long run.

3.4 Week 4

This week focuses on developing two new campaign pages, taking a concept with a design, building the implementation and deploying these new services.

Monday

Today's goal is to investigate and create a new campaign page for a service in the platform.

In the morning I attended a meeting welcoming a new member joining the team. After that I spent some time helping another colleague set up the development environment in their machine. As most of the resources needed for Project Y are only available inside the client's domain, a VPN is needed to access them. After getting my colleague up and running, I focused on my development work. I started by gathering the specifications for the task at hand.

I meet with the sales manager and obtained the information needed for the task. The main goal of the task is to create two different pages where a user can input a unique code and then can choose and activate a service with a campaign offer. Since this task required two different implementations, I started by focusing on product one. I fetched and parsed the data in the desired format and was able to display the data in the front-end, one thing I noticed is that when entering a detailed view of each service one key value was incorrect. The item data shown in the detailed view was the correct item, but the campaign offer was not applied.

I investigated where this data was being fetched from and I was able to find the root cause. The component for the detailed view had a bug related to a type-o in one of the props passed to the component. I pointed this out to a senior engineer, and he confirmed my hypothesis and asked me to correct this and deploy the fix to production. After fixing the bug I was able to get core functionality working for the first part of my task and remove this bug in another part of the platform in the production environment. I will continue this task tomorrow as the design and style have not been set. Overall today I feel really good and productive as my contributions have crucial impact in the project's performance.

Tuesday

Today's goal is to continue the campaign page feature by implementing the core functionality for product 2.

As the designs for page I worked on yesterday have not been delivered, I proceeded to set up the core functionality for the second page. I was able to use most of the applied logic created yesterday for the product one page. I fetched the data and visualized it in the front-end however I did run into a blocker. In the main view of the page when pressing the purchase button, the price would not apply the correct campaign. I spent most of my day reverse engineering getting a deeper insight on how the data from the page was being sent to the web shop platform.

By using the built-in browser developer tools, it was able to see what was being passed and received when the page went to the cart. In the parameters passed to the web shop cart, the one related to the campaign was not being detected. I checked that the names of the parameter in question and realized one key parameter didn't match what the cart was expecting. Correcting the name of that parameter the data was passed successfully. At the end of the day I understood how it worked and was able to correctly apply the campaign offer to reflex the correct price.

Wednesday

Today's main goal is to add content for the campaign pages and update the add-on page.

In the morning I was informed that I should update the product and content of the add-on purchase option I built last week. This had a deadline of going live at 11 AM, after getting the new product details and content I focused on this task. Fortunately, I was able to accomplish the update and publish the changes to production by 10:30 AM. The managers and peers congratulated me for being able to make the change in time. After lunch I obtained the design from the designer for the campaign page regarding product one and for the rest of the day, I continued to further develop the campaign page for product one, focusing on the style and layout of this variant.

Thursday

Today's main task is building the UI and style the campaign page for product one.

Today I focused on taking the designed layout and implementing the styles on the page. Using a mobile first approach, I started by creating the mobile version of the layout. By using the CSS flex box layout, I was able to the ground structure set up. I figured out that I needed icons for some parts of the design and confirmed with the sales managers which ones would be the correct ones. For the rest of the day I spent creating the layout with CSS.

Friday

Today was a quite different Friday, the focus for today was in internal actives with *Company X*.

Together with the Helsinki team, we travelled to Sweden to visit the new branch in Stockholm. The day was filled with internal activities to get a bigger picture of the company, current goals and future plans. I got to know the team better and also get to see how other branches are doing things. Even though this time wasn't focused on client-based work, I feel this experience was quite beneficial for the whole international team.

Weekly analysis

During this week my main focus has been on creating new pages in the platform and updating existing code instances that I've made in the past. I faced some obstacles while developing the core functionality for the new pages but by reverse engineering the existing codebase I've been able to further understand how the existing utilities function. Throughout the development of the mentioned tasks I've been able to identify and fix various bugs in the existing codebase which were not known to the team. These observations and fixes were praised by the team and it made me feel like I've contributed in a positive way.

I've gotten to interact and learn about new tools inside the browsers built in developer toolset. This occurred while debugging the cause of why one of the products values was not consistent in the UI and in the web shop cart. I've used the developer tools in the past, mostly using the console and network inspector. With these it was possible to view the status of the data in a certain point and understand the incoming and outgoing HTTP requests being made. For this use case, a deeper understanding of the user's session was needed.

According to (W3C, 2016) there are two ways of storing and tracking a user's session. The first one is using client-side *cookies* which work best in a place where a single transaction is made. Cookies have been used in the past for general purposes, but critical flaws have been identified in use cases such as when a user has multi window sessions running. By performing said scenario, if the user triggers a tracking action on multiple windows one of these actions would be lost from one window to another. This would lead to critical bugs in a system causing serious problems to the client and provider.

Based on the article, the best practice would be to use the *sessionStorage* attribute. This allows us to add and track data to a user's session which can be used in multiple pages and once the last page is closed the session ends (MDN Web Docs, 2019c) . While investigating this new attribute, along the way I've learnt to use a new built in feature in the browser known as *local storage* This is a place where web applications store data locally inside of the user's browser. (W3 Schools, 2019c)

The users cart session was being stored locally using the mentioned attribute of *sessionStorage*. Knowing where this was located and how to access it was critical information needed for the task.

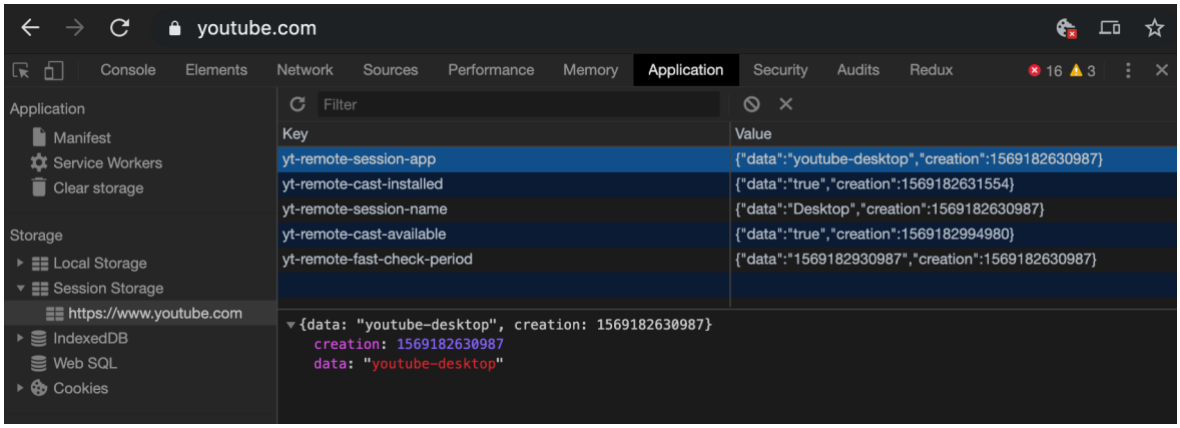

Figure 11 Example view of a Session Storage Object - Zapata, 2019

By combining the theoretical literature about user sessions and hands on official documentation (Basques, 2019) I was able to know how to create, view, update and delete data inside of the session storage object. After getting familiar with the workflow, I was able to pinpoint the value that was missing the session and why it was missing. Exploring the developer tools was definitely helpful for the task and in the future will drastically improve my technical performance.

3.5 Week 5

This week focuses on developing campaign pages and A/B tests.

Monday

Today's main goal is to complete the campaign pages for product one and two.

I spent most of my day creating and developing the style for the new campaign pages. Using a mobile first approach I was able to achieve a nice and responsive layout that could be used for both of the pages. In this task I also had to add analytical tracking, at first, I thought it was working correctly but once I published the first page, I realized that it was not tracking so I proceeded to debug the issue. I found out that I had not passed down the correct tracking information to analytical function.

I was informed by one colleague that a A/B test agent needed an update, this task was changing a color of an element which wasn't being applied on the mobile view of the site but on desktop it was. I investigated the JavaScript injection and saw that the actual element shown in the mobile version of the site was a completely different than the one shown in desktop, so I applied the color changes to both elements used in the two environments. By the end of the day I published both of the campaign pages to the live site and updated the A/B test agent.

Tuesday

Today's main goal is to create a new campaign page where the user can activate and choose a service plan.

In the morning, together with a fellow engineer we studied and analyzed the results of an A/B test which I made on August 20th. The hypothesis of the test consisted of changing the color of a filter button. What we wanted to find out was if this change would increase the features usage. Using Google Analytics and Conductrics it was possible to pinpoint the number of users who used the filters in both the original A and modified B version. The conclusion was that the usage of filters did increase with a positive conversion rate, I documented the findings and the test came to an end.

I had a meeting with the sales manager where we checked the two campaign pages published yesterday and they were according to the expectation. He wanted to add more text content and styling to one of the pages. Once I had all the new changes in order, I updated the pages and for the rest of the day I worked on a new campaign page. This page consisted of similar functionality that I did last week. By having my code in a modular set up, I was able to re-use the core logic and general UI components. After getting the core functionality up and running I spent the rest of the day developing style of the UI based on the design and specs given.

Wednesday

Today's main goal is to create a new page for an A/B test.

I spent the morning planning and brainstorming future stories with the team. After the brainstorming session ended, I had an introductory meeting with the new growth manager from *Client Y*. I got to know him better and also understand what his role is going to be in the team. After lunch I obtained a design for a new UI and the technical requirements for a new variant page. This page originally lets the user purchase services, the change that I should implement is to add one step in the purchase funnel.

This extra step consists of a modal dialog which should offer an add on service and then proceed with the funnel. The modal component has already been created in the codebase and it's the one I need to use it. I developed the page and got the modal showing but based on the requirements, the modal should close after the user clicks a button inside the modals dialog. With the current status of the modal component, if a button is clicked inside of it an action will be performed but the modal would stay open unless the user clicks the exit button.

In this use case, the modal needs to close after the user clicks the buy button. I spent the rest of the day trying different ways of achieving this functionality, at the end I got a working solution that would conditionally render the modal based on a Boolean value inside of the parent component. This works well for the situation and can be used in similar future scenarios.

Thursday

Today's main goal is to continue the developing the A/B test variant page.

I tested possible scenarios making sure the modal was working correctly and proceeded to take develop other parts of the UI based on the design provided. During this part of the development I had to add a border line to a specific element without changing its size. Adding the line to the element directly showed the style but the goal was to get the line separate from the element. By using the CSS property margin, I was able to add space between the element and the border line without changing the actual size of the original element.

Once I completed the UI for the content inside the modal, I proceeded to investigate how to make a button add an item to the cart. Since I'm still new to the codebase and how the flow of purchase works, the best way to understand this was done programmatically was to reverse engineer existing parts of the code that contain similar functionality. By the end of the day I got most of the UI and styles completed and managed to build a working buy now button inside the modal. For the time being the data passed to the cart is hard coded test data which I plan to continue tomorrow.

Friday

Today's main goal is to implement the functionality of being able to add an extra product from a pop-up modal.

I continued to develop the buy now button that I started yesterday, by understanding were the product data was fetched and passed down. I removed the hard-coded data I had and implemented the data fetching for the add on product. One challenge I did face was to make the pop-up buy button add two items in the cart. By using the browsers developer tools, I was able to identify that the items were indeed being added to the cart data set but in the UI, only one item was being reflected.

Checking the data structure, I identified that what I needed was an object with an array for each product. I replicated the data structure for mi use case, but it still didn't update the UI. To get an example of the correct data layout, I went on the production platform and added two different items to the cart. I overcame this challenge by having the button which opens the modal add the first item to the cart and then the optional button adds the second item if the user chooses it. In order to get the button to open the modal and send an item to cart was quite challenging but I managed to do it by using React Ref.

Weekly analysis

This week's focus has been around developing new pages and A/B tests. I've also had the opportunity to study and analyze the results of the A/B test that I've made when the diary began. By completing the A/B test cycle from start to finish I've gotten a better understanding of the value these tests bring to the project and also how it impacts the client's revenue. I've been also exposed to Google Analytics tools which feel a bit more familiar each time I use them. I've been able to identify workflow patterns in each of the tasks completed. In each new page I've been building there seems to be a minimal requirement of creating a new campaign code which is used to track sales and analytical data. Implementing the UI and functionality and either publishing the page via the CSM or Conductrics tool.

From the technological side I've gotten to solve multiple challenges with success, I've gotten to interact with different technologies related to front-end development and data structures. One of my lacking skills in the past have been CSS, I enjoy building and implementing functionality. Throughout the week I've created several UI components and styled them with CSS. One particular skill I've obtained is how to add style to a certain element by creating space between the element and the style without having to add additional HTML elements. For showcasing purposes, the goal is to move the red line to the top of the page without moving the element content.

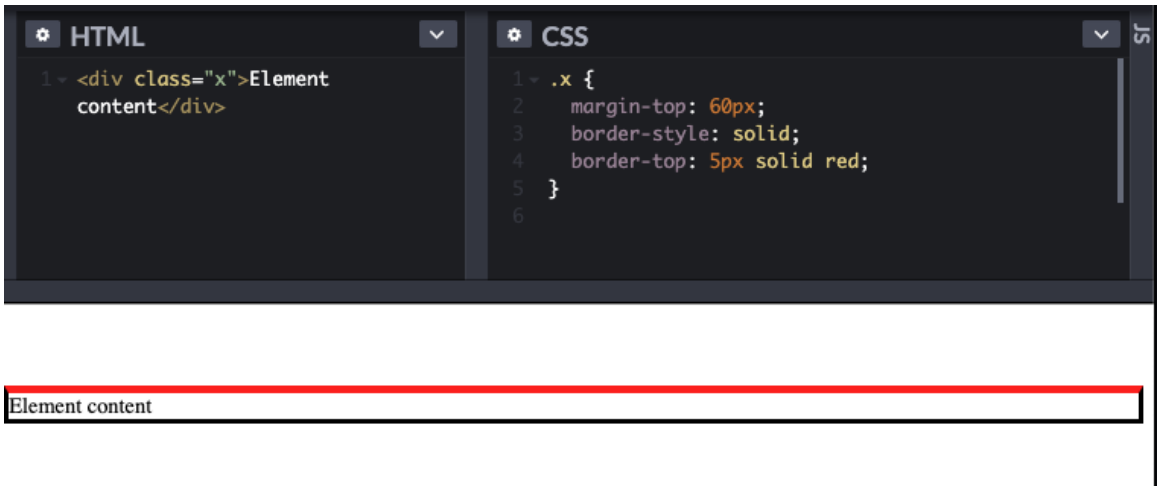

Figure 12 usage of margin-top - Zapata, 2019

My first approach to this problem was to add a margin style property which uses space around the element and outside its border (W3 Schools, 2019a). As seen in the image the element is being spaced but the red style is in line with the content. According to (Crozier, 2013) the correct way to achieve this style is by using the padding-top property which creates spaces around the content of the element (W3 Schools, 2019b).

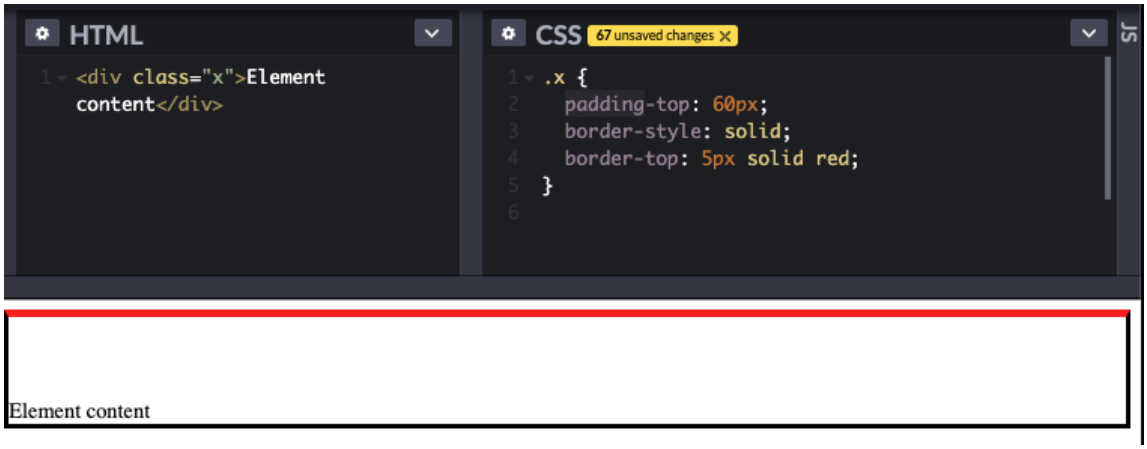

Figure 13 usage of padding-top - Zapata, 2019

By using the padding property, it was possible to add the required style without adding extra HTML elements to create extra space. this approach of using CSS instead of adding unnecessary elements is significantly better as it keeps the codebase clean.

While developing in React, using a modal component a significant amount of time was spent figuring out how to programmatically open the modal from outside of the component. In a traditional React scenario, this would be accomplished by the parent component passing props to the children. The change in the props would trigger a re-render and cause changes in the DOM (React JS Documentation, 2019b). In my scenario I would have a button which should trigger the opening of the Modal. The blocker here is that the button is inside the parent element and the access to one of the modals methods would be needed to open it.

This would mean that the child should expose its methods and have them available to the parent, in essence this required the workflow to be from bottom to top. This is something I've never came across and after trying multiple different ways of solving this. I've came to learn about React Refs. A ref is placed inside the constructor of a component which creates an instance property which then can be referenced throughout the component (React JS Documentation, 2019b) By creating refences in both the modal and the parent component both of them are able to communicate and expose their raw DOM node and its methods. By using the ref is was possible to achieve the required functionality of calling a function from the child in the parent component.

Overall this week has been quite productive, I've learned a lot about the project and feel more familiar with the workflow each time a task is implemented. My skills are progressing and I'm getting more familiar with the technologies used every day. I've learned to ask more questions from my peers and get an insight from their experiences.

3.6 Week 6

This week focuses on adding more functionality to previous work made and creating new A/B tests.

Monday

Today's task is to figure out how to programmatically execute two click events in one button.

Continuing the task from Friday, it came to my attention that the interaction from the modal caused an undesired behavior in the cart view. Once the user performed an action inside the modal, it caused the cart view to be closed. The morning was spent debugging the cause of this behavior. With the help of a senior developer we came to a conclusion, the behavior was a global click event handler being executed when there was a click in the modal or in the page. To get around this I decided to change the flow of the modal making the purchase button add two items simultaneously. It was new and quite interesting to understand how these click event handlers can be monitored and identified.

Tuesday

Today's task is to develop a UI concept based on a design sketch.

I finished implementing the functionality of the add on purchase with the modal. I tested all possible use cases and once I felt confident with workflow, I showcased the progress to the designer and sales manager in person. They were happy with the functionality and progress made; this was a great chance to clarify some design questions that I had in mind. After the discussion we agreed on a few changes to the design, once I obtained the updated sketch, I proceeded to implement the changes for the rest of the day.

Wednesday

Today's task it to combine two different pages and programmatically serve one variant randomly.

I completed the new style changes and created a new Conductrics agent to create an A/B Test on the page. In the previous tests I've set up, the changes between variants have been in a way that the base of the page is the same.

Presenting a small to medium change in the B variant. In this instance the change between the variants was a completely different page and logic. In order to do this efficiently I encapsulated each different page in a root React component.

This way I could use the value from the Conductrics API and conditionally render a different page based on the variant type. This kept both pages separated along with different logic and styles. During the day I interacted with my co-workers and asked multiple questions which helped me to achieve my goal and understand how different segment groups see the same product but receive different offers.

Thursday

Today's main goal is to update styles in the A/B test created yesterday and update content in existing JavaScript injections.

In the morning I was approached by a designer which suggested a few styles changes to the page that I deployed yesterday. A block of text should be broken in smaller pieces and in small device screens the text was overflowing. I spent some time updating the styles based on the requirements and pushed a fix to production. I tested these with different browsers and viewports making sure the text was showing up correctly.

After lunch I was given the task of updating content in 4 different pages by the growth owner. While doing the second page I realized that one of the images was being overwritten causing inconsistency. I spent most of my time investigating this and consulted my colleague, we both debugged the page and found out there is are 3 JavaScript injections in the page. One injection is from our team and the other two are from different teams.

Friday

Today's main goal is focused on internal actives in Company X and learning.

The day was filled with various learning opportunities, I attended the dev by-weekly meet up in Company X's headquarters. The main topic focused on web accessibility, as a developer it's our responsibility to make websites that are accessible for everyone. I obtained key pieces of knowledge on how to implement accessibility in existing sites. In the afternoon I attended another tech tack in Company's X parent company.

This focused on API testing using dynamic data and unit tests with the help of a third-party library. The second topic discussed was focused on React Native, I got introduced to new features in the framework, performance upgrades and tips and best practices.

Weekly analysis

This week has been filled with new knowledge and learnings. At the start of the week, a critical blocker was experienced. After finalizing the functionality of the additional purchase via a pop-up modal, it was realized that if the user declined the action, a separate cart view was closed at the same time as the modal. The first approach was to use CSS and JavaScript. I learned that is possible to identify an element by its CSS class name using the *Document.querySelector() Method,* this method returns the first available element that matches the parameter (MDN Web Docs, 2019L).

Once the element was identified it was possible manually trigger the opening of the cart view with JavaScript. Writing a function which triggered when the modal closed, forcing the cart view to stay open. This worked but if disabled the existing functionality of the cart view opening dynamically. By having the benefit of having fellow developers nearby, suggestions for this problem were asked. The team was happy to help, a senior developer debugged the cause of the behavior together with me. Along the way learning a new way of viewing the event listeners using the built-in developer tools in the browser.

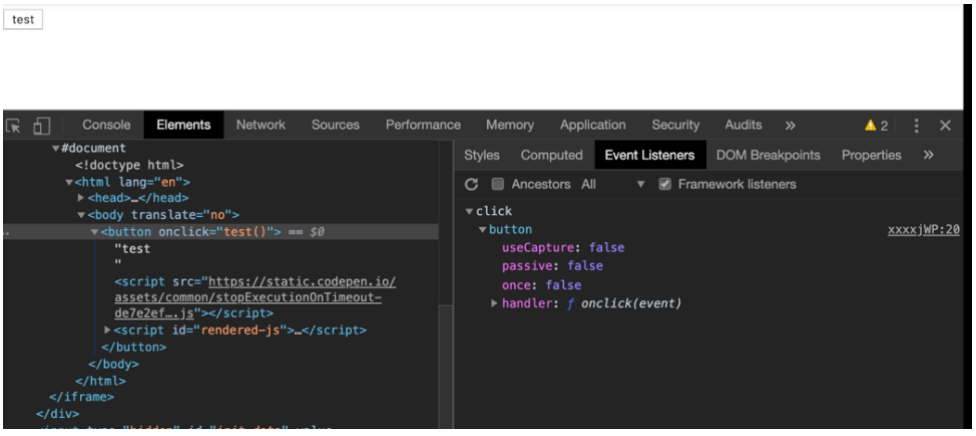

Figure 14 View of the Event Listeners for a button element - Zapata, 2019

Selecting an element on the page and using the Event Listeners tab, it's possible to view all events (Umar, 2019). It's also possible to view the actual script related to the listener in question.

```
20 <button onClick="test()">test
21 <script src="https://static.codepen.io/assets/common/stopExecutionOnTime
22 <script id="rendered-js">
23       const test = () => {
24   console.log('event executed');
25 };
26       //# sourceURL=pen.js
27     </script>
28 </body>
29 </html>
```

Figure 15 In depth view of script linked to an Event Listener - Zapata, 2019

The usage of this new tool made it possible to identify why the cart view was closed. There was an existing global event listener added by the Web shop platform on the body element of the page. This could not be overwritten for two reasons; the code was out of our reach as it part of another team and this listener was being used in all of the platform.

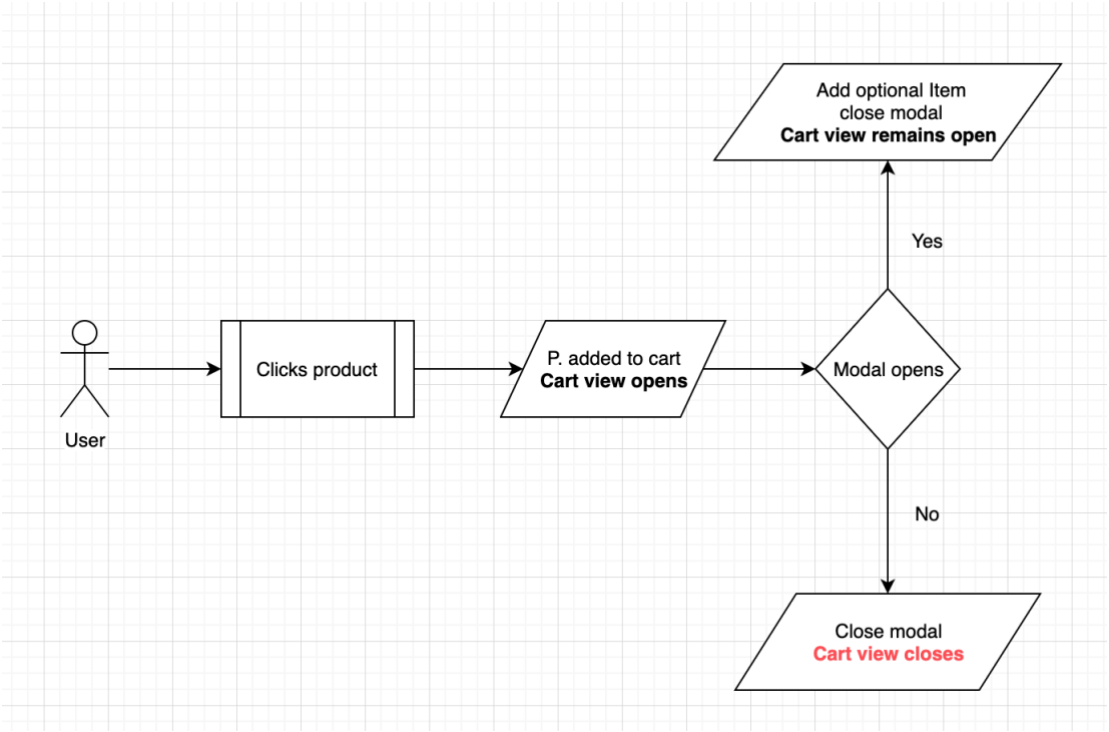

Figure 16 Current implementation of the add on purchase - Zapata, 2019

Another approach was need, going back to the drawing board it was time to change the functionality. The current add to cart components logic takes as input a single product. My goal was to somehow bypass this rule and add two items at time without affecting other parts of the code that depend on this set logic.

Recalling the debugging session with a senior developer, he showed me how to simulate a click on any element with JavaScript. I researched and found the *DOM click() method* can simulate a click on a DOM element (MDN Web Docs, 2019k). Taking that piece of knowledge, a new approach came in mind.

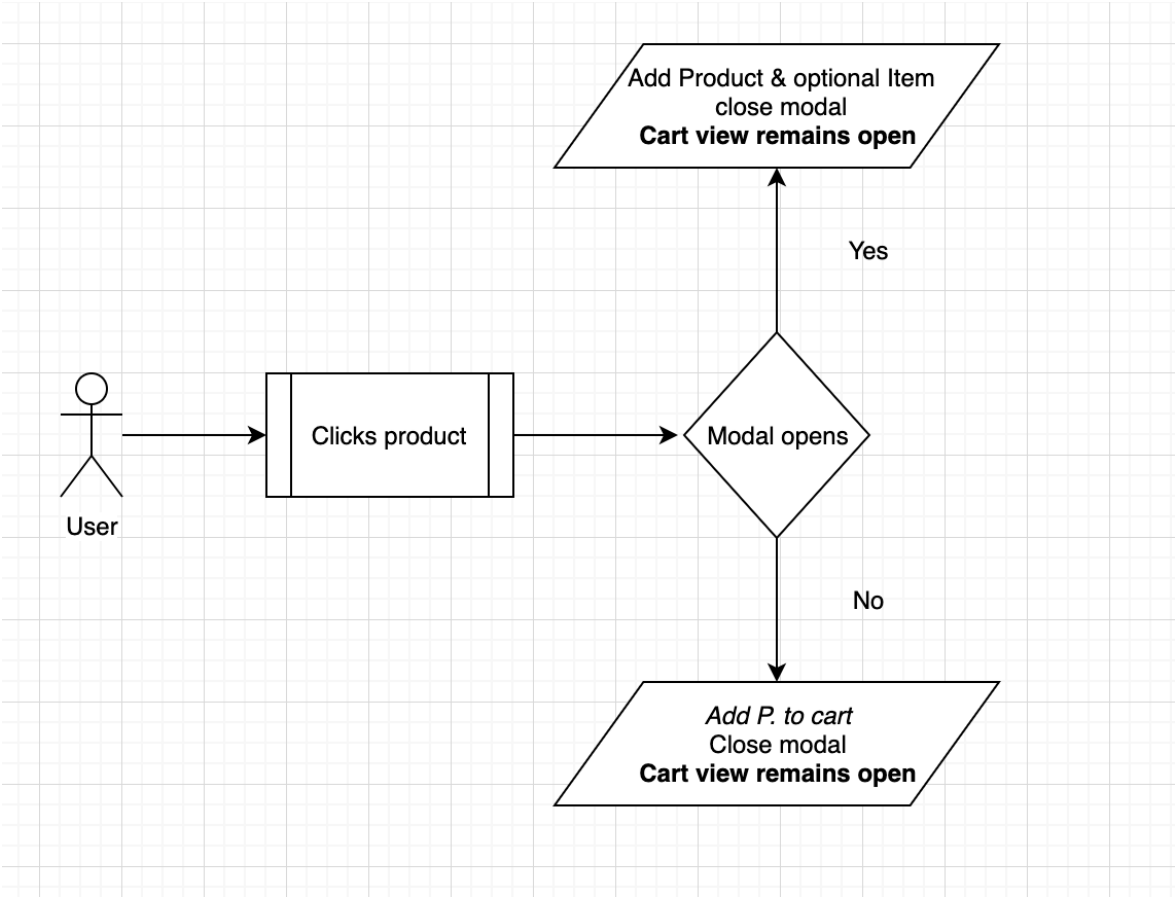

Figure 17 Updated add-on implementation - Zapata, 2019

When the user clicks the initial add to cart button nothing will be added, instead this button would only open the modal. Once the modal is shown there would be two visible buttons prompting the user if they'll want the add-on purchase. Visually the user would see the same elements as in the first approach but under the hood there is actually three add instances of the add to cart button in the DOM.

- Original product
- *Hidden* add-on item
- Original product

I learned about a new CSS property known as *display: none*; by applying this the element is visually removed from the UI (W3 Schools, 2019h) but it still present in the DOM. it was possible to have the add-on instance of the add to cart button hidden for the user. Keeping the functionality of the component available in React. The add to cart button accepts a prop for a callback function. This executes right after the item is added to the cart. In the first case the user wants the add-on item with the original product.

The original item would be added to the cart and that component would trigger a JavaScript callback function. This simulates a click on the hidden add-on item then the modal closes and the cart view remains open. In the case the user would not want the add-on product, only the original product would be added then the modal closes and the cart view stays open. The open cart view is only triggered when an item is added and is closed once the user clicks out of focus of that view. Keeping that in mind I understood why the first approach didn't work as expected.

When the user clicked the product button, the original product would be added to the cart opening the cart view in the background. Then the modal with the add-on prompt would open. If the user chooses to add the additional item everything worked as expected. In the other case of rejecting the add-on product, the modal be closed via a click event. By further investigating the cause with the developer tools, I was able to pinpoint a being triggered event listener that closes the cart view if the click is out of that view. This happened because there was an event handler at the top level of the DOM.

I came to learn that this concept is known as *GlobalEventHander* which it at the top level of the DOM mixing the Document, Window and Element handlers (GlobalEventHandlers, 2019m). This gave me insight on understanding the problem and the cause of the unwanted behavior. By combining CSS, JavaScript and outside the box thinking it was possible to achieve the end goal. Even though I'm not the fastest to complete the development tasks this week has an overall good feeling. At the end of this week I've improved my debugging and logical skills by problem solving. I've also gotten more familiar with my team; I've learnt to not be afraid of asking questions and speaking to them face to face.

3.7 Week 7

This week's focus is on developing new features, JavaScript injections and analyzing data from an A/B test.

Monday

Today's goal is to add more functionality to an existing page.

After the daily meeting in the morning I was given a time sensitive task by the growth owner. The task consists of adding a permission toggle option for both variants of a running A/B test page I created last week. The user ID's who accepted the permission would be collected into the platforms database. This feature had to be ready by the end of the day as it was planned to be used tomorrow in a new campaign. More information regarding the task was shared via Slack but it was quite broad, and I needed to clarify with the designer and sales manager critical details in order to get the full picture of the feature.

After having a long thread conversation, I decided to ask in person the remaining details from the growth owner. This was the first time interacting with the platform in such way, so I consulted my team mates to get a technical insight of what was needed to accomplish this task. Even though it was a long hard day, at the end I was able to complete the task at hand within the time frame specified.

Tuesday

Today's main goal is to update an A/B test and team building activities.

The morning consisted of an internal meeting with the sales manager and the development team. In the meeting it was decided to postpone the permission feature which I had built yesterday. A new campaign was scheduled to be released in a couple of hours affecting the page where this feature was located. I removed the code for this feature in both variants of this page and released the update to production. Once these changes were made, the campaign was launched successfully.

In the afternoon the whole team from client Y headed to Company X's headquarters for a retrospective meeting. This was a great learning opportunity to identify what the team is excelling at, possible risks and target areas which could be improved. The rest of the day was filled with team building activities, I had fun and felt I've bonded more with the team.

Wednesday

Today's main goal is to create a new feature for an upcoming promotional campaign.

I was given the task to create a form inside of the sub navigation bar of the platform. This element was part of the web shop codebase and I didn't have direct access to modify the source. The best possible approach was to create a JavaScript injection that would target the navigation bar element and insert a React element. I managed to target the correct element using Vanilla JavaScript, but I did notice that if the navigational bar was toggled the injected content was visible only in the first render. I investigated the cause and found out that the navigation bar was a React component which changed the HTML structure dynamically when the nav bar was toggled.

I spent quite some time investigating how to keep the injected content shown at all times, but I didn't find any suitable solution. After a couple of hours, I decided to ask my peers and I was pointed in the right direction. Since JS injections is common practice in this project, the team had a set of existing helper functions in the codebase that could be used in this scenario. By the end of the day I created a basic JS injection that kept the inserted element in place in the desktop view.

Thursday

Today's main goal is to continue developing the JS injection which I started yesterday.

While the desktop version of the injection was behaving correctly, I realized that in mobile view the element tree structure was different. I expanded the implementation to toggle between these two and make the content injection in the right place. Manipulating the *DOM* manually with JavaScript is a different approach from what I'm used to doing. I learned a lot about new methods provided by the *DOM* node object which I used during this day and also got more familiar with the helper utilities in the project. At the end of the day I feel I made some vast progress in this task which will be continued.

Friday

Today's main task is to re-implement the feature built on Monday.

I spent the morning analyzing the test data from an A/B test started earlier this week. Based on the sale rate of each variant it was clearly visible that the original implementation of the page created more sales. I presented these findings to the team where it was decided to stop the test. Keeping the original variant of the page in production. Shortly after that I received a message from the growth owner requesting, the addition of the permission feature to this page.

He mentioned this task was time sensitive and had to be completed by the end of the day. After adding the feature, I identified a problem running it in production which required changes in the current implementation of the permission feature. This was definitely a challenging day but the end of the day I was able to get a working solution of this feature and meet the deadline successfully .

Weekly analysis

This week has increased the difficulty of the tasks and responsibility in the project. I've been able to further develop my communicational skills and development skills by learning to interact with the DOM manually.

The main place where the team synchronizes is in the daily team meetings, specifically the Monday meeting. This gives a general picture of what the focus of the week will be. Other general internal announcements are made via Slack. This flow of communication is quite efficient and great to broadcast to a larger audience. However, in situations such as a narrow-scoped discussion it can be hard to communicate the same idea to every party. This was the case during the week, as multiple clarifications were required to get everyone in the same page.

A task was assigned with a broad description that involved various stakeholders. Asking direct questions on Slack helped but not all parties voiced their opinion. Before developing can start its crucial to understand all requirements and specifications. By utilizing the benefit of the team's location, it was possible to have a face to face discussion with all the parties involved.

It’s crucial to ask questions that are effective yet keeping them open-ended to get a better insight from the respondents (Ray, 2016). I’ve learned that it’s a good practice to ask as much questions needed using a friendly approach with words that are easy to understand from everyone’s point of view.

From a development perspective there were two problems faced this week. The first one was regarding development of a feature which should affect parts of the platform outside the project’s codebase. The task itself was something manageable but having no access to the source code raised the level of complexity. In cases like these the teams suggested approach is to use JavaScript injections to manually modify the DOM. The goal of the task consisted of adding content inside of a nested navigation bar.

By asking questions to fellow colleagues it was possible to discover and get a deeper understanding of the existing helper utility functions inside of the project. By combining helper functions and built in DOM object methods it was possible to modify the navigator without the source code. Using methods such as *querySelectorAll( )* which returns a static NodeList of all the matching elements passed as parameters (MDN Web Docs, 2019f) and utilizing the *querySelector( )* which checks the HTML document from top to bottom and return the first Element that matches the selector (MDN Web Docs, 2019g). I’ve gained a basic understanding how these methods worked, with them it was possible to target specific parent elements in the DOM. While using these I’ve learnt that it’s also possible to target nested elements which was a requirement to successfully target the inner elements of the navigator.

Getting up and running took time as it was important to learn how to properly use the tools mentioned. Once the specific elements were identified and found with the query selectors the injection was able to successfully inject new elements. This was just the tip of the iceberg, right away it was noticeable that if the navigator was toggled the injected content was not visible anymore. Using the browsers developer tools really helped in this situation, making it possible to pinpoint what elements changed when the navigator was toggled. The current implementation of the injection was not displaying the injected content in smaller viewports.

In theory this would mean that there should be two injections with the same code in the page. According to (Diomidis, 2006) having code repetition is considered bad practice as it leads to unexpected bugs in the codebase and it goes against the single point control principle. This states that each function or method should be encapsulated and have a single responsibility over the program. (Martin, 2006).

By following this principle it's possible to keep the code in a modular set-up by breaking the main function into smaller methods that are easier to maintain, read and re-use. In order to avoid code repetition in this scenario I've created a check to see if the root element was from the desktop view or mobile.

By storing the results of the targeted elements in JavaScript variables it was possible to create Boolean conditions. By using these it's possible to check if an element existed and programmatically execute code. Inside the conditional check an inner method would be called which takes in a parameter for an elements class and in return, it injects the correct content to the DOM. This brought the functionality needed in both use cases by keeping the codebase free of code repetition. This is a new skill that will certainly be useful in similar scenarios in the future.

The second problem encountered was regarding a feature which consisted of a permission toggle box. If the user checked the box, once the purchase button is clicked a call-back function would make a *POST* request to the platforms database and redirect the user to the cart page.

Building the UI React component with the toggle to the page was straight forward and the added functionality worked in the local development environment. This task was completed early in the week but was postponed by management. Later in the week it was decided that this feature should be added. This particular feature request occurred almost at the end of the day when the rest of the team had left the office. After deploying the solution to production and checking the database, the permission data was not being stored.

Upon further investigation, using the browsers network inspector tools, it was possible to see that the permissions request wasn't being executed. Investigating this issue took time, coming to a conclusion that the redirect happened faster than the time it took for the call-back function to execute. The best scenario would have been to delay the redirect which required more time to implement. This was a time sensitive task and it required a fast-innovative approach. To get this to work the functionality of the toggle had to be changed so that once the user checks the box and the request was sent immediately.

This task was quite tricky as the problem only came once it was in production. In the future the deployment times should be targeted at the start of the day to have more time and resources to tackle possible problems like these. Even though the week was hectic I've managed to contribute to the project while learning along the way.

3.8 Week 8

This week focuses on developing JavaScript injections and new landing pages.

Monday

Off work.

Tuesday

Today's main goal is to develop a JavaScript injection that inserts content in the navigational bar.

By the end of the day I learnt to attach a call back function to click event listeners which are triggered in specific events in the DOM. When certain elements are pressed this would trigger a method that inserts the new elements in the page. There is still development to be done as the injection works to an extent. Depending on the viewport size the correct target element is selected and the event listener is attached but if there are changes in the viewport size after the injection runs the content is not inserted.

Wednesday

Today's main goal is to further expand the navigational bar injection and remove obstacles in the implementation.

I spent the morning investigating and brainstorming possible ways to overcome the challenge of dynamically running the injection if the viewport size changes. When the viewport size is changed, HTML elements are being discarded and regenerated by React thus the injected content from the injection was not present. To overcome this blocker, I came to a conclusion of adding an event handler to the window object of the page. Each time the viewports size changes a method would be called. This first checked if the elements were from a desktop or mobile viewport then checked if the injected elements existed and if they didn't it would proceed to insert the desired content.

Thursday

Today's main goal is to develop a new landing page.

At the start of the day I received confirmation that an upcoming product will be released and required a landing page. My task was to gather all leads and content requirements along with a fresh design template from a designer. By the end of the day I was able to develop the base structure of the page based on the design and requirements.

Friday

Today's main goal is to continue the development of a landing page.

I spent most of the day implementing the design template styles and content into the page. In the afternoon it was decided that a couple of more images would be added to the page and I was able to implement them in time. By the end of the day the landing page was successfully deployed and finished.

Weekly analysis

This week has been filled with new challenges and learnings along the way. There are noticeable improvements in communicational skills with interaction and dialogue with members of the team. By being in daily contact with front-end technologies such as HTML,CSS and JS, building new landing pages seem straightforward and efficient. There are challenges in situations when it's not possible to have direct access to the source code of all the parts of the platform. Adding or modifying all parts of the platform is possible but it can be complex sometimes.

During this week I've focused on a task that consisted of adding content to the navigational bar. Our team doesn't have direct access to the source code to that section in question. There seems to be a repeating pattern of using JavaScript injections in the project and each time one is needed it's a great learning opportunity filled with challenges along the way. It came to my attention that the navigational bars elements were managed by React. Although I'm proficient in the framework, changing the content of elements managed by the framework turned out to be unstable.

The JavaScript injection ran once when the page loaded and inserted content in the navigational bar. This worked correctly but if the navigational bar was toggled, React would discard the elements when closed and re-create them when opened. When this happened, the injected content would be discarded and not inserted again. Investigating work arounds for this issue took some time. After trying different approaches and techniques with no success I decided to ask for help.

By reading about the topic of communication and understand the different ways of asking the right questions (Ray, 2016) I've started to put in practice these gained skills. Being open with my teammates and ask more questions when I'm stuck in a problem. In this case a senior engineer pointed me in the right direction and introduced the *addEventListener()* method by creating a small example. This method is available by the browsers HTML DOM and provides the ability to attach a custom call-back function to any of the available events such as a click, hover, etc. event to an element in the DOM tree (Hope Computer, 2018). I've come to understand that a call-back function is passed as a parameter to another function which then utilizes the past function (FCC, 2019). In this case the call-back is only executed when the event is triggered. In my scenario this fit perfectly as it made it possible to run the injection only when the navigational bar was toggled.

By expanding the injection further, it was possible to not only insert but clean up the inserted content when the navigational bar was closed to prevent a memory leak. The injected being a React based component, I learned to use the *ReactDom.unmountComponentAtNode() method*. This method removes a React component from the DOM and once finished it returns a Boolean value which can confirm if the element was removed or not (React JS Documentation, 2019f). After implementing these changes, a new challenge came up when resizing the page. The injection would work correctly only if the size of the viewport stayed the same but if it was resized from desktop to mobile the injected content would be removed.

The root for this behavior was that the parent navigational element was different in mobile and desktop. Taking some time to learn further about event listeners in general, I've learnt that it's possible to add a *resize* event listener to the window element (W3 Schools, 2019f). The call-back function then takes care of checking if the injected content was already there and if it was not present it would insert accordingly to the viewport size. By utilizing the event handlers, it was possible to complete this task with all its challenges that rose along the way.

3.9 Week 9

This week focuses on further development of a JavaScript injection in the navigational bar.

Monday

Today's main goal plan future team objectives and update a campaign page.

During the morning the team focused on discussing what has been done last week and planning what will be done the upcoming week. Showcasing a clear vision on the team goals and brainstorming how we will get there. After the meeting I was assigned the task for updating a campaign page. By having a verbal discussion with the stakeholders, I was able to grasp the main points of what the update would require and include. For the rest of the day I focused on implementing the required changes in the campaign page.

From the technical perspective, this task focused on front-end development. By being in constant interaction with the platforms UI, I feel I'm getting more familiar with CSS styling and JavaScript. By the end of the day I was able to complete the update and deploy the new campaign.

Tuesday

Today's main goal is to further develop the navigational bar JavaScript injection.

During the morning team meeting I was informed of a change in the requirements on a previous task I completed last week. This was concerned the existing navigational bar injection. Based on these changes the injected script should only execute for users who are known competitors in the platform. I haven't done any similar tasks before, so I reached out for help. By speaking with my team members and explaining my current situation I was introduced to a couple of available methods available from the web shop platform.

I experimented with the new set of methods introduced and examined the returned data from the web shop in each one. I found a particular method that returned data regarding the existing user session. This included the IP address of the visiting user and other meta data. This was exactly what I was looking for, With this data at hand it would be possible to programmatically execute the script only for the targeted user segment.

I spent the rest of the day expanding the implementation and adding more functionality according to the new requirements. At the end of the day I was able to integrate the IP address check to the injection. This guaranteed that the injected script would only execute for competitor users who visited the page and not for known customers.

Wednesday

Today's main goal is to continue developing the JavaScript injection, showing unique offers to selected competitor segments.

After the daily meeting more requirements were determined for the navigational bar feature. The sales team agreed that the injection should show unique offers for each competitor segment. The current implementation showed the same offer for all competitor user segments. I needed a way to differentiate each user segment in order to be able to display unique offers. I decided to use the IP address method and further expand the identification process of the user's segment.

Utilizing the available utility functions provided by the web shop platform I learned that it was possible to identify which group segment the competitor user belonged to. I made incremental changes to the way the injection worked. Now the first thing the injection does is to execute an IP check. Passing the retrieved IP address to a second method which checked if the user belonged to a particular segment. Once the user's group segment was identified, the data is passed as a prop to the React component in the injection script.

Once the component is rendered in the UI, using the provided prop the appropriate offer is fetched in the lifecycle method of *componentDidMount()* and displayed to the user. By the end of the day I was able to implement the new required functionality and test the dynamic offer rendering for each competitor segment.

Thursday

Today's main goal is to create a style based on a template design and learn how to implement a required feature.

I spent the morning discussing a new design for an existing page with designer. Once all the displayed data and design were agreed upon, I obtained a design sketch from my colleague. The changes were design driven focusing on highlighting certain pieces of data and improving the color scheme.

After lunch the growth owner suggested a major change in the workflow of the navigational bar injection. The idea proposed would alter the functionality of the feature in multiple ways.

In the current solution, once the user submits a form this triggers a page re-direct. In the new page the user can choose to continue with the offer product and complete the purchase. The proposed idea would be to let the user add the offered product directly to the cart once the form was filled and submitted. The new approach would make the purchase flow more direct. I spent the rest of the evening studying parts of the codebase to get a deeper understanding on how to implement the requested functionality.

Friday

Today's main goal is to learn and implement a one click buy feature.

In the morning I continued learning by reverse engineering similar implementations of instant purchases in the codebase. After understanding the current implementations, I had a clear picture of how the logic and components worked. Taking the current product data available, I created my own instance of the cart order object. Once the data was parsed in the appropriate format, I moved on to implement the one click purchase button functionality. I had to manually test the functionality assuring the solution worked in all possible use case scenarios for all targeted user segments.

Now that there are two different implementations of the injection, I decided to encapsulate shared logic and create smaller and reusable React components of common elements. This made the code modular and easier to maintain in the future. By the end of the day I was able to accomplish the task at hand and create new business orientated opportunities.

Weekly analysis

This week has revolved around a feature for the navigational bar. Requirements and complexity have increased for this task, pushing me to learn more about the project's existing resources and workflow. I've been getting more familiar with manual DOM manipulation and creating styles with CSS. I've also learnt to ask for clarification from key stakeholders and obtain help from my colleagues when needed.

The main problems faced during this week were development orientated. In the middle of the week there was a major requirement change which affected the workflow of the feature. Changing from an option-based purchase flow to a one click purchase button. This change required me to develop a new solution from scratch. I had an idea how to accomplish this but there were many crucial details that were unknown.

By expressing my concerns and questions, it was possible to obtain help from other colleagues. They guided me towards the right path by showing me similar implementations of what needed to be accomplished. Facing a similar scenario, a couple of weeks ago, I would've spent more time trying to figure this out before asking for help which would slow down the development progress. I've learnt to count on my teammates learn from their shared knowledge.

The task at hand required that the script should persist the injected content in all pages in the site where the navigational bar is present. By reverse engineering other injections in the codebase and asking clarification of how these worked I learned about available utilities and was able implement the appropriate functionality for this use case. When the page changed, a call-back function would execute and clean up any existing injected elements in the previous page. Once the new page loaded another call-function would re-run the injection script in the new page and insert the injected content. This persisted the injection and its content in all pages in the platform.

While developing this task and manipulating the DOM I had to bind a function to three child elements inside of a parent *DIV* element. The first approach was to add bind the function three times manually which worked but wasn't efficient because there were three instances of repeated code. I was able to overcome this first targeting the parent element using the querySelector() method which checks the document elements and returns the first Element matched (MDN Web Docs, 2019g). Then storing the result as a refence variable called *parentElement* in JavaScript and then iterating through the child elements.

The goal was to bind a function to each element in an efficient way. In JavaScript it's possible to use the *forEach()* method which iterates over an array of elements and executes a provided function for each element (MDN Web Docs, 2019i). First trying to iterate over the result of the *querySelector()* did not work, an error in the console stated that the forEach() method is not a function. The iteration was possible by accessing the parent elements *childNodes* property. This returned a live list of nodes which can be individually accessed and iterated (w3 Schools, 2019g).

HTML

```html
<div id="parent">
  <button id='btn1'>child 1 </button>
  <button id='btn2'> child 2</button>
  <button id='btn3'>child 3 </button>
</div>
```

CSS

JS 100+ unsaved changes

```javascript
const firstApproach = () => {
  const element1 = document.querySelector("#btn1");
  const element2 = document.querySelector("#btn2");
  const element3 = document.querySelector("#btn3");
  element1.addEventListener("click", () => {
    console.log("manual click Event Listener");
  });
  element2.addEventListener("click", () => {
    console.log("manual click Event Listener");
  });
  element3.addEventListener("click", () => {
    console.log("manual click Event Listener");
  });
};

const secondApproach = () => {
  const parentElement = document.querySelector("#parent");
  console.log(parentElement);
  parentElement.childNodes.forEach(childElement => {
    childElement.addEventListener("click", () => {
      console.log("click Event Listener");
    });
  });
};
```

Figure 18 Code snippet comparing two function approaches - Zapata,2019.

As seen in the code snippet above both approached worked but the first approach was manual and repetitive with a long set of code lines while the second approach is drastically shorter, functional and easier to read and maintain.

By the end of the week there were two different implementations of this feature. Both implementations shared similar pieces of logic and UI elements only differentiating on the actual purchase flow. It came to my attention there was pieces of code repeated in each implementation. Keeping in mind that code repetition is a bad practice (Diomidis, 2006) I decided to pin point all shared elements and abstracted them into smaller functions of code which then can be re-used in both places. By doing the code refactoring now, it will be easier to maintain in the future.

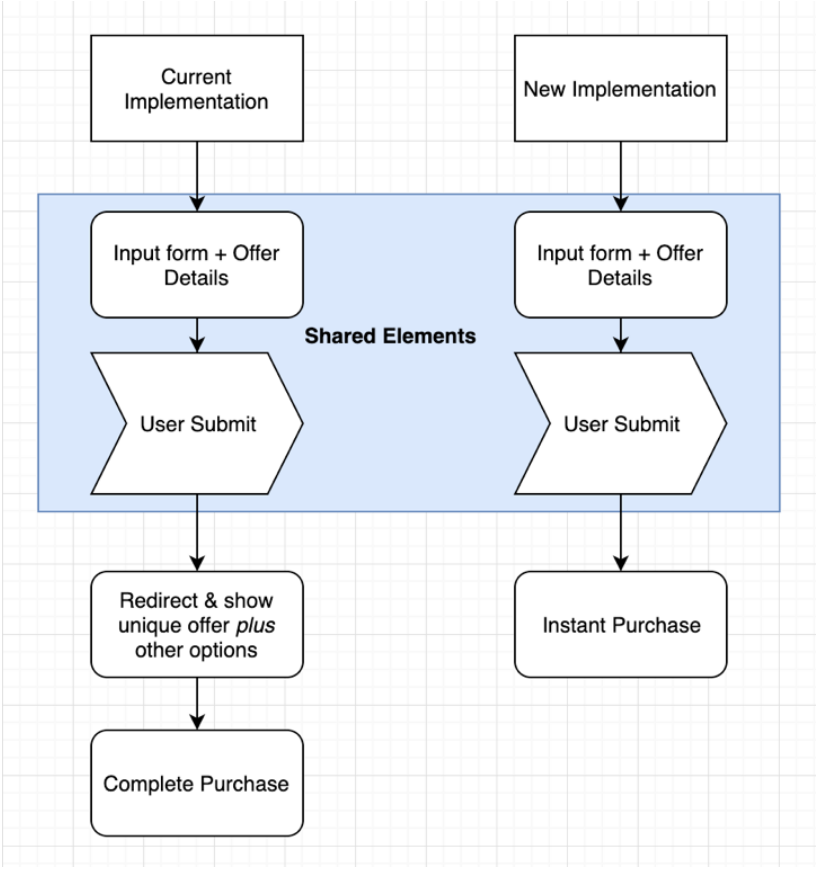

Figure 19 Diagram of two different implementations & shared elements - Zapata, 2019.

Although it a couple of hours to refactor the code of the injection it will bring positive results in the long run. According to (Peter Andreae, 1993) there are multiple advantages of having smaller chunks of reusable code, the biggest one is to have a single source of truth. Preventing inconsistences between each instance that uses that piece of code and saving time when there is a need for an update. This made the whole file a lot easier to read for myself and other developers. Now both implementations could be separate yet still share their common elements without having redundant code. Taking that in account this opens a new possibility that this injection will become an A/B test in the future.

3.10 Week 10

This week's focus is on further development of JavaScript injections, developing new features and A/B tests.

Monday

Today's main goal is to improve the navigational bar injection and setup a new A/B test.

I started the day by testing the usability of the navigational bar with the injected content. I noticed an inconsistent behavior that occurred when the user would perform the purchase action. By reviewing the injection script together with a colleague, we were able to identify the root cause of the inconsistency and I implemented an appropriate fix.

Once I tested the fix, I shifted my focus on creating a new A/B test for an existing page. The test includes visual changes in the UI. Modifying the color scheme and styles of certain visual elements with a goal to see if these visual changes will impact user interaction. By the end of the day I was able to fix the unwanted behavior in the navigational bar and deploy the new A/B test.

Tuesday

Today's main goal is to update an old Git branch and create a new promotional banner.

The morning consisted of bringing a three-month-old git branch up to date with the latest's changes in the develop branch. Being a relatively large team, there was a vast amount of incoming changes with a large number of merge conflicts to be handled. I carefully tested incoming changes and fixed them according making sure not that these changes wouldn't affect other parts of the codebase. By the middle of the day I was able to get the git branch working together with the latest changes from the main git branch in the source control tool.

In the afternoon I meet with a sales manager and a designer for a brainstorming session. During this meeting it was agreed that there will be a new design for a banner which would replace a generic product card. Taking in account all requested requirements and wishes from the stakeholders for this feature, I got a clear scope of the goal of the new banner.

Wednesday

Today's main goal is to develop the promotional sales banner.

I spent the day developing a new react component for the feature discussed yesterday. My focus was getting the main functionality working so I started by creating a new campaign code with a POST request to a specific endpoint in the platform. Inside of the requests body I specified a set of visibility groups and set the amount of available uses. With the campaign code ready I wrote a function which took in two parameters, a product ID and a campaign Code. This function returned a JavaScript object of product data with the correct price and offer details based on the specified campaign.

I moved this function and used it in the *componentDidMount()* life cycle method of the component to fetch the product data for the sales banner. Once the main functionality of the banner was working, I focused on styling. By taking the design sketch obtained yesterday I developed the styles accordingly. By the end of the day I finished the new promotional banner and deployed it to production.

Thursday

Today's main goal is to plan the development of a new feature.

During the daily meeting I was introduced to a product page in the platform which required development work. My task was to develop a new feature which consists of adding a three-step input field with instant validation. I took notes on all the requirements and wishes from the stakeholders and proceeded to plan the implementation. By having an open discussion with my colleagues, I was able to find possible API's that I could use for input validation and fetching data for the suggestion feed. I spent the rest of the day reverse engineering the use of the API's and planning how to combine and parse both validation and suggestion data.

Friday

Today's main goal is to continue developing yesterday's task by integrating input validation.

I spent the morning investigating how to combine both API's needed for the input validation. I came to a conclusion that with the current available data returned from the API was not enough to generate suggestions data. I shared my findings about the data restriction on the API with a senior engineer and he agreed with my conclusion. In order to obtain suggestion data based on the input it will require changes in the API's functionality. The API in question is developed and maintained from an external team.

By having a meeting with the growth owner and sales manager it was agreed that the feature will continue with the input validation and the suggestions will be postponed. I contacted the external team and made a request for required changes in the API. I spent the rest of the day developing the UI elements and adding the input validation. By the end of the day I completed the feature and deployed the update to the production environment, completing this task.

Weekly analysis

This week has been very productive with a varied mix of different tasks at hand. These tasks have further improved my development as a professional. Obtaining new technical knowledge as well as improving communicational and team working skills.

The major technical problem encountered during this week was regarding the usage of the web shops APIs. A feature required the validation of input data on the client side using an API. When the task was assigned, I knew the goal of what needed to be done but the technical details of which API to use in the integration was not specified. I was shown two existing places in the platform were input validation was being used. By reverse engineering the example implementations two API endpoints were identified.

It was possible to understand how they work in general but not their full usage. The lack of documentation on these services made this information restricted. My initial thought was to combine both services to accomplish the task. After running multiple test use cases it came to my attention that both API's returned similar data but in certain scenarios there was no data returned. This would be result in inconsistent results if API number one would not return data and the second one would need that data to achieve the required validation.

There were also major differences in important parameters such as the ID attribute. By counting on my teammates, it was possible to get a clear scope on why these differences occurred. Thanks to the discussion it was possible to understand what the real use cases for each API service were offered. By explaining these limitations of the API services to the growth owner and sales manager the task requirements were changed and it was possible to complete the feature.

According to (Jeffrey, 2000) code without documentation causes wrong assumptions, unnecessary in-depth examination of the code and loss of a vast amount of time. This is exactly the scenario that was presented this week. In future I'll be adding documentation code which I'm an author. This will provide valuable insight to the next person who will encounter it. Making it easier to get an understanding on what the code accomplishes and what features it provides.

At the time when a new software feature is being planned it's important to analyze and truly understand its requirements. Getting a deeper insight of what the current problem and how it could be solved to achieve a greater goal. Each stakeholder should have at least one representational member present in this session. This should include a software architect which understands what is technically possible to achieve in order to prevent unachievable requirements (Jäälinoja, 2004).

Once the requirements are agreed upon and set, it's important to ensure that a proper validation is conducted in each requirement (Boota, Ahmad, & Masoom, 2014) . This should be done after the brainstorming session and before the start of the concept's implementation. This will assure that the requirement is still relevant and its valid. Over the time span of my presence in the project there have been many instances where the specifications of requirements have not been clearly set. By the time a concept reaches a development some requirements are not relevant or are missing important details.

This has caused a vast amount of time being spent constantly trying to obtain the correct specifications. Questioning what the real goal is and what are the correct project utilities to be used. As the project's platform is large with multiple different parts and utilities, there are many places which haven't been explored. Even though I'm technically proficient in programming and the technologies used in the project. It's not possible for me to know how legacy parts of the systems work or API's which haven't been used before.

The lack of information on the requirements can be linked to communication as it plays an important role in software development. During the week there were multiple occurrences where technical clarification was needed. Each new task assigned touches different resources and parts of the platform. By verbally expressing possible blockers to my teammates it was possible to get a deeper understanding of the situation and understand what needed to be done to overcome the blockers. This helped understand details that were unknown before and increased the speed of development.

There were tasks that needed further clarification from the stakeholders as well. One particular occurrence was on Tuesday's task which was about the development of a new promotional banner. The initial requirements and specifications of the task given were quite broad. Once the task was completed and deployed, the sales manager and designer didn't agree on the placement of the banner. By creating a group message on Slack, it was possible to have a detailed discussion involving all parties.

Being the starter of the conversation, my first thought was to write a direct question demanding a specific answer. By recalling what (Ray, 2016) stated about asking friendly, clarifying questions. It was possible to observe a positive response from both parties. Each mentioned their valid points defending each other's view on the subject. Instead of favoring one side over the other, I decided to state follow-up questions on each valid point made. This created a great flow of ideas and discussion between all parties. Once all stakeholders agreed on the placement of the banner, the feature was successfully implemented.

4 Discussion and conclusions

Over the period of the diary entries I've came to learn substantial knowledge as I've grown as a professional and as a person. The scope of this chapter will be on analyzing the what has been learned, methods used for learning and highlighting issues along the way.

4.1 Personal development

I feel I've developed positively in countless ways throughout my time in project Y. Being a new member to an existing enterprise level project was an interesting and compelling challenge. From the variety of tools and technologies used in the project, I was primarily familiar with React and modern JavaScript. These technologies have been a solid foundation which I've been able to expand drastically. At the beginning of the project I felt my skill level was between a novice actor and a skillful performer.

The tasks assigned were generally understood and for the most part, I was proficient in how to technically implement the solution required to accomplish the task. Although most required substantial guidance and clarification in both technical and business aspects. By keeping an open mind I've learned to try new ways of working throughout project. It took me multiple weeks to get familiar with the unknown tools, technologies and methodology used in the project. These included concepts such as A/B testing, understanding how analytical tracking worked and manual DOM manipulation using JavaScript injections.

Looking back in my diary entries I can see that the most technical challenges were manual DOM manipulation with vanilla JavaScript and implementing analytical tracking. By implementing tasks which utilized known technologies I was able to further expand upon them and integrate the new tool set. Combining key literature about the subject, online resources and examining the existing platform code. It was possible to build upon and improve my existing knowledge with new technical approaches and best practices. These will be of great value in the future when similar scenarios are present.

This resulted in excellent progress to the team as my contributions brought value to the project and also to myself. On a higher level there have been multiple instances where I questioned what problem I was truly trying to solve. There were many unclarities as the specified requirements were occasionally general and vague. This related to miscommunication and poor requirement planning.

4.2 What I learned

I've come to realize that problem solving is the core of software development. During this period, I have found that before actually starting to code there are certain elements that must be taken in account. It's best to start by focusing and truly understanding what we're trying to solve and achieve with the code. We must remember that code boils down to binary instructions which are executed by a machine. If the instructions given to the author who will be writing the code are not clear, the code interpreted by the machine will certainly be unclear as the machine only reads and executes what is given in the instructions.

When a requirement or specification given is too broad or not clearly defined this leaves a grey area of uncertainty. With no clarification or details the developer must work with that he/she is given and provide a result. When the stakeholders see that the end result is not what they expected or wanted, the task must be rectified. This involves starting the development cycle all over again. Spending extra time, effort and resources from the developer and the team.

To minimize scenarios like the one described it's important to truly understand the root of the problem and its causes. Once the problem is understood it's possible to brainstorm on and what needs to be done to solve the problem. Starting this at a high level of the planning phase, involving all stakeholder members. By having a detailed planning session with all the affecting parties it's possible gain insights from all aspects and understanding what we're trying to solve. The goal of this phase is to clearly define the task with validated business and technical requirements.

During my time in the project I've come to understand the emphasis of understanding the problem before starting to develop a solution. I've become active in the planning sessions and increased my involvement in the discussions with the stakeholders. Using this opportunity of having everyone there to clarify and rectify the requirements. Taking notes of the planned requirements and validating them with experienced team members have increased my productivity drastically. This also gives stakeholders a clear overview of what can and can't be technically done based on their business requirements.

By utilizing the learned approach, it has saved time and resources from both the business and engineering stakeholders. It has given me a clear scope of the final goal, making it evident what I need to do to get there and what I will be using along the way.

I have found that communication plays a key role in being successful in my workplace. Daily interaction between colleagues and other teams in the organization have made me realize what I needed to improve in order to communicate efficiently. I've come to learn that by asking open ended questions in a friendly approach it's possible to have an easy-going discussion with all involved parties no matter what their background is. This has resulted in richer discussion sessions and better answers to my initial questions. When I needed to explain a complex problem to someone with no background information on the subject.

I've learned to put myself in their shoes and break down the complexity of the subject. Utilizing easy to understand vocabulary and real-life scenarios as a reference I've been able to get my point across to the other party. Practicing these skills has improved how I express ideas and thoughts in a clear efficient way. Both verbally and electronically.

4.3 Conclusion

I've come to realize that in the beginning of the diary I used to spend a significant amount of time trying to figure everything out by myself. Although I did manage to figure out and learn most things independently, there were multiple topics which were specific to this project. No matter how much time I spent trying to understand them it was just not feasible. Throughout the middle of the diary I came to overcome these issues by learning to cooperate with my team members. By ask them questions and advice in what I didn't know. I've learned that it's okay to ask as many questions needed and get help if I'm stuck.

From now on I will keep on being pro-active and ask questions when needed. By doing this I've been able to save time, increasing my productivity and contributions in the project. This not only helped me in my daily tasks, it has also created a great bond of trust and communication between the team and myself. Now that I'm more familiar with the project and my peers I can also help them in subjects that I have insight on. By the end of the observation period I was able to get familiar with the project, all the new technologies used and ways of working by learning by doing.

Even though it took time to get up and running, I'm now confident enough to develop larger tasks and take more responsibility in the project. My soft and technical skills have improved drastically, and I consider myself a skillful performer. I now understand the importance of being part of the tasks defining process. Express my thoughts and ideas more clearly and when clarification is needed, I know how to properly proceed. Most importantly I now see that technology is a tool which can be combined with a business to serve a greater purpose.

4.4 Future plans

I plan on continuing my professional career with Company X. I will remain working in Project Y as a full-time member of the growth hacking team. Taking in account technical, business and soft skills as base. I plan to continue to learn, taking these findings and further expand them as I grow as a professional in the field.

References

Basques, K. (2019). *Inspect Network Activity In Chrome DevTools.* Retrieved from https://developers.google.com/: https://developers.google.com/web/tools/chrome-devtools/network

Basques, K. (2019). *Local Storage With Chrome DevTools.* Retrieved from https://developers.google.com/: https://developers.google.com/web/tools/chrome-devtools/storage/localstorage

Boota, M. W., Ahmad, N., & Masoom, A. H. (2014). *Requirement Engineering Issues and Their Solutions.* ERPublication. Retrieved from https://www.academia.edu/9777149/Requirement_Engineering_Issues_and_Their_Solutions

Crozier, J. (2013). *CSS element border include margin-top.* Retrieved from https://stackoverflow.com/: https://stackoverflow.com/questions/19694784/css-element-border-include-margin-top

Di Marco, R. (2009). *Usage of git rebase and git merge.* Retrieved from stackoverflow: https://stackoverflow.com/questions/804115/when-do-you-use-git-rebase-instead-of-git-merge

Diomidis, S. (2006). *The Bad Code Spotter's Guide.* Pearson Education.

Dixon, E., Enos, E., & Brodmerkle, S. (2011). *A/B Testing.* Retrieved from Patents Google Directory: https://patents.google.com/patent/US20060162071

Dzmitry, A. (2019). *Locating sessionStorage.* Retrieved from StackOverflow: https://stackoverflow.com/questions/30320693/where-is-the-session-stored/30320916

Ellis, S. (2017). *Hacking Growth: How Today's Fastest-Growing Companies Drive Breakout Success.* Virgin Digital.

Elrod, J. (2012). *JavaScript Injections.* Retrieved from StackOverflow: https://stackoverflow.com/questions/9987248/what-is-javascript-injection-and-how-it-could-be-use-in-software-testing

FCC. (2019). *Callback Functions.* Retrieved from freecodecamp.org: https://guide.freecodecamp.org/javascript/callback-functions/

GlobalEventHandlers. (2019m). Retrieved from https://developer.mozilla.org/: https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers

GraphQL Foundation. (2019). *GraphQL Foundation.* Retrieved from GraphQL Foundation: https://foundation.graphql.org/

Hope Computer. (2018). *Computer Hope.* Retrieved from Event Listener: https://www.computerhope.com/jargon/e/event-listener.htm

Jäälinoja, J. (2004). *Requirements implementation in embedded software development.* Espoo: VTT Publications 526. Retrieved from https://www.vtt.fi/: https://www.vtt.fi/inf/pdf/publications/2004/P526.pdf

Jeffrey, K. (2000). *Source Code Documentation: An Engineering Deliverable.* IEEE Computer Society Washington, DC, USA ©2000.

Koen, P. (2013). *Vanilla JavaScript.* Retrieved from StackOverflow: https://stackoverflow.com/questions/20435653/what-is-vanillajs

Kraft, R. (2016). *Robert's Rules of Coders: #11 Separate User Interface Logic From Business Logic.* Retrieved from https://www.codeproject.com/: https://www.codeproject.com/Articles/1111498/Robert-s-Rules-of-Coders-sharp-Separate-User-Inter

Lelonek, K. (2016). *The art of asking questions for developers.* Retrieved from Medium: https://blog.lelonek.me/the-art-of-asking-questions-for-developers-cd88351b9e87

Lodash Official Documentation. (2019). *Lodash Official Documentation.* Retrieved from Lodash: https://lodash.com/

Martin, R. C. (2006). *Agile Software Development, Principles, Patterns, and Practices.* Prentice Hall PTR.

MDN Web Docs. (2019a). *CSS: Cascading Style Sheets.* Retrieved from Mozilla Developr Network - Web Docs: https://developer.mozilla.org/en-US/docs/Web/CSS

MDN Web Docs. (2019b). *Promise.all( ).* Retrieved from Mozilla Developr Network - Web Docs: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/all

MDN Web Docs. (2019c). *Session Storage.* Retrieved from Mozilla Developr Network - Web Docs: https://developer.mozilla.org/en-US/docs/Web/API/Window/sessionStorage

MDN Web Docs. (2019d). *Specificity.* Retrieved from Mozilla Developr Network - Web Docs: https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity

MDN Web Docs. (2019e). *Using Promises.* Retrieved from Mozilla Developr Network - Web Docs: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises

MDN Web Docs. (2019f). *DOM Method querySelectorAll().* Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelectorAll

MDN Web Docs. (2019g). *DOM Method Document.querySelector().* Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector

MDN Web Docs. (2019h). *Promise.All( ) Method.* Retrieved from Mozilla Developr Network - Web Docs: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/all

MDN Web Docs. (2019i). *MDN Web Docs.* Retrieved from Array.prototype.forEach(): https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach

MDN Web Docs. (2019j). *HTTP GET Method.* Retrieved from MDN Web Docs: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET

MDN Web Docs. (2019k). *HTMLElement.click().* Retrieved from https://developer.mozilla.org: https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/click

MDN Web Docs. (2019L). *Document.querySelector().* Retrieved from https://developer.mozilla.org: https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector

Mohan, J. (2019). *Naming Conventions.* Retrieved from Devopedia: https://devopedia.org/naming-conventions

Moore, K., Njeru, E., & Pocevicius, M. (2019). *Object Oriented Programming Classes.* Retrieved from Brilliant Org.: https://brilliant.org/wiki/classes-oop/

Peter Andreae, R. B. (1993). Understanding Code Reusability: Experience with C and C++. Wellington, New Zeland. Retrieved from http://www.mcs.vuw.ac.nz/comp/Publications/archive/CS-TR-93/CS-TR-93-12.pdf

Ray, Z. (2016). *COMMUNICATION STRATEGIES.* Retrieved from Entrepreneur Europe: https://www.entrepreneur.com/article/254264

React JS Documentation. (2019a). *React Hooks.* Retrieved from React JS: https://reactjs.org/docs/hooks-intro.html

React JS Documentation. (2019b). *Refs and the DOM.* Retrieved from React JS: https://reactjs.org/docs/refs-and-the-dom.html

React JS Documentation. (2019c). *Rendering Elements.* Retrieved from React JS: https://reactjs.org/docs/rendering-elements.html

React JS Documentation. (2019d). *State and Lifecycle.* Retrieved from React JS: https://reactjs.org/docs/state-and-lifecycle.html

React JS Documentation. (2019e). *Conditional Rendering.* Retrieved from https://reactjs.org/: https://reactjs.org/docs/conditional-rendering.html

React JS Documentation. (2019f). *React JS Documentation.* Retrieved from https://reactjs.org/docs/: https://reactjs.org/docs/react-dom.html#unmountcomponentatnode

Starr Roxanne, H., Kenneth, J., & Murray, T. (1986). *Experiments in Group Decision Making: Communication Process and Outcome in Face-to-Face versus Computerized Conferences, Human Communication Research, Volume 13, Issue 2, December 1986, Pages 225–252,.* https://doi.org/10.1111/j.1468-2958.1986.tb00104.x.

Umar, H. (2019). *Event Listeners.* Retrieved from Google Developers: https://developers.google.com/web/updates/2015/05/easily-jump-to-event-listeners

W3 Schools. (2019a). *CSS Margins.* Retrieved from W3 Schools: https://www.w3schools.com/css/css_margin.asp

W3 Schools. (2019b). *CSS Padding.* Retrieved from W3 Schools: https://www.w3schools.com/css/css_padding.asp

W3 Schools. (2019c). *HTML5 Web Storage.* Retrieved from W3 Schools: https://www.w3schools.com/html/html5_webstorage.asp

W3 Schools. (2019d). *HTTP Request Methods.* Retrieved from W3 Schools: https://www.w3schools.com/tags/ref_httpmethods.asp

W3 Schools. (2019e). *JavaScript Array concat() Method.* Retrieved from W3 Schools: https://www.w3schools.com/jsref/jsref_concat_array.asp

W3 Schools. (2019f). *W3 Schools.* Retrieved from onresize Event: https://developer.mozilla.org/en-US/docs/Web/API/Window/resize_event

w3 Schools. (2019g). *HTML DOM childNodes Property.* Retrieved from w3 Schools: https://www.w3schools.com/jsref/prop_node_childnodes.asp

W3 Schools. (2019h). *CSS display Property.* Retrieved from https://www.w3schools.com/: https://www.w3schools.com/cssref/pr_class_display.asp

W3C. (2016). *Web Storage - Second Edition.* Retrieved from https://www.w3.org/: https://www.w3.org/TR/webstorage

w3resource. (2019). *w3resource.* Retrieved from JavaScript: typeof operator: https://www.w3resource.com/javascript/operators/typeof.php