NHL STENDEN
university of
applied sciences

Internship Report

Robert Răchită

host company
krumedia GmbH
Karlsruhe, Germany

29th November 2023

General Info

NHL Stenden University of Applied Sciences

Course	Information Technology
Place	Emmen, Netherlands
Student Number	4859367
Host company	krumedia GmbH
Place	Karlsruhe, Germany
Duration	September 2023 - January 2024

Contact Info

Intern	krumedia Supervisor	NHL Stenden Supervisor
Robert Rachita	Michael Reetz	Elise Siersema
robert.rachita	michael.reetz	elise.siersema
@student.nhlstenden.com	@krumedia.com	@nhlstenden.com

Version Management

Version	Date	Description
0.1	22/09/2023	Layout and initial drafting
0.2	27/09/2023	Company Research
0.3	13/10/2023	Technical Research
0.4	3/11/2023	Activity overview draft
0.5	29/11/2023	Final draft (including Ch. 4 & 5)
1.0		Peer Review and feedback
1.1		

This work placement report is written by me, the student/intern. The report is created as part of the assignment of the course programme, with the purpose of explaining and detailing the activities of said work placement, that has been conducted between September 2023 till January 2024.

Special thanks go to krumedia and Dr. Michael Krutwig for offering me the chance to work with them, as well as all to my supervisor and the staff, that have continuously showed warmth and professionalism and have helped me in my daily tasks whenever needed as well as to see the internship through.

I'd like to also express my gratitude to my NHL Stenden supervisor, Ms. Elise Siersema, as well as Mr. Bert Meijerink, for their guidance and support.

Contents

Summary

Reason

The four-year Higher Technical Education study programme usually includes work placements. To this extend, the student is required to complete a work placement of 90 to 99 working days. The work placement is a part of the study programme and is supervised by the work placement supervisor as well as the educational institution, and is concluded with a work placement report. This report is a document that describes the work placement activities and the results achieved. To accomplish the goals of the work placement, support from the business community is required in the form of placement opportunities and transparent collaboration. Should it be a successful endeavour, the student earns 30 European Credits.

Objective

The objective of this work placement report is to provide an overview of the activities carried out during the work placement, as well as the results achieved. It also aims to provide a reflection on the work placement to the supervisor, as well as to any interested external parties, the lessons learned, and the skills acquired during this time. This report is written in accordance with the guidelines provided by NHL Stenden University of Applied Sciences in the course Handbook, and is a requirement for the completion of the work placement.

Research

This document consists of a research part into the host company, which is composed of two components, one focused on the business model and daily operation, and another one, diving into technical aspects and technologies used. The first part is based on interviews, observations and document analysis, while the second part is based on code review, framework documentation and development team collaboration.

Realisation

The realisation part of this document is a detailed overview of the activities carried out during the work placement. It discusses points such as the project, the tasks, the technologies used, the challenges faced, and the results achieved, by looking in-depth at the daily/weekly activities and tasks, as well at the projects that were carried out during the work placement itself. These projects include the development of new features for the company's main product, as well as various bug fixes and improvements.

Retrospective

The retrospective part of this document is a reflection on the work placement, and the lessons learned during this time, as well as this document itself. It discusses the challenges faced, the lessons learned, the skills acquired, and the personal growth that has been achieved during this time. It also discusses the future plans and the impact this work placement has had on the author's future career.

1 Introduction

In an era marked by rapid technological advancements and an increasing emphasis on sustainability, the role of creating and managing software has become paramount. To achieve this and to create the professionals of tomorrow, NHL Stenden, not unlike other places of Higher Technical Education, integrates a work placement as an essential component of the four-year study programme. This internship report aims to provide an insight into my learning experience at krumedia GmbH, a leading tech company in the dynamic and ever-changing landscape of the industry focused software development.

Located in Karlsruhe, Germany, krumedia pride themselves on offering premium energy management software for a wide array of companies, large or smaller, public services and institutions and more, with a focus on web application development that work "out of the box". Over the course of my internship, I had the opportunity of delving deep into the heart of this extensive software, immersing myself in its culture of excellence, and actively contributing to the development of software solutions that have a tangible impact on the energy landscape. This report serves as a documentation of the past five months, highlighting the valuable experiences, lessons learned, and the significant contributions I made during my tenure.

In the following sections, I will provide an overview of krumedia's mission and vision, a description of the energy management application that forms the core of their operations, an exploration of the industry standards and devices they work with, and a reflection on my role as an intern. Through this report, I aim to shed light on the intricate processes, collaborative endeavours, and technological advancements that make krumedia a "hidden champion" in the tech industry, as well as the valuable lessons I learned during my time there.

1.1 Reason

The decision to embark on an internship at krumedia was grounded in a convergence of personal and professional motivations that shaped my desire to contribute to the company's mission and leverage this experience for my own growth and development. I grew up in a fast changing world, where the drivers behind this change were computers and technology. Since my early exposure to this world, I have developed a passion for innovation and its potential to drive positive change.

As such, when time came to look for a work placement, once the opportunity to work for this company was presented to me, I decided to take it, as it presented a good and sustainable means of enhancing my professional skill set. Beyond the technical aspects, I was excited about the prospect of personal growth and skill enhancement. I believed that the challenges, responsibilities, and collaborative environment at krumedia, as well as

making the entire internship a cross-border experience, would push me out of my comfort zone, fostering adaptability, problem-solving skills, and teamwork that are vital in any professional journey. Working in a foreign country like Germany has allowed me to observe and embrace different work methodologies and cultural practices. It has deepened my appreciation for the importance of adaptability and cultural sensitivity in a globalized world.

1.2 Organisation

Krumedia is a leading tech company at the forefront of revolutionizing energy management through innovative software solutions. They have assembled a dedicated team of computer scientists, software developers, energy experts, and engineers to continually enhance and develop their flagship product, enerchart.

At the core of their portfolio, *enerchart*, is a premium energy management system (EnMS) that stands out for its unwavering commitment to excellence. Unlike other solutions in the market, enerchart is designed to meet the complex needs of end consumers in this market. It is a highly customisable, scalable, and future-proof solution that can be tailored to the unique needs of each customer.

Key Features of enerchart involve:

- Hardware-Agnostic: Enerchart doesn't rely on specific hardware manufacturers, giving it the flexibility to work seamlessly with various energy meters and data loggers.
- Scalability: It offers enterprise-class energy management, ensuring scalability in all directions, making it ideal for organisations with diverse energy needs and structures.
- Integration: Enerchart excels in integrating a wide range of energy forms and meters, even in complex, multi-location setups.
- Future-Proof: As an organisation committed to "energy management of tomorrow", krumedia ensures that enerchart is future-proof in terms of integration, technology, scalability, security, and cost-effectiveness. It supports ISO 50001 compliance and digitalization efforts. Customized Software Solutions

(krumedia GmbH, 2023a)

In addition to enerchart, krumedia offers customized software solutions tailored to the unique needs of their customers. Leveraging their expertise in energy efficiency and their immersion in the world of Industry 4.0 and the Internet of Things (IoT), krumedia provides agile project execution, allowing customers to actively influence the development of their solutions. With a dynamic and responsive team, krumedia can

often kick-start projects promptly, delivering high-quality results efficiently. They also capitalize on existing software components, such as those from enerchart, to enhance the quality and speed of their custom solutions.(krumedia GmbH, 2023b)

Furthermore, krumedia recognizes that energy management goes beyond software; it involves the digital transformation of organisations. They emphasize the importance of interconnecting companies with meters, sensors, and other components for measuring, controlling, and monitoring energy flows. In this context, they see the Internet of Things (IoT) as a key enabler for efficient digitalization efforts. Enerchart, purpose-built for this digital age, seamlessly integrates with IoT technologies, contributing to the digital transformation of businesses.

1.3 Reading Guide

This paper assumes the reader has a certain familiarity with current technological terminology. This chapter will briefly describe the format and layout of the thesis at hand.

1.4 Terminology and abbreviations

Agile development: An approach to software development that emphasizes iterative, incremental progress and adaptability to change.

API (Application Programming Interface): A set of rules and specifications that define how two pieces of software can communicate with each other.

Building Automation Systems (BAS): Computerized systems that control and monitor various aspects of a building's operation, such as heating, ventilation, and air conditioning (HVAC).

Bug: An error in software that causes it to produce unexpected results or behave in an unintended way.

CI/CD (Continuous Integration and Continuous Delivery): A set of practices that automate the software development process, enabling frequent and reliable releases of software.

Compiler: A computer program that translates code written in one programming language into another programming language or machine code.

Debugging: The process of identifying and fixing bugs in software.

DevOps: A set of practices that combines software development (Dev) and IT operations (Ops) to shorten the software development lifecycle and provide continuous delivery with high quality.

EMS (Energy Management System): A system that monitors, analyses, and optimizes energy consumption.

HVAC (Heating, Ventilation, and Air Conditioning): A system that provides heating, cooling, and ventilation for buildings.

OOP (Object-Oriented Programming): A programming paradigm based on the concept of objects, which encapsulate data and behaviour.

Industry 4.0: The fourth industrial revolution, characterized by the increasing use of digital technologies and automation in manufacturing and other industries.

International Organization for Standardization (ISO): An international standards organisation that develops and publishes international standards.

IoT (Internet of Things): The network of physical objects that are embedded with sensors and software that enable them to connect and exchange data with the internet.

JavaScript (JS): A scripting language commonly used for web development.

Kanban: An agile development method that visualizes the work process and limits the amount of work in progress.

Key performance indicators (KPIs): Measurable values that quantify the effectiveness of a particular activity or process.

LaminasPHP (formerly Zend): A full-stack PHP framework that provides a variety of tools and libraries for web development.

LoraWAN: A low-power wide-area network (LPWAN) protocol that is designed to wirelessly connect battery-operated devices to the internet.

Open source: Software that is freely available for anyone to use, modify, and distribute.

PDP (Product Development Plan): A document that outlines the goals, scope, and timeline for a product or personal development project.

RFC (Request for Comments): A document that describes a proposed internet standard or protocol.

SCRUM: An agile development framework that is based on iterative and incremental development.

Scalability: The ability of a system to handle an increasing amount of work without compromising performance.

Software framework: A collection of software libraries that provide a foundation for developing applications.

Smart meter: An electronic meter that measures and records electricity consumption.

Superset: A set that contains all the elements of another set and at least one additional element.

TypeScript (TS): A superset of JavaScript that adds static typing and other features.

2 Research

This chapter introduces the reader to the research done by the author, in order to gain more insight and knowledge about the tasks, approaches and methodologies used during the internship. While primarily an activity report, this document incorporates a research part. This research was carried out to investigate the way of working in krumedia, as well as a more technical aspect, involving the custom frameworks the company has built in-house, to ease the process of development. As such, there are two research questions that will be explored and answered in this document. They will be listed in the following subchapter.

2.1 Research Questions

1. How does Krumedia Operate and Approach Development?
2. What are the Tools used within the Company?

The first research question focuses on gaining a deep understanding of Krumedia's operational methodologies and approaches to software development. It aims to explore the company's work culture, project management practices, collaboration methods, and any other unique aspects that may distinguish Krumedia's approach from conventional software development practices. This question is researchable through interviews, an analysis of Krumedia's internal documentation and processes, and being a part of them for an extended period of time (direct observation).

The second research question delves into the technical aspects of Krumedia's custom frameworks, such as KrumediaData, which extends the Zend Framework. It seeks to identify the core features, functionalities, and advantages of these modules in simplifying the development process. Additionally, it aims to explore how these modules enhance efficiency, scalability, and code quality in the context of web application development. This question is researchable through a technical analysis of the framework, code reviews, and discussions with Krumedia's development team.

2.2 Research methods

To answer the research questions effectively, a combination of qualitative and technical research methods will be employed. These methods have been chosen to provide a comprehensive understanding of both the operational and technical aspects of Krumedia and its in-house frameworks. This subchapter will follow up on the previously mentioned methods, by explaining how they will be used.

1. Qualitative Research:
    a) *Interviews*: Structured interviews with key personnel at Krumedia, including project managers, software developers, and system architects, will be

conducted. These interviews will provide insights into the company's approach to development, work culture, and the role of the custom modules in their projects.

b) *Observations*: Observational studies within Krumedia's work environment will be conducted to gain first-hand experience of their day-to-day operations, collaboration practices, and project management methods.

c) *Document Analysis*: Internal documents, such as project documentation, process manuals, and company policies, if available, will be reviewed to extract information regarding Krumedia's work methodologies.

2. Technical Research:

a) *Code Review*: In-depth code reviews and understanding of internal projects will be performed, including the framework extensions. This analysis will focus on understanding the technical aspects of the framework, its integration with the Laminas/Zend Framework, and how it simplifies development tasks.

b) *Framework Documentation*: Code documentation and any relevant technical documents will be studied to gather detailed information about the framework's features, functionalities, and best practices for implementation.

c) *Development Team Collaboration*: Collaborative discussions and knowledge-sharing sessions with Krumedia's development team will be conducted to explore technical nuances, challenges, and benefits.

2.3 Research quality

Throughout the course of this research phase, the author has meticulously adhered to a set of guidelines to guarantee the appropriate and reliable use of research sources. This chapter delineates the evaluation criteria employed for assessing the research sources and demonstrates the compliance with the established guidelines for the proper use of research sources.

To appraise the quality and pertinence of the research sources, the following criteria were utilized:

- **Relevance**: The content from each source corresponded with the research objectives. Journal articles were examined by perusing abstracts and introductory paragraphs, and the publication date was verified to confirm its applicability to the research needs.

- **Timeliness**: The publication date of each source was checked, to ensure the information is up-to-date. Depending on the research requirements, both contemporaneous materials and current accounts of historical events were taken into consideration.

- **Reliability**: To establish the reliability of each source, the facts and data were cross-referenced with other documents addressing the same subject. This procedure facilitated the corroboration of the accuracy and dependability of the information presented. (Middleton, 2019) However, this was not always possible due to the existence of singular sources (for instance framework documentation).
- **Validity**: The author assessed the validity of each source by examining the author's credentials, the publication's reputation, and the presence of supporting evidence. The author also considered the objectivity of each source to ensure that the information was presented in a neutral manner.
- **Credibility**: The credentials of each source's author were evaluated to ensure their subject-matter expertise. Biographical reference sources and Google searches were employed to obtain information on the author's background and qualifications.
- **Purpose & Commercialism**: The purpose of each source was scrutinized to determine its congruence with the research objectives. The author analysed whether the source aimed to persuade, inform, provide an overview, or incite controversy. Similarly, it was assessed whether the source contained advertisements or other forms of commercialism that might influence the information provided. The impact of commercial pressures on the source's content was carefully evaluated.
- **Intended Audience**: The author appraised the target audience of each source to ensure its suitability for the research objectives. The source author's style and the intended audience of the source were considered.
- **Type of Source**: Each source was classified as popular, trade, scholarly or internal. Other types or sources were often not taken into account, due to their irrelevancy.
- **Source Selection**: The author ensured that the sources were selected from a variety of sources, including books, journal articles, and websites.
- **Conclusion**: The author ensured that the sources were used appropriately and that the information was cited correctly, in accordance with the APA Style guidelines.
- **Relevance**: The content from each source corresponded with the research objectives. Journal articles were examined by perusing abstracts and introductory paragraphs, and the publication date was verified to confirm its applicability to the research needs.

In the case of internal sources, information gathered during methods such as interviews, debriefings or collaborative development was deemed to be trustworthy due to the direct involvement of Krumedia's development team and employees, and their in-depth knowledge of the framework's design, implementation, processes and practical applications. Their insights and expertise provide a high level of credibility to the

technical research, ensuring that the information obtained accurately represents the intricacies and nuances of KrumediaData.

Research validity, reliability, and generalizability

In general terms, validity refers to the extent to which a study measures what it intends to measure. (Nikolopoulou, 2022) In this research validity is ensured by selecting the appropriate research methods for each research question and using the correct and right amount of sample. The research methods used in this research are interviews, observations, document analysis, code review, framework documentation and development team collaboration. The sample size is limited to the author and the development team at krumedia.

Limitations

While this research has been conducted to the best of the author's ability, there are certain limitations that need to be acknowledged. Firstly, the research was conducted in the context of a work placement, where the primary focus is to learn and contribute to the company's projects. As such, the research was conducted in parallel with the author's daily tasks, which limited the amount of time and resources available for the research. Furthermore, the author's technical expertise and knowledge of software development were limited, which restricted the extent to which the technical research could be conducted. Finally, the research was limited to the author's personal experience and the information provided by the development team at krumedia. As such, the research may not be generalizable to other companies or software development projects.

2.4 Research Findings & Results

As mentioned earlier, research has been carried out throughout the duration of the internship. The results of this research are written down and described in this chapter. The main focus of the research was to understand how work is being handled and successfully realised within krumedia. In order to get a better look at this aspect, the research was split into two parts, operational and technical. These topics have been answered in the upcoming pages, in the order mentioned here.

2.4.1 Company Research

How Does Krumedia Operate and Approach Development?

In order to understand how krumedia tackles daily operations, one has to first have a basic understanding of the company, its history, as well as current place in today's markets. Founded in 1999, the company has since grew to be a software house with a special focus on solutions in the field of energy efficiency.

Krumedia is primarily a software company. Due to the nature of the product offered, some hardware tinkering is needed from time to time (for instance, the usage of smart meters, or the installation of LoraWAN), however hardware solutions are not included in the company's portfolio. Inside the company there are multiple divisions, such as administrative, system administration, developing etc, however these departments are loosely defined and well integrated into one another, and it is not uncommon for a developer to also be involved in other tasks.

Their main offering is "enerchart" - a versatile and feature-rich energy management software. EMS is a general term and category referring to a variety of energy-related software applications which may provide utility bill tracking, real-time metering, building HVAC and lighting control systems, building simulation and modelling, carbon and sustainability reporting, IT equipment management, demand response, and/or energy audits.(Mahmud & Town, 2016) Managing energy can require a system of systems approach, and this is something that enerchart tries to cater to.

Energy management software gathers historical and/or real-time interval data at intervals ranging from hourly (or less) smart metre readings to quarterly billing statements. An EMS also gathers information on factors that affect energy consumption, such as the number of occupants in the building, the outside temperature, the quantity of produced units, and more. Interval metres, Building Automation Systems (BAS), utilities directly, sensors on electrical circuits directly, or other sources are used to collect the data. An analysis of previous bills can be used to compare energy consumption before and after the EMS. Throughout this analysis, as well as various other analytics, EMS tries to assist the users with analysing, tracking and forecasting energy consumption and

energy conservation measures. This helps in quantifying the success of the measures implements, as well as aids creating key performance indicators, calculate carbon footprint, greenhouse gas, and other metrics relevant in today's fast-paced business market.

According to the International Organisation for Standardization, in ISO 50001:2008, an EMS should provide at the very least the following features:

- Data Collection
- Data Analytics
- Reporting
- Monitoring and control
- Engagement

(for Standardization, 2018)

Enerchart is a prime example of such software. In addition to the list above, enerchart also offers a presentation feature, in the form of highly customisable charts, dashboards and web slideshows, a mobile app for meter reading, a wide array of connectivity and support for virtually all industry standards and protocols, data backup and export, and more.

Figure 1: Enerchart Dashboard

When it comes to the development process, krumedia chooses to implement the Kanban Agile way of working. Agile is an approach to project management and software development that emphasizes flexibility, collaboration, customer-centricity, and continuous improvement. It's a set of principles and practices that have been widely adopted in various industries, with software development being one of the primary areas where Agile methodologies are used. Agile methodologies were developed as a response to the limitations of traditional, rigid project management approaches.(Atlassian, 2022) Agile approaches, like Scrum, Kanban, or Lean, provide a number of benefits that make them excellent for putting knowledge management practices into practice:

- *Iterative and incremental approach*: This strategy enables the project team to concentrate on particular knowledge management goals within a set timeframe, encouraging a sense of progress and accomplishment. The team can receive input, make changes, and continually improve the use of knowledge management principles by working iterative.

- *Flexibility and Adaptability*: Agile methodologies promote adaptability and flexibility in response to changing project needs. New insights and requirements may surface as knowledge management programs progress. Agile enables the project team to act quickly in response to these alterations, modify the order of importance, and add new knowledge management procedures or tools as necessary. This adaptability makes sure that the realisation process keeps up with the project's changing goals and increases its efficiency.

- *Collaboration and Cross-Functional Teams*: Agile approaches promote cross-functional team participation and collaboration. Collaboration is essential for efficiently transferring knowledge in the context of knowledge management. Daily stand-up meetings, sprint reviews, and retrospectives are just a few of the platforms and times for collaboration that agile frameworks offer. These procedures promote knowledge exchange, encourage a culture of learning, and increase the general efficacy of knowledge management projects.

- *Continuous Improvement and Feedback*: Through frequent feedback loops, agile approaches encourage a culture of continuous improvement. This is in line with knowledge management's core principles of learning, adapting, and improving. Agile methodologies, such sprint retrospectives, offer chances to think back on the implementation process, pinpoint areas that might use improvement, and make the appropriate improvements. This loop of iterative feedback makes sure that knowledge management techniques develop and improve over time, producing better results. (Davenport & Prusak, 2000)

Kanban is a popular framework used to implement agile and DevOps software development. The word "Kanban" itself means "signboard" or "visual card" in Japanese, and the system revolves around visualizing work and workflow to optimise efficiency and

manage resources effectively. Implemented initially in the 1950s in Toyota's factories as a scheduling system for lean manufacturing, it necessitates complete transparency of work and real-time capacity communication. In software development, team members can always observe the status of every piece of work thanks to the visual representation of work items on a kanban board. It allows team leaders to leverage Just-in-Time principles by matching the amount of work in progress to the team's capacity. A kanban board, a tool used to visualise work and optimise the flow of it among the team, is the centre of activity for all kanban teams. Virtual boards are an essential component of any agile software development tool despite the fact that some teams prefer physical boards due to its traceability, ease of participation, and accessibility from many places. (Rehkopf, 2022)

The key principles of Kanban are:

- *Visualize Work*: One of the fundamental aspects of Kanban is visualizing the entire workflow on the "Kanban board," which is a visual representation of work items and their status. Each work item is represented by a card or sticky note.
- *Limit Work in Progress (WIP):* Kanban encourages setting limits on the number of work items that can be in progress at any given time. This constraint helps prevent overloading the team and ensures that work is completed before new tasks are started.
- *Manage Flow:* The focus in Kanban is on managing the flow of work items through the workflow. Teams aim to make the flow as smooth and efficient as possible, identifying and addressing bottlenecks and delays.
- *Make Process Policies Explicit:* The rules and policies governing how work is done and how items move through the workflow should be explicit and understood by everyone on the team. This transparency fosters consistency.
- *Feedback and Improvement:* Kanban promotes a culture of continuous improvement. Teams regularly review their processes, discuss what's working and what's not, and make adjustments to improve efficiency and quality.

The Kanban board is the central tool in the Kanban methodology, and it provides a visual representation of the workflow (see Figure 2 on page 14). It typically consists of the following components:

- *Columns:* The board is divided into columns representing the various stages or steps in the workflow. These might include "To Do," "In Progress," "Testing," and "Done," among others. Each column represents a different status of work items.
- *Work Items (Cards):* Work items are represented by cards or sticky notes. Each card typically contains information about the task or item, such as a description, priority, assignee, and possibly a deadline.

Figure 2: An abstract Kanban board

- *WIP Limits:* Each column may have a Work In Progress (WIP) limit, indicating the maximum number of cards that can be in that column at any given time. This limit prevents overloading and helps maintain a smooth flow.
- *Visual Signals:* Some Kanban boards use visual signals, such as colour-coding or icons, to represent different types of work or to indicate priority.
- *Progress Indicators:* Teams often use indicators, like stickers or codes, to mark the status of individual cards within a column, showing which items are actively being worked on, blocked, or completed.

(Daly, 2022)

So, the way Kanban works in software development teams is therefore quite simple - start by creating a Kanban board that represents the workflow for your specific project or process.(visualising the workflow). Here, the lead creates columns based on the need and requirements of the project, and the dev group, and also sets WIP limits for each of them, at every stage. Through the pull system, work items are pulled into the next stage as capacity becomes available, respecting the WIP limits. Finally, Kanban is an agile mean of working, meaning it values continuous improvement. The board can be reviewed periodically, in order to identify bottlenecks, delays, or issues, and to make the necessary adjustments to improve efficiency. Kanban provides a clear and flexible way to manage work, reduce waste, and optimise productivity. It is particularly useful for teams that have a steady stream of incoming tasks and need to maintain a high level of responsiveness and adaptability.

Inside krumedia, a typical Kanban board has the following columns:

- *Backlog:* In this column all the potential work items are collected. These items can be new features, bugs, fixes, or any task related to the software project. It is often high-level and may not be fully detailed
- *Concept Phase:* Here, work items undergo initial analysis and discussion. The focus is to define the scope and requirements of each item, and to set the boundaries and objectives before work starts.
- *Ready/Chosen for Development*: This column includes work items that have been sufficiently defined and are ready for development. It's a signal that a developer/team has committed to working on these items, and has enough information to do so. Its purpose is to prepare and schedule in advance of the developing phase.
- *Developing:* Represents work items currently being actively worked on by developers. This is where the actual coding and implementation happen. Work items move into this column when effective development work has started. The developer or team focuses on completing the coding and associated tasks. Each item in this column usually has the following "signals", or progress indicators:
  - *Open:* Items that are assigned but are not currently in work. Any item that was worked on in the past and has interrupted developing at the moment also falls under this category.
  - *In Progress:* Work is currently underway for this item.
  - *Information needed:* More information is needed, either from the customer or from the concept phase. Temporarily blocks the work on this item.
  - *Set for Review:* Waiting for Review, moves the item to the next column.
- *Review:* In the "Review" column, work items are placed when development is complete, and they are ready for assessment and validation. This stage involves code reviews, testing, and quality assurance.
- *(Client) Acceptance:* The "Acceptance" column is where the work items await formal acceptance by stakeholders or clients. It's a checkpoint where the client or product owner confirms that the work meets the requirements and expectations. This stage ensures that the work aligns with the client's needs and the project's goals. Client or stakeholder approval is often a prerequisite for marking the task as "Completed."
- *Completed:* The "Completed" column is the final stage in the Kanban board. It represents work items that have been successfully finished, reviewed, and accepted. Work items in this column are considered done and meet the agreed-upon criteria for completion. They are ready to be deployed or released to end-users.

It should be noted that the exact implementation of a board varies depending on the project and the specific requirements and needs of clients.

2.4.2 Technical Research

What are the Tools used within the Company?

In the previous chapter, we have discussed the software products of krumedia, its use-cases and features. In this chapter, we will look at the technical aspect, the development process, programming languages, techniques and tools used in the creation of krumedia applications. Additionally, a description of the in-house frameworks used will be provided, again, including the technical aspects, as well as features and the benefits of using these frameworks.

Enerchart is a web application of Laminas Framework, using PHP 7.4, OOP style, and TypeScript. In the following paragraphs, we will briefly look at each of these, as well as the toolset used.

PHP

PHP is a general-purpose scripting language, geared towards web development. ('PHP Documentation', 2023) Created in 1993 and released in 1995, it was originally an abbreviation of *Personal Home Page*, but now the name stands for the recursive initialism *PHP: Hypertext Preprocessor*. A fast and flexible programming language, PHP powers 77% of all the website with known server side language, as of September 2023. (see Figure 3 on page 17) ('Usage statistics of PHP for websites', 2023) Of these, 62% are using Version 7.x.x. Some of the most notable websites written in PHP are Facebook, Wikipedia or WordPress, which powers almost all the personal blogs around the web. It is usually processed on a web server by a PHP interpreter, but it can be used for various tasks outside the web context, such as standalone application, or command line execution.

PHP generally follows C syntax, with exceptions and enhancements for its main use in web development, which makes heavy use of string manipulation. Variables must be prefixed by "$", which allows PHP to perform string interpolation in double-quoted strings, where backslash is supported as an escape character. PHP also supports a C-like `sprintf` function. Code can be modularised into functions defined with keyword function, as well as categorised with namespaces. PHP supports an optional object-oriented coding style, with classes denoted by the class keyword. Functions defined inside classes are sometimes called methods. Control structures include: `if`, `while`, `do/while`, `for`, `foreach`, and `switch`. Statements are terminated by a semicolon (;), not line endings. ('PHP Documentation', 2023)

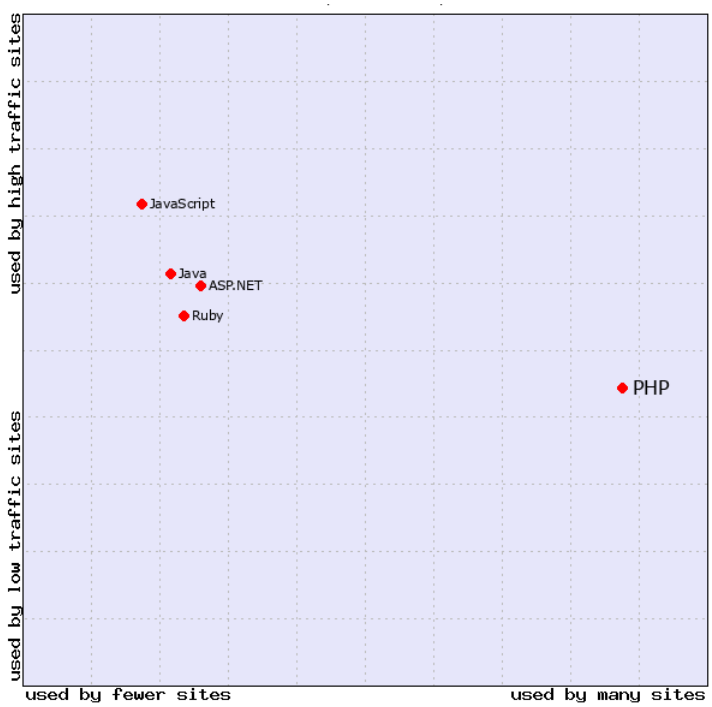

Figure 3: PHP Market Position, 23 October 2023, ('Usage statistics of PHP for websites', 2023)

```php
<?php
/* This is a comment */
$greetings = "Hello World!\n";
while(true){
    $aNumber = 10;
    break;
}
if($aNumber === 10){
    echo $greetings;
}
?>
```

Listing 1: PHP example

Basic object-oriented programming was added in PHP 3, but completely rewritten for PHP 5, expanding the feature set and enhancing performance. In the new approach, objects are referenced by handle, and not by value. PHP 5 introduced private and protected member variables and methods, along with abstract classes and final classes as well as abstract methods and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model, as well as interfaces and

abstraction.

Laminas Project

Laminas, formerly known as Zend Framework, is an open source, object-oriented web application framework implemented in PHP 7. A web application framework is a software framework, designed to help, support and standardise the development process of web application, sch as web services, web resources and web APIs. ('Programmer's Reference Guide — Zend Framework', 2007). Built originally in 2006, it has been rebranded in 2019, following the transition into an open source project hosted by the Linux Foundation. Laminas follows configuration-over-convention approach and does not impose any particular application structure. Laminas does not impose any specific application structure and instead adopts a configuration-over-convention approach. There are skeleton applications for Zend-mvc and Zend-expressive that come with everything needed to run programs and make for excellent starting points.

Applications built with Laminas can run on any PHP stack that satisfies the technical specifications. In order to run Laminas applications, Zend Technologies offers a PHP stack called Zend Server (or Zend Server Community Edition). Offering a modular, decoupled structure, it allows developers to only use the components needed in their projects. ('Laminas Documentation', 2023) As mentioned before, Laminas is governed by an open source community and operates under the Linux Foundation's governance model. This ensures collaboration and innovation, assures quality contribution subjected to rigorous testing and standards, and has security as a key element.

TypeScript

TypeScript is an open-source programming language, a statically-typed superset of JavaScript, developed by Microsoft and released to the public in October 2021. It builds upon JavaScript by adding static typing to the language, making it more predictable and less error-prone. TypeScript code is transcompiled into JavaScript, which can run in any web browser or server environment. The TypeScript compiler is itself written in TS and compiled to JavaScript. It has gained popularity among developers for its ability to enhance code quality, maintainability, and developer productivity. TypeScript may be used to develop JavaScript applications for both client-side and server-side execution. The key features of this language include:

- *Static Typing:* TS allows developers to specify data types for variables, function parameters, and return values. This static typing helps catch common programming errors at compile-time rather than runtime.
- *Interfaces:* TS allows developers to define custom data structures using interfaces. This feature promotes code clarity and facilitates code documentation.
- *Strict type comparisons and null checks:* TS introduces strict null checks to help prevent null and undefined errors, as well as dropping implicit conversion and type coercion from JS.

- *Code Maintainability:* The static nature of TS code makes it easier to understand and maintain as projects grow. It enhances code readability, making it a popular choice for large-scale applications.

(Microsoft, 2023)

In short, the advantages that TypeScript brings to the table are improved code quality, which in turn allows for enhanced developer productivity and reduced debugging time. Below, code examples will be provided to highlight the key differences between the JS and TS, in terms of classes and objects instantiation.

```
// JavaScript class definition (ES6)
class PersonJS {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }
}

// JavaScript object instantiation
const personJS = new PersonJS("John", 25);
console.log("JavaScript Person:", personJS);

// Strong type comparison in JavaScript
const isJohn = personJS.name == "John";  // Returns true (string comparison)
const isAdult = personJS.age == 25;      // Returns true (number comparison)
const isStringEqual = "25" == 25;        // Returns true (JavaScript type
    coercion)
const isStringEqualStrong = "25" === 25  // Retunrs false (types comparison)
```

Listing 2: JavaScript example

```
// TypeScript class definition
class PersonTS {
  name: string;
  age: number;

  constructor(name: string, age: number) {
    this.name = name;
    this.age = age;
  }
}

// TypeScript object instantiation
const personTS: PersonTS = new PersonTS("Jane", 25);
console.log("TypeScript Person:", personTS);

// Strong type comparison in TypeScript
const isJane: boolean = personTS.name === "Jane";
// Returns true (string comparison)
const isYoung: boolean = personTS.age === 25;
```

```
// Returns true (number comparison)
const isStringEqualTS: boolean = "25" == 25;
// Returns false (TypeScript does not perform type coercion)
const isStringEqualStong = "25" === 25;
/* Retunrs false (types comparison) and the following warning:
   This comparison appears to be unintentional because the types 'string' and '
    number' have no overlap.*/
```

Listing 3: TypeScript example

As it can be seen, there are some key changes between the two examples. First, the class definition. In JS, classes are defined using constructor functions. Properties are assigned to the object using the `this` keyword within the constructor, whereas in TS classes are explicitly defined using the class keyword. Properties are declared with their types within the class body, and a constructor method is used to initialize the object. Classes can inherit from interfaces or implement abstract classes now. TypeScript provides static typing by explicitly declaring the types of class properties, which is not present in JavaScript. Furthermore, in TypeScript, objects are also instantiated using the `new` keyword, but the type of the object can be explicitly specified, as shown with `const personTS: PersonTS`. TypeScript uses type annotations to declare the types of properties and variables. In the TypeScript example, it can be seen that the name property is of type string, and the age property is of type number. Lastly, on the topic of type coercion, JavaScript is known for its type coercion, a feature that allows the language to automatically convert one type to another in certain operations, such as comparisons. When it compares values of different types without using the strict operator (`'==='`), JavaScript will attempt to convert one or both of the values to a common type to make the comparison. This behaviour can lead to unexpected results in some cases. TypeScript, on the other hand, is designed to provide static typing and strong type checking, aiming to catch type-related errors at compile-time rather than runtime. TypeScript does not perform implicit type coercion when comparing values of different types. It enforces strict type checking to ensure that values are compared only when their types match.

Vagrant

Vagrant is a powerful open-source tool that simplifies the management of virtualised development environments. (HashiCorp, 2023) Developed by HashiCorp, and released initially in 2010, Vagrant aims to streamline the setup, configuration, and sharing of development environments among software developers, as well as simplifying the software management aspect of virtualisation, making it an essential tool for modern software development practices. (HashiCorp, 2023). Vagrant is compatible with various operating systems, including Windows, macOS, and Linux. It abstracts the underlying VM technologies, allowing developers to create and share consistent development environments across platforms. Vagrant was initially offered just for Oracle's VirtualBox, but version 1.1 added support for other virtualisation software such as VMware and KVM, and for server environments like Amazon EC2, with later updates bringing support for Docker containers, which in some cases can serve as a substitute for a fully

virtualised operating system. It is written in Ruby, but it can be used in projects written in other programming languages such as PHP, Python, Java, C#, and JavaScript.

Vagrant uses a simple, human-readable configuration file known as the Vagrantfile. This file specifies the virtual machine's settings, provisioning scripts, and software requirements. With configuration as code, developers can maintain version-controlled, reproducible environments. A Vagrant "box" is a pre-packaged Virtual Machine image. These can be highly customisable and easy to share between team members, thus making sure that everyone works with an identical development environment and reducing "it works on my machine" issues. It can also facilitate the simulation of clients' hardware, as well as facilitating seamless integration with tools for continuous integration and deployment (CI/CD).

ORM Doctrine, Database, MySQL

Doctrine is a widely-used Object-Relational Mapping (ORM) tool for PHP applications. ORM simplifies the interaction between PHP code and relational databases by mapping PHP objects to database tables. One of the most prominent features of Doctrine is its ability to work seamlessly with various database systems, including MySQL, enabling efficient database operations and enhancing software development. Doctrine allows developers to define the structure of their application's database schema using plain PHP classes. These classes, known as entities, represent database tables and their relationships. Doctrine then maps these entities to the corresponding database tables, easing database interactions and eliminating the need for direct SQL queries. ('Doctrine ORM — DBAL', 2023)

```php
class Order
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer");
     * @ORM\GeneratedValue(strategy="AUTO")
     * @var int
     */
    protected int $id;

    /**
     * @ORM\Column(type="string");
     * @var string
     */
    protected string $orderId;

    //rest of the code, constructor, getters and setters, methods etc.
}
```

Listing 4: Doctrine ORM example

As shown in the example above, Doctrine entities are plain PHP classes with properties and methods. These classes are mapped to database tables using annotations, which are special PHP comments preceded by the @ symbol. Annotations are used to

define entities, properties, and relationships. The `@ORM` annotation defines the entity class and its properties. The `@ORM\Id` annotation indicates that the ID property is the entity's primary key, and the `@ORM\Column` annotation specifies the column type for each property. The `@ORM\GeneratedValue` annotation specifies that the ID property is auto-generated. Doctrine provides a powerful Query Builder API that can be used to query the database without writing any SQL code. The Query Builder API allows developers to build SQL queries using PHP methods and functions. The Doctrine Query Language (DQL) is an Object-Oriented query language that allows developers to write database queries in a syntax similar to SQL. DQL is translated into SQL by Doctrine's query engine.

Other tools

Some of the other tools used within krumedia will be briefly discussed in the following list:

- **PHPStorm**: PhpStorm is a popular, integrated development environment (IDE) specifically designed for PHP development. It's developed by JetBrains and provides a comprehensive set of features for coding, debugging, and testing PHP applications. PhpStorm offers code analysis, version control integration, and a wide range of plugins and extensions to enhance productivity and streamline the PHP development process.

- **Gulp**: Gulp is a JavaScript-based task runner used for automating various development tasks. It's particularly valuable for front-end web development. Gulp allows developers to define and run a series of tasks such as minification, image optimization, and file concatenation, making it easier to streamline workflows and optimise the performance of web applications. It's known for its simplicity and speed. One of the use cases at krumedia is the use of module bundling using `webpack`.

- **Composer**: Composer is a popular command-line dependency management tool for PHP applications. It allows developers to declare the libraries and packages that their projects depend on. Composer then installs these dependencies in the project and loads them automatically. This simplifies the process of managing dependencies and ensures that all the required libraries are present in the project.

- **Git**: Git is a popular open-source version control system (VCS) used for tracking changes in source code during software development. Git is a distributed version control system, which means that each developer has a copy of the entire codebase on their machine.

- **Jira**: Jira is a popular issue and project tracking software that can be used for agile software development. It's developed by Atlassian and provides a wide range of features for managing software projects. Jira allows developers to create and track

issues, plan sprints, and manage backlogs. It also provides a web-based interface for managing projects and teams.

- **Gerrit**: Gerrit is a free, web-based code review tool that integrates with Git. It provides a platform for reviewing and approving source code changes before they are committed to the main codebase. Gerrit facilitates code reviews by allowing developers to submit changes for review, comment on other developers' changes, and approve changes before they are merged into the main codebase.
- **Jenkins**: Jenkins is a popular open-source automation server that can be used to automate various parts of the software development process. It's primarily used for continuous integration (CI) and continuous delivery (CD) of software applications. It can also be used to automate tasks such as building, testing, and deploying software applications. Jenkins provides a web-based interface for managing and monitoring the software development process. It's highly extensible and provides a wide range of plugins for enhancing its functionality.
- **CLI, various developing tools**: The Command Line Interface (CLI) is a text-based interface used to interact with a computer or operating system through typed commands.

Now that we have seen the technologies and tools used within krumedia, let's look at the in-house frameworks used.

Krumedia Frameworks

Krumedia has developed multiple frameworks over the years, each with its own use case and purpose. The most notable ones are:

- **KrumediaData**: A framework used for data handling, such as database connections, data manipulation, form data binding, etc.
- **KrumediaCommon**: A collection of `util` classes and methods, used for various purposes, such as `DateTime` manipulation, string manipulation, file handling, etc.
- **DBUpdater**: A utility used for updating the database schema, as well as the data itself. Works in accordance with Doctrine ORM.

Most of these frameworks are built on top of the Zend Framework, and are used in the development of krumedia products.

2.5 Research Answers

The research questions posed in the beginning of this chapter have been answered and discussed in the previous sections. The answers are summarised in the following subsections.

2.5.1 Analysis

Through this research, a better understanding of the company was gained, about its products, and the way it operates. We have also gained insight into the technical aspect of the company, the tools used, the frameworks developed in house, and the way development is handled. This information will be used in the next chapter, to analyse the deliverables of this internship, and the way they were realised.

2.5.2 Company Research

Krumedia, founded in 1999, has evolved into a software house with a specialized focus on energy efficiency solutions. The company's main offering, "enerchart," is a versatile energy management software designed to cater to the complexities of energy-related applications. Krumedia operates primarily as a software company, with multiple integrated divisions such as administrative, system administration, and development. The fluidity between these departments allows for a collaborative and dynamic work environment.

The central product, enerchart, falls under the category of Energy Management Software (EMS). It provides utility bill tracking, real-time metering, building control systems, customisable graphs and more. Enerchart's capabilities align with ISO 50001 standards, emphasizing data collection, analytics, reporting, monitoring, control, and engagement.

The company works Agile, using Kanban. The choice aligns with the iterative and collaborative nature of knowledge management practices, promoting cross-functional teams and fostering a culture of continuous improvement and feedback.

2.5.3 Technical Research

At the moment of writing, krumedia primarily employs PHP 7.4 with Laminas Framework and TypeScript for its products. The development process for enerchart and other software offerings is facilitated by tools such as Vagrant, Composer, Gulp, and Git. The company has also developed multiple in-house frameworks, such as KrumediaData, KrumediaCommon, and DBUpdater. These frameworks are built on top of the Zend Framework and are used in the development of krumedia products.

PHPStorm serves as the integrated development environment (IDE), while Gulp automates various development tasks. Composer is used for dependency management, and Git is used for version control. Jira is used for issue tracking, Gerrit for code reviews, and Jenkins for continuous integration and delivery. Doctrine is used for object-relational mapping (ORM), and MySQL is used as the database management system (DBMS).

2.5.4 Conclusion

In this chapter, we have looked at the research carried out during the internship. This research was split into two parts, company research and technical research. The company research was focused on understanding the company itself, its history, products, and the way it operates, whereas the technical research was focused on the tools used within the company, and the way development is handled. The results of this research will be used in the next chapter, to analyse the work done during this internship.

3 Overview of internship

This chapter will provide the reader with an overview of the internship, including the activities carried out, the methodologies used, and further information about the delivered products. The chapter is split into three parts, the first one being a general overview of the internship's methodology, the second one being a more detailed account of the activities, and finally a description of the deliverables.

3.1 Methodology

This section emphasizes the manner in which my daily activities were planned and conducted throughout the duration of the internship. As mentioned in the chapter Company Research on page 10, krumedia makes use of Kanban to keep track of their work. (please refer to that chapter for an in-depth explanation of Kanban and its methodology, and how it is applied within krumedia). Here, we shall look only at how this applied to, and impacted, me.

As mentioned before, krumedia uses a board (Figure 2 on page 14) that is divided into multiple columns. Each column represents a stage in the development process, and each card represents a task. The board is used to visualise the workflow and track the progress of tasks as they move through the workflow. Each task is assigned to a developer and has a deadline. The board also indicates the priority of each task. The Kanban board is updated regularly to reflect the current status of tasks. Tasks are moved to the appropriate column as they move through the workflow. Each column has a Work In Progress (WIP) limit that specifies the maximum number of tasks that can be in that column at any given time. This ensures that the team does not take on more work than they can handle, and that tasks are completed before new ones are started. The Kanban board is used to visualise the workflow and track the progress of tasks as they move through the workflow. Each task is assigned to a developer and has an estimated time of completion, in hours. The board also indicates the priority of each task. The Kanban board is updated regularly to reflect the current status of tasks. Tasks are moved to the appropriate column as they move through the workflow.

As an intern, I was initially assigned tasks with a lower priority, which were then moved to the "Selected for development" column. When I started working on a ticket, it

was moved to the ”In Progress” column. I usually only had one ticket active at a time, although later on this changed, as I had to go back and make necessary changes to some submissions. When I finished working on a task, I moved it to the ”Review” column. The task was then reviewed by a senior developer, who either approved the task or sent it back to me with feedback. If the task was approved, it was moved to the ”Acceptance” column, where it was reviewed by the client and either accepted or rejected, and once the client approved the task, it was moved to the ”Completed” column. Should the client have rejected the task, it was moved back to the ”Developing” column, and the ticket description updated with the new requirements. The typical flow of a ticket can be consulted in Figure 4 on page 26

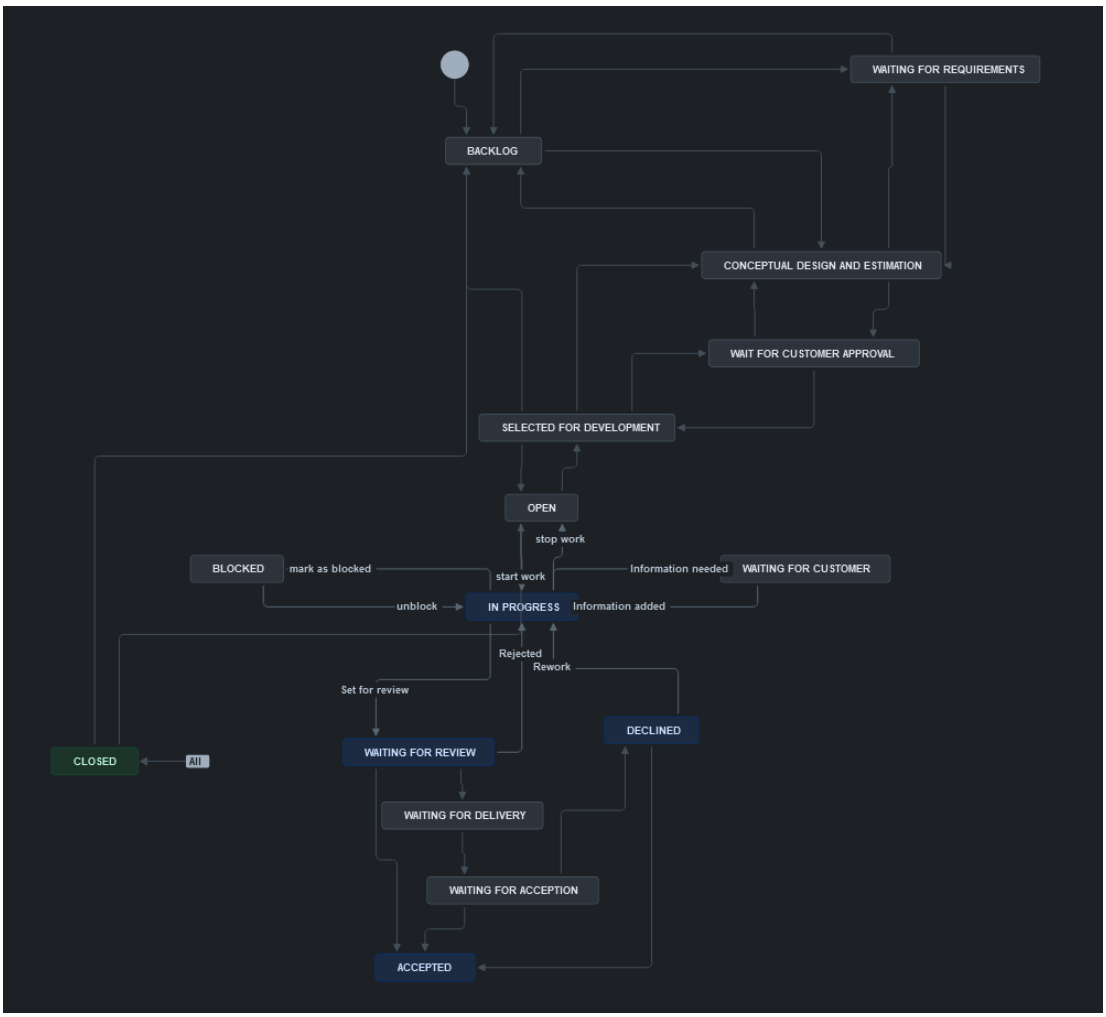

Figure 4: Typical flow of a ticket

This process was followed throughout the duration of the internship. The tasks assigned to me were gradually increased in complexity and priority, as I became more familiar with the company's products and development process.

3.2 Detailed account

This chapter will provide the reader with an in-depth account of the activities and educational process undertook during this work placement. This will mirror to some extent the weekly reports that can be found attached to this document. More information about the reports can be found in chapter 3.3 Deliverables on page 31, or in the Appendix 7.3 Weekly Reports on page 41.

Internship Objectives

The objectives for this work placement encompass a spectrum of general, technical, professionalization, and research-oriented goals. I will make a distinction here, between general internship objectives, as mandated by the course and its handbook, and personal objectives, as defined in the Personal Development Plan, although these may overlap from time to time, in regard to the process of accomplishing them. The more general objectives refer to getting myself acquainted with professional practices, namely to get immersed in the actual professional world and to get a feeling for the future profession. Naturally, in the tech industry, this can be realised by broadening and deepening professional knowledge already acquired, applying the theoretical knowledge laid in the earlier year of studying and thus bringing a gap between the academic theory and real world problem-solving.

This internship has encouraged the honing of observation, reporting and creative thinking skills, through the tasks that were assigned to me. These include managing tickets, features, fixes as well as organisational responsibilities within the professional setting. Furthermore, in order to strengthen my previous knowledge, as well as to prepare me for the upcoming graduation project, a research competence objective was observed. This internship offered a good opportunity to research the ins and outs of a leading tech company, as well as to serve as a basis for substantial technical research. The results of these can be consulted and reviewed in the earlier parts of this document, namely in chapter Research Findings & Results on page 10.

Lastly, the objectives that I have set for myself, at the start of this internship, and that I have worked on for the past five months, are expanding my technical knowledge about databases technologies and new web development frameworks, as well as strengthening cross-functional collaboration skills. These goals were chosen in accordance with the structure of the work that I was (then) about to do.

These objectives, mirrored in the PDP, were as follows:

- **Enhance Technical Proficiency in Database Management**: Strengthen my technical proficiency in database management, by learning about new technologies and techniques, as well as hands-on experience with the existing database.

- **Enhance Technical Proficiency in Database Management**: Enhance my technical proficiency in web development frameworks, by learning about new technologies and techniques, as well as hands-on experience with the existing frameworks, most notably the ones used within krumedia.
- **Strengthen Cross-Functional Collaboration Skills**: Strengthen cross-functional collaboration skills. The effectiveness of my cross-functional collaboration skills are determined with the means of feedback from team members, the successful completion of cross-functional projects, and my ability to handle and resolve conflicts and/or difficulties encountered within the internship period.

These objectives have been completed by the end of the internship, following the planning and guidelines/activities set in the PDP. For more information about these, please consult Appendix 7.2 Personal Development Plan on page 41.

As mentioned previously, the internship took place in a company that specialises in software development, more specifically in the energy management software field. As such, the training and supervision was provided by the company itself, as well as by the assigned mentor, Michael Reetz, but also the developer in charge of mentoring new employees, the team leader that was assigning tickets to me, as well as various other developers that I could turn to anytime I had questions. The training and supervision was done in a gradual manner, as I was introduced to the company, its products and the way it operates in the first couple of days, then dived into the code directly. I was also given access to the company's internal wiki, which contains information about the company, its products, and the way it operates.

Roles and Responsibilities

I have started the internship on a Monday, 4^{th} of September 2023 to be more precise. During this time at krumedia, I was entrusted with a diverse array of tasks, projects and responsibilities that collectively contributed to a holistic and enriching professional experience. These tasks were chosen based on my level of understanding of the products and technical capabilities, and increased slightly in complexity as time went on. Each task was reviewed some time after completion, sometimes requiring additional actions from my end, other times being approved right away.

A primary responsibility and/or task of my internship revolved around software development. As a new member of the team, I was assigned a mentor, who was also my direct superior at the beginning, and who would provide me with the necessary guidance and support, as well as work tasks. The first week was spent familiarising myself with the company, its products, and the way it operates. I was actively involved in coding, debugging, and testing software for various projects. I have started with coding a small, custom implementation of Conway's Game of Life, as a test for my current (at the time) technical skill set. The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970. It is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further

input. One interacts with the Game of Life by creating an initial configuration and observing how it evolves. It is Turing complete and can simulate a universal constructor or any other Turing machine. (Conway et al., 2015).

Later in the first week, I have started working on my first actual task at krumedia. This task was the implementation of a calendar feature in an internal calendar and employee management software, used primarily for tracking sick and holidays inside the company. This software, aptly called "acksense", is an internal-built piece of software written in PHP/Laminas and Typescript. Through this shift, I could gain a foothold into how such projects are created and how they work, what is the code structure and so on.

Such a role fell well within my academic pursuit, as it mirrored a lot of the activities I have been engaging in for the previous three years of study, when it came to the realisation of Projects. It has also allowed me to develop the personal objectives, namely the database and framework points.

Following the passage of time, I have soon after moved onto working on enerchart (for more information about this software, please consult chapter Company Research on page 10). In small increments, I have gradually become more familiar with how the code behind the web app functions, and my productivity increased accordingly. My first task on this product was retrofitting a data entry feature. In short, users can import data from various sources, and process this data while importing. In the past, the user could choose to ignore values from the source, but there was a need for a data correction feature, that would replace certain values (specified by the user) with a static value, also up to the user's needs. This has proven to be an excellent first task on enerchart, due to how simplistic it appeared initially, but in reality extending to various sub-forms and secondary pages, Wizards and components in the app. Notably, this task also required me to work with the database and the ORM Doctrine part.

Due to how it grew in time complexity, one could consider this feature addition task as a mini-project, especially in the educational context. It required me to expand the data importing feature in additional software components, such as the krumediaData framework, which was explained previously. This framework is used to handle data manipulation, database connections, form data binding and so on. This task was a great way to get familiar with the krumediaData framework, as well as to get a better understanding of how the database works, and how it interacts with the rest of the app. It was also a great way to learn about the ORM Doctrine part, and the Gerrit review process, as I had to refactor parts of the code I've submitted, in order to make the value normalizer more abstract, after receiving feedback. The final feature patch-set was over 2000 lines of code added, and it was a great learning experience, as well as a great way to get familiar with the codebase and the way the app works.

After this, I have worked on a series of smaller tasks, such as bug fixes, feature additions, and so on. These tasks were assigned to me by a team leader, and they were chosen based on my level of understanding of the products and technical capabilities. Each task was reviewed some time after completion, sometimes requiring additional

actions from my end, other times being approved right away. Details about these smaller tasks can be found in the weekly reports, in Appendix 7.3 Weekly Reports on page 41.

Throughout the internship, I've encountered a series of challenges and problems, that provided me with invaluable opportunities for problem-solving and personal growth. Some of these challenges were more demanding than others, nonetheless they all served as a catalyst for enhancing my skills and resilience. In the following lines, I will outline some of these and the strategies I've employed in navigating them.

One of the very first challenge was simply the breadth of the software that I was to familiarise myself with, and work with for the following twenty weeks. As a highly sophisticated application, working with enerchart required in-depth technical knowledge, not only about the technologies used, but the app itself. Being comprised of thousand of files, tens of thousand of classes and hundreds of thousands of lines of code, navigating this *maze* of code can prove intimidating at the beginning. To address and overcome this, I've engaged in comprehensive code reviewing and code reading, fuelled also by implementing various features, as described previously. I have also sought guidance from senior colleagues whenever needed, and meticulously tested chunks of code, to see what does what.

Adapting to the new technologies was a secondary challenged that I've encountered. As mentioned earlier in these documents, krumedia employs several programming languages and frameworks, so keeping up with these while continuing to meet project and internship objectives required continuous learning and adaptability. This was done mostly through self-guided learning, exploring the emerging technologies and asking for guidance from my supervisor or other experienced team members. This approach not only allowed me to stay current but also reinforced my ability to adapt to technological changes and to communicate effectively with other team members.

Speaking of communicating effectively, this brings me to Collaboration and Teamwork aspect. These were fundamental aspects of the internship. While my workflow, as well as code documentation and other resources were mostly in English, the primary language of communication within krumedia remained German. This presented some unique challenges and opportunities for me to further develop my language skills, primarily in a business oriented way, but also to enhance my cross-cultural communication skills. I have also had the opportunity to work with a diverse team of developers, each with their own unique skill set and background. This allowed me to gain a better understanding of the importance of teamwork and collaboration in the workplace.

3.3 Deliverables

In this subchapter, the focus is on the description of the delivered products. In order to assess the quality of said deliverables and to also determined if they are in-line with the guidelines and rules set by the course, one has to first know what these deliverables are. To this extent, the following list was put together, in no particular order:

- Internship Thesis
- Weekly reports
- Personal Development Plan
- Work planning
- Contract, agreement and feedback forms

And of course, the software the was produced during the internship proper. As no single, larger project was assigned during this work placement, the software aspect consists of various features and bug fixes implemented in several projects, namely enerchart, acksense, and a few of the internal ones.

The thesis is the main deliverable of this internship, and it is the document that is being read right now. It is a comprehensive, in-depth account of the internship, the activities undertaken, the research carried out, and the results of said research. It also contains a retrospective and a discussion of the results, as well as a conclusion and recommendations.

The weekly reports are a series of documents that were submitted to the course coordinator on a weekly basis, summarising the activities undertaken during that week, as well as any other relevant information. These reports were also shared with the work supervisor, and they were used as a basis for this report as well. The weekly reports were used to track the student's progress and to ensure that the work placement was on track with the objectives and the deliverables.

The Personal Development Plan (PDP) is a document that was created at the beginning of the internship, outlining the objectives and goals for the internship. It was used as a basis for the weekly reports, and it was updated regularly to reflect the progress made. The PDP was also used to track the student's progress and to ensure that the work placement was on track with the objectives and the deliverables.

The work planning document was created at the beginning of the internship, creating a rough overview of the activities that were to be undertaken during the placement itself.

The contract and agreement forms were signed at the beginning of the internship. These constitute the administrative and/or legal aspect of the internship, and they are not relevant to the deliverables of the internship itself. The feedback form was filled out by the work supervisor at the end of the internship, and it was used to evaluate the student's performance and to provide feedback.

The Weekly reports, the PDP, the work planning document as well as the feedback form can be found in the Appendix section of this document, in Appendix Weekly Reports on page 41, Appendix Personal Development Plan on page 41, Appendix Planning on page 38, and Appendix Feedback Form on page 41 respectively. The contract, agreement, and feedback forms are not included in this document, as they are not relevant to the deliverables of the internship itself. Similarly, the software that was produced during the internship is not included, as it is the intellectual property of krumedia. Finally, other documents such as the deliverables of the Internship Review Day or various presentations are not included either.

4 Conclusion & Recommendations

In this chapter, an overall conclusion will be drawn from the research and the realisation phase of the project, alongside a retrospective and a recommendation subchapter to evaluate the possible areas where the product and/or the research can be improved.

4.1 Conclusion

This internship report has provided a detailed account of my experiences and contributions during my internship at krumedia. Throughout my time there, I have gained valuable insights into the software engineering field and have had the opportunity to apply my technical skills to real-world projects.

In this document, we have looked at the research carried out during the internship and discussed the company's history, products, and the way it operates, as well as the technical aspect of the company, the tools used, the frameworks developed in house, and the way development is handled. I have also discussed the research findings carried out during this internship, which were split into two parts, company research and technical research. The results of this research were then used, in the next chapter, to analyse the work done during this internship. In this concluding chapter, I will summarise the key takeaways from my internship experience and reflect on the skills and knowledge I have gained. I will also discuss the potential impact of my work on krumedia and the future directions for the project.

The research carried out during this internship has provided me with a better

understanding of the company, its products, and the way it operates. It has also allowed me to gain insight into the technical aspect of the company, the tools used, the frameworks developed in house, and the way development is handled. This information was then used to analyse the work done during this internship.

During the internship, I was entrusted with a diverse array of tasks, projects and responsibilities that collectively contributed to a holistic and enriching professional experience. These tasks were chosen based on my level of understanding of the products and technical capabilities, and increased slightly in complexity as time went on.

As part of the Internship Review Day, which happened past the halfway point of the work placement, I have presented my work and progress at krumedia to the course coordinator, as well as to my fellow students, through the means of a video and a business presentation. This presentation was a great opportunity to reflect on my progress and to receive feedback from my peers and the course coordinator, as well as to share my experiences with my fellow students, and to compare them with theirs.

To wrap up, I strongly believe that this internship has proven to be a valuable learning experience, allowing me to gain a better understanding of the software engineering field and to apply my technical skills to real-world projects. I can say without a doubt that I am now better prepared for my future career as a professional in the field of software engineering.

4.2 Recommendations

The work done during this internship had a positive impact on krumedia and its products. The features and bug fixes that I have implemented have improved the functionality and usability of the products, as demonstrated by the successful deployment of these additions. The work done during this internship has also contributed to the company's knowledge base, as it has been documented and shared with other team members. This will allow the company to build on the work done during this internship and to continue improving its products. Additionally, I have decided to continue working with krumedia for my graduation project, which will allow me to further build on the work and knowledge gained during this internship.

Looking back at this internship, I can say that I am satisfied with the work that I have done. However, there are a few things that I could have done better. For example, I could have been more proactive in seeking feedback from my supervisor or work colleagues.

5 Evaluation & Reflection

Throughout the internship, I've encountered a series of challenges and problems, that provided me with invaluable opportunities for problem-solving and personal growth. Some of these challenges were more demanding than others, nonetheless they all served as a catalyst for enhancing my skills and resilience. In the following lines, I will outline some of these and the strategies I've employed in navigating them.

The purpose of this internship was to gain practical experience in the field of software development and to apply the theoretical knowledge acquired during the academic years. The internship has offered a well-rounded and synergistic experience that aligned with both my academic goals and career aspirations. The tasks and projects I undertook were carefully crafted to enrich my technical skills, bolster my understanding of project management, and fortify my abilities in data analysis, quality assurance, and collaborative teamwork. These roles have not only contributed to my academic growth but have also laid a strong foundation for my future career in the field of software development and technology.

In summary, my roles and responsibilities during this internship were thoughtfully designed to offer a well-rounded and synergistic experience that aligned with both my academic goals and career aspirations. The tasks and projects I undertook were carefully crafted to enrich my technical skills, bolster my understanding of project management, and fortify my abilities in data analysis, quality assurance, and collaborative teamwork. These roles have not only contributed to my academic growth but have also laid a strong foundation for my future career in the field of software development and technology.

Throughout my internship, I had the privilege of being mentored by experienced professionals in the organisation. I actively sought guidance and feedback to enhance my technical skills and professional competencies. This mentorship process was a critical component of my academic and career development, allowing me to refine my understanding of the industry's best practices and emerging technologies.

My internship at krumedia has provided me with a comprehensive understanding of the software development process and the various tools and technologies used in the industry. I have also learned the importance of teamwork, communication, and problem-solving in the software engineering field.

Key takeaways from my internship experience include:

- The importance of thorough planning, design, and testing in software development.
- The effectiveness of using agile methodologies for managing software projects, such as Kanban.

- The value of continuous integration and continuous delivery (CI/CD) practices.
- The importance of documentation and code maintainability and abstraction.
- The need for effective communication and collaboration among team members and stakeholders.
- The importance of continuous learning and professional development.
- The importance of teamwork and problem-solving in the software engineering field.

I have had the opportunity to develop and enhance my technical skills in areas such as PHP, Laminas, TypeScript, MySQL, and Git, and more. I have also gained valuable experience in areas such as project management, software testing, quality assurance, as well as the importance of documentation, code maintainability and abstraction. These skills will be invaluable in my future career as a software engineer.

6 References

Atlassian. (2022). *What is the agile methodology?* https://www.atlassian.com/agile

Conway, J. H., Izhikevich, E. M., & Seth, A. (2015). Game of Life [revision #150735]. *Scholarpedia*, *10*(6), 1816. https://doi.org/10.4249/scholarpedia.1816

Daly, L. (2022). *Kanplan: Where your backlog meets kanban*. https://www.atlassian.com/agile/kanban/kanplan

Davenport, T. H., & Prusak, L. (2000). *Working knowledge: How organizations manage what they know*. Harvard Business Review Press.

Doctrine orm — dbal. (2023). The Doctrine Project. https://www.doctrine-project.org/

for Standardization, I. O. (2018, August). Iso 50001:2018 - energy management systems — requirements with guidance for use. https://www.iso.org/standard/69426.html

HashiCorp. (2023). *Vagrant documentation*. https://developer.hashicorp.com/vagrant/docs

krumedia GmbH. (2023a). *Enerchart - energy management for the future*. https://www.enerchart.com/en/

krumedia GmbH. (2023b, October). *Our portfolio - krumedia* [Accessed 29-09-2023]. https://www.krumedia.com/en/our-portfolio/#Premium-energy-management-with-enerchart

Laminas documentation [Accessed 23-10-2023]. (2023, October). https://docs.laminas.dev/

Mahmud, K., & Town, G. E. (2016). A review of computer tools for modeling electric vehicle energy requirements and their impact on power distribution networks. *Applied Energy*, *172*, 337–359. https://doi.org/https://doi.org/10.1016/j.apenergy.2016.03.100

Microsoft. (2023). *The typescript handbook*. https://www.typescriptlang.org/docs/handbook/intro.html

Middleton, F. (2019, July). *Reliability vs. validity in research — difference, types and examples*. https://www.scribbr.com/methodology/reliability-vs-validity/

Nikolopoulou, K. (2022, October). *What is generalizability? — definition & examples* [Accessed 29-09-2023]. https://www.scribbr.com/research-bias/generalizability/

Php documentation [Accessed 17-10-2023]. (2023, October). https://www.php.net/docs.php

Programmer's reference guide — zend framework [Accessed 23-10-2023]. (2007, June). https://web.archive.org/web/20090211210131/http://framework.zend.com/manual/en/introduction.html

Rehkopf, M. (2022). *What is a kanban board?* https://www.atlassian.com/agile/kanban/boards

Usage statistics of php for websites [Accessed 17-10-2023]. (2023, September). https://w3techs.com/technologies/details/pl-php

List of Figures

Listings

7 Appendixes

7.1 Planning

2023-2024

Internship - Robert Rachita @ krumedia

September '23

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

19

October '23

M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

22

November '23

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

December '23

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

16

January '24

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

11

February '24

M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29			

March '24

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

0

April '24

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

0

May '24

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June '24

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

0

July '24

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

0

August '24

M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Legend

Start day
Worked day
End day
Day not worked

Holiday/company closure

Total:	Out of	Remaining
90	90	0

Details:

Beginning of the internship : 4/09/2023

Company closure :

-3rd of October : German Unity Day (national holiday)

-10 days, from Christmas up to one week after NYE.

Expected end date: 19th/22nd of January 2024

Days planned not to work - 21st of September = MongoDB convention and meeting in Frankfurt. I have an invitation and I'd like to take part in the event.

Summary and planned activities

These milestones represent just a glimpse of my commitment to making this internship a successufl experience, both for myself as well as the host company. I am looking forward to working towards achieving this goal, by keeping in mind the following:

September

4 -> 8th September: Introductory week. Get to know the workplace, the team and familiarise myself with the workflow

11 -> 15 September: Work on initial projects and smaller assignments

18 -> 29 September: Gradually work towards intergrating with the main synergy of the team.

October & November

By the end of the second month, I plan to make substantial progress and demonstrate a clear grasp of the project's technical aspects. Throughout my internship, I will actively seek opportunities to collaborate with different teams and departments, fostering a holistic understanding of the company's operations.

By the end of the third month, I hope to have made meaningful connections and contributed to cross-functional projects.

December

I am dedicated to continuously learning and improving my skills. I will regularly attend company training sessions, workshops, and utilize online resources to enhance my technical and soft skills. Past the halfway point of my internship, I aim to have acquired new skills and applied them effectively in my daily tasks.

January & End

Spend the time finishing up my report, besides the work that will be assigned to me at that point in time. By the conclusion of my internship, I anticipate having a clearer vision of my career path and the necessary steps to achieve it. I trust that I will be prepared to go onto my graduation phase of the study by then.