Information Systems: Report on the International Practical Placement at the Usability Lab of Karlstad University

Jonas Frei and Timo Sökler

Karlstad Business School / Karlstad University
Information Systems – ISGC99
Date for Submission 2026-03-31

Table of Contents

1. Introduction

This final report documents the experiences, insights, and outcomes achieved by Timo and Jonas during their International Practical Placement at Karlstad University (KAU). The placement was conducted from January 19th until March 27th, 2025, as part of the course ISGC99 at Karlstad Business School.

Over the course of approximately nine weeks, the placement offered the opportunity to engage with a range of applied projects at the intersection of information systems, computer science, and digital humanities. The primary working environment was the usability laboratory within the university, which served as the base for collaborative and interdisciplinary work.

Three main projects defined the placement: the development of a 360° Laboratory Viewer designed to provide an immersive virtual tour of the usability facilities, the migration and improvement of LuwiaNet, a convolutional neural network for Luwian hieroglyph recognition, from Scala to Python, and the summarization of student usability assignments to consolidate key findings from prior experiments.

In this report, we reflect on the weekly progress made across these projects, summarize the key tasks and deliverables, and discuss how our prior academic studies in Business Information Systems and Data Science at DHBW Stuttgart related to the practical challenges we encountered. We conclude with reflections on what we have learned and how this placement has contributed to our personal and professional development.

2. Placement

Report Calendar Week 4

The first week of our placement at Karlstad University centered on familiarizing ourselves with the usability laboratory and its digital infrastructure. We conducted an initial audit of the laboratory's existing web presence and technological capabilities to understand what tools and resources were available for supporting interdisciplinary research projects. As part of our introduction we were assigned the task to add a description of the Labs new VR headset and Insta360 X5 360° camera so that students can inform themselves about the tools that the usability laboratory can provide. To further increase visibility of the labs capabilities we worked out to create a 360° viewing webpage.

Report Calendar Week 5

This week was dedicated to making tangible progress on the 360° Laboratory Viewer. The first step was to become proficient with the Insta360 X5 camera. After installing the required desktop and smartphone applications, we found the process of connecting to the camera via Bluetooth and capturing images to be straightforward. Importantly, the remote control feature allowed us to take pictures without an operator standing near the camera, which was essential for creating clean, unobstructed panoramic shots of the usability laboratory.

With all images captured, the next challenge was to find a suitable way to display them on the web. Together with John Sören, we conceptualized a Google Street View-like browsing experience and evaluated several approaches ranked by their level of immersion: a JavaScript-based viewer embedded directly into the KAU website, a viewer hosted on a separate linked page, a 360° YouTube video, or a standard rotation video. We prioritized the most immersive option and began developing a Proof of Concept.

During our research, we discovered Pannellum, a lightweight JavaScript framework for rendering 360° images in a 2:1 equirectangular format. Pannellum supports hotspots for connecting different scenes and information points for adding contextual descriptions of equipment and rooms. The PoC was developed quickly and confirmed that the intended Street View experience was technically feasible. However, we noted that the initial photographs could be improved by including open doors and depicting the lab in use, with testers and observers present, to give future viewers a more realistic impression of the usability facilities' workflow.

Report Calendar Week 6

After coordinating with Pierre and the IT department, we obtained a hosted server on which we could publish the 360° viewer website. The initial deployment was based on our earlier PoC. However, we encountered connectivity issues: some devices were unable to reach the URL, a problem we would continue to troubleshoot in the following weeks. Plans for the next iteration included retaking the photos with people actively working in the lab, adding detailed information points via Pannellum, and linking these to the official Ozlab KAU website for extended descriptions. We also conceived the idea of a toggle between an "in-use" view and an "empty lab" view, implemented through hotspot-based scene switching.

A significant new project emerged this week: LuwiaNet. We joined two video calls with Annick Payne, Lorenzo D'Alfonso, and Giulia Morra to learn about an existing image recognition system for Luwian hieroglyphs. LuwiaNet, originally written in Scala by Giulia, was capable of recognizing some of the most common hieroglyphic signs, but it had notable limitations, including an incomplete sign inventory and potential challenges related to data scarcity and the variability of hieroglyphs across time periods and locations. We received access to the full codebase and the associated Google Drive containing training data. The team explained the manual annotation pipeline: photographs of hieroglyphic tablets are traced onto 2D outlines, then individual signs are tagged using VLT before being fed into the recognition model.

Report Calendar Week 7

On Monday, we received the Scala source code for LuwiaNet and began an in-depth analysis. The codebase proved challenging to navigate: it contained two separate neural network architectures (LuwiaNet and GiuliaNet), much of the code was either undocumented or documented in Italian, and the project relied on the Metaliquid machine learning library running on a Java Virtual Machine, a library now considered outdated. Since the code could not simply be uploaded to an AI tool for automated analysis, understanding its structure required considerable manual effort, especially given our lack of prior experience with Scala.

After evaluating the situation, we decided that a full migration to Python using modern machine learning libraries such as PyTorch or TensorFlow would be the most sustainable approach. These frameworks support all functionalities provided by Metaliquid while making the project accessible to a wider range of collaborators. We scheduled a code walkthrough session with Giulia for the following week to clarify remaining open questions before committing to the migration.

In parallel, we discussed the need for server infrastructure with John Sören, who agreed to investigate what computational resources KAU could provide for the project. On the 360° viewer side, the connectivity issues reported in the previous week had not yet been resolved. We contacted Pierre again to seek assistance.

On Friday, Malin introduced us to a new task: summarizing student assignments in which KAU students had used the usability facilities to evaluate the e-services of the Karlstad municipality. The assignment reports included usability tests and conclusions, and our role was to extract and consolidate the key findings regarding the tools used, the methodology, and the resulting recommendations. This task also served as a practical introduction to the usability facilities capabilities and typical use cases.

Report Calendar Week 8

Week 8 saw progress across all three main projects. On the 360° Laboratory Viewer, we continued improving the Pannellum-based interface, successfully integrating basic info points as interactive markers and implementing two distinct viewing modes: an Explorative mode for free navigation and a Live mode for synchronized or guided viewing. A dedicated GitHub repository was created to manage the codebase and track future development.

For LuwiaNet, we conducted a comprehensive deep dive into the existing Scala codebase to map out the architecture and document the specific functionality of each file. An overview document was produced summarizing the system's logic. In parallel, we performed background research on convolutional neural networks (CNNs) and character recognition methods relevant to the project. The first steps of the Python migration were taken: we developed a Proof of Concept demonstrating how the LuwiaNet architecture could be structured using PyTorch. On the data side, we were still awaiting a response regarding the folder structure of the training images and sent a follow-up email. We also contacted the IT desk to discuss server resource provisioning; Sebastian from IT indicated that exchange students could likely be granted access.

The summarization of student assignments was formally kicked off this week. We began reviewing materials and defining task goals, but encountered several issues: Project 4 was entirely missing, and Project 5 (Group 10) was incomplete. After communicating these gaps, Malin provided the missing files. We also reported access permission issues for certain required documents.

Report Calendar Week 9

This week included an important Ozlab onboarding session on Thursday, during which we were walked through the full workflow: starting the lab systems, managing passwords, accessing relevant websites, operating the Wizard interface, and setting up the test person environment. Timo served as a test person during the demonstration. On Friday, we assisted master students during their own Ozlab session and, together with John Sören, demonstrated the setup to two additional students.

We also participated in individual consultations with five master students who were planning their usability projects. These conversations covered a range of topics: one student planned to continue a prior course project involving data storage questions, another was designing a group ordering feature for a food delivery context, and a third, Charles, working on a project called NutriStudent, was developing an app to help students manage grocery shopping and meal planning based on budgets and expiry dates. We supported Charles with his Ozlab onboarding and provided an overview of available features.

On the LuwiaNet front, we reviewed the repository structure in preparation for a meeting with Giulia on Friday afternoon. During the call, Giulia clarified which images had been used for training, enabling us to finalize our Python PoC. She also explained the role of the recognition dataset used for fine-tuning and the baseline MNIST-trained model that preceded it. We discussed the possibility of using a pretrained MNIST model as a starting point for our own fine-tuning pipeline.

For the student assignment summaries, we completed the summarization of the fifth assignment on Wednesday, adding content to sections two through four. The next step was to align the formatting with the existing document style.

Report Calendar Weeks 10 & 11

Week 10 was largely dedicated to a university excursion to Arctic northern Scandinavia, providing a mix of cultural and outdoor educational experiences. As the trip spread over two weeks we decided to combine those two reports. We visited Narvik and learned about the historically significant iron ore mine in Kiruna, including its railway connection and the city's occupation during World War II. Local cuisine, specifically Torrfisk, was sampled. The program included a snowshoe hike through the Arctic landscape, an Arctic survival skills workshop covering fire-making and shelter construction in sub-zero conditions, and an excursion to Abisko National Park. We met reindeer herders who introduced us to Sámi culture, including traditional herding practices, a Joik performance, and an overview of indigenous heritage. A husky sledding experience near Kiruna and an overnight Arctic survival exercise, sleeping in a self-built snow shelter under the Northern Lights, were among the highlights of the trip.

Upon returning, project work resumed. On the 360° viewer, the access problem with the hub.cse.kau.se server was resolved with help from CSMA support. We discussed the report structure with Malin and John Sören, who advised following the academic or semi-academic writing style used by previous placement students. A follow-up photo session with master students in the usability laboratory was scheduled.

For LuwiaNet, we achieved a major milestone: the first PoC model was created. Built on EfficientNet-B0 (pretrained on ImageNet), the classifier takes a cropped image of a single hieroglyphic sign as input and outputs the corresponding sign class. A Git repository was established for collaboration and version control. Initial training was conducted locally, and we planned to compare performance against the server environment once GPU access via hub.cse.kau.se was confirmed which it was during this period. The train-test accuracy of the first model was notably high, prompting us to plan a careful review of this result during further development.

On the summarization front, Malin delivered the remaining missing files for Assignment 4 and Group 10 of Assignment 5, and we began working on the corresponding summaries. Formatting issues, including a broken table of contents and inconsistent table styling, were identified and flagged for resolution.

Report Calendar Week 12

This week produced significant results, particularly for LuwiaNet. We conducted a comprehensive benchmark evaluation of the first trained model using a dedicated set of benchmark images. The key results were encouraging. Overall accuracy reached 93.1% (27 out of 29 correct classifications), with zero errors among high-confidence predictions (threshold of 0.85 or above). Of the 29 predictions, 14 (48.3%) fell into the high-confidence category, and the average confidence across all predictions was 0.79. Two classes exhibited low confidence (Class 35 at 0.43 and Class 209 at 0.41), while 14 out of 16 tested classes achieved perfect accuracy. Based on these results, we recommend applying a confidence threshold of 0.85 or higher for automated classification, as this yielded zero errors in our benchmark.

Motivated by these results, we emailed the wider team – Giulia, Lorenzo, and Annick – to request fully cropped and labelled tablet images, with the goal of extending the system from single-sign detection to full tablet detection. The team responded promptly, confirming that students would be assigned to carry out the labelling work. We followed up with a detailed specification of the required data format, including bounding box identifiers, sign identifiers, normalized coordinates, and unique filename-based pairing between images and CSV annotation files. We explicitly highlighted the importance of clean labelling to prevent data leakage.

For the 360° viewer, we conducted a photo session together with John Sören and master students, capturing new images for integration into the viewer. The viewer was subsequently reworked with the new photographs, including adjusted hotspot positions, improved coordinate detection, and informational overlay boxes describing the purpose and usage of each room.

The student assignment summarization was also finalized: all outstanding formatting issues were resolved, and the document was brought into alignment with the style and layout of the pre-existing content.

Report Calendar Week 13

The final week of the placement was characterized by consolidation and documentation across all projects. On Monday, the updated 360° Laboratory Viewer was sent to Pierre for deployment on the university servers, marking the conclusion of active development on that project. In parallel, work continued documenting the current status of LuwiaNet 2 to ensure that future collaborators could pick up where we had left off.

Toward the end of the week, we conducted a systematic benchmarking study to quantify the impact of different computing environments on model training performance. Runtime measurement was integrated directly into the training script to enable reproducible comparisons across three setups: a local machine, the KAU standard server, and the KAU GPU server. The results were clear and instructive. Training on the GPU server completed in approximately four minutes, making it by far the most efficient environment. The local machine required around 100 minutes for the same task. The KAU standard server, by contrast, did finish training after nearly a full day of runtime, a consequence of the significant performance throttling applied to general-purpose server nodes. The output files generated during this benchmarking exercise are later referred to in this report.

These findings reinforced a key practical insight: access to GPU resources provides a decisive advantage even when working with comparatively small neural networks. A local machine equipped with modern hardware represents a viable fallback, but relying on a standard university server for deep learning tasks is not advisable given the performance constraints involved.

3. Assignments during the practical placement

360° Laboratory Viewer

The goal of this project was to create an interactive, web-based virtual tour of the usability laboratory at Karlstad University. Using an Insta360 X5 camera, we captured 360° panoramic images of all three lab rooms. These images were then integrated into a custom-built viewer based on Pannellum, a lightweight JavaScript framework for equirectangular image rendering. The viewer supports navigation between rooms via hotspots and provides contextual information about equipment and room functions through interactive info points. Two distinct viewing modes were implemented: an Explorative mode for free browsing and a Live mode for guided or synchronized walkthroughs. The project was hosted on a KAU server and managed through a dedicated GitHub repository. Over the course of the placement, we iterated on the design multiple times, incorporating new photographs, including images of the lab in active use, and refining the user interface based on feedback.

Background information about the Insta360 X5

The Insta360 X5 is the latest flagship model in Insta360’s X series and is marketed as the toughest and most capable 360° camera the company has produced to date. At its core, it features dual 1/1.28-inch sensors with an F2.0 aperture, enabling 8K 360° video at 30fps and 72-megapixel still images. A Triple AI Chip system, consisting of one dedicated AI chip and two Pro Imaging Chips, handles real-time noise reduction and image processing, resulting in clear footage even in low-light indoor environments such as the usability laboratory. The camera is certified IP68 waterproof to 49ft (15m) and weighs approximately 200g. One of its most practically relevant features for lab documentation is FlowState Stabilization combined with 360° Horizon Lock, which ensures steady footage even without a tripod or gimbal. The X5 also introduces a replaceable lens system, allowing damaged lenses to be swapped out in the field. For connectivity and control, the camera pairs with both the Insta360 mobile app and the desktop application Insta360 Studio, which provides full manual editing and export capabilities at the highest quality. A 2400mAh battery supports up to 185 minutes of recording in power-saving mode and can be fast-charged to 80% in approximately 20 minutes.[1]

Background information about Panellum

Pannellum is a free and open-source panorama viewer for the web, developed by Matthew Petroff and hosted publicly on GitHub. It is built entirely using HTML5, CSS3, JavaScript, and WebGL, making it completely plugin-free and deployable on virtually any standard web server. The viewer can be integrated into web pages either via an <iframe> element for quick embedding or through a more powerful JavaScript API that allows tighter programmatic control. Configuration is handled via a JSON-based file, which enables fine-grained control over the starting view (pitch, yaw, and horizontal field of view), auto-rotation behavior, scene transitions, and UI elements such as the fullscreen button and compass. Pannellum supports three panorama types: equirectangular (the format produced by the Insta360 X5), cubemap, and a tiled multiresolution format suitable for very high-resolution images. For virtual tours, multiple scenes can be defined in a single configuration file and linked through hotspots, with a firstScene property specifying the entry point. The JavaScript API additionally exposes methods for dynamically adding or removing hotspots at runtime, changing the active scene, querying the current view angles, and responding to events such as the completion of a panorama load.[2]

LuwiaNet: Hieroglyph Recognition with Deep Learning

LuwiaNet is a neural network designed to recognize Luwian hieroglyphs from annotated tablet images. At the start of our placement, the existing system was written in Scala using the outdated Metaliquid library. Our task was to understand the architecture, migrate it to Python using PyTorch, and improve its capabilities. After an extensive analysis of the original codebase and consultations with Giulia Morra, the original developer, we built a new classifier based on EfficientNet-B0 (pretrained on ImageNet). The model takes a cropped image of a single sign as input and outputs the predicted sign class. Benchmark evaluation on a dedicated test set yielded an overall accuracy of 93.1%, with zero misclassifications among high-confidence predictions. We recommended a confidence threshold of 0.85 for automated classification. By the end of the placement, we had initiated the next phase of the project: extending functionality from single-sign recognition to full tablet detection, for which we specified a detailed data annotation format and coordinated with the research team to begin labelling work.

Background information about the Anatolian hieroglyphs

Luwian hieroglyphs, also known as Anatolian hieroglyphs, are an indigenous writing system that was used across Anatolia and northern Syria from roughly the fourteenth to the early seventh century BCE. The script consists of approximately 500 unique signs and is semanto-phonetic in nature: signs can function as logograms representing whole words, as syllabograms representing individual syllables (typically a consonant-vowel pair), or as determinatives that indicate the semantic category of the following word. Texts are written in boustrophedon style, meaning the direction of writing alternates from line to line, and signs are typically arranged in vertical columns of two to four along horizontal lines. Unlike Egyptian hieroglyphs, with which they share a superficial visual similarity, Luwian hieroglyphs developed independently and have no demonstrated connection to any other known script. The writing system was used primarily for monumental stone inscriptions commemorating the deeds of rulers as well as on royal and official seals. A small number of inscriptions survive on more perishable media such as lead strips, which give a rare glimpse into non-monumental use of the script. The decipherment of Luwian hieroglyphs was a long process that unfolded through the twentieth century. A central challenge for automated recognition is the significant visual variability of the signs across time and region: individual hieroglyphs evolved differently depending on the scribal tradition of the area in which they were carved, meaning that the same sign can take on markedly different shapes in inscriptions from different centuries or geographic regions. This variability, combined with the relatively small and specialized nature of the surviving corpus, makes hieroglyph recognition a particularly demanding task for machine learning approaches.[3]

Training Runtime Comparison

To assess the practical feasibility of training LuwiaNet 2 in different computing environments, we conducted a systematic benchmarking study comparing three distinct setups: a laptop, a KAU GPU server equipped with an NVIDIA GeForce RTX 2080 Ti, and a KAU standard server operating on CPU only. Runtime measurement was integrated directly into the training script, which records per-epoch timing and throughput for both training stages: Stage 1, in which only the classification head is trained, and Stage 2, in which the full network is fine-tuned. All three runs used an identical configuration of 8 Stage-1 epochs and 20 Stage-2 epochs with a batch size of 32, applied to the same dataset of 4,023 training and 1,119 validation images across 28 classes.

The results revealed a performance difference of multiple orders of magnitude between the three environments. The GPU server completed the full training run in approximately 3.7 minutes (222.7 seconds), processing images at a mean throughput of around 1,217 images per second during Stage 1 and 460 images per second during Stage 2. The laptop, which ran PyTorch in CPU mode, required approximately 107 minutes (6,398 seconds) for the same task, yielding a throughput of around 47 images per second in Stage 1 and only 17 images per second in Stage 2. The KAU standard server -- despite having 20 physical CPU cores and 755 GB of RAM -- performed significantly worse, completing the full training run in approximately 21.7 hours (78,098 seconds), with a mean throughput of only 2.7 images per second in Stage 1 and 1.4 images per second in Stage 2. These results indicate that the GPU server is approximately 29 times faster than the laptop and roughly 351 times faster than the KAU standard server.

Environment	Total Time	Throughput (Stage 1)	Throughput (Stage 2)
GPU Server (RTX 2080 Ti)	~3.7 min	1,217 img/s	460 img/s
Laptop (CPU only)	~107 min	47 img/s	17 img/s
KAU Standard Server (CPU)	~21.7 h	2.7 img/s	1.4 img/s

Table: Training runtime and throughput comparison across environments.

The disproportionately poor performance of the standard server is particularly noteworthy. Although the machine is substantially more powerful in terms of raw hardware, featuring 40 logical cores and significantly more memory than either of the other environments, its effective throughput was the lowest of the three. This is consistent with the hypothesis that the KAU standard server applies significant CPU throttling on shared nodes, which severely restricts the available compute for individual processes. Notably, the server had PyTorch installed with CUDA support (torch version 2.10.0+cu128), yet CUDA was reported as unavailable at runtime, suggesting that no GPU was allocated to the job. The combination of CPU throttling and the absence of GPU access renders this environment unsuitable for iterative deep learning development.

Despite the substantial differences in training speed, all three environments converged to identical final model quality: each run achieved a best validation accuracy of 100% on the 1,119-image validation set, with comparable final loss values. This confirms that the hardware environment did not influence the model's ability to learn from the data, only the time required to do so. For future work, these findings strongly support prioritizing GPU-accelerated infrastructure for all training tasks. Given the confirmed availability of GPU resources on the KAU GPU server, this environment should be treated as the default for model training throughout the remainder of the project.

Summarization of Student Usability Assignments

This task involved reviewing and consolidating the results of usability experiments conducted by KAU students. In these experiments, students had used the tools available in the usability facilities to evaluate the e-services website of the Karlstad municipality. Our role was to read through the individual assignment reports, extract the most relevant findings regarding methodology and tool usage, and synthesize the outcomes into a structured summary document.[4] The work required resolving several data gaps, some assignments and group reports were initially missing and aligning the final document's formatting with pre-existing content. Completing this task also served as a practical orientation to the usability facilities and typical research workflows.

Background information about the student's assignments

The evaluations covered four specific e-services offered by the Karlstad municipality: Child Welfare Reporting, Equipment Rental, Senior Tech Support, and Civil Marriage Booking. A total of twelve student teams participated, each selecting one or two of these services to test. The course followed a structured five-assignment progression, with each assignment applying a different evaluation method. This multi-method approach allowed findings to be cross-validated and refined across successive rounds of testing. The resulting summary document, which we co-authored with Malin Wik and Karolina Popek, organized the outcomes into structured tables covering usability findings and design recommendations across three dimensions: navigation, information structure, and graphic design. Among the most recurring issues identified across teams were the use of unclear terminology (such as the Swedish phrase "helgfri lördag" in the marriage booking form), insufficient visibility of key details such as location, time, and pricing, and a general absence of integrated help elements to support less experienced users. Based on these findings, the report concludes that improvements to navigation clarity, information hierarchy, and accessibility would meaningfully raise the usability of the municipality's digital services for citizens, entrepreneurs, and organizations alike.

Ozlab Onboarding and Master Student Support

Throughout the placement, we contributed to the Ozlab's operations by participating in onboarding sessions and supporting master students with their usability projects. This included learning the full workflow from system startup and Wizard configuration to test person management and subsequently demonstrating these processes to other students. We provided individual consultations to several master students, helping them plan their experiments and navigate the Ozlab environment. This supporting role deepened our understanding of usability testing methodology and the practical challenges involved in setting up controlled experiments.

Background information about eye-tracking in the Ozlab

The usability lab is equipped with eye-tracking technology, particularly well-suited for usability testing of web pages and graphical elements of interactive programs. It is also possible to record where a person looks when watching films and video commercials. The eye-tracker monitor is manufactured by Tobii Technology, a world leader in eye-tracking and eye-control solutions. The accompanying software, Tobii Pro Lab, allows test leaders to set up experiments, register test participants, select different stimulus types (e.g. images, web pages, movie clips), and analyse recorded gaze data — for instance by defining Areas of Interest (AOIs), generating heat map visualisations, and replaying individual gaze patterns. Since 2025, two dedicated workstations for eye-tracking are available in the control room, enabling data processing to run in parallel with ongoing recording sessions. The eye-tracker and a computer with the necessary software can also be transported for studies outside the lab environment.[5]

4. Relation to earlier studies

Our academic background at DHBW Stuttgart is in Business Information Systems with a specialization in Data Science. The curriculum combines modules in informatics, business administration, mathematics, and key qualifications. Throughout this placement, we identified several meaningful connections between our prior coursework and the practical challenges we encountered at Karlstad University.

Machine Learning and Data Science

The LuwiaNet project drew directly on knowledge from our Data Science studies, particularly in the areas of neural network architectures, image classification, and model evaluation. Concepts such as convolutional neural networks, transfer learning, and the CRISP-ML(Q) framework, which we studied concurrently during a Data Exploration course at DHBW were directly applicable. The process of migrating the model from Scala to Python using PyTorch mirrored the practical coding exercises from our programming courses, while the benchmark evaluation required a systematic approach to performance metrics and confidence calibration that we had covered in our statistics and analytics modules. The project also confronted us with challenges that we already experienced in our assignments and practical deployments, especially when working with datasets that were not specifically created for a classroom:

- A small and domain-specific dataset
- Understanding an undocumented legacy codebase
- And making deliberate architectural decisions under uncertainty.

The project also gave us direct exposure to the full machine learning lifecycle, from data preparation and model design through training, evaluation, and handover planning, in a way that closely mirrors what the CRISP-ML(Q) process model describes.

Web Development and Software Engineering

The 360° viewer project required proficiency in JavaScript, HTML, and CSS, as well as the ability to evaluate and integrate third-party frameworks. These skills were developed in our Web Programming course at DHBW, where we built web applications using Spring Boot and JavaScript. The evaluation of third-party frameworks and the downsides and upsides of using Panellum or another displaying option was a skill we had learned in multiple courses already. Additionally, version control with Git, which we used extensively for both the viewer and LuwiaNet projects, was a core competency from our software engineering modules, where we had worked with branching strategies, pull requests, and repository documentation. The experience of hosting a web application on a university server and troubleshooting connectivity issues also connected to our understanding of client-server architectures. What the placement reinforced was the ability to apply these skills in an iterative development process. The viewer went through multiple rounds of redesign, requiring us to manage an evolving codebase with shifting requirements while keeping the deployment stable. Designing a GitHub repository structure suited to a long-running research project, and coordinating changes between two contributors over the full duration of the placement, gave us practical experience with collaborative software engineering that coursework submissions had introduced us to in our case-study course.

Usability and Human-Computer Interaction

While our Data Science curriculum does not focus heavily on usability engineering, our Marketing module at DHBW introduced us to concepts such as user experience design, stakeholder analysis, and methods for gathering user feedback. These concepts proved relevant when summarizing the student usability experiments and when supporting master students with their Ozlab projects. From our studies we were already familiar with the idea that systems should be designed around the needs and behaviours of their users, the placement gave us a concrete and methodologically structured experience of what that looks like in practice. The placement significantly expanded our practical knowledge in this area, particularly regarding the Wizard-of-Oz technique, controlled experiment design, and the interpretation of usability test results.

Scientific Writing and Research Methodology

The task of summarizing student assignments required the ability to identify key findings, synthesize information from multiple sources, and produce a coherent summary document. These are skills we developed during our Basics of Scientific Work modules at DHBW, where we learned to manage references, structure academic texts, and formulate research questions. The placement reinforced these competencies and gave us exposure to the research environment at a Swedish university, including interactions with master students and their supervisors. Co-authoring the summary document further introduced us to the collaborative dimension of research writing, including the need to align on structure, and scope across contributors and how easily formatting errors can happen especially when using two different programs or more than one person editing the same document at the same time.

5. Analysis of IS Application in Practical Work

In this section, we consolidate our experiences to offer a broader perspective on how information systems theory can and sometimes cannot be directly applied in a practical working environment.

360° Laboratory Viewer

The viewer project illustrates how IS theory around web-based systems, user interface design, and iterative development translates well into practice. The decision-making process for selecting a suitable framework (Pannellum), the evaluation of hosting options ranked by immersion level, and the iterative improvement cycle based on user feedback all reflect standard IS development methodologies. However, the project also revealed practical friction points that textbooks rarely emphasize: server connectivity issues that persisted for weeks, the dependency on institutional IT support for deployment, and the challenge of coordinating photo sessions with multiple stakeholders. These operational realities required patience, persistence, and informal communication skills that go beyond formal project management frameworks.

LuwiaNet

The LuwiaNet project provided a compelling example of applying data science and machine learning theory to a real-world, interdisciplinary research context. The theoretical knowledge from our studies, CNN architectures, transfer learning, model benchmarking, was directly applicable and instrumental in achieving a 93.1% accuracy rate on the benchmark. At the same time, the project highlighted aspects that are underrepresented in academic coursework: the difficulty of working with undocumented legacy code in an unfamiliar programming language, the importance of clear data annotation standards to prevent leakage, and the need to coordinate with domain experts (in this case, archaeologists and linguists) who bring entirely different perspectives to the problem. The interdisciplinary nature of this collaboration was both the most challenging and the most enriching aspect of the project.

Summarization of Student Assignments

This task was more analytical than technical, but it nonetheless connected to IS principles around information extraction, knowledge management, and structured documentation. The ability to read through heterogeneous reports, identify common patterns, and produce a unified summary is a core competency in information systems work. The practical challenge lay in dealing with incomplete data, missing assignments and access issues which required proactive communication and follow-up. This experience reinforced the importance of data quality and completeness as prerequisites for any meaningful analysis, a principle that applies equally to database management and to qualitative research synthesis.

6. Final reflections

Looking back on the nine weeks of our practical placement at Karlstad University, we are grateful for the breadth of experiences and the depth of learning that this period has afforded us.

From a technical standpoint, the placement allowed us to apply and extend our data science skills in a meaningful way. The LuwiaNet project, in particular, challenged us to go beyond classroom exercises and engage with a real research problem that carried genuine academic significance. Achieving a 93.1% classification accuracy and laying the groundwork for full tablet detection gave us confidence in our ability to contribute to applied machine learning research. The 360° viewer project, meanwhile, strengthened our web development skills and taught us the value of iterative design and user-centered thinking.

Beyond the technical dimension, we gained substantial experience in academic and interdisciplinary collaboration. Working alongside researchers from archaeology and linguistics on LuwiaNet, supporting master students in the Ozlab, and navigating the institutional landscape of a Swedish university all contributed to our growth as professionals who can operate effectively across domains and cultures.

The Lappland excursion added a memorable cultural dimension to the placement. Experiencing Arctic survival conditions, learning about Sámi heritage, and witnessing the Northern Lights from a self-built snow shelter are experiences that transcend academic or professional categories and will remain with us for a long time.

On a personal level, the placement reinforced the importance of adaptability. Whether it was debugging server connectivity, deciphering undocumented Scala code, or adjusting project plans when data was missing, the ability to remain flexible and solution-oriented proved essential. These are qualities that cannot be fully taught in a lecture hall but become second nature through practice.

We extend our sincere thanks to John Sören, Malin, and the entire Information Systems team at Karlstad University for their warm welcome, consistent support, and the trust they placed in us throughout this placement.

List of sources

[1] “Insta360 | 360 Cameras - Official Website.” Accessed: Mar. 29, 2026. [Online]. Available: https://www.insta360.com/

[2] “Pannellum.” Accessed: Mar. 29, 2026. [Online]. Available: https://pannellum.org/

[3] L. d’Alfonso and A. Payne, “The Paleography of Anatolian Hieroglyphic Stone Inscriptions,” *J. Cuneif. Stud.*, vol. 68, p. 107, Jan. 2016, doi: 10.5615/jcunestud.68.2016.0107.

[4] T. Sökler, J. Frei, M. Wik, and K. Popek, “Usability Review of e-services: Evaluations by Information System students (KAU), September - October 2025.”

[5] “Eye tracking | Karlstad University.” Accessed: Mar. 29, 2026. [Online]. Available: https://www.kau.se/en/ozlab/usability-lab-and-systems/eye-tracking