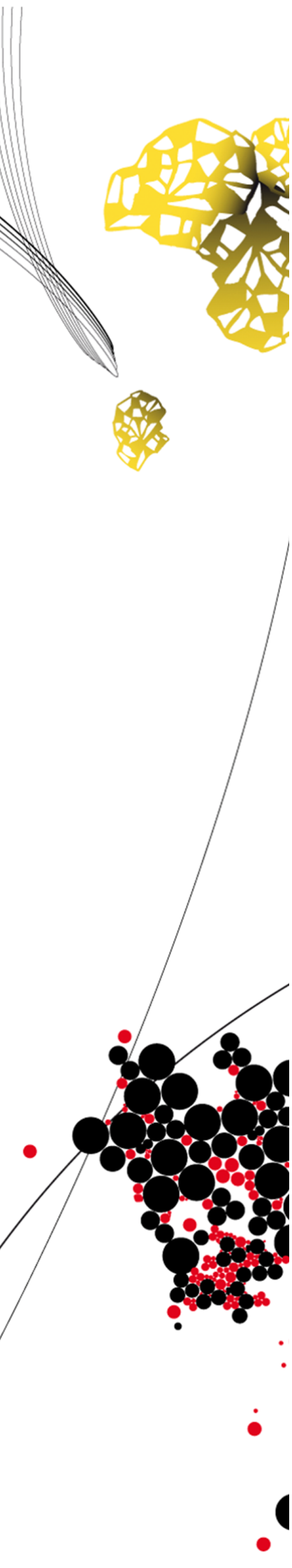

UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

From Software Data to Actionable Insights: Designing a Data Dashboard

Chenhao Yi
M.Sc. Interaction Technology

December 2023

First supervisor:	Dr. M. Theune (Mariet)
Second supervisor:	Dr. A. Karahanoglu (Armagan)
Co-supervisor:	Dr. C. Moral (Cristian) (Universidad Politécnica de Madrid)
Company supervisor:	Kaj de Hoop

Summary

In this era of data explosion, many companies want to use data for insights. This study focuses on the creation of a data dashboard to utilize software usage data, originating from the case company Movella. The company aims to gain useful insights from collected software data to better understand their customers and make informed decisions. Therefore, the research centered on designing a data dashboard that uses software usage data while meeting the diverse needs of Movella's professionals.

In my research, I adopted a user-centered design approach. Initially, I conducted user interviews to understand the needs, motivations, and preferences of different target roles. By analyzing the interview results, I identified the data and features to beavailable on the dashboard.. After that, I developed a low-fidelity prototype of the dashboard. This prototype was evaluated during usability testing, where it received positive feedback and suggestions for improvements. Subsequently, I implemented a Proof of Concept (PoC) for the data dashboard, exploring a viable development approach. This involved integrating software usage data into a database and connecting it to a visualization platform to create a dashboard tailored to user needs. The functional data dashboard received positive feedback during its final presentation at Movella, indicating its potential effectiveness in practical applications. Future directions will focus on real-time data updates and visualization improvements. In summary, this study lays the groundwork for developing a data dashboard that visualize software usage data.

Contents

Chapter 1

Introduction

1.1 Case company

Movella is a company that provides products to digitize movement.[1] Customers use Movella's products for sensing, capturing, and analyzing motion. Movella's customers come from various industries, including entertainment, health and sports, automation and mobility, and research and education. In the entertainment industry, Movella primarily engages in projects related to film production, gaming, and virtual reality.[2]. In the health and sports sectors, Movella focuses on areas such as sports performance and training management.

Figure 1.1: MVN motion capture hardware[3]

[1] *Motion Capture — Movella.com*. Available at: `https://www.movella.com/products/motion-capture` (Accessed on August 2, 2023).

[2] *Motion Capture — Movella.com* Available at: `https://www.movella.com` (Accessed on August 2, 2023).

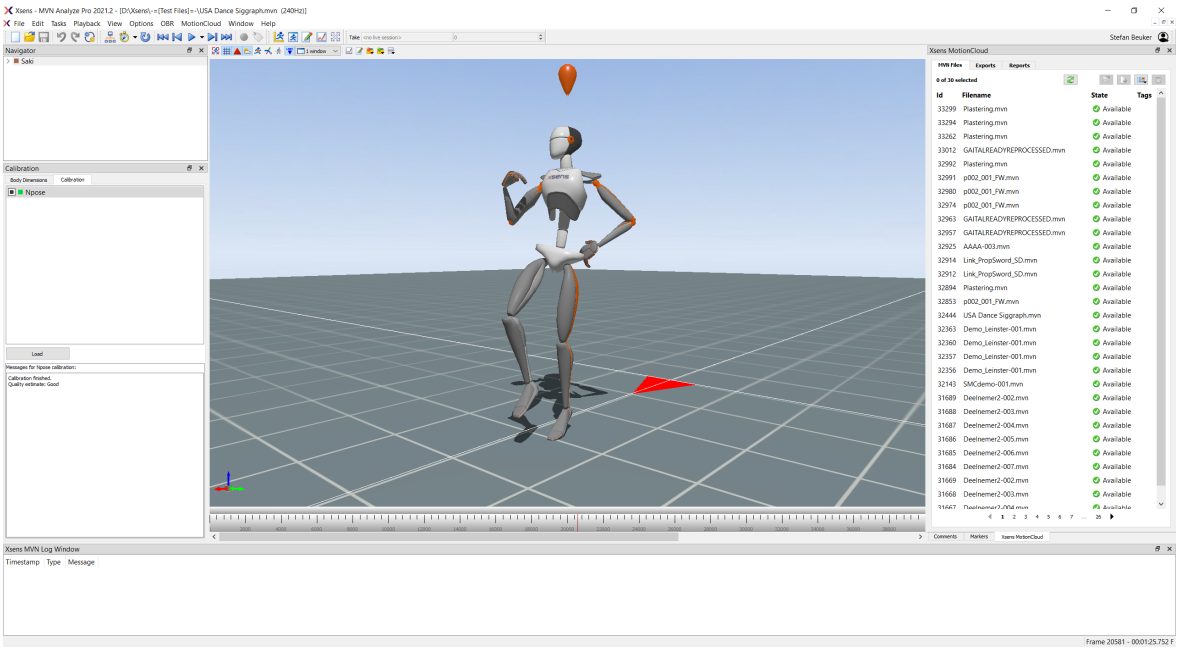

Figure 1.2: A screenshot of MVN software[4]

One of the key products offered by Movella is the MVN inertial motion capture system. Figure 1.1 showcases the hardware lineup of MVN motion capture products. The system is designed for capturing full-body human motion, leveraging miniature inertial sensors, wireless communication, and sophisticated biomechanical models. Comprising both hardware and software components, the MVN system includes specialized sensors, suits that facilitate motion capture and companion software. The MVN software is used in conjunction with the motion capture device. A screenshot of MVN software can be seen in Figure1.2. MVN software can be used to control the motion capture device and perform their tasks. Some typical tasks for which users use MVN software include calibrating the motion capture suits, starting and stopping recording of motion movement, monitoring the real-time movement and so on.

1.2 Problem statement

1.2.1 Introduction of the software usage data

Since 2018, Movella started to collected anonymous usage data that are generated when people are using the MVN software. The collected data consists of two parts.

[4] *Choosing xsens motion capture — Movella.com*. Available at: `https://www.movella.com/resources/blog/choosing-the-best-xsens-motion-capture-setup` (Accessed on September 2, 2023).

The first part presents basic data related to activities, such as the chosen version, operating system license. The second part focuses on recorded event data. Events are predefined actions that happen or are triggered in the MVN software. For instance, when a user initiates the MVN software, an 'MVNstarted' event is logged. With user consent, Movella logs these data generated during user interactions with the MVN software in log files. These log files can be periodically sent to Movella.

1.2.2 Current solution

At Movella, a script is currently executed to consolidate the gathered data, with the results displayed in an HTML file. An blurred example of this HTML file can be seen in Figure 1.3. This file encapsulates basic summaries of usage data, such as the number of users for each license on a monthly basis. However, employees barely use this HTML file as a tool to extract insights from the data. They find the tool to have limited readability, consisting merely of numbers and text. It becomes challenging for them to observe trends over time or compare multiple specific data to derive meaningful insights. Therefore, a new solution was needed to utilize the software usage data.

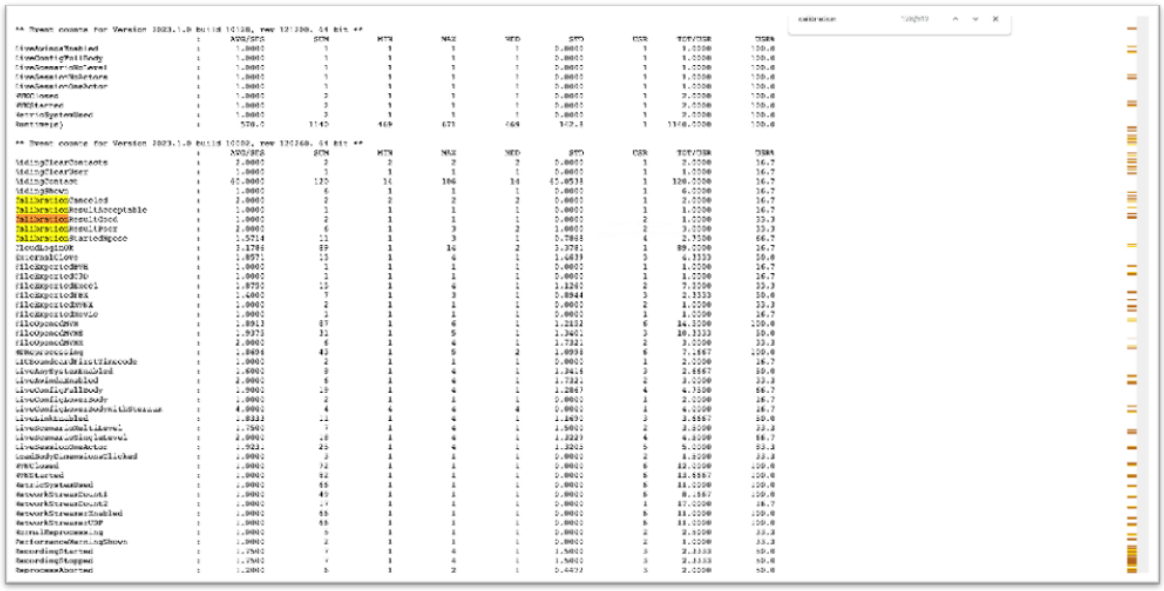

Figure 1.3: Current solution: a HTML file to present the software usage data

1.2.3 Utilizing software usage data

This research focuses on utilizing the software data generated when people are using MVN software. The utilization of these software data poses certain challenges.

Firstly, there is a scarcity of academic research on using software usage data, making it difficult to find academic achievements or solutions to serve as references. Secondly, cross-functional teams, consisting of professionals such as developers, designers, quality testers, marketers, and product managers, face challenges in deriving different insights from user statistics relevant to their respective workflows. Each role has different data interests, and designing a solution to utilize software usage data that meets everyone's needs presents a challenge. Furthermore, dashboard design is not a straightforward process [1]. MVN software has complex features and numerous data, and deciding which specific data generated during its usage to collect and analyze poses a challenge. Hence, this research aims to determine how to present data to meet the needs of the product development team.

1.3 Motivation

The motivation of the research can be summed up as follows. The first motivation is to inform the redesign decision of MVN software. The MVN software has been on the market for more than ten years, and it is currently undergoing a redesign. The interface and functionality of the software need to be updated and improved to align with the user needs and latest aesthetic trends. The software data might provide insights into which features are frequently used by users and which ones are barely used. This information serves as a valuable reference for decision-makers to determine whether to keep, remove, or expand specific software functionalities in the redesign process.

The second motivation is to understand users more and at a lower cost. Because it is expensive to collect real user feedback through different user research methods, such as user interviews, questionnaires, and so on. This is often costly in terms of manpower, budget and time. The number of users that can be reached is also limited. The MVN software development team want to get to know their customers from software data, aiming to design the MVN software that aligns closely with customer needs. Thirdly, the demand for software usage data is emerging within software development and quality testing processes. Utilizing software usage data facilitates the team's ability to conduct enhanced testing, maintenance, and iterative processes [2].

1.4 Scope and limitations

The main aim of the research is to design data dashboard for using software statistics to help the process of software design, development and maintenance. This research will focus on the needs of different professionals who want to use software

data. It will also focus on the way they want to interact with data and how they want to access the data to complete their tasks.

However, different companies and teams have different purposes for using software data, based on the software being used for different purposes. This study will focus on providing a useful solution for the case company. However, efforts will be made to create a generalizable solution and to give insight to other people that also want to use software data.

1.5 Research questions

The main research question of this study is:

How can a data dashboard be designed to utilize software usage data and cater to the distinct needs of various professionals within the product development team?

The following questions serve as sub-questions to address the main research question in greater detail:

1. What are different professionals needs and motivations when it comes to utilizing software usage data?
2. What key data and features should be included in the data dashboard?
3. What layout and organization of the dashboard should be designed to accommodate the needs and preferences of different professionals?
4. What is a useful approach to implement the data dashboard?

The aim of this study is to develop a solution that utilizes software usage data to cater to the needs of diverse individuals. This is because creating a separate solution for each type of user who wants to utilize software data would require an excessive amount of resources. Additionally, a singular solution is a specific requirement of the case company, Movella.

1.6 Overview

This study aims to explore how to design data dashboards to use software usage data, offering insights about MVN software usage. Chapter 2 discusses literature of related works and explores the methodology that can be used to achieve the research goals. In this research, I adopted a user-centered design approach, and from Chapter 3 to Chapter 5, I detailed the process from user research to prototype design and usability testing. In Chapter 7, I designed and implemented a proof of

concept for the data dashboard, studying feasible implementation solutions. Chapter 8 then dives deep into the contributions and challenges of the research, and limitations. The study concludes with a summary of the entire research. Through this study, I hope to provide Movella company with an effective tool to better utilize software data, understand software usage, and assist in data-driven decision-making. Additionally, I hope this research can offer a reference framework for other individuals or organizations who want to use software usage data.

Chapter 2

Background work

2.1 Data dashboard

Data dashboards are a visual display of data information, including graphical elements and data visualization [3]. Sarikaya et al. categorized dashboards into two types: one represented by static charts and numbers, and another allowing for interaction with real-time data [4]. Dashboards aim to help users understand various aspects of data, supporting data-driven decision-making [5]. They are widely used across multiple industries. For instance, in the field of information security, Pavel Yermalovich described a series of dashboard visualizations based on standardized role tasks in information security, enhancing response speed to security incidents through detailed visualizations [6]. In healthcare, Cannavacciuolo et al. demonstrated how data dashboards are used to assess the decision-making quality of nurses in emergencies [7]. In the e-banking sector, Buananta and Chowanda explored the use of business intelligence dashboards to understand user payment and purchasing behaviors, informing product promotion decisions [8]. The literature indicates that data dashboards are widely and effectively used across various industries, demonstrating their flexibility in adapting to different scenarios, needs, and audiences in terms of data utilization and presentation. However, I also noticed a near absence of literature on practices involving data dashboards using software data. Existing literature on data dashboard practices tends to focus more on public service industries such as healthcare and education. This may be due to software companies' reluctance to reveal their methods of utilizing software data, preferring not to provide competitors with insights. I argue that in-depth research into data dashboard design for specific domains is essential. Taking the software data field in this study as an example, research tailored to specific domains can provide concrete guidance. For instance, at the strategic level, we can explore which data can effec-

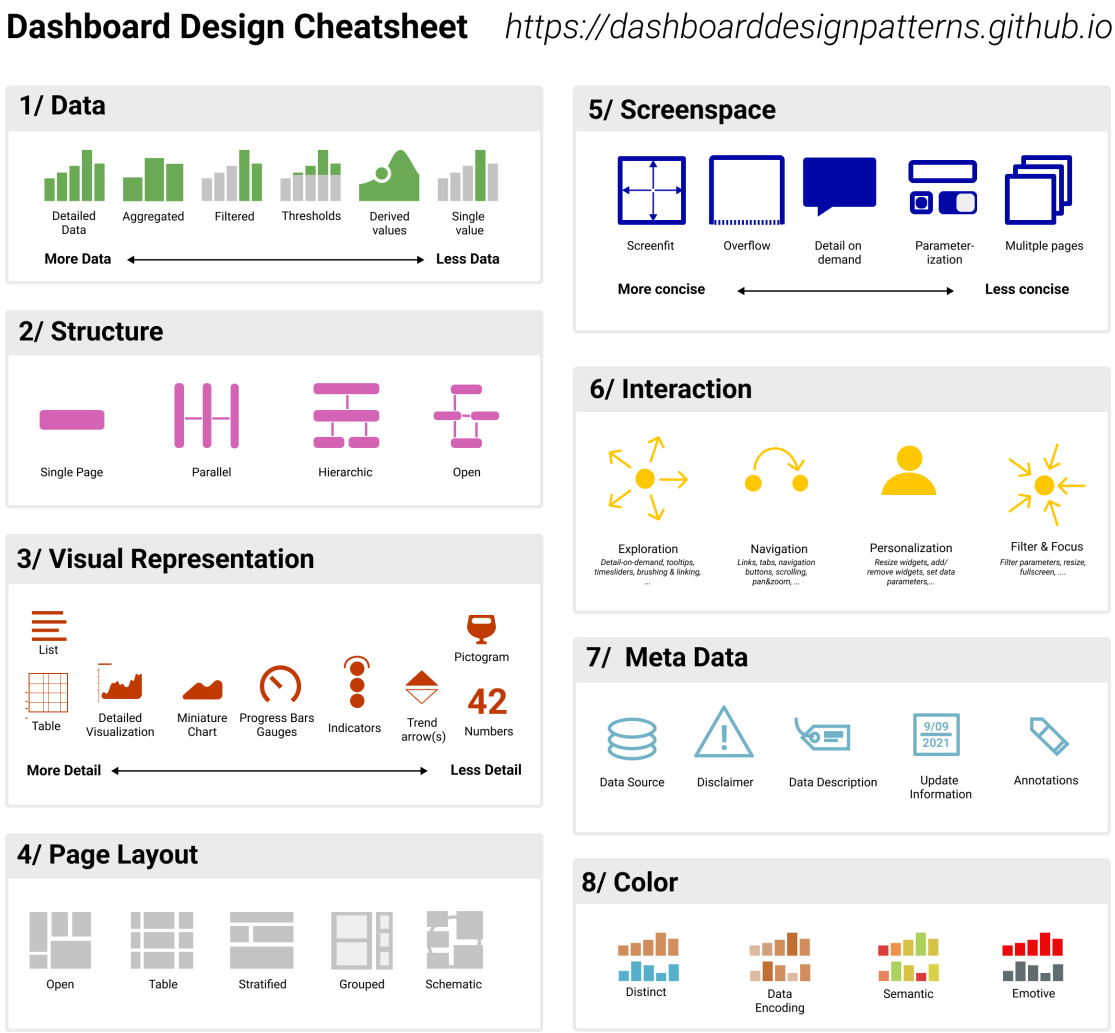

Figure 2.1: Dashboard Design Cheatsheet. From Bach, B. Dashboard Design Patterns [1].

tively measure software performance [9]; at the operational level, we can investigate which data can guide the daily tasks of front-line employees.

The design of data dashboards is a complex and multifaceted process that encompasses data selection, application of visualization techniques, design patterns, and consideration of user experience. Regarding data selection, Janes et al.'s introduction of the GQM (Goal-Question-Measurement) model in their research provides a pivotal insight into this process. This method, which involves defining questions based on user goals and subsequently selecting corresponding data, has informed the primary step in my dashboard design approach: to first clarify user goals and needs, ensuring that the choice of data displayed is driven by these collected user requirements. In the design of user interviews for collecting user goals that follow, I have drawn upon the approach of the Goal-Question-Metric (GQM) model. This involves clarifying the goals of the target role, identifying the problems they aim to solve, and determining the data capable of answering these questions. The GQM model has also become a popular method in designing and evaluating dashboards. Additionally, Wu et al. introduced a deep learning-based approach in their research to streamline the dashboard creation process. This method automates the selection of the most suitable visualization types for specific datasets, significantly simplifying the task [10]. Similar automation features are also present in mainstream data visualization platforms like Qlik Sense. The task of selecting data and fitting visualizations typically demands substantial effort, yet these algorithmically generated visualizations and dashboards can serve as practical references and sources of inspiration.

Regarding visualization methods, Toasa et al.'s research, which introduces commonly used visualization techniques such as automatic tables, correlation matrices, network diagrams, and word clouds, and discusses their functions and application scenarios, serves as a comprehensive reference. After gathering user requirements and determining the data to be displayed, their study provides a valuable guide to identify the most appropriate visualization techniques. This research, akin to a lexicon of visualization options, is instrumental in refining the final design of the dashboard, ensuring it meets the specific needs and preferences of the users. Regarding visualization techniques, platforms like Tableau, Google Analytics, and PowerBI provide a comprehensive range of options. These studies provided support and reference for selecting suitable visualizations in my design process.

Regarding design patterns, there are many general guidelines available to assist in the design process. As pointed out by Janes et al., understanding the complexity of these visualizations and the time it takes for users to interpret them is an important consideration [11]. I should choose appropriate visualizations based on audience's

understanding ability. Moreover, Janes et al. emphasized the significance of aesthetics and usability in dashboard design. Employing various design components, such as color and shape, to represent key information and draw user attention is a practical approach [11]. This approach ensures that the dashboard is not only informative but also visually appealing and user-friendly.

In the process of dashboard design, many decisions and trade-offs need to be made. Dashboard design patterns are needed, guiding us to solve more specific design questions, such as when to adopt a screen-fit layout, how to structure data, and the extent of data abstraction. Sarikaya et al. systematically reviewed 83 data dashboards across various fields, analyzing them from multiple design aspects including functionality, purpose, audience, and data [4]. This study provided me with a design space that expanded my design perspective, helping me explore possible dashboard design solutions and ideas. Additionally, Bach et al. thoroughly reviewed 144 dashboards and outlined eight groups of design patterns, as depicted in the Figure 2.1. These patterns cover essential aspects such as data, structure, visual representation, page layout, screen space, interaction, metadata, and color [1]. This systematic categorization of design elements is very useful, especially when starting a dashboard design from scratch. It offers a range of common layouts and structures, making the selection process for a design that suits my dataset more straightforward. The work of Sarikaya and Bach has laid a solid foundation for research in the field of dashboard design patterns.

2.2 User-centered design

User-centered design (UCD) is a design approach that emphasizes involving end users throughout the design process to create a product or service that meets their needs [12]. Core principles of UCD include early and ongoing user involvement, iterative design, and usability evaluation [13]. UCD approach enables designers to gain a deep understanding of user needs, preferences, and behaviors to create solutions that are both user-friendly and effective [14].

UCD methodology can be achieved through several techniques of design thinking [15]. Design thinking is a problem-solving and innovation approach that focuses on understanding users' needs, generating creative solutions, and iteratively refining those solutions through prototyping and testing. It is a human-centered and iterative process that combines analytical thinking with empathy and experimentation. Design thinking encourages a collaborative and interdisciplinary approach to problem-solving, involving individuals with diverse perspectives and expertise [16].

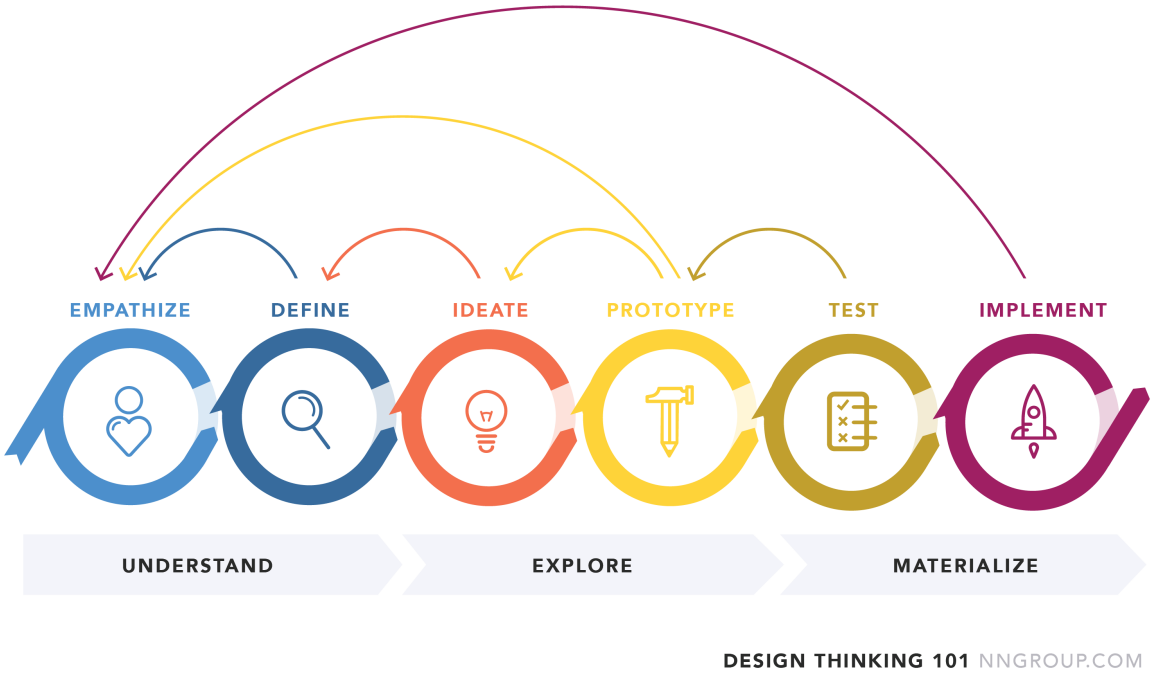

Figure 2.2: Figure: Design thinking process.
(Source: https://www.nngroup.com/articles/design-thinking/)

There are various iterations of design thinking process and principles [17]. Figure 2.2 serves as an illustrative example of a design process and its stages, which I find particularly relevant to the structure of my research involving the utilization of software data. It comprises 6 steps that can be followed in the solution design and development process.

1. **Empathize**: This stage involves a deep understanding of the users' needs, experiences, and motivations, primarily through user interviews and observations. For software data, this might mean understanding which data users are interested in and why.

2. **Define**: This step involves organizing insights gathered during the user research phase and clearly defining the problem to be solved. It entails reconstructing the problem from the users' perspective and identifying potential issues, such as challenges in interpreting data and the complexity of data processing.

3. **Ideate**: At this stage, brainstorming takes place to come up with multiple solutions for the defined problem. This may involve methods of collecting, analyzing, and presenting data. It encourages a divergent thinking approach to explore multiple possibilities and potential solutions.

4. **Prototype**: This step refers to creating concrete design prototypes or representations of potential solutions, such as physical models, digital mock-ups, etc. An example is the preliminary design of a data dashboard interface. The goal is to quickly and iteratively test and refine ideas based on user feedback.

5. **Test**: Involves having users test the prototypes and collecting their feedback and insights. The feedback is used for further refinement and iteration of the design.

6. **Implement**: This final step involves transforming the prototype into a usable product, such as developing an actual usable data dashboard.

In the design of data dashboards, a key consideration is the understanding and addressing of user needs. Dashboards that fail to meet these needs may not be fully utilized. For instance, a major issue in applying visualization in agile software development is usability issues. This arises when development teams focus more on the development aspect of information visualization, with less consideration for usability, potentially reducing the overall quality of the design [18].

Take, for example, a case at the University of Missouri Hospital's medical ICU. A dashboard designed for team communication was underutilized because it did not fully meet user needs. Nurses reported difficulty in reading the dashboard and expressed a desire for it to cover data needed for daily monitoring in the ICU [19]. This underscores the importance of User-Centered Design (UCD) in enhancing the user experience in dashboard creation [20]. Researchers have successfully incorporated UCD in dashboard design, achieving positive results. For instance, Tsangaris et al. developed a platform called imPROVE, which supports patient-reported outcomes in breast cancer care. This platform was designed through a user-centered approach and agile development. This process led to the creation of a patient mobile app and a clinical dashboard [21]. These case studies demonstrate that UCD helps designers better understand user needs, enabling them to create more intuitive and effective data dashboards for data-driven decision-making processes.

In my research, I aim to employ UCD methods to create a data dashboard that truly addressed the needs of users. Additionally, the data dashboard I designed was intended for use by employees from various departments and with different technical backgrounds at Movella. Utilizing UCD methods enabled me to understand and adapt to the diverse needs of these user groups.

2.3 Data-driven decision-making

Data-driven decision making (DDDM) involves using data and analytics to inform decision-making processes. It emphasizes the importance of objective, evidence-based decision making rather than relying solely on intuition or subjective judgments. Software vendors aim to analyze this data to gain insights into enhancing user engagement and improving product usability [22]. Employing data-driven decision-making has a positive impact on a company's performance [23]. Research suggests that organizations can make decisions in response to market changes, such as identifying new business opportunities and understanding customer needs, by analyzing and visualizing data, thereby enhancing their competitive advantage [24]. For instance, marketers can make informed decisions about where to place advertisements based on consumer response data collected from previous ad campaigns [25]. Similarly, UX designers can evaluate the effectiveness of user experience by analyzing user behavior data, such as click-through rates and browsing duration [26].

In this research, a primary motivation for utilizing software data is to aid in decision-making. In the realm of software development, key figures such as project and product managers, along with technical team leaders, face numerous factors and high complexity. Gathering and analyzing data offers a quantitative and more scientific method compared to relying solely on intuition and instinct. Currently, there are several frameworks for data-driven decision-making, particularly in educational settings [27]–[29]. Gill et al. proposed a high-level and generalized theory of action and organizational supports for data-driven decision-making aimed at enhancing student performance [28]. This process involves acquiring high-quality raw data, followed by data analysis, and ultimately using relevant and diagnostic data to inform instructional and operational decisions. There is also a wealth of literature focusing on specific practical cases, demonstrating the analysis of large-scale data to support data-driven decisions in areas such as emergency management [30], urban planning [31], healthcare [32], and more. However, research on the framework of data-driven decision-making in the context of software development and usage is limited. A study posed that DDDM can be used as a tool to improve productivity in software development [33]. Svensson et al.'s study indicates that the widespread use of DDDM in agile software development is currently minimal, with practitioners holding a positive attitude towards the potential of DDDM in software development [34]. This indicates that implementing DDDM in the field of software development holds considerable potential, offering opportunities to enhance productivity.

2.4 Conclusion

This chapter provides an overview of key concepts relevant to the research. It introduces data dashboards as tools for data presentation and insight acquisition. The discussion extends to how existing research addresses essential aspects of dashboard creation, including data selection, visualization technique choice, common design patterns, and consideration of user experience. These serve as a solid foundation and good reference for my dashboard design. Furthermore, the chapter highlights the effectiveness of user-centered design (UCD) methodologies in ensuring dashboards meet user needs effectively. Additionally, it underscores the importance and driving forces behind data-driven decision-making models. At Movella, current solutions utilizing software data are scarcely applied. To design a dashboard that truly satisfies user needs, I have adopted the UCD approach. In subsequent chapters, following the design thinking process outlined in this chapter, I embark on the dashboard design journey, starting with the collection of user requirements.

Chapter 3

User research

3.1 Interview objectives

The objective of user interviews is to gain a deeper understanding of the target users of the software usage data, including their preferences, thoughts, and expectations. The insights gathered from these interviews will serve as a solid foundation for designing solutions to make use of the software data.

Specific interview goals include:

1. Understanding the current workflow and job responsibilities of each professional. The target users come from various departments within Movella company, each responsible for different stages of the product development process.

2. Exploring the challenges professionals face in their work and the processes they employ to tackle the problems. I want to gain inspirations about how software statistics can potentially help address these challenges.

3. Gaining insights into the target users' perspectives on software statistics, including what data they are interested in and the corresponding potential applications.

By understanding the aforementioned aspects, I anticipate being able to align the software statistics more effectively with their workflow and assist in resolving their work-related challenges.

3.2 Participants

Within the case company, five different roles were chosen as the participants for this interview. They are UX designers, quality testers, product managers, software developers, and product specialists. This was due to the fact that these roles had more often requested and expressed their motivation to use the software data. To gather a comprehensive perspective, I invited at least two people for each role.

Recruitment of participants was done by invitation through mail. In the email, clear and straightforward language was used to explain the purpose of the research, the interview questions, the location, the time, and the expected duration. Additionally, an information letter and consent form were attached, providing them with most of the details about the interview, except for the specific interview questions, which were not shared in advance due to the research's purpose. Therefore, a general overview of the question content was provided.

Based on the responses and availability of participants, a total of 12 participants participated in the interview. The number of participants for each role is shown in Table 3.1. Only one quality assurance engineer participated in the interviews. In addition to the target roles, the product owner, who is interested in the topic of using the software data, participated in the interviews and shared many valuable insights.

Role	Number of participants
UX designer	3
Product owner	1
Quality assurance engineer	1
Software developer	2
Product support specialist	2
Product manager	3

Table 3.1: Interview participants

3.3 Interview questions

One-on-one interviews were chosen because this format allowed for a deeper delve into the thought processes of users. Follow-up questions tailored to individual users' responses were asked to further explore their thoughts and gather more specific and detailed responses. The user interviews were designed as semi-structured interviews. The interview scripts were categorized into three main topics for all partici-

pants. The first topic focused on discussing their past job responsibilities, workflows, and pain points related to their respective roles. The second topic aimed to explore their perspectives on software statistics and which data they found most interesting and relevant to their work. Lastly, the third topic involved asking participants about any potential barriers or concerns they had regarding the use of software statistics.

The interview questions for each role are listed in Appendix A.1. Different interview questions were designed for each role, using the three topics as the main framework. Follow-up questions were planned based on the participants' answers to uncover more useful information and understand the participants' mental models accurately. For example, for UX designers, they were asked about how they had been designing solutions to improve user experience and how using software data could help them understand users and make design decisions. For product managers, discussions revolved around how software data could be used to track product performance and impress investors.

By adopting this approach, the aim was to ensure that the interviews were more meaningful and directly relevant to the participants' professional backgrounds.

3.4 Analyzing interview results

3.4.1 Interview summary

With the informed consent of the participants, all interviews were audio recorded and transcribed. I organized the key phrases from the interviews and categorized them based on the predefined interview question framework. In this section, I will present the key interview results for each role in Table 3.2, including the workflow and challenges, and their insights software usage statistics.

Table 3.2: Interview summary for each role

Role	**Workflows and Challenges**	**Insights in Software Usage Statistics**

UX Designer	UX designers perform various tasks, including user research via interviews, observations, and tests to find usability issues. They manage research data, pinpoint user issues, and develop new interfaces for the MVN product. They collaborate with product managers and developers on design and prioritizing usability issues. However, they face challenges like limited data on usability issue frequency and difficulty identifying issues in MVN product. Additionally, global client distribution makes gathering user research costly.	UX designers aim to fully understand user interaction with the software to improve user experience. Their focus includes analyzing time intervals between events like software startup and recording initiation, to evaluate the software's performance. They also look at feature usage, such as body dimension inputs, to guide software redesign. By identifying patterns in usage data, they seek to refine user workflows. They plan to display these statistics on a website for quick access and also suggest creating automated reports for ease of use.
Quality Assurance Engineer	Their responsibilities include testing both software and hardware, involving manual and automated approaches like functional, performance, and security testing. They compile inputs from QA, product support, and product managers to form a thorough test selection, also using software usage data for this purpose. A key challenge is the absence of a quantified method for prioritizing test features, with decisions often based on human judgment and assumptions.	They want to know how extensive and whether to test a specific language, version, license, operating system, and a particular feature to conduct a rational and effective software testing process. Hence, they focus on the usage metrics of these elements and monitor the trends over time and across different software versions.
Software Developer	Developers handle both frontend and backend development, implementing new features and ensuring performance. They face challenges in utilizing statistics in development and extracting insights from internally developed tools.	They aim to determine the execution time of specific features like reprocessing on a user's PC to assess their algorithms' performance. They also seek to identify software errors, such as sudden stops in recording by multiple users, to improve software performance.

Product Manager	Product Managers at MVN are responsible for driving growth and success through business strategy, including targeting specific user segments with appropriate products and pricing. They handle new product development, acquisitions, and collaborations, making sure decisions align with the company's vision. They also focus on understanding feature usage and customer feedback, tracking MVN's performance through sales and revenue metrics, and prioritizing customer issues based on business impact.	Product Managers recognize the importance of software usage statistics in decision-making, using them to identify usage patterns and preferences for feature enhancements. They analyze feature usage to inform decisions in the software redesign process, determining which features to add, remove, or automate. They also monitor how new features or updates are received by users, using this data to refine strategies. Monitoring license usage trends helps them understand customer behavior and address potential issues, like declining renewals, to retain customers.
Product Support Specialist	Product Support Specialists manage pre-sales and post-sales support for MVN products, resolving technical issues for customers. They tackle recurring issues via stakeholder discussions and bug reports. They aim to minimize support cases and enhance user experience. They proactively mitigate customer issues by updating tutorials on the Movella website.	They aim to understand if users consistently use a feature and how they engage with the documentation, to carry out relevant maintenance and improvements.

3.4.2 Personas

Based on the interview result, I developed Personas for each role, as detailed in the Appendix A.3. These Personas present a comprehensive profile that helped me emphasize the distinct roles and guide better design decisions. They simplify the understanding of complex roles, aligning their needs with the insights derived from software usage data. Additionally, Personas provide a shared basis for communication, facilitating the explanation of design needs and rationale to different stakeholders with ease [35].

Figure 3.1: The persona for UX designer

An example of a persona is shown in Figure A.1. The top section contains basic information about the UX designer to make the persona more realistic and specific. This is followed by a statement summarizing the role's mission. I used labels to describe the role's responsibilities, highlighting this key information for easy reading and emphasis. For those interested in more details, the following sections cover the role's goals, motivations, and frustrations. The "ask the statistics" section outlines the questions this role would pose to the software data, aiding in achieving their goals. The 'System' section indicates the platforms most frequently used by this role, providing a reference for choosing platforms for future dashboard development.

Personas have been useful in the dashboard design process, offering crucial user perspectives. They have aided in identifying needed features and data in the dashboard, leading to the creation of more user-centric solutions.

3.4.3 Affinity diagram for analysing the interview results

During the process of organizing my interview notes, I noticed that the respondents' answers regarding their thoughts and expectations on using software data were quite varied. To systematically arrange the responses on this topic, I created an Affinity Diagram (Appendix A.4) to sort the interview data. An affinity diagram is a tool used to organize and categorize large amounts of data into groups based on their relationships [36]. The primary purpose of this method is to help me identify patterns and themes in the varied interview answers collected, making it easier to analyze and derive insights. Unlike before, where data was grouped by each role, this time I grouped data that spans across roles into three main themes, including data of interest, participants ideas for features and interface, and the things they want me to keep in mind when I am designing and implementing the data dashboard.

Table 3.3 presents an overview of an affinity diagram for quick reading. First theme is a list of all the data and visualizations that participants mentioned they were interested in and wanted to have access to. In terms of data and visualization preferences, three or more roles commonly expressed interest in the following data and visualizations:

- How many times each feature is used.
- User count for each feature.
- Session count of the MVN software.
- User count for the MVN software.
- User count for each license.

Data of Interest	Insights in Data Interaction and Visualization	Things to Keep in Mind
• User flow and behavior from statistics • Feature usage, e.g., body dimension fill out • Time interval between specific events • Changing trends of time intervals • Frequency of events/features used • Feature usage per session and per unique user • User's operating system, language, version, and license • Duration of certain features • Mouse movement of users • Sold and active software licenses • Usage duration for recording, reprocessing, and exporting • Upgrade rates after new releases • License usage over time and its correlation with revenue • Tracking feature quality, like calibration	• Filter to differentiate experienced and first-time users • Automatic report generation • Identification of unusual data patterns • Regular updates of data and graphs • Inclusion of both raw data and abstract information • Feature/data selection filters • Grouping events according to user flow • Visually appealing data representation • Simplified interpretation of statistics • Machine learning integration for problem identification • Dashboard viewing options like Tableau • Correlation analysis between feedback and statistics • ChatGPT-like features for data querying • Immediate web access to statistics • Abstracting data with filters, algorithms, or functions	• Removal of biased data, e.g., two-second recordings • Strategies for updating new events • General applicability of the solution • Inclusion of internal test data • Accuracy of timestamps in logfiles • Difficulty in estimating user experience • Documentation and naming consistency of events • Prioritization of statistics and exclusion of internal tests • Maintenance of statistics for new features • Limitations in indicating problem reasons • Insight extraction and comparison with customer complaints • Exclusion of internal MVN versions

Table 3.3: Affinity Diagram for analysing the interview

- User count for each version.
- Sequence of feature triggered when users use the MVN software.
- How long users spend on feature such as recording, reprocessing, and HDreprocessing.

Furthermore, distinct roles have specific data interests, as explained in Table 3.2.

Second theme presents participants' thoughts and expectations on interacting with the software data. Half of the respondents mentioned their preference for a data dashboard presentation. Such a format, being more visual, is easier to interpret compared to static text. They expressed the desire to access all raw data and have various filters to abstract the data, like a license type filter. Three respondents mentioned their interest in AI features, such as algorithms that can automatically identify problems from data and a search engine similar to ChatGPT, where users can simply input prompts and receive corresponding data or graphs. Two respondents highlighted the need for data and graphs to be updated regularly, be it daily, weekly, or even monthly.

The third theme highlighted considerations that participants wanted me to bear in mind when working with software data. Some respondents mentioned the need to remove biased data, such as recordings that lasted only two seconds. One participant expressed a desire for a more universally applicable approach to use software statistics, allowing other software and products at Movella to leverage the data similarly. Additionally, one participant emphasized the importance of distinguishing whether the software data generated from internal testing or actual users.

3.5 Conclusion

This chapter has presented the objectives, participants, questions, and processes of user interviews, as well as an analysis of the results. The primary purpose of the user interviews was to understand the preferences, motivations, and objectives of those who wish to utilize software usage data, with the goal of identifying the data that needs to be displayed and visualized on the dashboard. To achieve this, I employed a method based on the GQM model [37], which facilitated the analysis from user goals to the determination of necessary data. The analysis is displayed in Table 3.4.Through the interviews, I compiled the goals of each role looking to leverage software usage data, developed a set of questions to outline how software usage data can assist them in achieving their goals, and outlined a set of metrics to quantitatively answer every question. After thorough analysis, I have identified the

specific data that needs to be showcased and visualized on the dashboard, laying the groundwork for subsequent prototype design work.

Table 3.4: GQM Analysis of Interview

Role	**Goal**	**Question**	**Metric**
UX Designer	• Gain a better understanding of user behavior and preferences to inform design decisions. • Optimize the overall user experience. • Design interfaces that are user-friendly and allow users to complete tasks correctly.	• What is the common order of events during use of the application? • How many times is a certain feature used for each license and user group? • What are time intervals between two specific events and its trends?	• Track the sequence of user interaction with MVN software, logging the order of button clicks, menu selections and so on. • Calculate the number of usage for each feature, categorized by license type and user group. • Calculate the time elapsed between two predefined events.
Quality Assurance Engineer	• Ensure that hardware and software can be operated by users smoothly. • Be able to identify areas of the software and hardware that need to be tested and improved. • Efficiently plan and prioritize the testing.	• What are the operation system and language that users are using? • Which version and license are being using? • How extensive is a feature being used?	• Calculate the number of uses for each OS, language, version, and license. • Calculate the frequency of feature usage across all users.
Software Developer	• Develop software applications and algorithm that enable users to perform tasks effectively and efficiently. • Maintain and optimize the software performance by testing and bug fixing.	• How long it takes for a specific feature on the user's PC, such as reprocessing? • What is the number of users between versions? • What is the trends of the number of feature used?	• Measure the execution time for the reprocessing feature. • Calculate the number of users on each version of the software. • Display the trends in feature usage.

Table 3.4: GQM Analysis of Interview

Role	**Goal**	**Question**	**Metric**
Product Support Specialist	• Help customers get a comprehensive understanding of the product. • Help customers use the products effortlessly in their workflow. • Improve overall customer experience and lower the number of support cases.	• If a feature is used once, will it be used again? • What is the workflow when the user is looking for help documentations?	• Display the trends in feature usage. • Map out and time the steps users take to access help documentation in MVN software.
Product Manager	• Understand user behavior and product performance to make business strategies. • Monitor product performance to make product decisions. • Identify product issues and evaluate their importance, costs, and potential profits to decide which issue to tackle.	• How many active users are there and what are the trends over time? • Which license is used the most? • How many users upgrade to the latest version after a new release? • How does the last strategic decision impact the product usage?	• Calculate the number of uses for each license. • Calculate the number of users for the latest version of MVN software. • Calculate the number of uses and users for the new feature. • Compare the number of users and sessions before and after the last strategic decision.

Chapter 4

Lo-fi prototyping

In the last chapter, I have collected motivations, data interests, preferences, and recommendations regarding software data usage from target users through user interviews. The findings from the user research have further shaped the purpose of the data dashboard: to present visualizations of data that users want to see and to organize these visualizations in a manner that makes them easily accessible to users. As previously discussed, I followed a user-centered design approach and moved forward to the ideation process for designing the data dashboard.

4.1 Ideation

Drawing on the insights gained from extensive user interviews, I have compiled a list of functional requirements for the data dashboard. The source of these requirements stems from in-depth conversations with users, where I gathered information on the data they find most interesting and the features they wish to see in the dashboard, as illustrated in Table 3.4. For instance, both the Quality Engineer and Product mentioned they wanted to view the usage of licenses, which directly influenced the inclusion of presenting user count for each license in our requirements list. These user needs have been transformed into a comprehensive list, encompassing the types of data to be displayed, various visualization aspects, and interactive features of the dashboard.

1. Display each feature's usage.
2. Display the user count for each feature.
3. Present session count statistics for the MVN software.
4. Present the overall user count for the MVN software.

5. Present user count for each version, OS, language and license of the software.
6. Visualize the sequence in which features are triggered during software usage.
7. Display the duration users spend on specific features like recording, reprocessing, and HD reprocessing.
8. Ensure various filters to abstract the data, including a license type filter, Yearly and monthly filter and so on.
9. Incorporate AI features for automatic problem identification and a prompt search engine similar to ChatGPT.
10. Allow for regular updates of data and graphs, with options for daily, weekly, or monthly updates.

The lo-fi prototype serves as an initial representation of the solution, emphasizing functionality, layout, and the flow of user interactions. It serves as a minimally viable solution using relatively quick and simple resources and development time. It serves to validate that users can navigate and comprehend the data presented with ease.

This prototype follows a web-based format, a decision arrived at after discussions with the case company. Adopting a web format ensures that the data dashboard can be easily accessed and viewed. The design was created on the Figma platform, a predominant design tool at Movella. This choice ensures consistency in design software and fosters smoother team collaboration and reviews. Additionally, Movella's design system, encompassing frequently used web design elements like logos, colors, and fonts, resides on Figma. The current prototype design incorporates these design elements to maintain consistency with other designs within the company. Furthermore, this is a clickable prototype, allowing users to navigate through pages, apply filters, and perform other foundational functions.

4.2 Structure

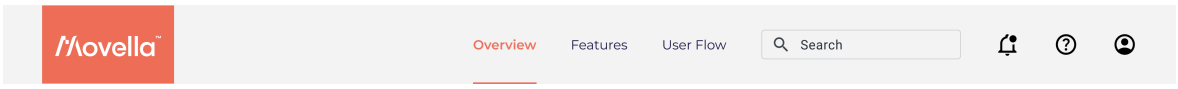

Figure 4.1: Navigation bar in the low-fidelity prototype

The data types and visualizations intended for display on the data dashboard are diverse and extensive. I divided the data to be presented on the dashboard into three categories: overview, feature, and user flow. Each data displayed on a different

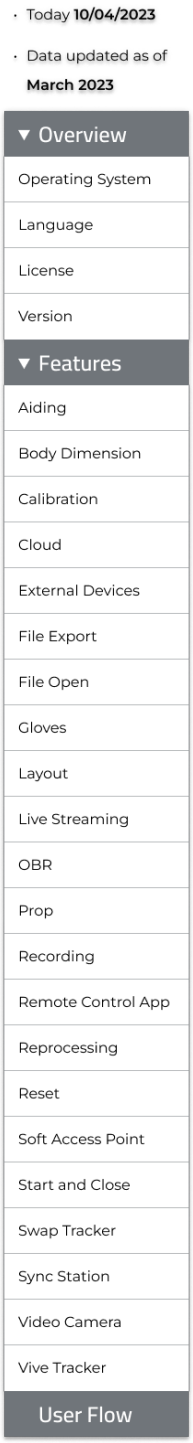

Figure 4.2: Navigation panel in the low-fidelity prototype

page of the dashboard. The structure of data can be seen in the navigation panel 4.2.

The 'Overview' page displays user and session counts, along with OS, language, license, and version usage metrics, offering a quick glance at the software's usage across various dimensions. It allows users to understand the overall software usage without needing to delve into complex details. The 'Feature' page shows event-related data, with events being predefined actions in the MVN software. During design, the 'Feature' page's sub-pages were structured for easy location of specific event data. Events are categorized by MVN's main functionalities, as illustrated in Figure 4.2. For example, 'calibration' events like 'calibrationresult' and 'calibrationstarted' are grouped under the 'calibration' sub-page, as decided with Movella's product managers and UX designers. The 'User Flow' page, unlike the others, illustrates a typical sequence of feature usage in MVN software, from calibration to recording, reprocessing, and exporting.

The categorization was based on two key considerations. First, data relevance and user perception were crucial, as suggested by Dastani [38], who emphasized that user-aligned data categorization enhances dashboard presentation. Insights from user interviews and Movella's data collection methods led to analyzing data set relationships and user categorization perceptions. Movella's data log file is split into user behavior data (such as language, license, version, and date) and event data (specific features triggered in the MVN software).

The second rationale for this categorization was visualization and filter types. On the 'Overview' page, same type of data charts were used for data on this page. The 'Feature' page mainly uses line charts for event-related visualizations, comparing event trigger frequencies. The 'User Flow' page, in contrast, presents more complex data abstractions with intricate diagrammatic visualizations, differing from the standard formats like line charts, tables, and pie charts used on the first two pages.

One approach to managing vast and multifaceted data is to categorize and divide it across separate pages, allowing users to access different pages and data sets through navigation [1]. Moreover, attention is paid to design a useful navigation system to effectively organize data and charts, helping users swiftly locate their desired data.

I implemented two navigation methods: a bar (Figure 4.1) and a panel (Figure 4.2). Both enhance user experience and data access efficiency. The bar provides an overview and quick access to main sections [39], while the panel offers detailed data exploration. This dual approach accommodates different user needs for effective dashboard navigation. Users can also search for data using keywords in the bar's

search box.

Overall, this structure simplifies data access, allowing users to easily locate specific information. The navigation bar enables quick page transitions, and the panel allows direct access to detailed data within categories.

4.3 Prototype

4.3.1 Overview page

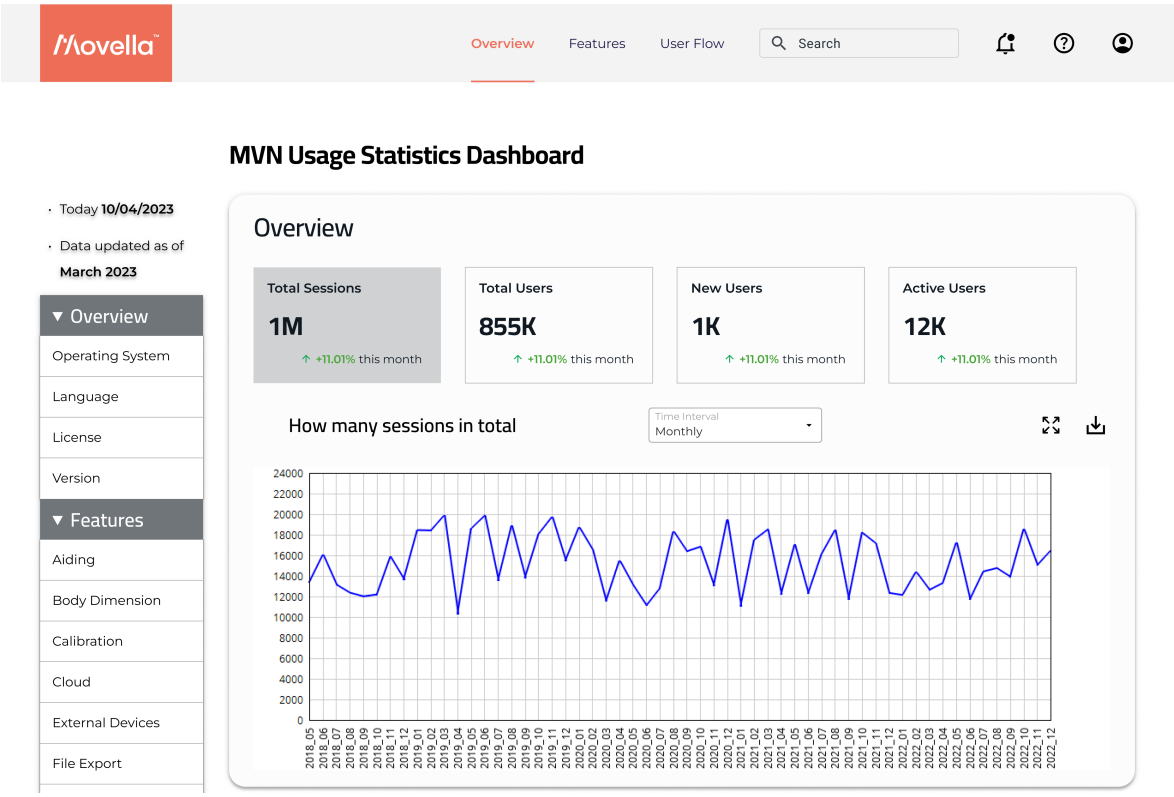

Figure 4.3: The upper section of the overview page

Figure 4.3 shows the upper section of the overview page. This section showcases the overview panel, where users can swiftly scan key data related to MVN software usage, including the number of times the MVN software was used, the number of users, the number of new users and the number of active users. The data is presented using both tabs and line charts. Initially, these four data points are displayed across four tabs. Each tab provides the current month's value for the data and its growth rate compared to the previous month. By selecting a data tab, a corresponding chart will appear below, illustrating the trend of that data over time. Note that in the lo-fi prototype, all data and charts are not real information for placeholder and illustrative purposes.

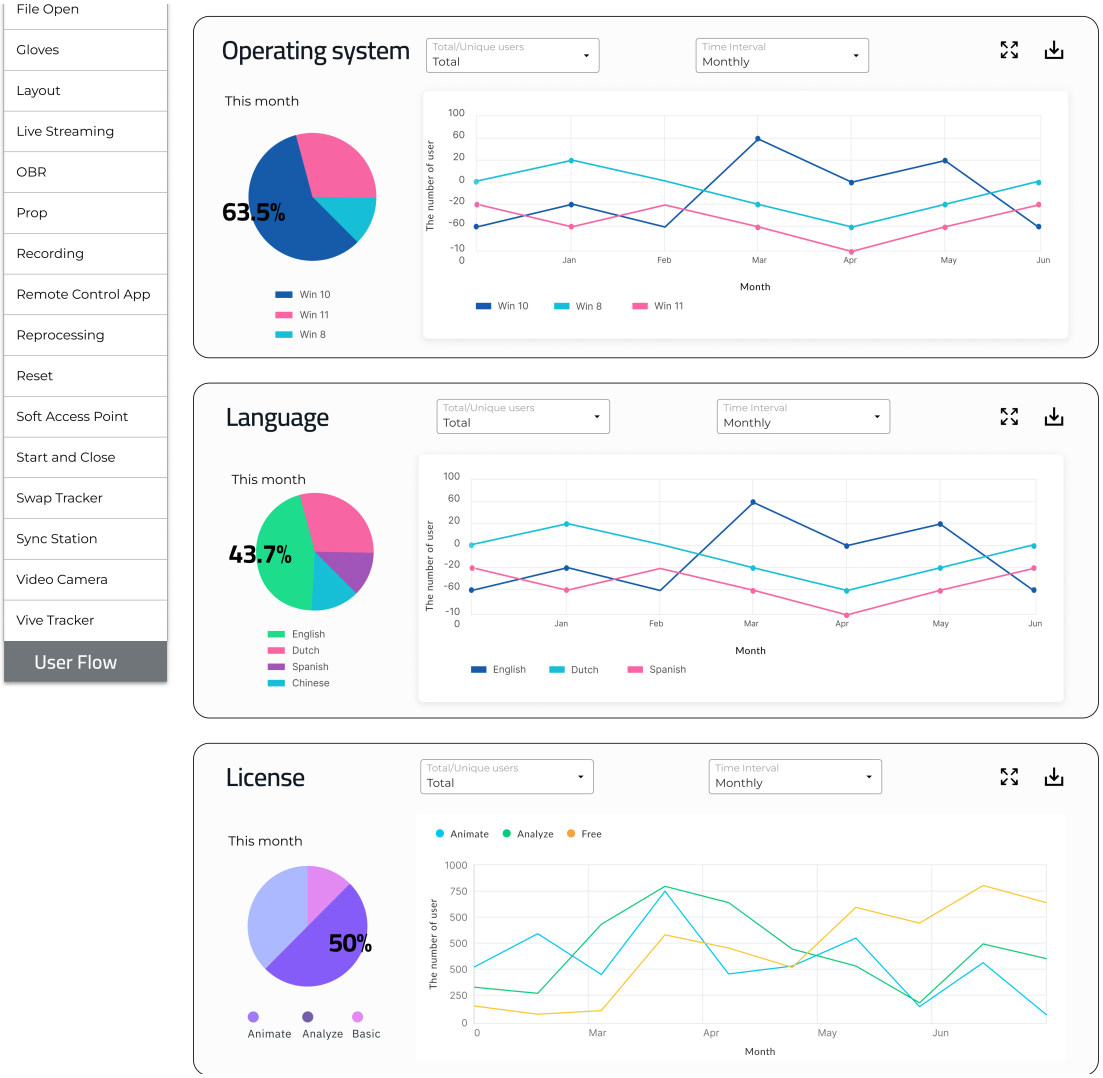

Figure 4.4: The mid-section of the overview page

Figure 4.4 illustrates the middle section of the overview page. As users scroll further down the page, they can view data related to the usage of the operating system, language, and license. Given that each operating system, language, and license type has a finite set of variations, and noting the feedback from our target users regarding the importance of understanding the proportional representation of each category, I have incorporated not only line charts depicting data trends over time but also pie charts to highlight the most dominant categories in each segment.

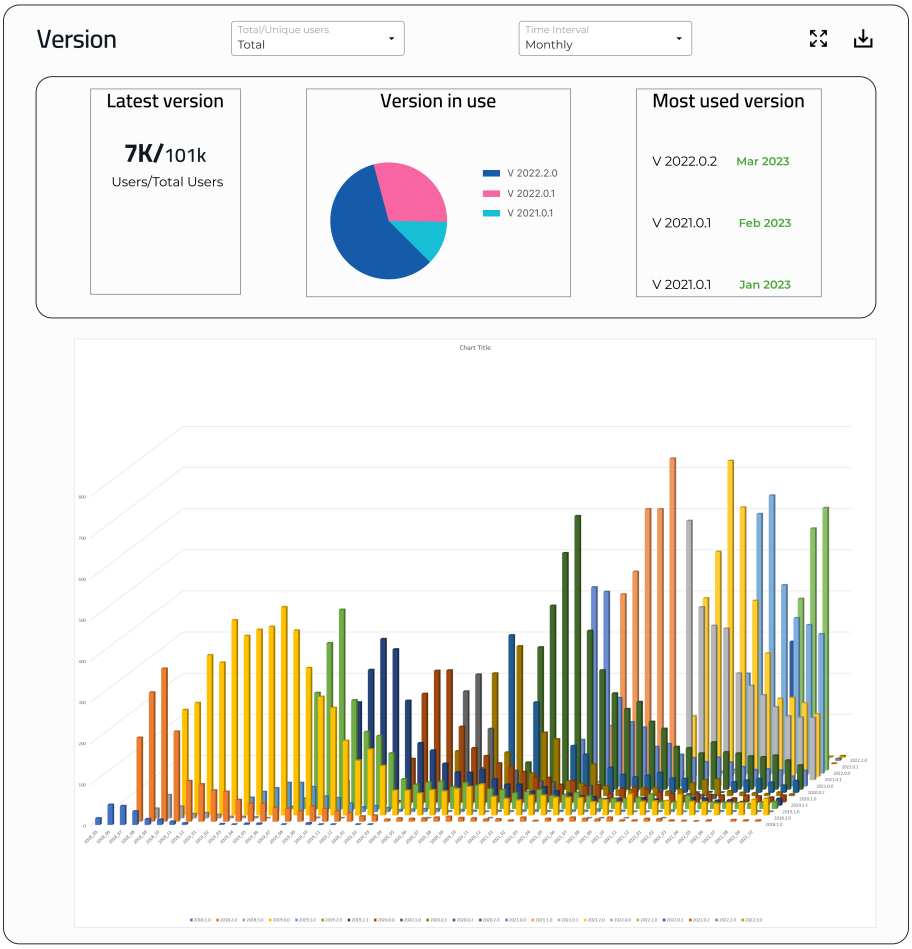

Figure 4.5: The bottom section of the overview page

Figure 4.5 shows the Overview page's bottom section, featuring diverse data visualizations for user needs. Product managers can track user transition to new versions, assessing business decision impacts. Quality engineers see which versions are actively used, aiding maintenance and testing plans. The multitude of software versions versus license quantities makes a 2D line graph representation challenging. A 3D bar chart is thus employed to display each version's usage count. The X-axis marks monthly timestamps from 2018 to 2022, the Z-axis categorizes MVN software

versions, and the Y-axis shows usage quantity. This chart helps identify trends and compare different version usages over time.

In alignment with the requirements listed earlier, this Overview page successfully delivers on providing session count statistics for the MVN software, presenting the overall user count, and detailing user counts across different versions, operating systems, languages, and licenses.

4.3.2 Feature page

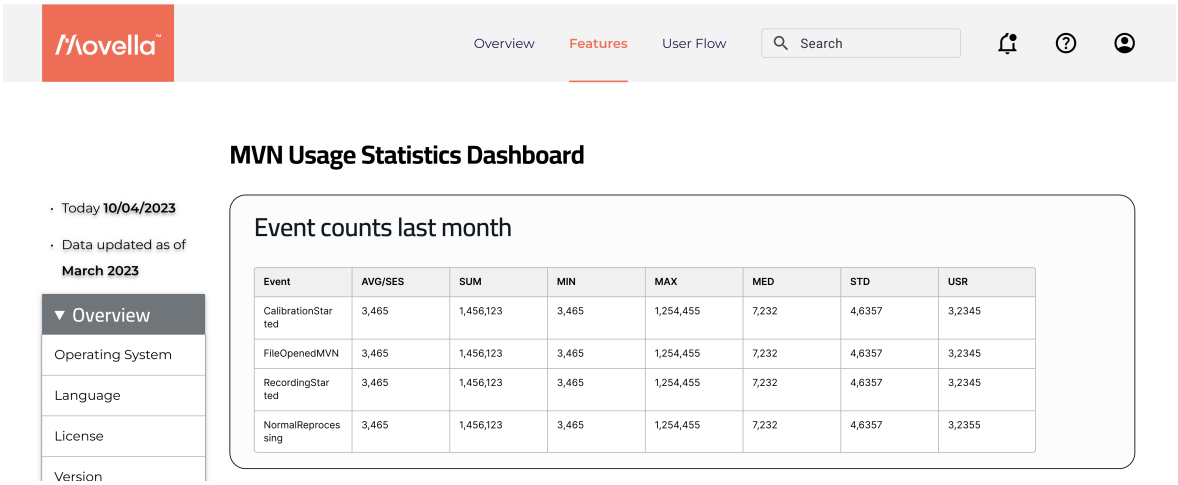

Figure 4.6: The upper section of the feature page

The Feature page displays event-related data. At the top, as Figure 4.6 shows, there's a table for events and their parameters, predefined by Movella's product managers and developers. These parameters, updated with new MVN software features and analytical needs, include metrics like event trigger counts, average, minimum, and maximum per session. These help users analyze event trigger patterns. For example, a quality tester might find a spike in event count, examining the maximum count in a session and user numbers. This analysis determines if the increase is due to frequent triggers by a single user or more widespread use. With over 100 events, the scrollable table provides comprehensive data access and raw data for reference.

The bottom section of the feature page, shown in Figure 4.7, organizes data visualizations over a hundred predefined events. Events in the MVN software are grouped by features in the left-side navigation panel, where a 'feature' is a specific MVN software functionality. For example, calibration-related events like calibrationstarted and calibrationresults are grouped together. This categorization, decided with Movella's product managers and UX designers, helps users find specific event data charts or

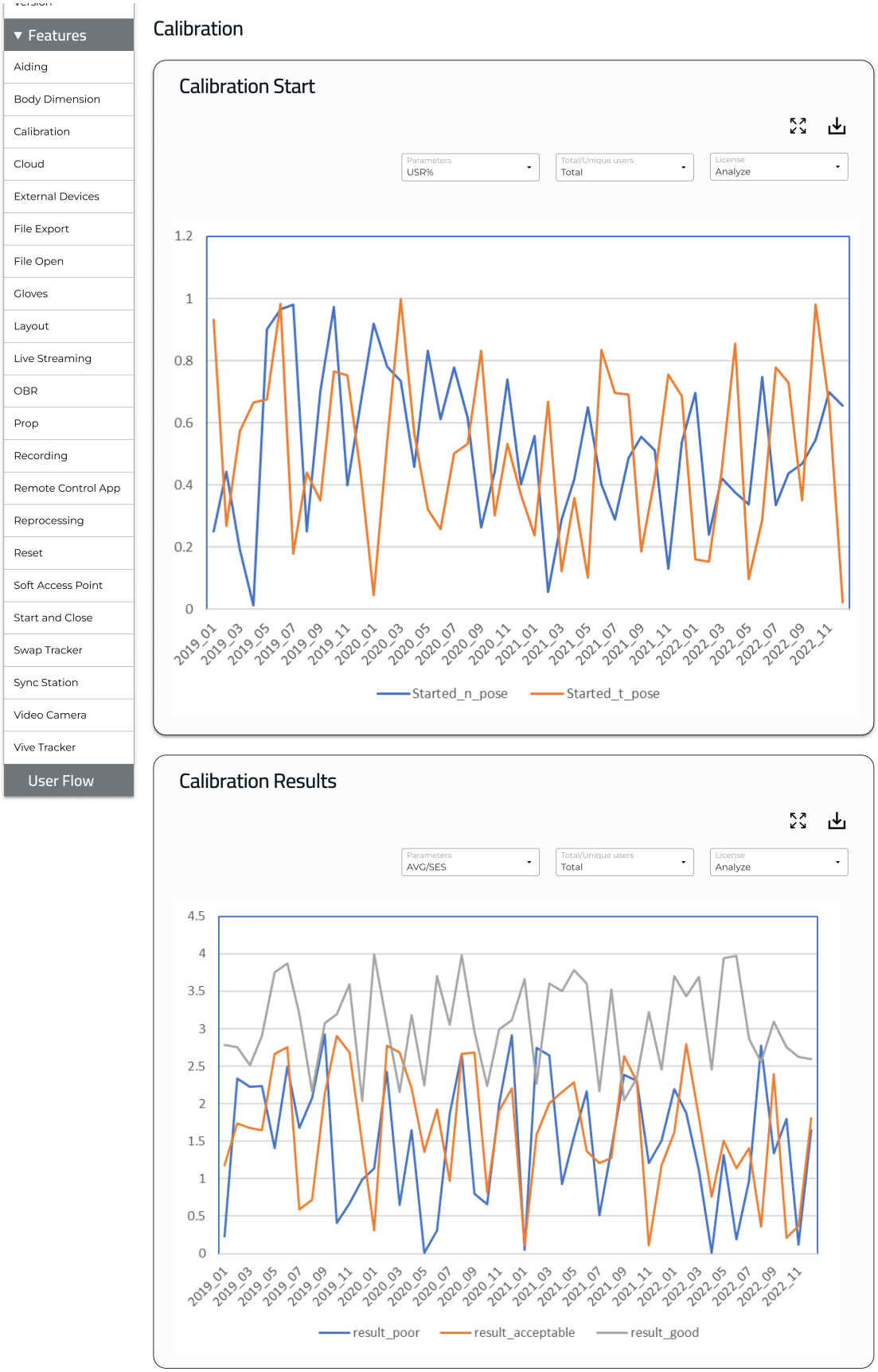

Figure 4.7: The bottom section of the feature page

compare several events based on the MVN software function. For instance, product managers interested in calibration quality can click the calibration tag in the navigation panel to see a line chart of calibration results, including events like calibrationresultgood, calibrationresultpoor, and calibrationresultacceptable. This chart tracks event counts over time, offering a detailed view of calibration-related events.

In short, on the Feature page, the usage and user count for each feature are prominently displayed, addressing the first two points from the requirement list. The page provides a comprehensive 'event counts' table at the top and categorizes event-related visualizations based on specific MVN software functionalities. This structured presentation ensures users can easily navigate and understand event data.

4.3.3 User flow page

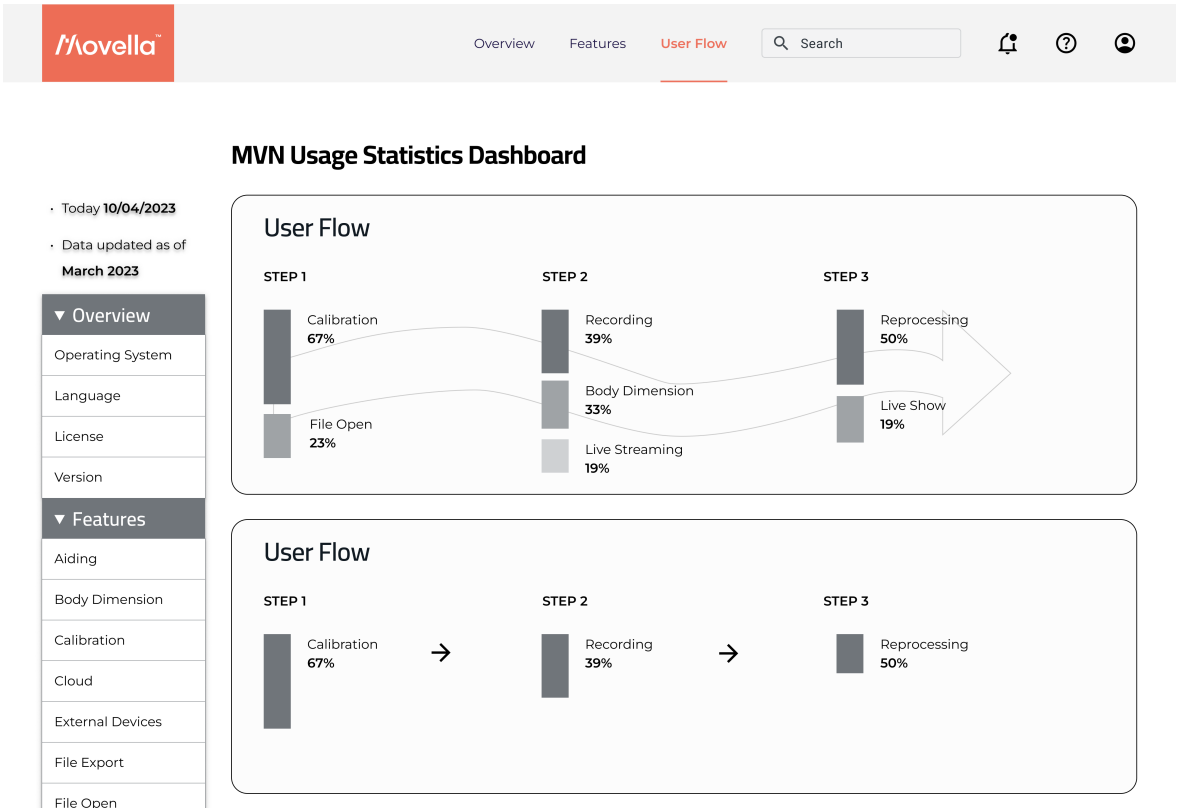

Figure 4.8: User flow page

Figure 4.8 illustrates the 'User Flow' page. For professionals, understanding the sequence in which users engage with features offers intriguing insights into user behavior. Compared to previous visualizations that utilized conventional chart types, illustrating the common user flow posed a challenge, as there wasn't a readily available data chart template to lean on. Consequently, I tried to design a new visualization type, aiming to represent the predominant patterns of feature usage within the

MVN software in a way that's both enlightening and user-friendly. It's informative to discern which features are initially accessed by users, and which set of features are often used in a specific sequence by a substantial number of users. As shown in the figure, the visual representation is ordered by the sequence of user actions. For the first step, 67% of users opt for the 'calibration' feature, and for the second step, 39% of users choose to start 'recording'.

4.4 Conclusion

This chapter outlines the development of a low-fidelity prototype for the data dashboard, addressing most requirements listed in Section 4.1, except for points 7, 9, and 10. The design includes a data table for feature usage, visualizations for user and session counts, and metrics for software versions, operating systems, languages, and licenses. It also visualizes user flow and feature activation sequences, with filters like time and license for data abstraction.

However, the prototype doesn't show the duration spent on specific features due to uncertainties in visualization and complexities in low-fidelity designs. AI integration for problem identification and regular updates, though requested, were not included in this early stage due to their complexity and implementation challenges.

The design focuses on accessibility and user experience, featuring a navigation bar and panel for easy access to overview, feature, and user flow categories, ensuring data is easy to locate and understand. Moving forward, the next chapter will delve into usability testing to evaluate and refine these design choices.

Chapter 5

Evaluation

5.1 Objectives

The objective of the usability testing is multifaceted, aiming to delve deep into the user experience of the prototype. Firstly, I intend to assess the layout and design, and to determine whether design is sufficient for our target users in accomplishing their tasks and locating the data they are interested in.

Secondly, I aim to evaluate the appropriateness of the chosen chart types and the methods employed to abstract and visualize the data. The overarching goal here is to ascertain if these visual representations can depict MVN statistics in a manner that is not only visually appealing but also easily comprehensible and accessible to users. It's essential that these visualizations not only present data but also enable users to derive meaningful insights efficiently.

Furthermore, I want to identify any usability challenges that users might encounter while navigating through the prototype. By pinpointing areas of pain, confusion, or any other obstacles, I can gather invaluable feedback that will be instrumental in guiding subsequent development phases and refining the prototype.

Lastly, I planned to collect the users' overall satisfaction with the prototype. I want to understand if the prototype aligns with their expectations and if they find it valuable in their workflow. This would help ensure that the final design is both effective and user-friendly.

5.2 Participants

Due to tight schedules and limited availability of participants, a total of four internal employees participated in this usability testing session. These participants were

selected to represent different departments within the company, with the following roles: Quality Assurance, UX Designer, Software Developer, and Product Manager. Their diverse roles within the organization are expected to bring varied perspectives to the evaluation, offering a well-rounded critique of the dashboard's functionality and design. To cater to the distinct motivations and job functions of each role, I designed unique usability scenarios and assigned varying tasks. I asked participants to engage with the lo-fi prototype I designed to accomplish these specific tasks. The scenarios and tasks used for each role during the usability testing can be found in the Appendix B.1.

5.3 Procedure

In the usability testing, I began by reading a welcome text to the participants. This introduction served to familiarize them with the purpose of the usability test. I also sought their permission to screen-record the session and encouraged them to speak their thoughts loudly throughout the test, a method known as ”think aloud.”

Once I received their consent, I initiated both screen recordings. I then provided the participants with instructions on the scenarios and tasks they were to perform, using a split-screen view with Figma. They were directed to engage with the prototype in Figma and carry out the tasks as specified.

As the participants interacted with the prototype, I took note of their verbal comments, my observations, and the status of task completion. I utilized a usability test observation form, which allowed me to systematically capture observational notes, verbal comments from participants, data related to task completion, and any issues or challenges they faced during the interaction.

Upon completion of the tasks, I conducted a brief interview with each participant. This post-test discussion aimed to gather their overall satisfaction levels, any issues they identified, and their insights regarding the prototype. According to different working content, different interview questions were designed, which can be accessed in Appendix B.1.

5.4 Results

5.4.1 Overall Findings

Table 5.1 shows task completion rates and times for each role, with most tasks successfully completed. The UX designer's uncertainty about 'license' impacted the

Table 5.1: Task Completion by Role

Role	Task Completion Success Rate	Total Completion Time
Quality Engineer	100%	30 mins
UX Designer	67%	20 mins
Software Developer	100%	8 mins
Product Manager	100%	18 mins

task completion. Participants provided valuable feedback during usability testing, with durations varying due to familiarity with MVN software data. The quality engineer and product manager, more interested in MVN data, thoroughly explored the prototype, offering insights. The UX designer, less familiar with MVN software, required more explanations.

Participants generally had a good impression on the prototype's design and layout, particularly appreciating the dual navigation options of the bar and panel for quicker data location and understanding. They were mostly satisfied with the data graphs and visualizations. Two participants favored the language and license charts, like the pie chart for quick language usage identification, crucial for testing plans. However, they found the 3D version usage graph confusing, suggesting a 2D graph for clarity.

Some issues arose during testing. Many found the user flow page's visualization complex, and there was confusion over terms like "session" and "license." One participant struggled to find the calibration visualization due to unfamiliar page categorization.

Overall, the participants had a positive impression of the lo-fi prototype and were encouraging about the progress made.

5.4.2 Results analysis

Regarding the overall design layout and data categorization, participants did not suggest any big changes. For all data charts, they recommended refining titles, subtitles, and labels. For instance, in the 3D chart displaying version usage, the lack of labels on the axes makes it difficult for people to understand the data graph. When it comes to data visualization, they expressed a preference for 2D charts over 3D ones. For the dashboard, participants expressed a desire for a feature that allows the generation and export of customizable reports. Additionally, their feedback and points of confusion were mainly directed at specific data charts, titles, labels, and so on.

On the overview page, two participants suggested dividing active users into daily

active users and monthly active users. Three users mentioned their wish to set a specific time range for the charts, from a particular start date to an end date. One participant noted that the term ”Total Runs” was ambiguous. Regarding the data visualization, two participants mentioned the possibility of reducing the chart size but increasing the font size for better readability.

On the feature page, two participants suggested using a pie chart to present calibration results, as it would offer a more intuitive comparison of each calibration result's proportion. Additionally, one participant expressed a desire for the dashboard to have a feature that calculates the correlation between two specific data sets to identify the root cause of any anomalies. For instance, if only Spanish users are using gloves, it would be beneficial to test the gloves specifically in Spanish.

On the user flow page, most participants indicated that it was their first time encountering such a chart, and they needed additional explanations to understand the visualization. They expressed interest in the content of the user flow and hoped for its continued refinement.

In the feedback, participants indicated acceptance of the design layout and data categorization. For various pages and charts, they provided valuable feedback and recommendations for adjustments. Additionally, they proposed advanced features such as report generation and correlation analysis.

5.5 Conclusion

In summary, all participants gave a positive evaluation of the overall design. They provided suggestions on how to improve the interface design. The issues encountered by participants during testing, as well as their proposed improvements, are detailed in Table 5.2. Overall, the design required no major changes, but some minor aspects, such as titles, labels, and choices of visualizations, needed refinement. Unfortunately, the number of participants in the usability testing was limited due to time constraints and respondent availability. The feedback and opinions collected will be incorporated into the next phase of dashboard development.

Issue	Improvement
The term ”Total Run” in the overview panel is not clear.	Change the term ”Total Runs” to ”Total Sessions”, as shown in Figure 5.1.
The time range of ”Total Runs” in the overview panel is not clear.	Specify the time range of ”Total Runs” to be monthly, as shown in Figure 5.2.
Have difficulty in understanding the “Most used version” panel, which is not clear and would be read the other way around.	Revise the title to ’Most Used Version Each Month’, and switch the columns of month and version to enhance readability and reduce the possibility of misinterpretation, as shown in Figure 5.3.
The percentage of users who have upgraded to the latest version is not clear enough.	Show the percentage of users who have updated to the latest version and change the title, as shown in Figure 5.4.
Unable to set the time range of the data charts.	Design a filter that allows users to set the start date and end date for the data charts.
The data chart displaying version information is a 3D bar graph, which is difficult to understand.	Replace the 3D bar graph with a 2D bar chart to show the usage of different versions.
Some data charts lack captions and labels for the x-axis and y-axis.	Add captions and labels to the x-axis and y-axis in the data charts.
The visualization on the User Flow page is difficult to understand.	Consider adding labels to the steps on the User Flow page and displaying more information when the mouse hovers over the text.
Some data charts were too large and got cropped, making it difficult to view them in their entirety.	Modify the prototype to be screen-fit and created multi-faced pages to prevent data charts from being cropped and eliminates the need to scroll to view the entire chart.

Table 5.2: Improvement list of usability testing

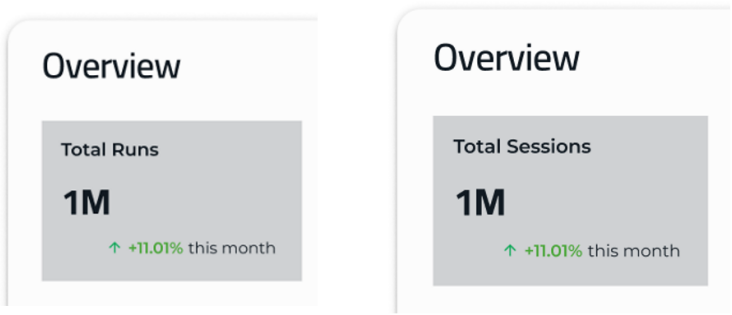

Figure 5.1: 'Total Run' changed to 'Total Session'.Left image: before the change. Right image: after the change.

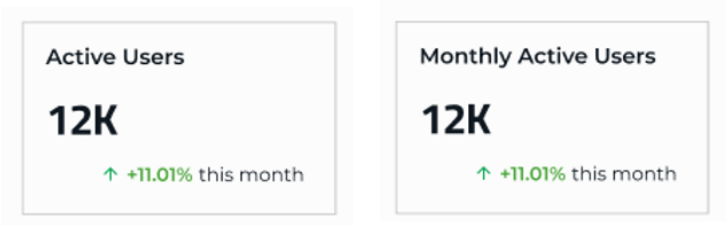

Figure 5.2: Specify the time range of "Total Runs" to be monthly. Left image: before the change. Right image: after the change.

Figure 5.3: Improve the readability of "Most used version" card. Left image: before the change. Right image: after the change.

Figure 5.4: Improve the readability of ”Latest version” card. Left image: before the change. Right image: after the change.

Chapter 6

Proof of concept implementation

6.1 Objectives

One primary objective of the Proof of Concept(PoC) is to explore a feasible development method for the data dashboard. This includes exploring strategies for storing and retrieving software usage data, along with selecting an appropriate platform for data visualization. The intention is to establish this PoC as a foundational model, enabling Movella to further refine and tailor the data dashboard based on their evolving usage needs and objectives. Another key goal is to present an functional data dashboard. I've found that using prototyping software like Figma for dashboard design and creating visualizations is not very efficient. Utilizing a dedicated visualization platform could likely lead to a more effective and faster dashboard design process.

In summary, the PoC is designed to validate the feasibility of creating a data dashboard that utilizes actual software usage data. This encompasses exploring a viable implementation method for the data dashboard and providing a functional representation of the data dashboard.

6.2 Approach

Every instance of MVN software use generates software usage data, which is recorded in log files. In other words, the current storage of software usage data is in text format across numerous log files. This storage method makes batch data management and data retrieval both difficult and inefficient. For example, to analyze the versions of software currently in use, I have to extract version information from each log file and store this extracted data for further statistical analysis.

I consulted Movella's product owner and developers, they argued that adopting a

Figure 6.1: The approach of PoC

database for storing software usage data and connecting it to a visualization platform is a practical approach. This change offers a more organized and structured way of storing data. During user interviews, the Product Manager also mentioned that similar visualization methods are being used for managing product's data and market-related data in Movella. One typical visualization pipeline introduced by Qin et al. involves importing data sources, cleaning and standardizing the data, manipulating and selecting the data for visualization, and finally visualizing the data [40]. After exploration of data visualization methods through literature review and incorporating insights provided by Movella, I have chosen to implement the approach depicted in Figure 6.1 for this PoC.

1. Database Design: The focus here is on creating a flexible, scalable, and maintainable database architecture. The design not only accommodates current needs but is also adaptable to future data growth and changes.
2. Data Preparation: This stage involves extracting data from log files and inserting it into database tables, ensuring accuracy and consistency. I aimed at addressing potential errors and inconsistencies by cleaning and standardizing the data, thus laying a solid foundation for data visualization.
3. Data Visualization: I connected the database to the Qlik visualization platform to create a data dashboard. The dashboard design follows my previous prototype design, achieving a functional data dashboard.

6.3 Database design

Currently, Movella records software usage data in log files. Specifically, every time an individual user uses the MVN software, a log file is generated, thereby capturing data related to that particular session. In this context, a 'session' refers to the period beginning when the MVN software is initiated and ending when it is closed after use. Each log file records predefined data, and includes various fields. For a thorough understanding of these fields, Table 6.1 offers a detailed overview, complete with descriptions and examples.

The fields recorded in log files inform the design of the database, consisting of two data tables. The structure of the database, along with the data fields and types, is

Field Name	Field Description	Example	Length (bits)
sessionid	Sessionid is used as the unique value to identify one log file which is generated from a specific user at a specific time.	mvn_evt_d09bb878-c6d3-4fbd-a06b-454c2d447fe6_1513938067889.log	62
version	It indicates the specific version of MVN used by the user.	2023.0.0	8
build	It refers to the build number, which represents a specific release or iteration of the software.	10035	4-5
revision	It refers to the revision number, indicating a specific revision or update of the software.	120521	5-6
Architecture	It indicates the software's architecture, such as 32-bit or 64-bit, defining its compatibility with system hardware.	64bit	6
userid	Userid is used as the unique value to identify an anonymous user.	37fb92ba-b6c7-4f25-bf5c-4bf003602c5e	28-36
os	It specifies the user's operating system, such as Windows 10, macOS, etc	Windows 10	9-20
license	It refers to the specific type of software license being used.	Animate Pro	4-12
licensekeyconfig	It refers to the configuration of the license key used.	haspsl-usermode	4-15
licensekeymodel	It indicates the model or type of the license key. In this instance, it is described as a "Certificate" model, suggesting that the license is based on a digital certificate that verifies the authenticity and validity of the license.	Certificate	4-11
licensekeytype	It indicates the type of license key, like 'HASP-SL', defining the licensing technology and method	HASP-SL	4-7
date	It records the creation date of the logfile, in YYYY-MM-DD format.	2023-04-17	10
language	It indicates the user's selected language, such as 'en' for English.	en	2
event_name	It refers to the name of the triggered event. An even is something triggered within of MVN, like user interactions, system messages and any other notable activity.	MVNstarted	9-40
timestamp	It records the corresponding timestamp (epoch format) when the event is triggered.	1687979815190	16

Table 6.1: Database Field Specifications and Descriptions

illustrated in the Figure 6.2. There are two data tables: the ”session” table and the ”event” table.The ”session” data table stores fundamental information about user activities in a session, including anonymous user id, software version, operating system, license, date, and language. The ”event” data table encompasses all the event names triggered during the session, along with their corresponding timestamps. The unique identifier for each session, represented as ”sessionID,” serves as both the primary key within each table and the foreign key that establishes a relational link between the two data tables. Apart from the ”Date” field, which uses the ”date” format, all other fields employ the nvarchar data type, capable of storing variable-length strings.

The design of the database primarily focuses on accessibility, maintainability, and scalability. Firstly, the storage format of the data tables aligns with the fields and sequence recorded in Movella's log files. This alignment ensures that Movella's developers and staff can comprehend the database structure, which is convenient for maintenance and updates.

Secondly, in the ”session” data table, each session contains only one value for data fields such as version, user ID, OS, license, date, and language. Conversely, multiple event names and timestamps can occur within each session. To prevent data duplication, event-related data is stored into a separate ”event” data table. This prevents the repetitive storage of ”session” data for each event name.

Additionally, this database structure aligns with the current update requirements of the MVN software. When new features are introduced in the MVN software, new events will be defined and recorded in the software data. With the increase in predefined events, there is no need to alter the data table structure. Furthermore, if there's a need to include additional data related to user activities, it can be easily incorporated as new fields in the ”session” data table, ensuring ease of expansion and maintenance. This data table design enhances functionality and adapts to software changes.

6.4 Data preparation

The primary goal of data preparation is to ensure that the database is ready for subsequent data visualization tasks. This involves assuring the data's integrity, accuracy, and consistency. In this phase, my approach involved several key steps: Firstly, with the support of the IT department, I established a database and corresponding data tables on the Movella server using SQL server database, adhering to our predefined database design. Secondly, I developed a Python script designed

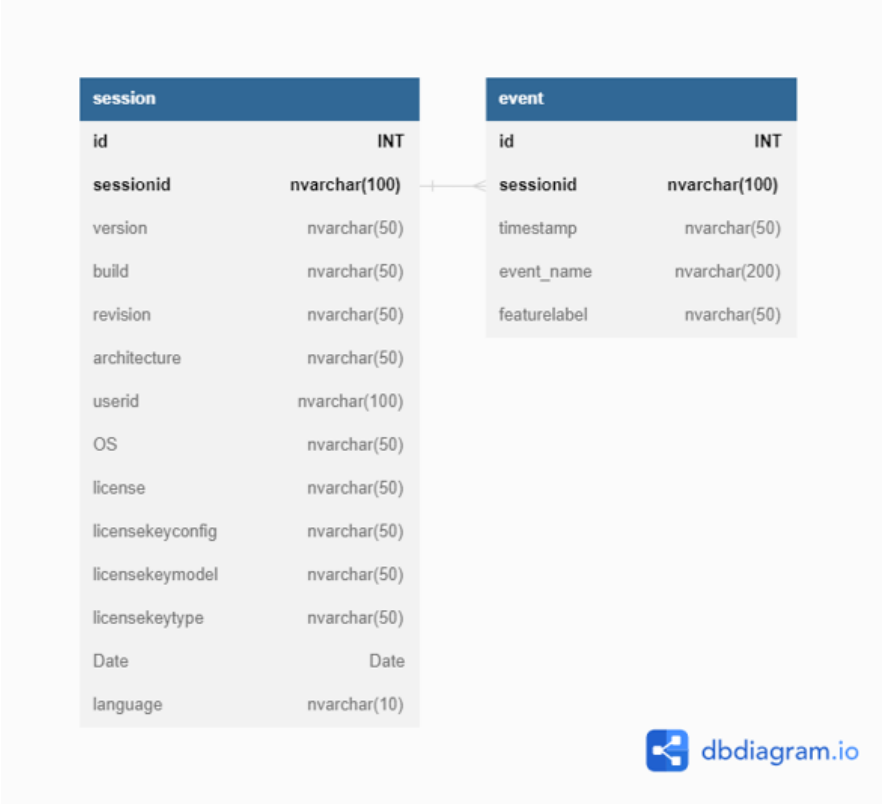

Figure 6.2: Data base structure

to systematically traverse all log files. This script was tasked with extracting relevant data fields through keyword matching. Finally, I ensured the extracted data was accurately inserted into the designated tables.

During this process, I encountered several challenges. The first was related to data integrity. My goal was to insert all data collected since 2018 up to June 2023. However, the volume of data was enormous, with at least tens of thousands of log files generated each month. Additionally, as the Python script ran locally on my computer, it had to first read log files from the Movella server and then insert the extracted data into the database located on the server. This data transfer with the server was quite slow. To address this, I implemented batch data insertion. The script records data extracted from log files in batches and then inserts it in batches (the default batch size is 5000 records). This batch processing approach reduces the frequency of data transfers with the server. Nevertheless, inserting one month's data still took between two to five hours, depending on the number of log files. Therefore, inserting all the data took approximately three weeks.

The second challenge was data accuracy. After inserting the data, I found many records that did not comply with the expected field length, format, or content. For example, although user data collection started in 2018, some records in the date

field showed dates as early as 2003. I lacked the resources and time to identify the causes of these error data. I wrote SQL scripts to filter out obvious error data, moving them to ”session_error” and ”event_error” data tables. The primary filtering criteria were: first, selecting records where the date field was before January 1, 2018, or after July 1, 2023, as these were dates outside the range of MVN data collection by Movella; second, selecting records where the event_name field contained more than three digits, spaces, or characters other than underscores, as many records in this field had symbols like ”& and %” or did not match predefined event names. I filtered these out based on the current naming convention for events (only letters, less than two digits, and no special symbols except underscores); third, selecting duplicate sessionid. Some log files appeared multiple times on the server for unclear reasons. I also moved these records to the ”session_error” data table.

Lastly, there was the challenge of data uniqueness, ensuring that each log file’s data was inserted into the database only once. Initially, the script would check whether the database already contained the same data before inserting new data. However, the data insertion process was already quite time-consuming. This approach further extended the duration of data insertion, leading me to remove this checking method for now. How to ensure data uniqueness while maintaining insertion efficiency remains a challenge.

6.5 Dashboard

After completing the data preparation, I connected the database with the Qlik visualization platform. The choice of Qlik was driven by two main reasons. Firstly, Qlik offers robust visualization capabilities and highly customizable dashboard designs. Secondly, as Qlik is already in use at Movella, opting for it ensured consistency with other products and facilitated a quicker acquisition of the Qlik license.

As described in Chapter 4 and 5, I created a prototype of the data dashboard using Figma and conducted a round of usability testing. When developing the Proof of Concept (PoC) on Qlik, I referred to the earlier low-fidelity (lo-fi) prototype design. This included similar elements such as the navigation bar, filters, data charts, and layout. However, the final design on Qlik did not completely align with the Figma prototype for some reasons. Qlik’s capabilities for interactive design elements are not as robust as Figma’s. While Figma allowed for the creation of an expandable navigation panel, Qlik lacks these detailed design features, making it challenging to replicate such interactive components precisely. As a result, the navigation panel and other design elements from the Figma prototype could not be exactly reproduced in Qlik. Furthermore, the design in Qlik integrated user feedback from the

usability testing of the lo-fi prototype. For instance, as illustrated in Figure 6.3, a filter was added to the top-left corner of the Qlik dashboard, enabling users to select specific time ranges for viewing data charts.

In the following content, I will present the data dashboard that has been developed on the Qlik platform. Figure 6.3 displays the overview page of the data dashboard. On this page, viewers can observe the frequency of software usage and its trends over time, as well as the number of users and their growth pattern. The text on the left highlights the current month's usage frequency and user count, along with the growth rate.

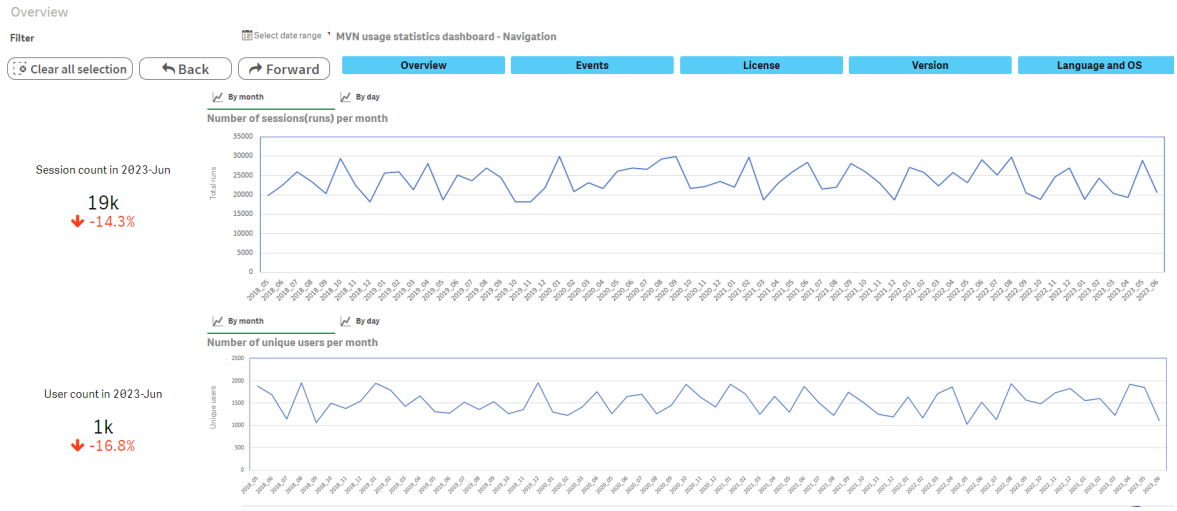

Figure 6.3: Overview page of the data dashboard

Figure 6.4 displays the license page of the data dashboard. On the left side of the page, a pie chart clearly illustrates the proportion of each license's usage. This visual representation allows users to quickly identify the most commonly used licenses. On the right side, a time-based view of license usage is presented, enabling users to observe trends over time and identify which licenses are increasing in popularity and which ones are declining in use.

Figures 6.5 and 6.6 display pages containing data on version, language, and operating system. Their design closely resembles that of the license page.

Figure 6.7 displays the Event Page of the data dashboard. At first glance, one is greeted by a data table presenting a detailed list of events, accompanied by respective parameters such as total count, user count, and average number. Beneath the data table, two visualizations are provided, each corresponding to the data outlined above. One is a bar chart, designed to highlight the variance in the usage frequency of different events. The other is a line chart, offering insights into the trends of event usage over time, illustrating the dynamic nature of user interaction with these

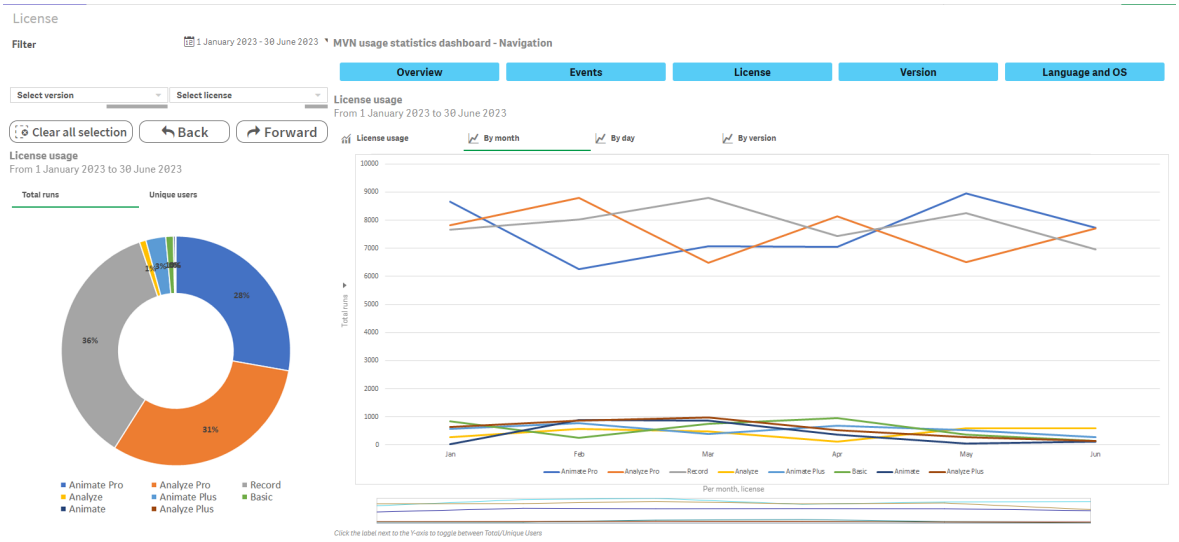

Figure 6.4: License page of the data dashboard

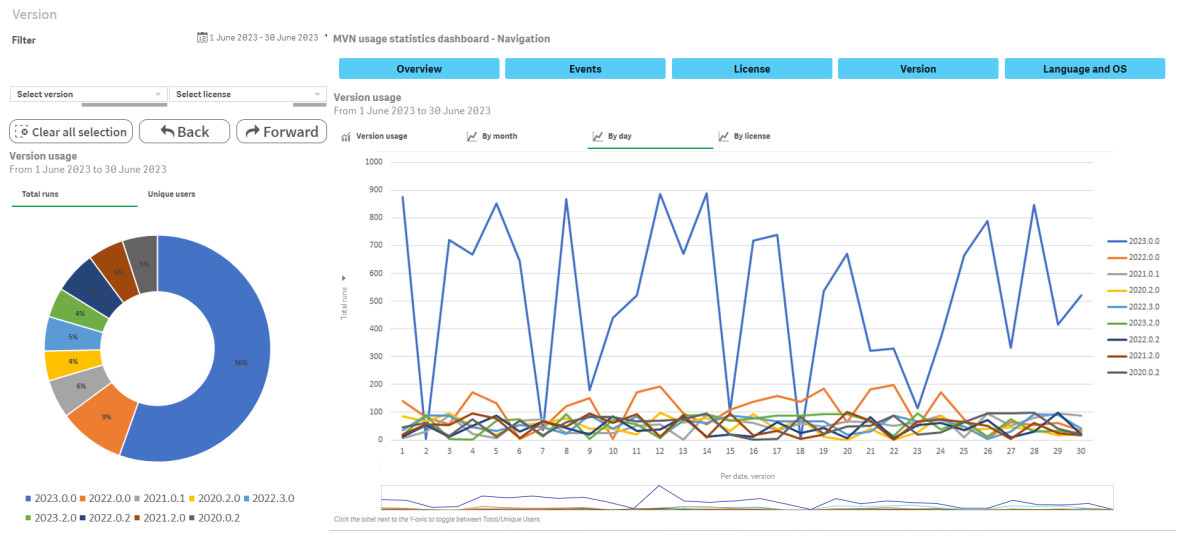

Figure 6.5: Version page of the data dashboard

Figure 6.6: Language and OS page of the data dashboard

events.

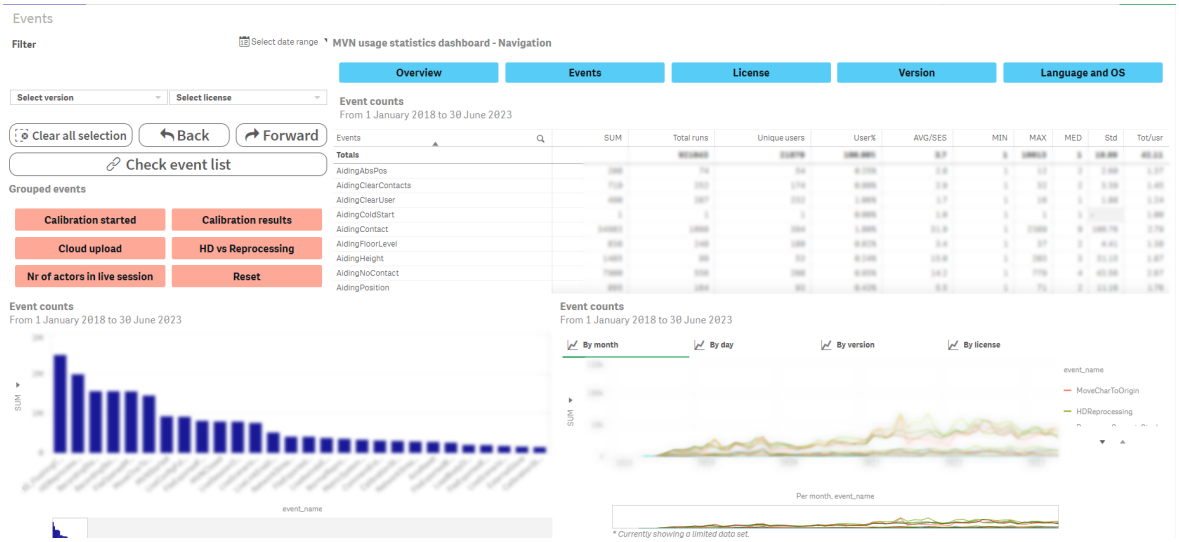

Figure 6.7: Event page of the data dashboard

6.6 Use case

In this section, I examine different user cases to illustrate the Qlik dashboard's practical use. It demonstrates how roles such as UX Designers, Product Managers, and Quality Engineers can utilize the dashboard in their daily tasks. For each role, a scenario is presented to showcase how the dashboard's features cater to their unique needs and facilitate their workflow. The goal is to clarify the dashboard's functionality for various job functions, highlighting its contribution to improved decision-making and operational efficiency.

User case - UX designer

A UX designer is curious about how users move the character to the starting point (reset) and seeks to identify the most preferred method of reset among users. This information is helpful for deciding whether to retain, modify, or enhance this specific feature. By clicking on the ”Reset” button on the Event Page of the data dashboard, as shown in the Figure 6.8, the designer can quickly assess the usage of different reset methods. The data reveals that the 'movechartoorigin' reset method is prominently used by people, with its usage exceeding other options like 'axisreset.' This insight aids the designer in determining whether to remove other reset methods, or to make the 'movechartoorigin' reset method more prominent, positioning it in a easily accessed area by users.

User case - Quality Engineer

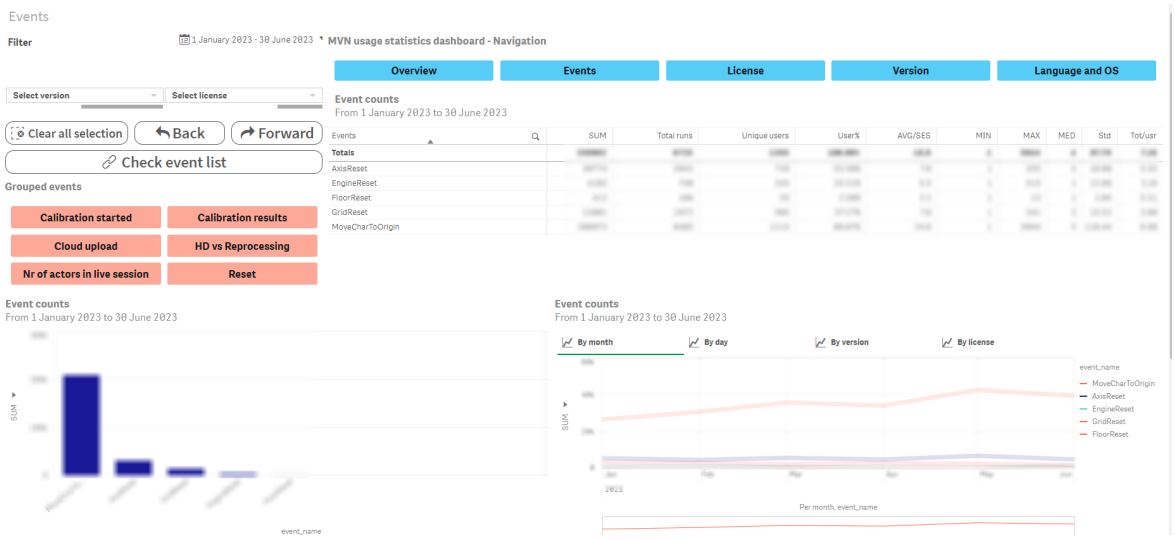

Figure 6.8: Reset buttion in the event page

A Quality Engineer needs to create a test plan and wants to know which operating systems users are primarily using. He goes to the OS Page (Figure 6.6) on the dashboard and quickly sees that most users are on Windows 10 and Windows 11 this year. With this information, he decides to include these two operating systems in every test to ensure the software works perfectly for the majority of users.

User care - Product Manager

A Product Manager is preparing for a meeting with stakeholders and needs to present data on the software's monthly usage, user count, and growth trends. He first visits the Overview Page (Figure 6.3) on the data dashboard to gather this information. With the needed data at hand, his attention shifts to the performance of the calibration feature, a topic slated for discussion during the meeting.

Navigating to the Event Page, he clicks on the "Calibration Results" button to analyze the users' experiences with this feature. The displayed data, as shown in Figure 6.9, indicates that a majority of users received the calibration results as "good." However, he notes that a considerable number of users received "poor" and "acceptable" results. With this data, the Product Manager is equipped to have a well-informed discussion with the stakeholders about the potential need to enhance the calibration feature to ensure consistently high-quality results for all users.

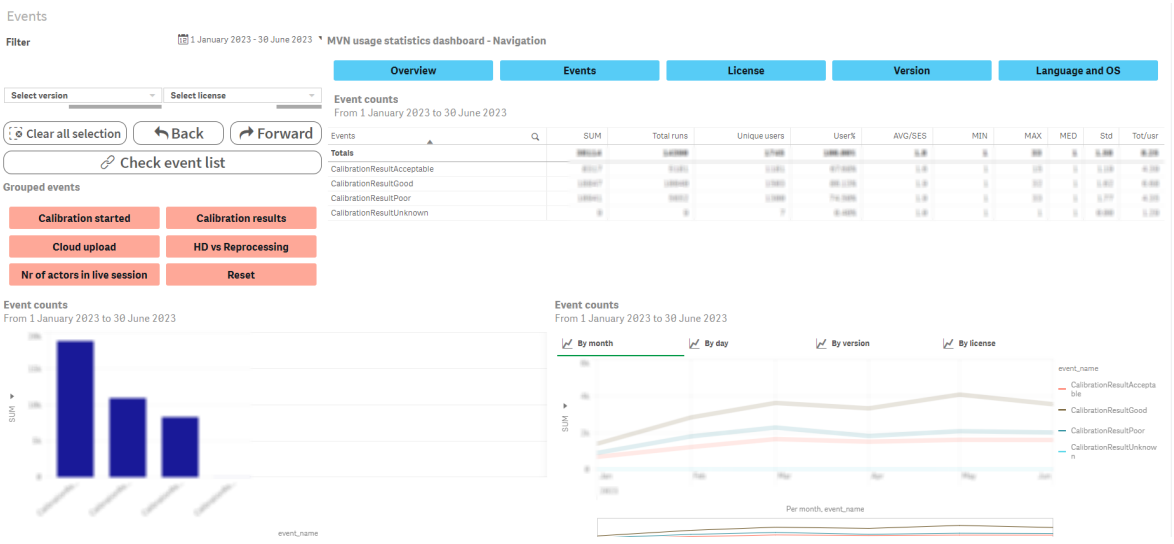

Figure 6.9: Calibration results

Chapter 7

Discussion

7.1 Challenges

During the research process, I encountered several challenges that I would like to discuss here. Firstly, during user interviews, I found that participants' responses were quite diverse, and their interests in data varied. However, the goal of this research is to design one data dashboard for all users. The diversity in responses made it challenging to extract a common pattern to inform the subsequent prototype design. To address this, I used an affinity diagram for text analysis, attempting to identify and emerge common themes and trends from these responses. I also compiled a requirement list summarizing the design needs of the dashboard to guide the subsequent prototype design. During the prototype design, I prioritized the features in the requirement list, focusing on the data most people wanted to see and the features most people needed based on their feedback.

Another challenge was encountered during the data preparation phase, where dealing with a large volume of data insertion was cumbersome. The data needed to be extracted from log files saved on the server to my local computer, inserted the database back to the server, and given the monthly quantity ranging from a minimum of ten thousand to as many as thirty to forty thousand log files, inserting data for a single month required ten hours. To tackle this, I employed a method of batch data insertion, inserting, for example, 5000 records into the database at a time, reducing the frequency of interactions with the database.

Additionally, I initially checked whether the data to be inserted into the database already existed before insertion. However, this significantly increased the data insertion time. As a result, I changed my approach to insert all data first and then performed data cleaning and delete duplicates to ensure data uniqueness. I would recommend this method only when there is a need for extremely high efficiency of

data insertion. This approach relies heavily on the structure and quality of the data; if there is plenty of poorly formatted data, the data cleaning process could become quite complicated.

7.2 Contributions

Despite the growing trends of utilizing software usage data in business decision-making, academic research in this field remains not extensive. This study introduces a set of steps for collecting, analyzing, and visualizing software usage data, providing a practical guide for this field. The method adopted in this research is not only beneficial for the case company but also has broader applicability for any entity looking to interact with software usage data.

Specifically, for Movella Company, this research has provided a usable dashboard. They can already begin using the data dashboard developed through this study, which allows them to visualize MVN software usage data and gain insights. For instance, as Movella Company is attempting to redesign their MVN software, the visualizations on the dashboard reveal which features are used more and which are barely used. This information informs whether to keep, expand, or remove certain features of the MVN software. The dashboard serves as a solid foundation, upon which further improvements and optimizations can be built. Furthermore, Movella can adopt this approach to other products within the company, enhancing their data-driven decision-making processes.

7.3 Limitation

This study is not without its limitations. One of the primary constraints was time. Due to the limited time and availability of test participants with the target roles, I wasn't able to conduct a comprehensive usability test on the developed data dashboard to evaluate and validate it thoroughly. My feedback collection was limited to the positive responses I received during my final presentation at Movella and the positive comments from my supervisor and UX team colleagues after they used the dashboard. A more rigorous usability testing of the developed data dashboard in a new round of study would enhance the completeness and scientific rigor of my research.

Furthermore, in terms of the implementation of the data dashboard, the study did not dive into the technical challenges associated with integrating various data sources and types into the dashboard. Issues such as ensuring the data uniqueness and security of the data, evaluating whether the current types of collected data are suf-

ficient, and identifying additional data that could be gathered were not thoroughly addressed. This oversight presents an opportunity for future exploration and could influence the real-world applicability and effectiveness of the data dashboard.

7.4 Future directions

Based on the insights and feedback gathered throughout this study, several directions have emerged for future research and development. These potential directions are aimed at addressing the limitations identified in the current work and building upon the initial findings. The goal is to further enhance the usability and effectiveness of the data dashboard, making it an even more valuable tool.

Thorough Usability Testing

Due to limited time and availability of participants, this study could not conduct usability testing on the data dashboard. Future efforts should focus on a thorough evaluation of the dashboard. Involving a wide number of participants from different departments within Movella Company will provide a more varied and comprehensive range of feedback and perspectives. Such testing is to ensure the dashboard meets various user needs and preferences.

More data and visualizations

The data and visualizations presented in the dashboard have more posibility. Future versions could include a more diverse range of data types and analyses, tailored to specific needs. For instance, visualizations could be generated using the 'event_name' and 'timestamp' fields, such as calculating the duration between 'recordingstarted' and 'recordingclosed', visualizing the time and trends of users using the MVN software for recording. Additionally, new data types can be collected, such as data related to software break off situations, which are useful for error diagnosing and repairing.

Optimizing Data Updates

The dashboard I created pulls data from Movella's server logs via a script on my local computer, then uploads it to a database on the Movella server. This method is inefficient, requiring at least two hours to process a month's data. To enhance efficiency, future improvements could involve running the script directly on the Movella server or inserting data into the database immediately after extraction from server logs. Also, strategies to ensure data uniqueness and integrity are necessary to avoid duplicate entries in the database.

Integrating AI

During earlier user interviews, some participants indicate they want to integrate artificial intelligence into the dashboard. For example, adding a feature similar to ChatGPT, where users can enter a prompt and receive the desired data and visualizations. This feature could even generate new visualizations based on user prompts. Additionally, AI algorithms could be incorporated to automatically diagnose potential issues in the MVN software. For instance, if there's an increase in the number of users experiencing interruptions while using the recording feature, the AI could highlight this issue and offer automated suggestions.

Chapter 8

Conclusion

In this thesis, I set out to answer the following research question:

How can a data dashboard be designed to utilize software usage data and cater to the distinct needs of various professionals within the product development team?

To answer this question, I took the following approach. To gain deeper insights into users' motivations, preferences, and data needs, I conducted interviews with five user roles. The goal was to understand their job tasks and gather insights into their preferences for using software usage data.

From the interviews, it became clear that all roles shared a common motivation to use software data to enhance their work processes and decision-making. They expressed a desire for data to be presented visually and interactively. The concept of a data dashboard was well-received, offering an easy-to-use platform for accessing data visualizations and applying filters for more in-depth analyses.

Regarding data preferences, there were both commonalities and differences among the roles. Shared interests included data on feature usage frequency, user counts for each feature, and trends in software usage. Developers showed a particular interest in technical performance and system stability data, while designers focused on user experience and interface interactions. Product managers looked at broader metrics like usage, active user count, and performance indicators.

With these insights, I designed a low-fidelity data dashboard prototype that caters to the diverse needs of different roles. The data was organized into three categories: 'Overview,' 'Feature,' and 'User Flow,' with user-friendly navigation and a variety of tailored data visualizations.

During the prototype testing phase, participants interacted with the prototype using

specific scenarios and tasks. Their feedback was generally positive, with valuable suggestions for improvement, which were integrated into the next design phase.

The final step involved developing a proof of concept, exploring the implementation of the data dashboard. Data was extracted from log files, inserted into a database, and linked to the Qlik data visualization platform. The result was a functional data dashboard that received positive feedback during its presentation at Movella.

In summary, this study presents a practical approach for creating a data dashboard to analyze software usage data. It serves as a reference for those interested in using software usage data. I've outlined the following steps:

1. Identify Data Sources and Collection Methods: This step involves recognizing and selecting suitable data sources that align with project objectives. Data collection methods may include automatic extraction from software systems, setting up log recording, or gathering feedback and input from users.

2. User Research: Gain a deep understanding of the target user group and their needs, preferences and pain points. Information can be collected through user interviews, surveys, observations and so on, and use this data to create user personas and use scenarios.

3. Define Design Goals: Based on previous user research and requirements analysis, clearly define the goals and requirement list for the dashboard.

4. Dashboard Prototyping: Start by designing prototypes of the data dashboard. Initially, create sketches or lo-fi prototypes to quickly iterate basic layouts and design elements. Subsequently, develop higher-fidelity prototypes, including more detailed interface designs, interactive features, and preliminary data displays.

5. User Testing and Feedback: Invite real users to test and evaluate the dashboard prototype. Provide them with testing scenarios and tasks. Gather feedback through observations, interviews, and task completion assessments. This feedback will be used for iterative design improvements.

6. Database Design: This includes determining the database, designing the database architecture, creating tables and relationships, and ensuring data security and privacy.

7. Data Preparation: Clean, transform, and integrate the collected data so that it can be effectively used for analysis and visualization. This may involve tasks like removing duplicate or incomplete fields, standardizing data formats, and more.

8. Data Visualization: Choose a suitable visualization platform; popular options include Tableau, Power BI, and Qlik. Then, start the design process by selecting appropriate charts and visualization elements, and create an intuitive and information-rich user interface.

In summary, this study developed a data dashboard for analyzing and visualizing software usage data, aimed at aiding Movella Company's MVN software. By collecting, analyzing, and presenting user interaction data, this research not only offered a solution for Movella but also provided a framework of steps applicable for any organization or individuals seeking to utilize software usage data. Future research directions include usability testing, expansion of data and visualizations, optimization of data updates, and integration of AI. These efforts are anticipated to further enhance the functionality and utility of the dashboard.

Bibliography

[1] B. Bach, E. Freeman, A. Abdul-Rahman, C. Turkay, S. Khan, Y. Fan, and M. Chen, "Dashboard design patterns," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 342–352, 2022.

[2] S. Suonsyrjä, "Data-driven software development with user-interaction data," *Tuni.fi*, 2019.

[3] S. Wexler, J. Shaffer, and A. Cotgreave, *The big book of dashboards: visualizing your data using real-world business scenarios*. John Wiley & Sons, 2017.

[4] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What do we talk about when we talk about dashboards?" *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 682–692, 2018.

[5] O. M. Yigitbasioglu and O. Velcu, "A review of dashboards in performance management: Implications for design and research," *International Journal of Accounting Information Systems*, vol. 13, no. 1, pp. 41–59, 2012.

[6] P. Yermalovich, "Dashboard visualization techniques in information security," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2020, pp. 1–6.

[7] L. Cannavacciuolo, C. Ponsiglione, and A. D'Ambrosio, "How to improve the triage: A dashboard to assess the quality of nurses' decision-making," *International Journal of Engineering Business Management*, vol. 13, p. 18479790211065558, 2021.

[8] S. E. A. BUANANTA and A. CHOWANDA, "Bi dashboard to support decision making on product promotion for payment/purchase transactions on e-banking," *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 15, 2021.

[9] W. W. Eckerson, *Performance dashboards: measuring, monitoring, and managing your business*. John Wiley & Sons, 2010.

[10] A. Wu, Y. Wang, M. Zhou, X. He, H. Zhang, H. Qu, and D. Zhang, "Multivision: Designing analytical dashboards with deep learning based recommendation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 162–172, 2021.

[11] A. Janes, A. Sillitti, and G. Succi, "Effective dashboard design," *Cutter IT Journal*, vol. 26, no. 1, pp. 17–24, 2013.

[12] R. W. Veryzer and B. Borja de Mozota, "The impact of user-oriented design on new product development: An examination of fundamental relationships," *Journal of product innovation management*, vol. 22, no. 2, pp. 128–143, 2005.

[13] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, "A survey of user-centered design practice," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2002, pp. 471–478.

[14] S. Adikari, C. McDonald, and J. Campbell, "Reframed contexts: design thinking for agile user experience design," in *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools: Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I 2*. Springer, 2013, pp. 3–12.

[15] C. S. G. González, E. G. González, V. M. Cruz, and J. S. Saavedra, "Integrating the design thinking into the ucd's methodology," in *IEEE EDUCON 2010 Conference*. IEEE, 2010, pp. 1477–1480.

[16] K. Tschimmel, "Design thinking as an effective toolkit for innovation," in *ISPIM Conference Proceedings*. The International Society for Professional Innovation Management (ISPIM), 2012, p. 1.

[17] D. Schallmo, C. A. Williams, and K. Lang, "An integrated design thinking approach-literature review, basic principles and roadmap for design thinking," in *ISPIM Innovation Symposium*. The International Society for Professional Innovation Management (ISPIM), 2018, pp. 1–18.

[18] J. Paredes, C. Anslow, and F. Maurer, "Information visualization for agile software development," in *2014 second ieee working conference on software visualization*. IEEE, 2014, pp. 157–166.

[19] H. Islam, X. Li, M. K. Z. Rana, K. Alaboud, T. Paul, A. Maruf, and A. S. M. Mosa, "An in-depth needs analysis to design a data visualization dashboard prototype for critical care setting," in *2022 Workshop on Visual Analytics in Healthcare (VAHC)*. IEEE, 2022, pp. 01–06.

[20] G. W. Young, R. Kitchin, and J. Naji, "Building city dashboards for different types of users," in *Sustainable smart city transitions*. Routledge, 2022, pp. 259–279.

[21] E. Tsangaris, M. Edelen, J. Means, M. Gregorowitsch, J. O'Gorman, R. Pattanaik, L. Dominici, M. Hassett, M. L. Witkowski, K. Schrieber *et al.*, "User-centered design and agile development of a novel mobile health application and clinician dashboard to support the collection and reporting of patient-reported outcomes for breast cancer care," *BMJ Surgery, Interventions, & Health Technologies*, vol. 4, no. 1, 2022.

[22] M. Goff, "Trends in software usage analytics and monetization 2021," Dec 2022.

[Online]. Available: https://www.revenera.com/blog/software-monetization/trends-in-software-usage-analytics-and-monetization-2021/

[23] E. Brynjolfsson, L. M. Hitt, and H. H. Kim, "Strength in numbers: How does data-driven decisionmaking affect firm performance?" *Available at SSRN 1819486*, 2011.

[24] M. M. d. Medeiros and A. C. G. Maçada, "Competitive advantage of data-driven analytical capabilities: the role of big data visualization and of organizational agility," *Management Decision*, vol. 60, no. 4, pp. 953–975, 2022.

[25] F. Provost and T. Fawcett, "Data science and its relationship to big data and data-driven decision making," *Big data*, vol. 1, no. 1, pp. 51–59, 2013.

[26] R. Y. Kim, "Data-driven user experience design," *Interactions*, vol. 30, no. 4, pp. 56–58, 2023.

[27] J. A. Marsh and C. C. Farrell, "How leaders can support teachers with data-driven decision making: A framework for understanding capacity building," *Educational Management Administration & Leadership*, vol. 43, no. 2, pp. 269–289, 2015.

[28] B. Gill, B. C. Borden, and K. Hallgren, "A conceptual framework for data-driven decision making," *Final Report of Research conducted by Mathematica Policy Research, Princeton, submitted to Bill & Melinda Gates Foundation, Seattle, WA*, 2014.

[29] E. B. Mandinach, M. Honey, and D. Light, "A theoretical framework for data-driven decision making," in *annual meeting of the American Educational Research Association, San Francisco, CA*, 2006.

[30] D. Wang, D. Guo, and H. Zhang, "Spatial temporal data visualization in emergency management: a view from data-driven decision," in *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Emergency Management using*, 2017, pp. 1–7.

[31] N. Ferreira, M. Lage, H. Doraiswamy, H. Vo, L. Wilson, H. Werner, M. Park, and C. Silva, "Urbane: A 3d framework to support data driven decision making in urban development," in *2015 IEEE conference on visual analytics science and technology (VAST)*. IEEE, 2015, pp. 97–104.

[32] D. Gotz and D. Borland, "Data-driven healthcare: challenges and opportunities for interactive visualization," *IEEE computer graphics and applications*, vol. 36, no. 3, pp. 90–96, 2016.

[33] M. E. Brown, "Data-driven decision making as a tool to improve software development productivity," Ph.D. dissertation, Walden University, 2013.

[34] R. B. Svensson, R. Feldt, and R. Torkar, "The unfulfilled potential of data-driven decision making in agile software development," in *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP*

2019, Montréal, QC, Canada, May 21–25, 2019, Proceedings 20. Springer, 2019, pp. 69–85.

[35] J. Pruitt and J. Grudin, "Personas: practice and theory," in *Proceedings of the 2003 conference on Designing for user experiences*, 2003, pp. 1–15.

[36] A. Lucero, "Using affinity diagrams to evaluate interactive prototypes," in *Human-Computer Interaction–INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14-18, 2015, Proceedings, Part II 15*. Springer, 2015, pp. 231–248.

[37] V. R. Basili, "Goal, question, metric paradigm," *Encyclopedia of software engineering*, vol. 1, pp. 528–532, 1994.

[38] M. Dastani, "The role of visual perception in data visualization," *Journal of Visual Languages & Computing*, vol. 13, no. 6, pp. 601–622, 2002.

[39] M. A. Hearst, "Uis for faceted navigation: Recent advances and remaining open problems," in *HCIR 2008: Proceedings of the Second Workshop on Human-Computer Interaction and Information Retrieval*. Citeseer, 2008, pp. 13–17.

[40] X. Qin, Y. Luo, N. Tang, and G. Li, "Making data visualization more efficient and effective: a survey," *The VLDB Journal*, vol. 29, pp. 93–117, 2020.

Appendix A

User interview

A.1 Interview questions for user research

Interview questions for product owner

Warm-up

Hello! My name is Chenhao Yi, UX intern at Movella. I am now writing my master thesis in the company to explore some ideas about the usage of MVN user statistics. Before we start, I would like to request your permission to audio-record this interview. It will be only for internal usage and will be deleted after a while.
Here is the consent form of the interview. If you have no other questions, could you sign it?

Role and workflow

- Could you tell me a little bit about your job responsibilities and the types of tasks you work on regularly?

MVN statistics features

- Can you please share with me the inspiration behind using the MVN user statistics?
- What data or results of the MVN statistics are you most interested in, that would help with the decision making in the Grogu project?
  - Can you walk me through the decision-making process?
  - Have you ever encountered any challenges or pain points in the decision-making process of Grogu project?
- How do you currently measure the performance of the Grogu project?
  - Any metrics or data used

- Can you think of what kind of person submitted the request to use the MVN statistics?
  - In what ways do you think the statistics could be helpful for the workflow of people from different departments correspondingly, like product manager, product specialists, software developers?
- What kind of person do you think might be good to interview regarding the MVN statistics?
- What features or capabilities do you think would be important for the usage statistics tool to have?
- Can you describe any specific scenarios or use cases where this tool would be particularly helpful?

Concerns and obstacles

- What concerns or challenges do you think it might have during the implementation of the statistics tool?
- Can you think of any potential issues or obstacles that might prevent people from using this statistics tool effectively?

Interview questions for software developers

Warm-up

Hello! My name is Chenhao Yi, UX intern at Movella. I am now writing my master thesis in the company to explore some ideas about the usage of MVN user statistics. Before we start, I would like to request your permission to audio-record this interview. It will be only used for research analysis and can be only accessed by me. And I will delete it after a while.
Here is the consent form of the interview. If you have no other questions, could you sign it?

Workflow and pain points

- Could you tell me a little bit about your job responsibilities and the types of tasks you work on regularly?
- How do you currently measure the performance of a specific feature or an algorithm?
- How do you communicate with stakeholders about the software performance?
  - do you think user statistics could help improve those communications?

- Can you think of any pain points or challenges in your current workflow? Especially those that statistics might help address.

MVN statistics features

- Have you used the statistics before? If yes, can you share with me how?
  - Do you know how the statistics and event sets will be updated if there is a new feature releasing?
- What kind of user statistics would be most valuable to you in your role as a software developer, and why?
- What features or capabilities do you think would be important for the usage statistics tool to have?

Concerns and obstacles

- What concerns or challenges do you think it might have during the implementation of the statistics tool?
- Can you think of any potential issues or obstacles that might prevent people from using this statistics tool effectively?

Interview questions for UX designers

Warm-up

Hello! My name is Chenhao Yi, UX intern at Movella. I am now writing my master thesis in the company to explore some ideas about the usage of MVN user statistics. Before we start, I would like to request your permission to audio-record this interview. It will be only for internal usage and will be deleted after a while.
Here is the consent form of the interview.[] If you have no other questions, could you sign it?

Workflow and pain points

- Could you tell me a little bit about your job responsibilities and the types of tasks you work on regularly?
- Can you describe your current process for identifying user behaviours when they are using MVN?
  - How do you determine if users are currently using the software in a desired way?
- Can you describe a time when you identified a usability issue of MVN, and how did you find the root cause of this issue?

- How do you prioritize the usability issues to address first, what metrics or data do you use?
- How do you involve stakeholders (such as product managers, developers, etc.) while identifying usability issues and making design decisions?
- What are the pain points that you have when identifying usability problems?
- Have you ever faced any challenges when trying to make design decision?

MVN statistics features

- What data or results of MVN statistics are you most interested in, that would help with usability identifications and design decision making?
- In what ways do you think that MVN statistics could potentially help to improve your current workflow and address some of the pain points that you've described?
- Can you think of any potential issues or obstacles that might prevent you from using the MVN statistics tool?

Interview questions for product specialists

Warm-up

Hello! My name is Chenhao Yi, UX intern at Movella. I am now writing my master thesis in the company to explore some ideas about the usage of MVN user statistics. Before we start, I would like to request your permission to audio-record this interview. It will be only used for research analysis and can be only accessed by me. I will delete it after a while.
Here is the consent form of the interview. If you have no other questions, could you sign it?

Workflow and pain points

- Could you tell me your job responsibilities and the types of tasks you work on regularly?
- What are some challenges you face when handling support cases, and how do you typically address those challenges?
- How do you currently prioritise the support cases? Is there any data and metrics used?
- How do you currently communicate with stakeholders about the product issues, is there any graphs used?

MVN statistics features

- Is there any data of user statistics you are most interested in?
- What features or capabilities do you think would be important for the statistics dashboard to have?
- How would you envision user statistics being integrated into your current workflow?
  - how do you think they could help improve the customer support process?

Interview questions for quality managers

Warm-up

Hello! My name is Chenhao Yi, UX intern at Movella. I am now writing my master thesis in the company to explore some ideas about the usage of MVN user statistics. Before we start, I would like to request your permission to audio-record this interview. It will be only used for research analysis and can be only accessed by me. I will delete it after a while.
Here is the consent form of the interview. If you have no other questions, could you sign it?

Workflow and pain points

- Can you describe your current workflow for managing quality assurance processes?
- How do you currently identify trends or patterns in quality issues?
- How do you prioritize quality issues, and how do you track them through resolution?
- How do you currently collaborate with other departments in Movella to improve quality?
- What are the challenges or pain points you face when ensuring MVN quality, and how do you typically address those challenges?

MVN statistics features

- What data or results of the MVN statistics are you most interested in, that would help with the current workflow?
- What features or capabilities do you think would be important for the usage statistics tool to have?
- How would you envision user statistics being integrated into your current workflow, and can you describe any specific scenarios or use cases where this tool

would be particularly helpful?

Concerns and obstacles

- What concerns or challenges do you think it might have during the implementation of the statistics tool?
- Can you think of any potential issues or obstacles that might prevent people from using this statistics tool effectively?
- What kind of people might be interested to talk to?

Interview questions for product managers

Warm-up

Hello! My name is Chenhao Yi, UX intern at Movella. I am now writing my master thesis in the company to explore some ideas about the usage of MVN user statistics. Before we start, I would like to request your permission to audio-record this interview. It will be only used for research analysis and can be only accessed by me. I will delete it after a while.
Here is the consent form of the interview. If you have no other questions, could you sign it?

Workflow and pain points

- Could you tell me your job responsibilities and the types of tasks you work on regularly?
- How do you currently track the performance of MVN?
- How do you currently find out the problems of MVN?
- How do you figure out how important the problem is?
- How do you determine the 'opportunity value' of a problem solution?
  - Sometimes in the voice-of-the-customer docs they put a number there where they estimate the value of the solution. how they make such an estimate.
- Is there any data and metrics that you use in the decision-making process?
- What are the challenges or pain points in your current workflow, especially those the statistics might help address?

MVN statistics features

- Have you tried to use the MVN statistics? What for? How?

- What are some features that you wish were available in a user statistics dashboard?
  - Or how would you envision the dashboard?
- What kind of data/visualizations do you find most helpful?

A.2 Information letter and consent form for user interview

Information Letter and Consent Form

We would like to invite you to participate in a research interview to collect insights and opinions on the proposed usage of MVN user statistics. This information will be used to inform the design of user statistics usage tool and ensure that it meets the needs of internal employees.

The interview will consist of a series of open questions related to current workflow, pain points and needs. We will also ask you about your knowledge and opinions on the proposed internal web tool, including how it might support your work and address any pain points or inefficiencies in your workflow.

The interview will be an audio-recorded interview for accuracy and research analysis lasting approximately 30 minutes. All information you provide during the interview will be kept confidential and are only accessible to the researcher to protect the personal information. Your participation is voluntary, and you may choose to withdraw at any time without having to give a reason. The recording of the interview will be transcribed as text anonymously and will be destroyed after analysis. Additionally, your responses may be quoted in the researcher's master thesis anonymously.

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the researcher Chenhao Yi(c.yi@student.utwente.nl) or the Secretary of the Ethics Committee Information & Computer Science at the University of Twente: ethicscommittee-CIS@utwente.nl.

Consent

By signing below, I confirm that I have read and understand the information provided in this consent form, and I voluntarily agree to participate in the research interview. I understand that I can refuse to answer questions and may withdraw my consent at any time without having to give a reason. I give permission that the interview will be audio-recorded. I agree that my information can be quoted in the master thesis anonymously.

Participant Signature: _____

Date: _____

UNIVERSITY OF TWENTE.

A.3 Personas

Figure A.1: The persona for UX designer

Figure A.2: The persona for quality engineer

Figure A.3: The persona for software engineer

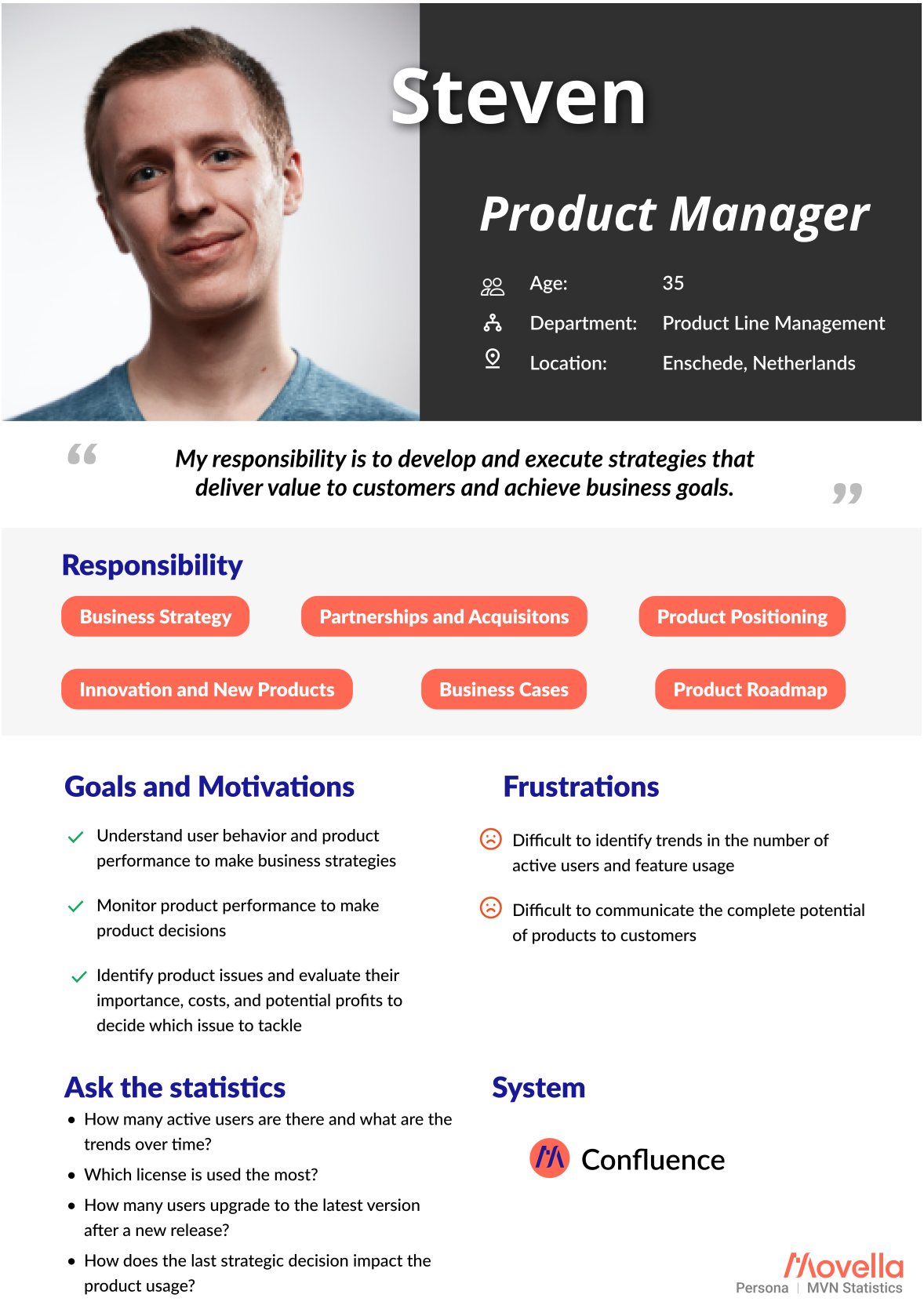

Figure A.4: The persona for product manager

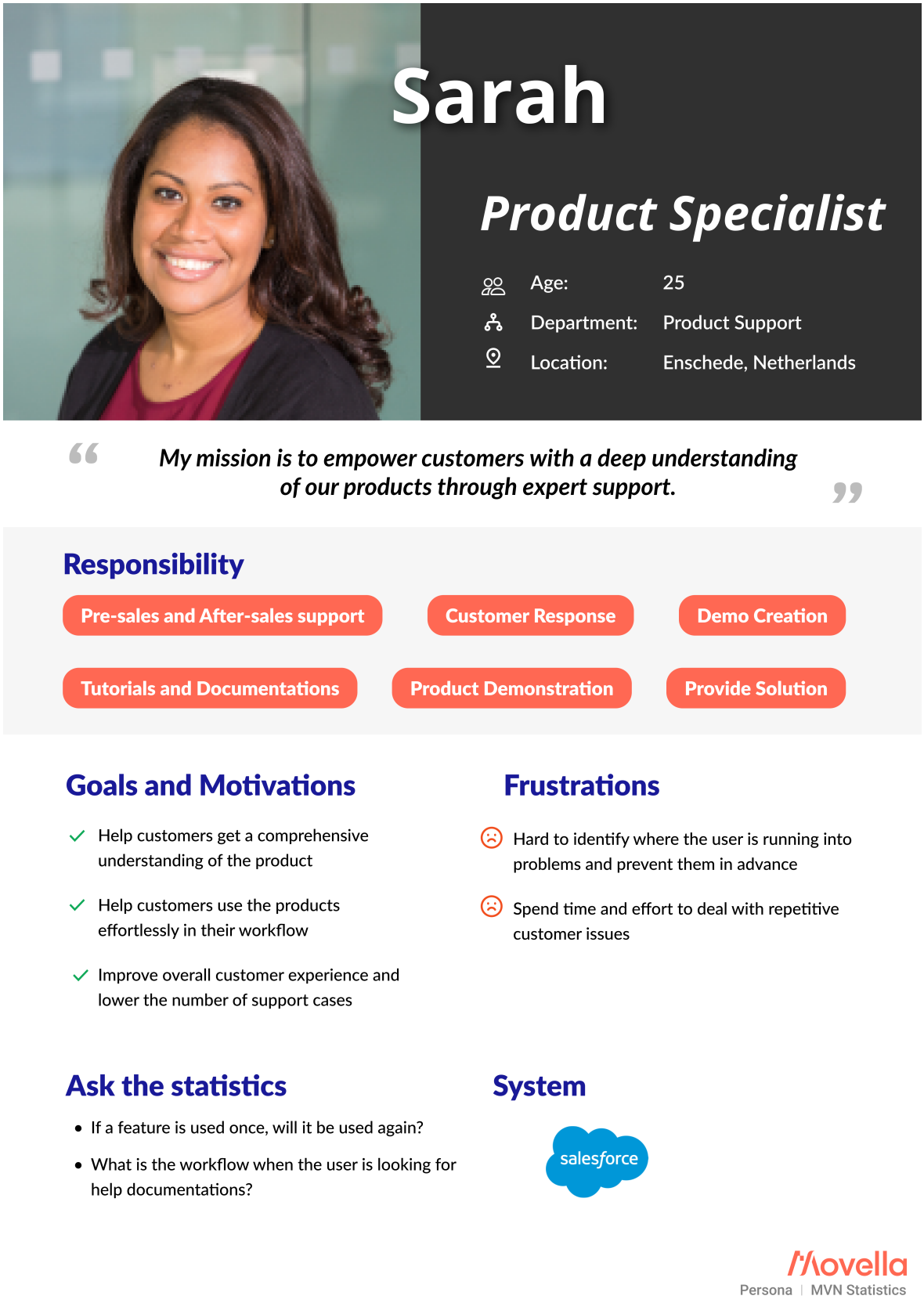

Figure A.5: The persona for product specialist

A.4 Affinity Diagram for Analyzing Interview Results

Appendix B

Usability testing

B.1 Materials used for usability testing

In this appendix, you will find the materials used for usability testing. Different materials are provided for each target role. These materials include a welcome text, scenarios, tasks for participants, and a short interview conducted after testing the lo-fi prototype.

Usability test with the quality assurance

Welcome text

Hello! My name is Chenhao Yi, UX intern at Movella. Thank your participating in this usability test of the low-fidelity prototype of MVN statistics dashboard. You'll be asked to perform the provided tasks using the prototype. Afterwards, I'll ask you a few questions to gather your feedback and insights.
Please remember to try to speak out your thoughts during the test, such as what you try to do, what problems you encountered, what you are satisfied or not and so on. It can help me to understand more accurately.
Before we start, I would like to request your permission to audio-record and screen-record this testing. It will be only for internal usage and will be deleted after a while.

Interview questions

The prototype:

- What are the main problems you have found while using this prototype?
- What is the part of the prototype that has been more difficult to understand? Why?
- Can you describe your overall experience with this prototype?

- What have you liked most of the prototype?
- What kind of data that you are interested in and how are you going to use them for?
- What other features and data do you expect that would be in this dashboard?
  Questions for developers/QA:
- How to identify the usage of a certain feature using the event counts? Event_Started?
- How to label the events?
- What is the usage of the identified parameters for event counts?
- Who is maintaining and updating the even list?

Scenario

You are a Quality tester in the company, and you are planning the quality test for MVN software, including whether and how extensive to test and support a specific OS, language, license, version, and features. You are going to find out the data using MVN statistics dashboard.

Task

1. Find out which language users have chosen the most in the last three month.
2. Find out which versions that users are still using in the last three month.
3. Find out the top two operation systems users have used in the last three months.
4. Find the number of users for each license and its trends over time.
5. Check the three most triggered events in the last month.

Usability test with the UX designer

Welcome text

Hello! My name is Chenhao Yi, UX intern at Movella. Thank your participating in this usability test of the low-fidelity prototype of MVN statistics dashboard. You'll be asked to perform the provided tasks using the prototype. Afterwards, I'll ask you a few questions to gather your feedback and insights.
Please remember to try to speak out your thoughts during the test, such as what you try to do, what problems you encountered, what you are satisfied or not and so on. It can help me to understand more accurately.
Before we start, I would like to request your permission to audio-record and screen-record this testing. It will be only for internal usage and will be deleted after a while.

Interview Questions

The prototype:

- What are the main problems you have found while using this prototype?
- What is the part of the prototype that has been more difficult to understand? Why?
- Can you describe your overall experience with this prototype?
- What have you liked most of the prototype?
- What kind of data that you are interested in and how are you going to use them for?
- What other features and data do you expect that would be in this dashboard?

User flow:

- Any suggestions about how to identify user flow out of the statistics?
- How to divide the features according to user flow?

Scenario

You are a user experience designer in the company, and you are in the process of redesigning the MVN software. And you are going to look for some data in the MVN statistics dashboard to inform your design decision.

Task

1. Find the number of users for each license and its trends in the last three months.
2. Find out how many times the 'Calibration' feature has been started by users, as well as its trends over the past three months.
3. Check the typical order of events that are triggered by the user.

Usability test with the software developer

Welcome text

Hello! My name is Chenhao Yi, UX intern at Movella. Thank your participating in this usability test of the low-fidelity prototype of MVN statistics dashboard. You'll be asked to perform the provided tasks using the prototype. Afterwards, I'll ask you a few questions to gather your feedback and insights.

Please remember to try to speak out your thoughts during the test, such as what you try to do, what problems you encountered, what you are satisfied or not and so on. It can help me to understand more accurately.

Before we start, I would like to request your permission to audio-record and screen-record this testing. It will be only for internal usage and will be deleted after a while.

Interview Questions

The prototype:

- What are the main problems you have found while using this prototype?
- What is the part of the prototype that has been more difficult to understand? Why?
- Can you describe your overall experience with this prototype?
- What have you liked most of the prototype?
- What kind of data that you are interested in and how are you going to use them for?
- What other features and data do you expect that would be in this dashboard?

Questions for developers:

- How to identify the usage of a certain feature using the event counts? Event_Started?
- How to label the events?
- What is the usage of the identified parameters for event counts?
- Who is maintaining and updating the even list?

Scenario

You are a software development engineer in the company. And you are going to find some data in the MVN statistics dashboard.

Task

1. Find out which language users have chosen the most in the last three month.
2. Find out which versions that users are still using in the last three month.
3. Find the number of users for each license and its trends over time.
4. Check the three most triggered events in the last month.

Usability test with the product manager

Welcome text

Hello! My name is Chenhao Yi, UX intern at Movella. Thank your participating in this usability test of the low-fidelity prototype of MVN statistics dashboard. You'll be asked to perform the provided tasks using the prototype. Afterwards, I'll ask you a few questions to gather your feedback and insights.

Please remember to try to speak out your thoughts during the test, such as what you try to do, what problems you encountered, what you are satisfied or not and so on. It can help me to understand more accurately.
Before we start, I would like to request your permission to audio-record and screen-record this testing. It will be only for internal usage and will be deleted after a while.

Interview Questions

- What are the main problems you have found while using this prototype?
- What is the part of the prototype that has been more difficult to understand? Why?
- Can you describe your overall experience with this prototype?
- What have you liked most of the prototype?
- What kind of data that you are interested in and how are you going to use them for?
- What other features and data do you expect that would be in this dashboard?

Scenario

As the product manager of the company, you are interested in tracking the performance of the MVN software, and thus, you are going to gather some data from the statistics dashboard.

Task

1. Check the number of active users out there and the trends over time.
2. Find out how many users updated to the latest version.
3. Check the number of users for each version and the trends over time.
4. Check the results of “Calibration” feature, the trends of good, acceptable, and poor results.

B.2 Information letter and consent form for usability testing

Information Letter and Consent Form

The MVN usage statistics refers to the data about the use of specific parts and features of MVN software. There were a few interviews conducted to collect people' needs to make use of the MVN usage statistics. According to it, I completed a lo-fi design of the statistics dashboard. And I would like to invite you to participate in a usability testing of the lo-fi design.

You as a participant will be asked to perform the tasks using the prototype. After that, I will ask you a few questions to gather your feedback and insights. The whole process will last approximately 30 minutes. You are encouraged to speak out your thoughts during the testing, such as what you try to do, what part you don't understand, what you are satisfied with or not and so on. It can help me to understand more accurately.

I would like to request your permission to screen-record without recording sound during the testing. The recordings will only be accessed by me for research analysis and will be deleted after analysis. All information you provide during the interview will be kept confidential and are only accessible to the researcher. Your participation is voluntary, and you may choose to withdraw at any time without having to give a reason. The notes taken during the testing and interview will be used only in this thesis research anonymously and will be destroyed after analysis. Additionally, your responses may be quoted in the researcher's master thesis anonymously.

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the researcher Chenhao Yi(c.yi@student.utwente.nl) or the Secretary of the Ethics Committee Information & Computer Science at the University of Twente: ethicscommittee-CIS@utwente.nl.

Consent

By signing below, I confirm that I have read and understand the information provided in this information letter and consent form, and I voluntarily agree to participate in the usability testing.

☐ I give permission for the data collected during the testing to be used for the purpose of this master thesis research.

☐ I understand that I can refuse to answer questions and may withdraw my consent at any time without having to give a reason.

☐ I give permission that the testing will be screen-recorded.

☐ I agree that the information I provide can be quoted in the master thesis anonymously.

Participant Signature: _____

Date: _____

UNIVERSITY OF TWENTE.